TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Computer Systems

IA70LT

Aleks-Daniel Jakimenko-Aleksejev 163447IASM

AUTOMATED CAMERA MOTION CONTROL SYSTEM FOR RHYTHMIC GYMNASTICS

Master's Thesis

Supervisor:	Siavoosh Payandeh Azad
	Researcher
Co-Supervisor:	Kristina Vassiljeva
	Associate Professor

Tallinn 2019

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

IA70LT

Aleks-Daniel Jakimenko-Aleksejev 163447IASM

AUTOMATISEERITUD KAAMERA JUHTIMISE SÜSTEEM ILUVÕIMLEMISE JAOKS

Magistritöö

Juhendaja: Siavoosh Payandeh Azad
Teadur

Kaasjuhendaja: Kristina Vassiljeva
Dotsent

Tallinn 2019

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, the literature and the work of others have been referenced. This thesis has not been presented for examination anywhere else.

Author: Aleks-Daniel Jakimenko-Aleksejev

2019-05-20

Abstract

High quality filming and broadcasting of local sport events is hard to achieve due to budget constraints. This thesis explores whether a low-cost solution with automated camera motion control can be developed. The work focuses on rhythmic gymnastics.

To answer the question, a complete solution was developed. The developed system includes a mechanical design, a PCB design, firmware, middleware, image processing and control software.

The solution was successfully tested at several local events where it was able to replace a camera operator for most of the duration of the events. The calculated autonomy of the system is 99.2 %. The system was evaluated against a human operator, and results show that it performs on a comparable level. Specifically, in 85.6 % of cases, framing and fluidity of motion are as good or better than the performance of a well-rested human operator. Therefore, it is clear that quality and cost-effectiveness of broadcasting and recordings can be drastically improved by using automated solutions.

This thesis is written in English and is 60 pages long, including 7 chapters, 18 figures, and 5 tables.

List of abbreviations and terms

CNC	Computer Numerical Control
ISR	Interrupt Service Routine
LANC	Logic Application Control Bus
PCB	Printed Circuit Board
PTP	Picture Transfer Protocol
PTZ	Pan-Tilt-Zoom
RG	Rhythmic Gymnastics
SPI	Serial Peripheral Interface

Table of Contents

List of Figures

List of Tables

1 Introduction

Creating high-quality recordings and broadcasts of small sports events is challenging. Unlike high profile events, there is not enough financing to cover the cost of a human production crew. The unfortunate consequence of this problem is that many events are not being recorded or broadcast at all, while others only get a simplistic single-camera broadcast and recordings filmed by a novice operator.

This issue is especially prevalent at local Rhythmic Gymnastics (RG) competitions. It is not uncommon for these events in Estonia to last up to 12 hours. Such long durations are due to relative popularity of the sport locally, as well as some organizational specifics like multiple competitions being often combined into a single seamless event. With durations that long, livestreaming is the most convenient solution for viewers because it allows each individual to only watch performances they are interested in, without having to attend the full event.

Currently, most videographers have to do everything manually. The labor includes not just setting up all of the cameras, but also actively operating them throughout the full duration of the event. Practice shows that although most event organizers are extremely interested in having a livestream, they can rarely afford the work required for a proper livestream of the full event. Autonomous robotic cameras can change the situation by drastically reducing the amount of worker hours required to cover an event, and in case of multi-camera setups, improve the overall production quality of broadcasts and recordings.

The presented work provides a complete low-cost automated solution that can produce human-like results. This solution solves the problems with recording and broadcasting of local RG competitions by reducing and easing the work of a camera operator. The described system was successfully used for commercial purposes at multiple Estonian sport events.

1.1 Background

The use of computer vision in sports forms a diverse research area [1]. A subset of this area is the use of autonomous cameras for broadcasting to do planning (where to look?), controlling (how to move?), and selecting (which camera should be on air?) [2].

Over the years, different solutions were attempted for different sports such as figure skating [3], hockey [4], soccer [5], [6], etc. More generic approaches were also proposed, one of which detects gaze directions of the audience in order to estimate the area of interest [7]. Different approaches have different challenges. For example, solutions that depend on camera calibration can have issues with keeping the camera calibrated [8]. Camera calibration involves figuring out its extrinsic (position and rotation) and intrinsic parameters (focal length, principal point), as well as lens distortion coefficients [9].

Control is typically done in one of two ways. Either the camera is moved in real time to keep the subject in frame, or the required frame is taken from a stitched high-resolution panorama. To keep the best of both worlds, hybrid solutions were also attempted to produce video by both roughly moving the camera and doing digital processing at the same time [10]. Overall, even if the planning is completely resolved, it can be challenging to control the camera in an aesthetic way [11], [12], [13].

On a higher level, there have been attempts to build production systems that require fewer crew [14]. The existence of such attempts indicates that not only it is often desired to reduce the size of the crew, but also that such optimizations are possible.

While most of the research focuses on video, taking photos in an automated way is also an interesting problem. For example, Newbury et al. propose a method of detecting good-quality pictures [15]. While a lot of inspiration can be taken from that area of research, video recording is an inherently different problem which requires different solutions.

Video processing is also used for evaluating performance of athletes in different sports. One such approach was used for rhythmic gymnastics [16]. However, its working principle heavily relies on optical flow and stable background conditions, which means the approach cannot be easily reused for camera control automation.

1.2 Problem statement

While many software approaches were proposed for specific sports throughout the years, most of these systems depend on expensive hardware like commercially available Pan-Tilt-Zoom (PTZ) cameras [17], [18], [19]. Even though good software can indeed reduce the amount of work required from the crew, expensive hardware still acts as a major roadblock for many scenarios. In other words, there is a need for a complete and practical solution that does not require substantial investments. Moreover, its design should be open so that it can further evolve and be adapted for other niches. To the best of my knowledge, no complete system was ever publicly documented. Some businesses provide proprietary solutions for certain sports (there is a famous case in American football [20]), but not for rhythmic gymnastics.

While the ePTZ (Electronic PTZ) approach (later described in 2.1) can be attractive because of its seeming simplicity, it does not allow for additional angles (judges, audience, etc.) and high-resolution close-up shots. The overall cost and need to buy additional equipment makes ePTZ solutions unfeasible. Moreover, ePTZ solutions will never result in aesthetic videos because of their inherent design. For example, with a regular camera that follows a fast-moving subject, the subject has minimal motion blur while the background is blurred. However, with the ePTZ approach, the effect of motion blur is reversed and the subject is blurred the most, which makes videos look unnatural [21].

1.3 Proposal

In this thesis I describe a complete affordable system that was successfully used at several rhythmic gymnastics events. In this solution, existing cameras can be mounted on a low-cost pan-tilt head. This avoids the expense of typical PTZ cameras.

To control the zoom, the system uses the LANC protocol which is supported by many cameras, ranging from cheap consumer devices to high-end production systems.

Instead of relying on camera calibration, I propose a method of detecting the floor area and a gymnast in relation to the current position based on the video feed from the camera. This makes the system indifferent to its position, orientation, current focal length and lens distortion.

To control the movement of the camera after a gymnast is detected, a PID controller is used. The coefficients of the controller were calibrated and tested at real events. Additionally, there is a multitude of special-cased rules that augment the behavior of the PID controller to make the video more visually pleasing.

All of the software components of the system are connected together using a lightweight custom-built middleware. The middleware enables automatic discovery and flexible data distribution using a publish-subscribe mechanism. This allows the output of the system to be used beyond its functional boundaries. For instance, messages from the image processing node can be interpreted in the streaming software to display overlays for the online audience whenever a gymnast enters the floor area.

The designed pan-tilt head can be built using primitive materials and tools, or mass-produced at a low cost.

In this thesis, neural networks will not be used since this area requires a separate substantial research that was done in another thesis [22].

1.4 Author's contributions

Contributions of this thesis are:

- CAD – mechanical design of a pan-tilt head which allows directional positioning of any camera placed on it.
- EDA – schematics and PCB design. The PCB allows to drive the motors and communicate with the camera over LANC.
- Firmware – code for the microcontroller on the PCB that performs pulse generation for the motors as well as implements LANC protocol.
- Middleware – a helper library based on Zeroconf and ZeroMQ which allows flexible setup and communication over the network.
- Computer Vision – image processing software specifically designed for Rhythmic Gymnastics competitions.
- Control – PID control loop that makes the output of computer vision software usable to control the system.

All components together form a complete system that can be used for camera motion control at rhythmic gymnastics events.

2 Proposed approach and system requirements

In this chapter, some basic requirements and design decisions are discussed. The goal is to set a baseline of what a good solution should be able to do.

2.1 Camera directional control

Rhythmic gymnastics is a very dynamic sport. The official rules have a requirement that the choreography must cover the entire floor [23]. This means that the gymnast is constantly moving, and a fixed camera can not be used to produce eye-pleasing results (Figure 1). Therefore, the camera must be able to pan, tilt and zoom during the whole routine while keeping the gymnast in frame.

In this work, only the corner cameras will be considered (Figure 2). That being said, the proposed approach can also be used for other use cases like front-view or top-view cameras.

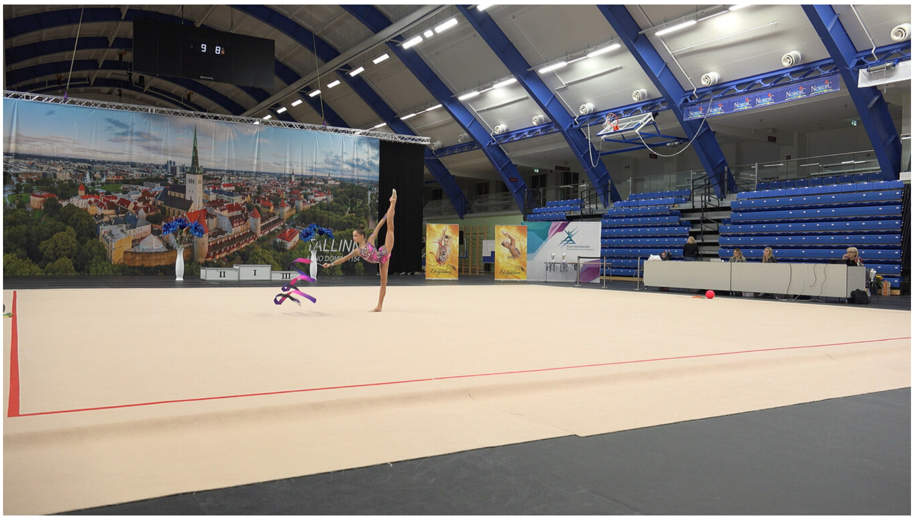


Figure 1. View from a camera placed at the corner. With no zoom, the gymnast looks too small in relation to the whole frame.

Three different classes of solutions exist for following a subject of interest: PTZ cameras, ePTZ approaches, and motorized pan-tilt heads.

A pan–tilt–zoom camera (PTZ camera) is a camera that can be electronically controlled

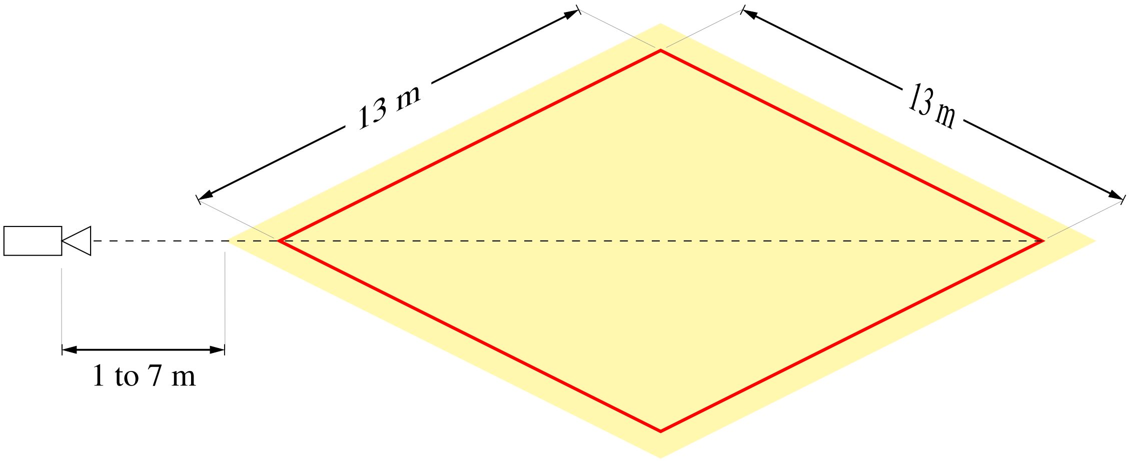


Figure 2. Illustration of the standard floor area and a corner camera.

to change the direction of the camera and its zoom. Many integrated PTZ cameras are available on the market. However, non-PTZ cameras with similar specifications are considerably cheaper and come in a much wider variety. Moreover, most videographers in the field have already invested in professional or semi-professional non-PTZ equipment, which means that the requirement to upgrade to a specific type of hardware is unrealistic. Therefore, commercially available PTZ cameras will not be used in this work.

The electronic PTZ (ePTZ) approach was also considered. With ePTZ, pan-tilt-zoom actions are performed digitally by first capturing a wide-angle high resolution video and then using part of it accordingly. Such approach is possible in the context of this thesis, but it requires cameras capable of capturing video of a much higher resolution (at least 8K) which are much more expensive than what will be proposed in this work. The same problem arises when a stitched panorama from feeds of multiple cameras is used. Similarly to PTZ cameras, this approach is not desirable for videographers who have already acquired quality equipment, hence it is not considered in this work.

Motorized pan-tilt mechanism can be built separately from the camera. In this case, PTZ controls (at least for pan and tilt) are also available, but with less vendor lock-in and no requirement to use a specific camera. Systems like this are also commercially available. For example, OpenMoCo (not to be confused with OpenMoko which will be mentioned later in this thesis) was an attempt to develop an open-source photographic camera motion control system [24]. While the project disappeared many years ago, it is now available as a commercial product [25]. Looking at the currently available models, it seems that the hardware can potentially satisfy the goals of this thesis, but at a very high price of over 1400 €. There is another cheaper project that costs around 900 € for a single camera setup [26]. Under a closer inspection, it seems that it relies on a very basic motorized pan-tilt head that costs around 100 € [27]. That pan-tilt head is rather limited, with no way to control the acceleration profile, and no way to operate at very low speeds. Filming at rhythmic gymnastics competitions often happens at high zoom levels where smoothness of operation is crucial. In other words, it is hard to adapt existing pan-tilt heads for filming rhythmic gymnastics competitions because of either high price or limited hardware features.

If a motorized pan-tilt head is used, a separate solution is required to control the zoom. Many cameras have servo zoom and are able to control it according to the user input (zoom ring encoder or zoom rocker), and a subset of them have a way to control zoom from external devices (zoom control is discussed in section 2.2).

In this work I will present a solution with a custom made pan-tilt head for directional control.

2.2 Camera zoom control

There are several options for controlling the zoom of a camera. Logic Application Control Bus (LANC) [28] is a communication protocol that was initially introduced by Sony, but is present on many cameras of various manufacturers. Different cameras may support different features, although zoom control is more or less universal across all cameras that are capable of controlling it on a hardware level.

LANC is not the only way that can be used to control the zoom of a camera. One of the alternatives is to use gPhoto [29], which is a software library and tool that can communicate with cameras over USB using Picture Transfer Protocol (PTP). Unfortunately, the PTP standard focuses on DSPDs (Digital Still Photography Devices) [30], and as a result most video cameras do not support it, especially when it comes to control. In case of cameras that I had available, all of them do support Media Transfer Protocol (MTP) which is based on PTP, but features required for controlling the camera are not available.

Many cameras also allow control by other means (IrDA, WiFi, Bluetooth, etc.). Typically, in each case the protocol is incompatible with other cameras, and comes with its own set of restrictions. For example, throughout this thesis I have tested WiFi control of the Sony AX100, and it turns out that the camera is unable to record video when it is in WiFi mode. Moreover, the protocol has very few commands available, far fewer than provided by LANC.

Another alternative is to mechanically couple a zoom ring of a camera with an external motor. This is also known as FIZ (Focus-Iris-Zoom) control. This approach is not very elegant because many small cameras do not have a zoom ring that is physically connected to the lens system, and instead have a rotary encoder which simply acts as a user input interface for controlling the servo zoom digitally. In some situations this may be the only approach available, for example if there is no other way to control an existing lens. I would like to explore this in the future, but in the scope of this thesis a FIZ-like approach will not be used.

In this work I will present a solution that relies on LANC to control the zoom of a camera.

2.3 Required angular speeds

The system should be able to handle low angular speeds without producing visible stepped motion. At the same time, the maximum speed should not be too limited. If these requirements are not met, the system will not enable smooth motion or might not be able to follow a fast-moving gymnast.

In order to estimate angular speed requirements, an angle of view needs to be calculated for the highest zoom level. For doing the calculations, I will assume a maximum focal length of 300.0 mm (35 mm equivalent). 300.0 mm is a sweet spot because lenses with that focal length are widely available yet satisfy most of the needs for a typical rhythmic gymnastics event.

Calculating the angle of view precisely can be challenging, but a simple model can give a good rough approximation [31]. A formula used to calculate the angle of view is shown in Equation (1). It relies on the assumption that the focal length does not change (and hence the angle of view) when focusing, which is mostly true for video cameras. Lens distortion is also not taken into account.

$$\theta = 2 \cdot arctan\left(\frac{h}{2f}\right) \tag{1}$$

Plugging the numbers ($h = 36\ mm$ – frame width of a full frame sensor, $f = 300\ mm$ – focal length), we get $6.87°$ as an estimation for the horizontal angle of view. Assuming that the recorded videos are in Full HD (1920x1080), it is fair to say that on average to cause a 1 pixel shift in the video, a $3.58 \cdot 10^{-3}$ (°) angular rotation is required. This value will be taken into account when discussing the actual achievable speeds later in this thesis. That being said, there does not seem to be a clear line that can be drawn. For example, a human operator also introduces some camera shake even if a fluid head[1] is used, and the amount that is considered acceptable is subjective.

2.4 Other requirements

Due to long durations of competitions, the system must be able to run continuously for up to 12 hours. However, battery-powered operation is not required, because in most real-life scenarios it is possible to connect the system to grid power.

The system should not interfere or impose restrictions on other parts of setups that are already in use. For example, anyone who does livestreaming already has some means of capturing the video feed from the camera (USB3 capture cards, PCIe capture cards, built-in camera WiFi functionality, NDI, etc.), and forcing some particular way will make the system less useful.

The target audience of this work is videographers who have enough technical knowledge to build and maintain a complete system, as well as researchers who would like to explore new ways to control PTZ cameras.

* * *

The rest of the document is organized as follows. First, I will introduce the designed hardware (mechanical and electronic design). Then, I will describe the development of firmware and middleware. Finally, I will show how image processing was implemented and how it can be used to control the system. Afterwards, the system will be evaluated by comparing its performance to a human operator.

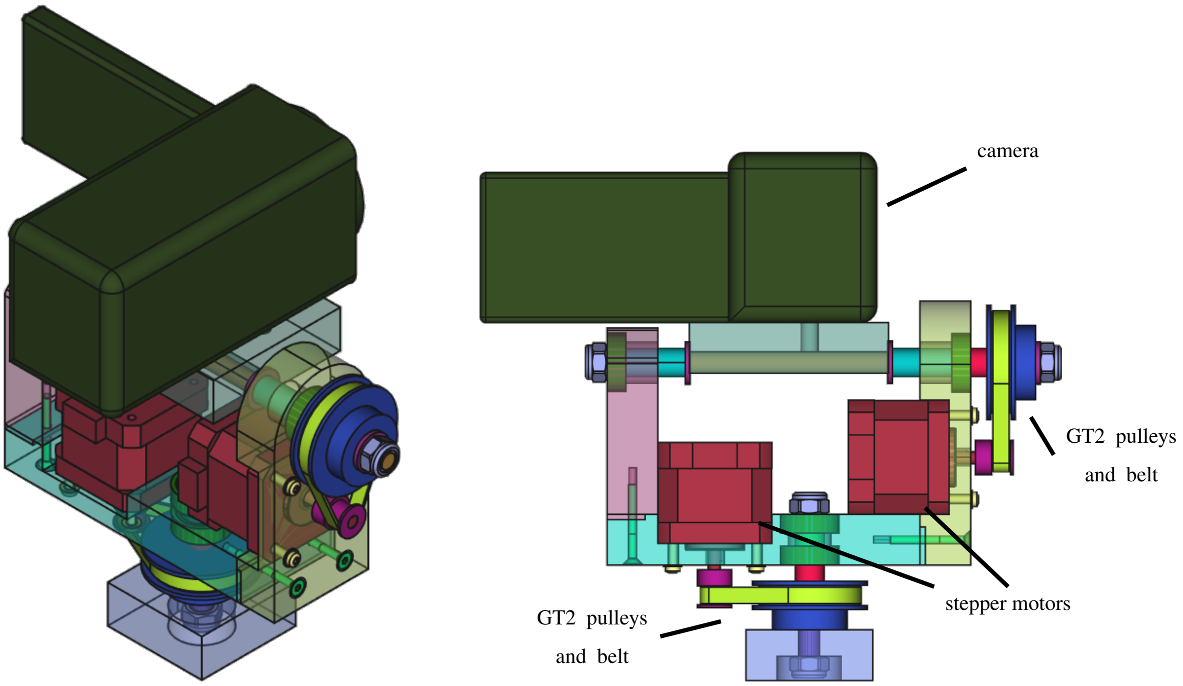
[1] Fluid head is a special type of a tripod head specifically designed to dampen vibrations from user input

3 Mechanical and electronic design

The hardware developed for this thesis sets the foundation for further software development. In this chapter, the mechanical and electronic designs will be discussed.

3.1 Details of the mechanical design

This section describes how I designed and created the mechanical part of the system. Figure 3a and 3b show renderings of the developed hardware.



(a) Rendering from a composite angle.

(b) Rendering from the back.

Figure 3. Renderings of the mechanical part of the system.

The design uses 2 stepper motors to control two axes of rotation (pan and tilt). To couple the axis shafts with motors, GT2 timing belts and pulleys were used. GT2 is specifically designed for reversible motion with minimal backlash. For both axes, a combination of a 16T driving pulley (on the motor shaft) and a 60T driven pulley is utilized, which gives a 3.75x reduction. To minimize the complexity of the design, a 152T closed loop belt is used to couple the pulleys, and the motor is placed at a fixed distance from the axis without any tensioning mechanism.

The design is built around M8 hardware, meaning M8 bolts, nuts and washers, and 608 bearings (commonly known as “skate bearings”). The 8 mm rods that are used for shafts are supposed to have a thread which can be added using an M8 thread cutting die. While deep-groove ball bearings are typically associated with radial load, they have approximately equal thrust-load capacities [32], which is why they are also used to handle axial load on the tilt shaft. Nylon-insert lock nuts are shown in the figure and are recommended, but are not obligatory (any M8 nut should work).

Most of the chosen hardware parts are commonly available in hardware or hobby stores, and are also often used for projects like 3D printers [33].

The design does not call for any specific tripod mounting solution. Most manufacturers of tripods tend to have their own proprietary mechanisms, and designing the system for some specific tripod type would be too limiting. Although designing for ¼ or ⅜ inch screws may result in the most universal solution, in many cases this would mean that the developed pan-tilt head would have to be mounted on top of a manual pan-tilt head. This has implied disadvantages of additional weight and cost, as well as bulkiness and lesser structural rigidity.

State-of-the-art industrial automation control systems use brushless motors. Unfortunately, the cost of using brushless motors is at least 10 times higher than an alternative system with stepper motors [34]. Since stepper motors are capable of fulfilling all requirements of a pan-tilt head, it was therefore decided to use them. Stepper motors are perfect for high torque at low speeds (below 1000 rpm), which is exactly what is required for camera directional control. Given that they are often associated with relatively coarse movement due to stepping, it is important to discuss their applicability in this project. Typical stepper motors have 200 steps per revolution, which is only 1.8° per step. However, modern stepper motor drivers can do microstepping [35]. The drivers that are used in this work (this will be discussed in more detail in chapter 3.2) are capable of 256 microsteps per native step, which results in approximately 0.007° per step. The design uses a combination of 60T and 16T pulleys, both of which are commonly available. This set of pulleys gives a 3.75x reduction. Therefore, the total resolution of the system is:

$$\Delta\theta = 360 \div 200 \div 256 \cdot \frac{16}{60} \approx 1.88 \cdot 10^{-3} \ \ ^{\circ}/\text{step} \tag{2}$$

As mentioned in 2.3, this value should be compared to $3.58 \cdot 10^{-3}$ °/step. Because the actual resolution of the system is less, it means that at the maximum considered zoom level each microstep causes just a small change in the angle, significantly less than what is required to “shift” the image by 1 pixel.

Because the value is so small, a curious reader may wonder if it would make sense to sacrifice some of the low speed performance in favor of higher maximum speed (after all, these speeds will not be used for continuous motion in practice). However, these changes would come with disadvantages. Additionally, a very low minimum speed by itself is not a drawback, and higher speeds can be achieved by using built-in step interpolation in stepper drivers. Major examples of disadvantages of increasing the minimum speed are: making the reduction ratio smaller will amplify the nonlinearity effects of microstepping; reducing the number of microsteps per full step may introduce vibrations, noise and inaccuracy.

3.1.1 Accuracy of microstepping

While microstepping indeed allows increasing the resolution of a stepper motor, it is known that microsteps in most drivers are not very accurate [36]. The accuracy depends on many factors, such as motor driver’s design, the operating voltage and current, and other configurable settings of the driver. In order to evaluate the applicability of microstepping, I conducted the following experiment to measure its accuracy.

The driver being used in this work is TMC2130. The voltage was 19.5 V, and the current in the driver was set to 800 mA (RMS). The drivers were configured to use 256 microsteps. In this test only the stealthChop mode was tested (see Chapter 3.2).

Because at that point the system was already built, it was easy to test microstepping using the actual hardware. By mounting a camera on the assembled pan-tilt head, it was possible to calculate the angle by taking images from camera of a known reference object.

The camera used had a 30x optical zoom. As a reference, a measuring tape was placed 5.25 meters away. Digital zoom was used for convenience, so that it would be easier to point the camera. To make reading the measured value less difficult, a line was rendered over the frames in post-processing using ffmpeg [37]. One of the frames can be seen in Figure 4. All values were manually extracted from the frames and written down. Figure 5 shows the resulting graph.

It is worth noting that with the given setup any small disturbances in the surroundings (like people walking a few meters away) were resulting in large fluctuations in the reading because the deformation of the floor is perfectly noticeable at such high zoom levels.

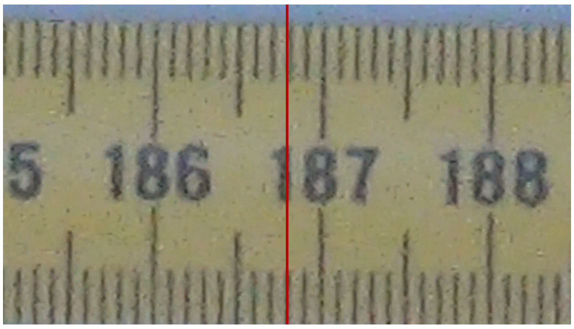


Figure 4. One of the frames from the microstepping accuracy test.

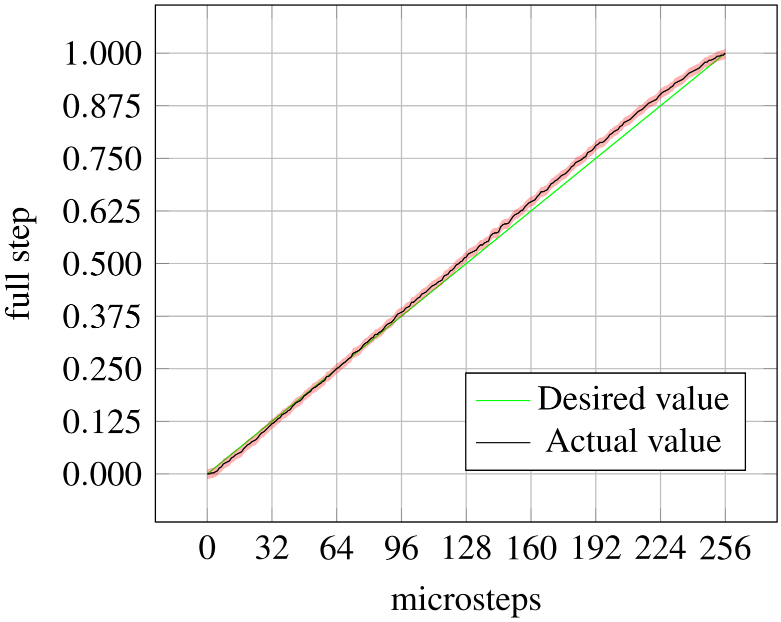


Figure 5. Linearity of microstepping when using TMC2130. Values are normalized, ±0.012 (±0.5*mm*) uncertainty is shown in red.

Arguably, this goes to show that the resolution of the system is high enough that issues like non-linearity of microstepping are mostly irrelevant.

Looking at the graph, it is clear that a high number of microsteps can be used to increase smoothness of the operation and overall resolution of the system. The graph is mostly linear, although there is an insignificant deflection from linearity after the half step, which is currently unexplained. If necessary, performance of microstepping can be improved by raising the current or voltage. Even if the microstepping pattern was too non-linear, it would have been possible to do some compensation in the firmware. That being said, the average deviation in the data above is only 1.4 %, which suggests that the system can be used as is for recording rhythmic gymnastics.

3.1.2 Design and manufacturing

To create a mechanical design for this work, I used FreeCAD, which is a free and open-source general-purpose 3D modeling tool that not only supports constructive solid geometry (CSG), but also allows parametric feature editing methodology [38], [39]. Constructive solid geometry is a way of creating complex objects by using boolean operations on primitive objects. On the other hand, with feature editing methodology, a 3D shape is constructed by creating a series of 2D constraint sketches and operations that work on these sketches. In either case, all operations form a dependency graph, meaning that any constraint or shape can be modified later and anything that depends on it will be recomputed. This should make it easier for anyone to change the design according to the available materials and tools.

Prototyping and small-scale production typically falls into one of two categories: additive manufacturing and subtractive manufacturing. Additive manufacturing is a way of creating an object by adding material to it. Subtractive manufacturing is a way of creating an object by means of controlled material removal, where chunks of existing material are machined away from the stock to “reveal” a required part inside of it.

A common way of doing additive manufacturing is 3D printing. 3D printing is cheap and nowadays widely available. The most commonly available process is fused filament fabrication (FFF), where the material (typically plastic) is deposited in layers. The filament is usually heated and then extruded through a nozzle in order to fuse it to previously placed layers. This is easier to achieve if the material has low glass transition temperature. For example, two of the most common materials used for filament are PLA and ABS. Their glass transition temperatures are around 60-65 °C and 105 °C respectively. However, mechanical strength relative temperature index (RTI) of pure ABS is only 60 °C (or 80 °C with common additives). RTI characterizes long term thermal degradation of plastic materials. Even though it is commonly not a big concern, special care should be taken when designing long-lasting parts that will be used near heat sources (e.g. stepper motors operated at high power). Because the goal of this work is to produce a sturdy design that will not degrade over time, the use of 3D printing will be avoided. Still, the reader should feel free to use 3D printing for creating the designed parts.

Subtractive manufacturing usually involves using Computer Numerical Control (CNC) machining, but can also be carried out manually. In this thesis, designed parts have simple features and while I did most of the work on a lightweight CNC milling machine (Figure 6), the parts can also be created by using simpler power tools. Many features of the model are optional (fillets, chamfers, some of the blind holes, alignment lips).

This approach allows quick prototyping and small-volume manufacturing, while being expandable to medium-volume production.

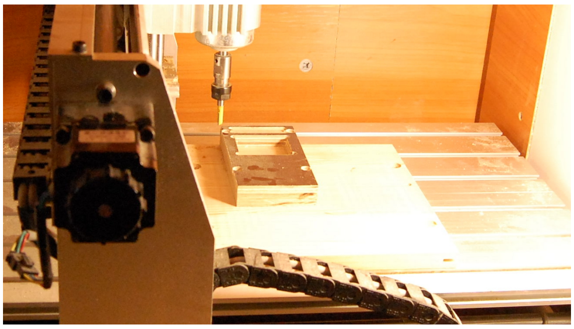


Figure 6. Photo of a part being cut on a CNC mill (testing some pocket operations).

In the future I would like to try creating a similar design using aluminium profiles (T-slot or V-slot) and other commercially available parts so that no machining operations are needed to build the system. While that would noticeably increase the total cost of materials and parts, for many people such design can be more accessible.

3.1.3 Results

The designed pan-tilt head was built and taken into use. Full plans are provided in Appendix 2. Figure 7 shows a fully assembled pan-tilt head. FreeCAD source files are available under a free/open-source license and a link can be found in Appendix 1.

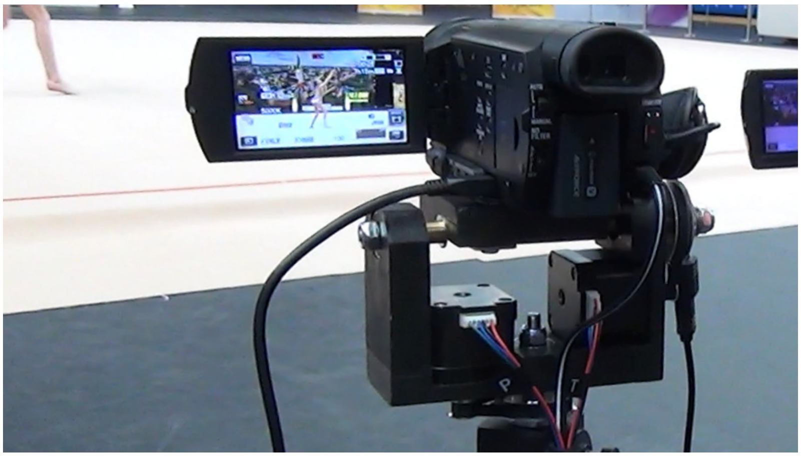


Figure 7. Photo of the assembled pan-tilt head.

The motors are driven by a control board which also interfaces with a camera. Section 3.2 describes the details of its schematic diagram and Printed Circuit Board (PCB) design.

3.2 Schematics and board design

To control the motors and communicate with the camera, a PCB had to be created. This chapter discusses the schematics and board design. The resulting PCB is shown in Figure 8.

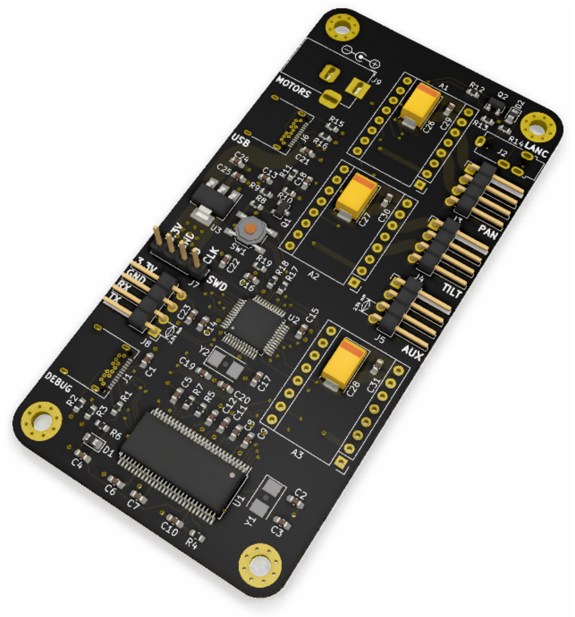


Figure 8. Render of the designed PCB.

The system was first prototyped on a simple development board with ATmega328 microcontroller. Despite that, the final design features an STM32F103C8T6 microcontroller instead. There were several reasons for the switch. One is that the new microcontroller can be clocked up to 72 MHz instead of just 16 MHz. Because of the high microstepping count, resolving fast multi-axis camera acceleration profiles requires high performance of the microcontroller in order to maintain the right timing between step pulses. While it is possible to use hardware PWM and other workarounds, the higher operating frequency allows keeping the firmare simple and leaves more room for additional functionality. As a bonus, the new microcontroller has a USB peripheral which makes it easy to use the developed board as a proper standalone device.

To control the stepper motors, I used TMC2130 stepper motor drivers [40]. TMC2130 offers a completely noiseless current control mode (stealthChop). The noiseless operation is important in some circumstances because the board will be located relatively close to the camera, and the chopper hiss can be picked up by the microphone. Moreover, TMC2130 allows using up to 256 microsteps, as well configuring the driver on the fly via SPI. The combination of these factors make it the perfect driver for this project.

To control a camera over LANC, a simple 2.5 mm jack was added. Although some cameras may have a different connector, in most cases the connection to the 2.5 mm port can be achieved via a couple of available adapters. The timing of LANC protocol is similar to RS232 at 9600 baud rate, but the communication happens over a single wire [28]. The camera acts as a master and pulls the line low for the start bit of every byte. Depending on the number of the byte in the telegram sequence, a slave should read or write data. For example, the first 2 bytes of a telegram are writable and are used to send commands to the camera. Some information from the camera can also be retrieved. The voltage on the data line can be up to 8V, it is therefore important to drop it to a lower level (3.3V in this case, although pins of the microcontroller are 5V-tolerant) with a Zener diode (Figure 9). The protocol can be implemented via bit banging. The firmware can also make use of the USART peripheral that is available on the same pins.

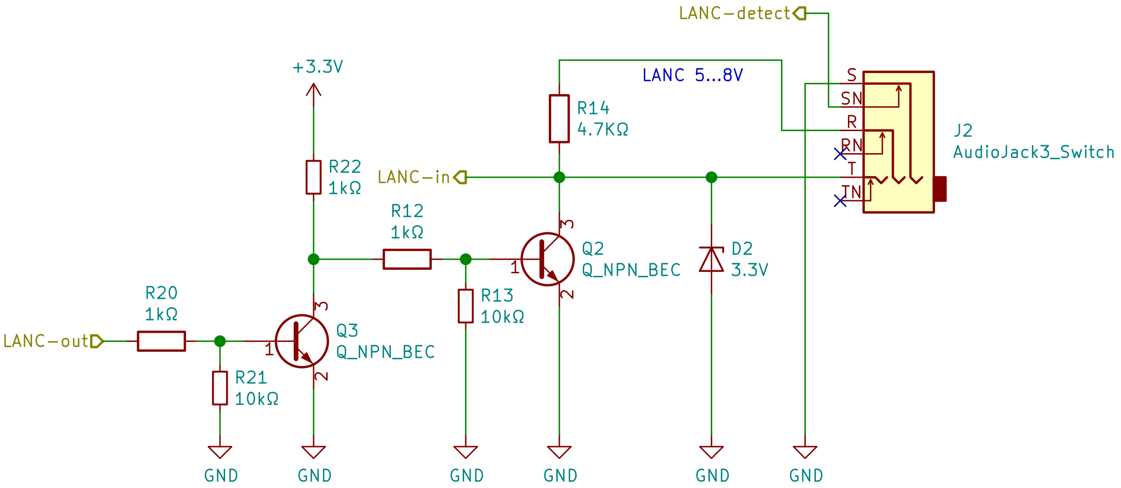


Figure 9. Schematics: using the LANC.

Besides LANC, the board should also provide an interface to a PC and motors. The choice of other connectors was driven mostly by availability of corresponding cables and other parts. For example, for connecting the board to a PC via a USB 2.0 interface, I used USB-C receptacles. For the motors I used simple 2.54 mm pitch pin headers, since many stepper motors available to consumers come with corresponding cables.

The stepper motor drivers are not integrated on the board, and instead are supposed to be plugged into the appropriate headers. This makes it possible to replace and upgrade the drivers if needed.

Because system requirements tend to change faster than the hardware can, the board has some optional footprints that do not have to be populated, but were added in order to simplify software and firmware development in case new functionality is needed in the future, or if extra debugging is required. One such feature is an additional socket for a stepper motor driver. While currently unused, it may later be required to have a third motor for a third rotation axis (roll), for controlling zoom on cameras than have no servo zoom, or for adding translational motion control (as in a motorized camera dolly). An optional CY7C68013A microcontroller [41] can be placed for either debugging purposes, or to aid reverse-engineering of the LANC data sent by the camera. When used with sigrok [42], it acts as a built-in logic analyzer [43]. Sigrok is a free/open-source signal analysis software suite.

Table 1. PCB layer stack.

Layer	**Function**
Top	Signals
Inner 1	Ground plane
Inner 2	Power plane
Bottom	Signals

The designed board has four layers. The layer stack is shown in Table 1. While it is possible to route the same design on two layers, the cost savings are not worth the extra complexity of the PCB design. Four-layer boards are roughly twice as expensive per area, but two-layer boards require extra care when routing and are more susceptible to noise [44]. The board was routed manually in KiCad [45].

3.2.1 Results

The designed PCB was assembled, tested and incorporated into the system. Full schematics are provided in Appendix 3 and gerber files in Appendix 4. Figure 10 shows a fully assembled board. KiCad source files are available under a free/open-source license and a link can be found in Appendix 1.

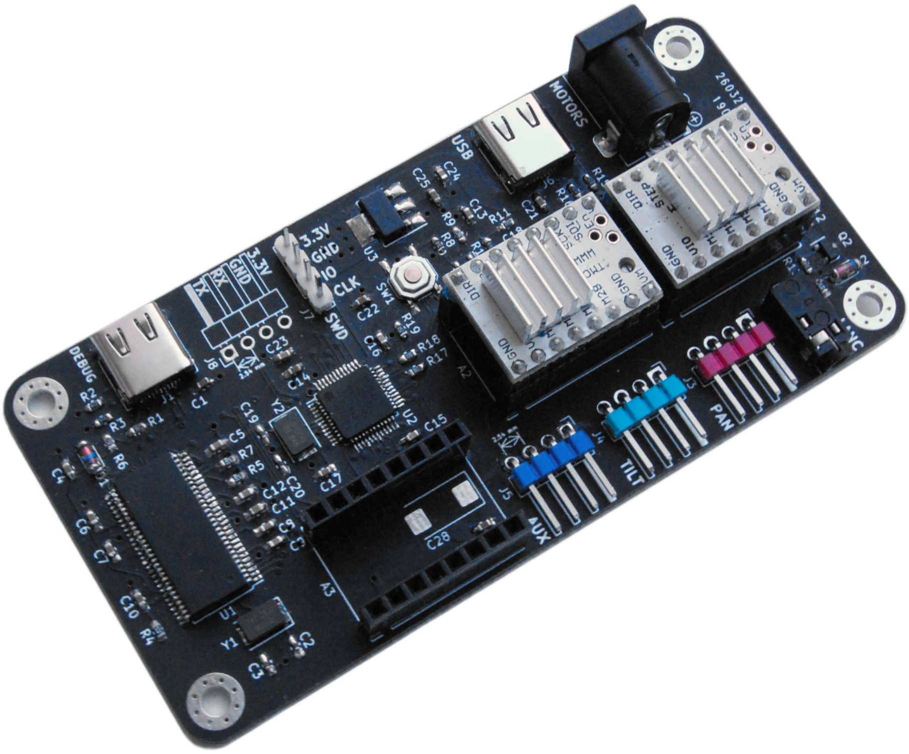


Figure 10. Fully-assembled PCB.

To make full use of the designed PCB, a firmware for the microcontroller is needed. Development of the firmware will be discussed in the next chapter, specifically in Section 4.1.

4 Firmware and software development

In this chapter, firmware and software will be discussed separately.

4.1 Firmware

The firmware communicates with a computer over a USB interface and drives the motors accordingly. Moreover, it implements the LANC protocol in order to control the zoom of a camera.

To write the firmware, LibOpenCM3 [46] was used. LibOpenCM3 is an open-source firmware library meant to replace proprietary HALs (Hardware Abstraction Libraries) provided by microcontroller manufacturers. To compile the firmware I used GCC and arm-none-eabi toolchain. Other open-source tools that were used are stlink [47], stm32flash [48], and openocd [49].

Stepper motors are controlled by sending a pulse to the driver's "step" pin, and the direction of rotation is determined by the level on the "dir" pin. Parameters of the drivers are adjusted over Serial Peripheral Interface (SPI) [40]. The implemented pulse generation is tickless. That is, the timer is configured to trigger an interrupt when the next pulse would need to be sent. In the Interrupt Service Routine (ISR), the pulse is sent and the timer is reconfigured for the next pulse. Currently, the firmware only implements a trapezoidal profile mode (constant acceleration), where the acceleration value can be set from the software. In other words, jerk control is currently not implemented [50], [51]. Most of the required motion is very slow and the way the stepper motors are used does not seem to cause immediate rigid motion, suggesting that 3rd-order (and beyond) motion planning is not strictly required. A proper real-time implementation of jerk-controlled motion typically requires a carefully written assembly code [52], meaning that there is a trade-off between the simplicity of the firmware and the resulting smoothness of acceleration/deceleration.

Reading from LANC is implemented via hardware USART features of the microcontroller, and datagram-related logic (e.g. which byte of the datagram is being read right now) is handled in the ISR that is called when a byte is received. For writing (i.e. sending commands), two implementations were written. In both of them, the start bit of every byte triggers an interrupt which is then disabled (so that subsequent bits do not trigger it again). This interrupt is then re-enabled from a USART ISR when a byte is received. The first approach implements writing using hardware USART features, and the second one is implemented with bit-banging where a timer is used to trigger an ISR regularly for shifting out the bits. Both approaches are viable, and even though it is generally preferable to use hardware features of the microcontroller, using hardware TX of the USART peripheral was shown to be unreliable. On the given microcontroller, the transmission can only start on the next baud clock edge (within the duration of 1 bit) [53], and when it comes to implementing LANC, it is too late half of the time. This makes it unreliable, which is why an implementation using bit-banging is preferred and is used. A new datagram is detected if the time between two start bits was large enough. If there is a command that needs to be sent to the camera on that particular byte, the line will be pulled low accordingly. Because LANC has only a single wire for communication, the data sent from the microcontroller on the output pin will also be visible on the input pin (Figure 11).

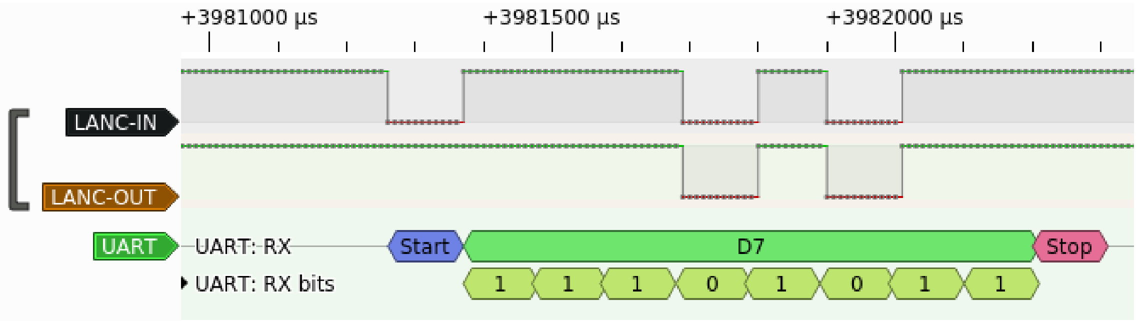


Figure 11. LANC communication captured with the built-in logic analyzer. The start bit is sent by the camera. Low signal on LANC-OUT is echoed back on LANC-IN. This is the first byte in the datagram, and, when inverted, the value reads 0b0010_1000 which is a “Special command to video camera” [28].

Communication with a computer is possible via both the USB and the UART interfaces. Currently, the USB function implements USB CDC ACM (Communications Device Class Abstract Control Model). CDC ACM is a vendor-independent protocol for emulating serial connections over USB [54]. To satisfy the immediate needs of the project, a custom plaintext protocol was created. In the future it is planned to use PTP. In order to make the device recognizable by the OS, a USB VID/PID (Vendor and Product IDs) is required. Because getting your own VID is costly [55], and the project is open-source, a PID from OpenMoko will be requested [56].

4.1.1 Results

The firmware was developed and taken into use. The source code is available under a free/open-source license and can a link can be found in Appendix 1.

To make full use of the firmware, higher level software should be able to communicate with it. This is typically mediated by a separate software layer often called “middleware”. Development of the middleware will be discussed in the next Section (4.2).

4.2 Middleware

To allow different control implementations to communicate with hardware in a flexible way, a middleware is required. While first prototypes were using simple named pipes, it later became clear that a better approach is required. The solution must provide discovery (nodes can be anywhere on the network, not necessarily on the same machine) and flexible message distribution.

To solve the discovery problem, ZeroConf [57] is used. ZeroConf allows programs to discover services on the network without using manual preconfiguration or relying on a configuration server. This means that nodes on the network can come and go freely, and they will be discoverable even if their IP address changes.

ZeroMQ [58] is used to address the message distribution problem. ZeroMQ is a messaging library that works without a dedicated message broker. Among others, ZeroMQ implements a publish–subscribe data distribution pattern.

During development, ROS (Robot Operating System) [59] was considered. However, unlike the proposed solution, ROS requires a master node which creates a single point of failure. Even if the network is down, a free-standing automated camera should be able to keep recording and following a gymnast. This is especially important in multi-camera setups, where some minimal communication between cameras is needed, but otherwise they should remain independent. By default, ROS does not allow having multiple masters on the same network. This problem was solved in ROS2 by eliminating the ROS master, but the current maturity of ROS2 [60] and its dependency on DDS (Data Distribution Service) would make installation and use of the system rather difficult. In contrast, the implemented solution is very lightweight and satisfies all current and future project needs.

A graph with all nodes of the network can be seen in Figure 12, and every node is explained in Table 2.

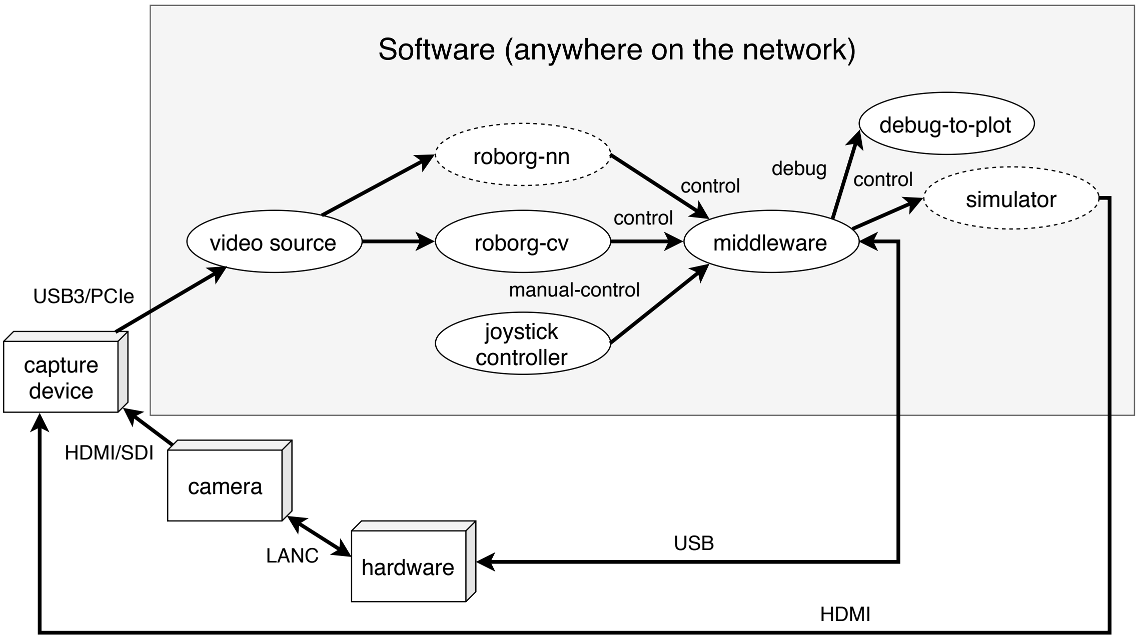


Figure 12. Nodes on the network, other software, hardware and their relations. Dashed lines indicate nodes that are out of scope of this thesis. Each node is described in Table 2.

Table 2. Description of nodes in Figure 12.

Unit	**Description**
roborg-cv	Image processing node as described in 5.1
joystick controller	Node for manual control using any USB joystick
middleware	Communication with the hardware and a control loop as described in 5.2
debug-to-plot	Simple real-time plotter of values using matplotlib [61]
video source	v4l2 video source on Linux [62]
hardware	Control board as described in 3.2
camera	Any compatible camera
capture device	Any device capable of capturing the feed from a given camera
roborg-nn	CNN and CNN+LSTM control as developed in [22]
simulator	Virtual pan-tilt-zoom simulator as developed in [22]

4.2.1 Results

This section discussed the developed middleware which allows different software nodes on the network to communicate. The source code is available under a free/open-source license and a link can be found in Appendix 1.

Together with the middleware, a simple node that reads the input from a joystick was created. This node can be used by a human operator to use the system as a remote PTZ camera. That alone makes the system fully usable, although not in an automated way.

To automate the system operation, some image processing software is needed. Development of this software will be discussed in the next chapter, specifically in Section 5.1.

5 Vision and control

In this chapter, software components related to the camera motion control logic are discussed. The work is divided into two parts: Computer vision and Control.

5.1 Computer vision

This section describes how image processing was approached. The developed software is robust and can reliably detect a gymnast in real life scenarios. The method is independent of the type or color of clothing worn by the gymnast, and can consistently filter out unrelated objects like backup apparatus placed on the edges of the floor area.

Figure 13 shows the visualization produced by the developed image processing software. The software outputs accurate horizontal position of the gymnast as well as the position of gymnast's contact with the floor.

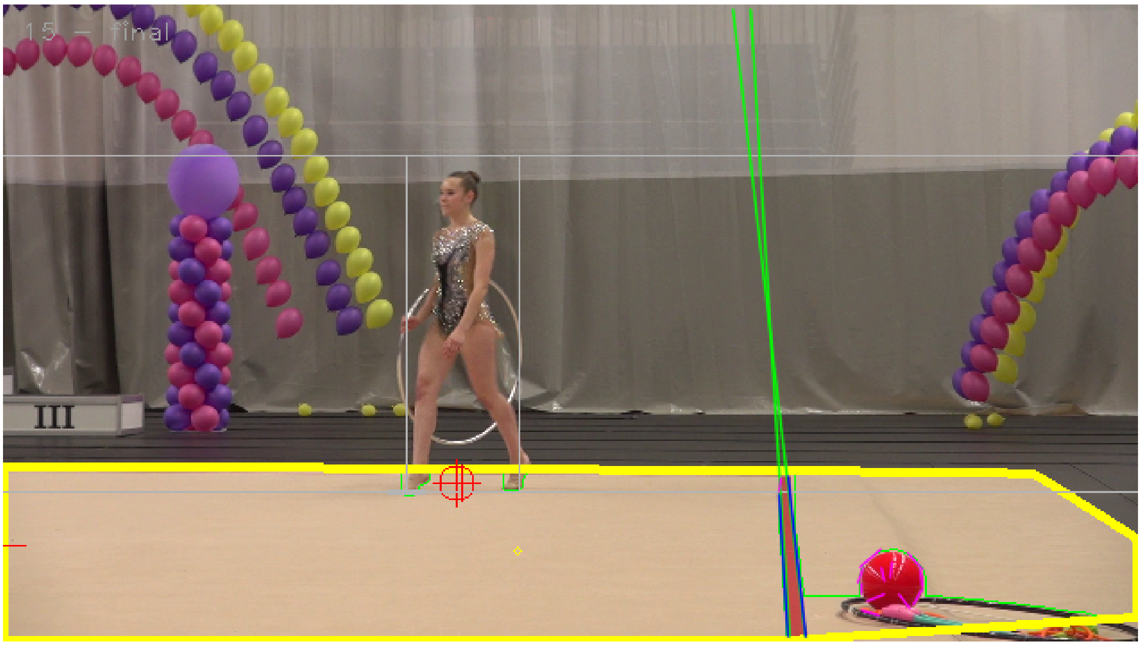


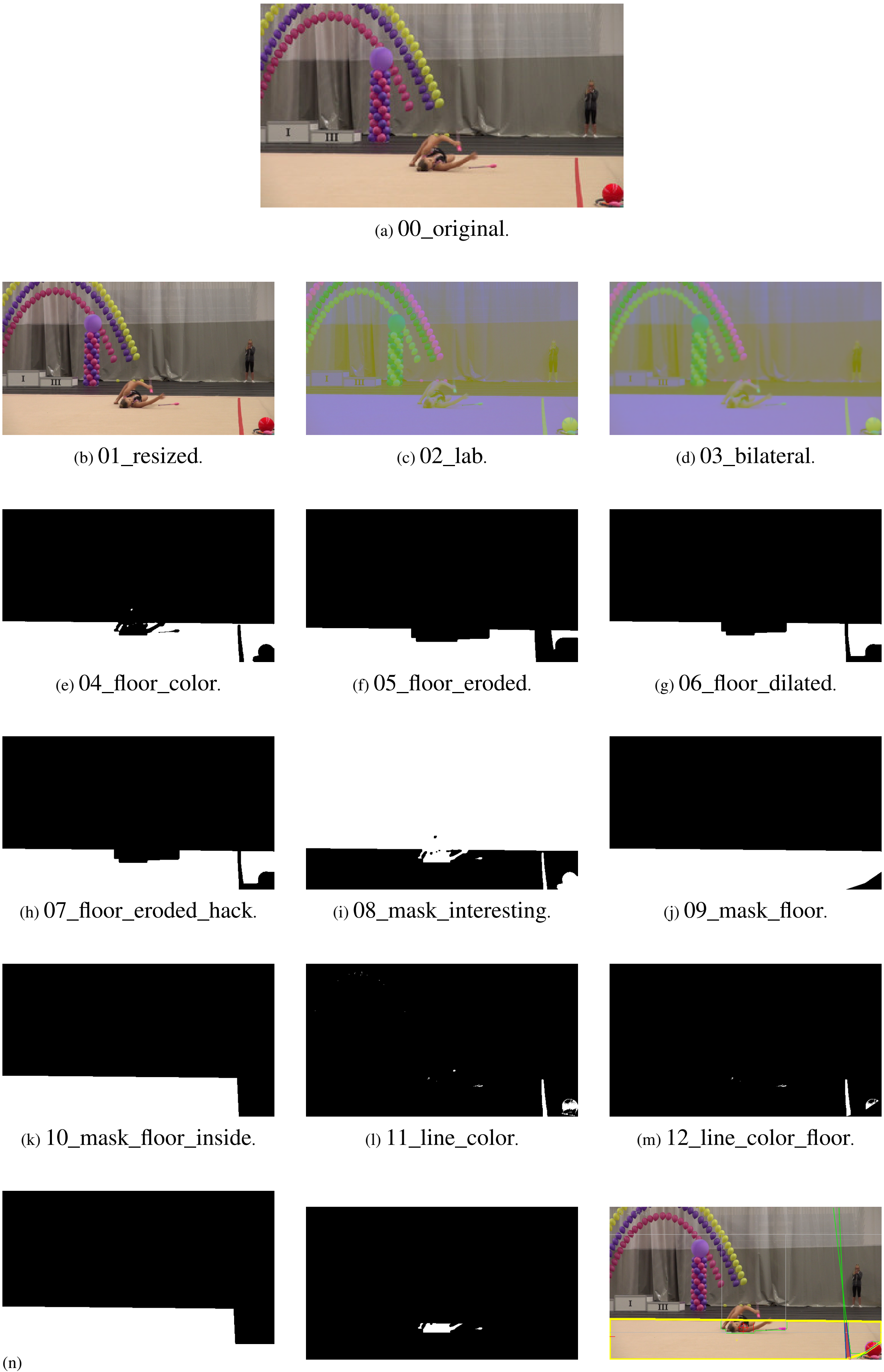
Figure 13. Output visualization of the image processing software. Red cross represents the center of mass of detected areas, and a gray horizontal line on the bottom indicates the bottommost position.

For working with frames, OpenCV [63] was used (specifically OpenCV Python bindings). OpenCV is a popular computer vision library that provides many building blocks for computer vision applications.

Normally, before working with images or video, it is required to perform camera calibration. Calibration is the process of figuring out camera extrinsic parameters (location and orientation), intrinsic parameters (focal length, principal point) and distortion coefficients (e.g. the extent of barrel distortion). If camera calibration is performed correctly, it should then be possible to accurately map 3D coordinates in space to pixel locations on captured frames, and vice versa. In most cases this makes image processing significantly easier. While camera extrinsic parameters in this system are always known[1], it is unfortunately not possible to get the current focal length because LANC only provides open-loop control over the zoom. Moreover, it is the zooming speed that is being controlled, and not position. Due to camera limitations, the same sequence of commands may result in a different final focal length, which makes tracking it by using the history of sent commands impossible. Because distortion coefficients are different at different focal lengths, they are also not known. Fortunately, camera calibration can be ignored in this system altogether. The system can work in terms of relative change from any starting focal length, and video cameras that are typically used to record rhythmic gymnastics events have very minimal barrel distortion. For example, line detection in most discussed cases will work as is, and even if a slightly curved line results in multiple line segments being detected, it does not impact the performance of the system.

Figure 14 demonstrates all steps of the pipeline. First of all, the image is resized from the original (Figure 14b). This is done purely for performance reasons. Generally, the goal is to use the smallest resolution that still has enough details to successfully perform all image processing operations in real time. In this case the resize operation goes from Full HD (1920x1080) to 576x324.

[1] Pan-tilt angles at any given point are controlled as accurately as possible for an open loop system. Issues will only arise if a step is skipped on the motors, which should never happen during normal operation.



(a) 00_original.

(b) 01_resized.

(c) 02_lab.

(d) 03_bilateral.

(e) 04_floor_color.

(f) 05_floor_eroded.

(g) 06_floor_dilated.

(h) 07_floor_eroded_hack.

(i) 08_mask_interesting.

(j) 09_mask_floor.

(k) 10_mask_floor_inside.

(l) 11_line_color.

(m) 12_line_color_floor.

(n) 13_mask_floor_inside_shrinked.

(o) 14_mask_target.

(p) 15_final.

Figure 14. Full image processing pipeline.

Next, the image is converted to another colorspace (Figure 14c). Many colorspaces exist, but when it comes to image processing, there are two notable categories: colorspaces that separate luminance and chrominance, and those that do not. Luminance represents how light or dark a pixel is (as in a black and white image), and chrominance defines the color. RGB and BGR (BGR is used by default in OpenCV) are examples of colorspaces where luminance is not separated from chrominance. While this is convenient for displaying images, for many other applications it is suboptimal. For example, YCbCr is a family of colorspaces used in video systems (Y is a luma signal, and Cb and Cr are chroma components) which allows saving bandwidth via chroma subsampling, making use of the fact that human the eye is less sensitive to changes in chroma [64], [65]. This is also the reason why resizing the image is justified for color-sensitive tasks, because frames with 4:2:2 or 4:2:0 chroma subsampling can be 0.5x downscaled without losing any chroma information. For image processing, it is common to rely on CIE-Lab colorspace [66].

Then, the frame is processed with a bilateral filter (Figure 14d). The goal is to remove as much noise as possible while preserving sharp edges in the image. Note that the bilateral filter should only be applied to colorspaces where Euclidean distances make sense, so if HSV colorspace was used, it would be more appropriate to use a bilateral filter earlier in the pipeline (before colorspace conversion). In case of CIE-Lab colorspace there is no such issue. Applying bilateral filter makes it much easier to carry out color-based operations further down the pipeline because less noise is present, while still preserving information that is required for edge detection.

After that, a mask representing the floor is extracted by thresholding the components of the frame in Lab colorspace (Figure 14e). The result of that operation is a binary image where all pixels have a value of 1 if the color falls within the specified window, or 0 otherwise. The window is specified in a config file, and the values will be different in different scenarios depending on the actual color of the floor, the lighting, camera settings, capture device in use and other factors. These values can be tweaked while the system is running using the GUI. Even though after this operation the outline of the gymnast is relatively obvious to a human eye, it cannot be used directly to find the position of a gymnast.

Detected masks may include small spots and regions that can be described as noise which needs to be filtered out. Erosion and dilation are two of the fundamental operations in image processing. They are typically executed one after another, in which case it is called opening (or closing if the order is reversed). When applied together, these operations can successfully remove noise and other unwanted features from images. During this step, the goal is to remove any small objects by assuming that the gymnast takes a relatively large area and will not be removed after aggressive opening (Figure 14f and Figure 14g). The resulting image does not have a detailed outline of a gymnast, but it is not needed to determine their position.

To get an accurate outline of the floor, a convex hull can be found for the floor mask. This alone proved to be somewhat unreliable in rare cases. To detect the whole floor area reliably, the color window has to be relatively wide. This has a downside because large areas with similar color will end up being detected as the floor. For example, it can be an issue when the ceiling of the building or the walls have a yellow-red tint and no texture. To combat this problem, a set of special-cased rules is applied to the floor mask (the result is shown in Figure 14h, although that step was a no-op for that particular frame). One of such tricks is to remove anything in the upper part of the frame, because it is not physically possible for the floor area to be located on the ceiling. Other potential artifacts can also be filtered out in a similar manner. For example, effects of vignetting can be combated by removing the bottom corners.

The floor used in rhythmic gymnastics is rectangular and cannot be of any other shape. Given that a rectangle always remains a convex shape no matter in which orientation it is observed, a convex hull of the detected mask will give an accurate and useful outline of the floor (Figure 14j). Lens distortion can affect the outline of the floor, but in practice it causes no significant problems.

While the base of the floor mask was detected by finding all pixels of some particular color, an opposite operation can be used to find objects of interest (Figure 14i). The resulting mask may includes unrelated pixels like the red line on the floor and backup apparatus that is placed beyond it. To successfully filter that out, two different cases need to be considered. In one case we only focus on the closest lines (usually two if the camera is placed at the corner of the floor area, see Figure 2). These lines are clearly visible even when the image is downscaled. To detect them, we can use the usual approach to extract pixels of the required color, and to remove some noise ahead of time we can AND these pixels with the detected floor area (Figure 14l and Figure 14m). By performing line detection (blue lines in Figure 14p) we can detect their position and orientation. In this work, I used a recent line detection algorithm proposed by Gioi et al. [67]. However, the lines are not always reliably detected along the full length, so for that reason we have to assume that they are straight and extend them to the image boundaries (green lines on Figure 14p). After that, the task is seemingly simple, because all that is required is to cut out everything beyond the line. The challenge is to figure out from which side of the line the operation has to be performed. In this work, I assume that the biggest visible area of the floor is the main area in the middle, and everything beyond that is not of our interest.

Therefore, by finding moments of the floor mask, we can then derive the “center of mass” and perform a cut in a direction from the center of mass to the line. This turned out to be a resilient mechanism that works in practice (Figure 14k). Another problem that needs to be accounted for is the lines that are further away. These are arguably more challenging, because the perspective makes them indistinguishable even at high resolutions. On the other hand, backup apparatus even at that distance causes a noticeable hole within the floor mask, potentially resulting in false-positive detection of a gymnast. To resolve that, I apply additional erosion in order to simply cut out the topmost part of the floor (Figure 14n). The downside of this approach is that if the subject stands on the furthest edge of the floor, the erosion may cut off the needed parts of the gymnast.

Finally, the gymnast can be reliably detected by AND-ing the resulting floor mask and the mask of objects of interest (Figure 14o). Afterwards, all of the important information is combined and visualized (Figure 14p) to make calibration easier for the operator.

5.1.1 Results

The image processing software that can reliably detect a subject of interest in rhythmic gymnastics was developed, tested and taken into use. The source code is available under a free/open-source license and a link can be found in Appendix 1. The described method can be easily applied to other sports by following the documented steps.

To make use of the developed image processing software, a control loop is needed. Its development will be discussed in the next section (5.2).

5.2 Control

For panning, the output of the image processing software is a value that corresponds to the horizontal position of a gymnast in a given frame. However, this value cannot be used directly (as in an end-to-end approach) and requires a control loop. Without a proper control loop, it is nearly impossible to keep a gymnast in frame, and without applying smoothing it is hard to achieve aesthetic camera movement. Moreover, practice showed that addition of some ad-hoc rules can significantly improve the visual performance of the system. This chapter discusses the PID controller, smoothing, and ad-hoc techniques used in the system.

For zooming, no sophisticated control or filtering is strictly required. Practice shows that slow changes in focal length produce a result that aesthetically looks good enough. Additionally, it seems that there is a significant range of zoom levels that provides visually correct framing. Therefore, this chapter will focus on panning only, although similar techniques are used for zoom.

5.2.1 PID

A proportional–integral–derivative controller (PID controller) is a widely used control algorithm. It consists of proportional, integral and derivative components which, when adjusted by setting the corresponding coefficients, can result in an optimal response. The output of a PID controller depends on the error value which is the difference between a desired setpoint and a measured process value.

In this system, the process value is the current position of a gymnast within a video frame. The setpoint is generally set to the middle of the frame, but this is not strictly the case (see 5.2.3). The system is well-behaved for using a PID controller: it is a minimum-phase system that is open loop stable. There is some non-linearity that is introduced by lens distortion, but its effect is fairly minimal in video cameras used for recording rhythmic gymnastics events. If necessary, lens distortion can be combated by calibrating the camera and undistorting the video stream (see 5.1). Additionally, acceleration profiles on the hardware can add to non-linearity, although this should only be prominent when commands to change the speed by a large amount are sent rapidly. The system does have a significant delay (around 150 ms), and that makes getting an optimal response challenging [68].

Unfortunately, there was no model of the system. Yet, by making a few observations and combining them, it is possible to get a natural feel to tune the coefficients manually.

In parallel to this work, a simulator that uses an ePTZ (Electronic PTZ) approach was developed by another student [22]. This work can be potentially used for tuning the PID controller. Video output of the simulator is just a small fraction of a full 4K frame (depending on the zoom level), meaning that line detection will not work properly. That being said, it should be possible to augment the output to make line detection easier. This will be researched in the future.

When tuning the coefficients, several observations about the system behavior and perceived smoothness of motion were made. First, any oscillations are undesired because they result in non eye-pleasing motion (human operators never generate oscillated movements). Second, a steady-state error is not an issue as long as the gymnast is roughly in the middle of the frame (human operators never keep dynamic subjects precisely at some position). Third, any fast movements or corrections should be avoided (this is one of the problems of fatigued human operators). Fourth, the hardware is capable of very fast movements, much faster than what is needed during operation, meaning that it is relatively hard to saturate the system.

Moreover, it is clear that different types of movement of a gymnast require different camera operation styles. For example, during periods of slow movement it is better to have an over-damped response, whereas faster rise time and possible overshoot can be preferred for fast motion. This means that gain scheduling is required. That is, instead of having just one PID controller, multiple controllers with different sets of coefficients can be used.

Using all this information, a set of coefficients was created (Table 3). All coefficients were tweaked on a live system by trial and error. Action of the derivative component is sensitive to measurement noise, and even though the measurements in this system are relatively stable, the movement of a gymnast and the way it is detected may appear erratic. This can cause the derivative part of the PID controller to come through as noise, and hence non-aesthetic visual results. When tuning the PID controller, it was noticed that very low coefficients (close to 0 or equal to 0) result in smoother, more eye-pleasing camera motion. Moreover, a coefficient of 0 (which essentially makes it a PI controller) does not have significant adverse effects.

A parallel form of a PID controller was used (3), and the base of the resulting implementation is shown in Figure 15.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\mathrm{d}t + K_d \frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{3}$$

Table 3. Gain scheduling and PID coefficients used in this work. The rules are processed from top to bottom until appropriate PID coefficients are found.

Threshold	K_P	K_I	K_D	**Description**
< 0.15	-			Deadband, setpoint is reset to zero, error is ignored
< 0.20	0.30	0.00	0.00	The gymnast is still in frame, slow proportional-only control for fluid motion
< 0.30	0.45	0.00	0.00	
< 0.40	0.60	0.00	0.00	
$\geqslant 0.40$	-			Setpoint is moved to add visual lead room
< 0.50	0.70	0.10	0.00	Medium-speed movements, integral control is added
$\geqslant 0.50$	-			Zooming out is forced
other	1.00	0.20	0.00	Fast movements

```
whenever interval(iteration-time) {
   error = last-reading-with-smoothing - setpoint;
   integral += error × iteration-time;
   derivative = (error - error-prior) ÷ iteration-time;
   p = KP × error;
   i = KI × integral;
   d = KD × derivative;
   output = p + i + d;
   error-prior = error;
   ...  // gain scheduling, smoothing, ad-hoc rules, etc.
}
```

Figure 15: Pseudocode of the PID algorithm used in this thesis. Smoothing and ad-hoc rules are not shown.

5.2.2 Smoothing

Small changes in the scene can lead to relatively large changes in the output of the image processing software. To make the resulting camera motion aesthetic, some smoothing is required.

Moreover, the system should behave in a stable way even if the input has drastic unpredicted changes. For example, a person walking in front of a camera (blocking the view) may make the gymnast detection produce arbitrary values. Also, the way image processing works in this system allows for short periods of time with no output, for instance when the gymnast performs a jump. All of that needs to be handled during smoothing.

The goal is to filter out any high frequency spikes and preserve a smooth low frequency signal. In other words, a low-pass filter is needed. Many implementations of low-pass filters exist, and many of them can be used for this application. However, achieving eye-pleasing results while still keeping the system responsive is challenging. Another challenge is to make the camera movement pleasing to humans, because the way they perceive it is hard to express in technical terms. At first, a simple moving average filter was tested. While wide windows resulted in very smooth motion, the system was not punchy enough. That is, if a gymnast started running from a standstill position, the system was unable to react in a timely manner and as a result a gymnast would get out of frame. Smaller window sizes did not have that issue, but the resulting camera movement was jerky and not visually pleasing. Since I was unable to find a good balance between smooth motion and camera responsiveness, I decided to try exponential smoothing (Equation 4). Exponential smoothing is also known as exponential moving average.

$$S_t = \alpha \cdot X_t + (1 - \alpha) \cdot S_{t-1} \tag{4}$$

In exponential smoothing, the most recent values have more effect on the output. On the other hand, older values quickly diminish in their influence (exponential decay). Depending on the coefficient, this can result in a nice balance between smoothness and responsiveness.

While moving average is a finite impulse response (FIR) filter and exponential smoothing is an infinite impulse response (IIR) filter, the difference between FIR and IIR filters is unlikely to have any effect on the perceived smoothness of motion. As long as the filter has a large enough window and values do not drop out abruptly (by falling out of the window), there should be no noticeable adverse effects.

It is worth noting that most mechanical systems will dampen high frequency signals even when fed with unfiltered input. This is the result of the nature of motion (desired speed can never be achieved instantly), and most hardware systems will have their own processing in terms of acceleration profiles. In case of this particular system, there is indeed a maximum acceleration value that can be controlled from software. This value is kept at a relatively low level to make long-term use of the hardware safe, and also to maintain motion smoothness even if spikes happen to pass through the filtering.

In the end, to filter the output of the image processing software, exponential smoothing was used. For panning, I used $\alpha = 0.3$ with sampling time of 25 frames per second. Sampling time corresponds to the framerate of the image processing, and α may need to be changed (or scaled) if a different framerate is used.

5.2.3 Additional ad-hoc rules

Human operators do not normally compensate for minuscule errors in the desired camera orientation. If a gymnast is approximately at the right place within a frame, no camera movement is needed. Therefore, a window of acceptable values was implemented and used for decaying the error when the values are within that window. This works similarly to a commonly used deadband technique, where the frequency of activation within a deadband is reduced. Technically, this shifts the desired setpoint to the position of the process value. Similar logic is used for integral anti-windup. To prevent overshoot during fast motions, the integral is aggressively reduced once the gymnast is within the acceptable window.

Because gymnasts often switch between types of motion that are inherently different (balances, leaps, etc.), gain scheduling is used to adapt to changes on the scene by changing the coefficients of the PID controller. Most of the logic is based on crossing certain error thresholds, in which case PID coefficients are gradually ramped to corresponding preset levels.

Lead room (nose room) compensation is another solution that is closely related to gain scheduling. By detecting when the gymnast starts moving, it is possible to change the setpoint by shifting it into the direction opposite to the movement. This serves two purposes. One is that well-composed shots should generally have space in the direction the subject is moving [69]. Another is that because of the latency, a gymnast can get out of frame if there is not enough lead room. In other words, changing the setpoint allows getting more aesthetic results and minimize the occurrence of lost tracking.

5.2.4 Results

Implementation of the controller completes the system, meaning that when all parts of the project are now combined, the system can be used to perform automated filming of rhythmic gymnastics events. Performance of the system will be evaluated in Chapter 6.

6 Evaluation

The implemented system was used in production at several Estonian events. Its use varied because the system was at different stages of completeness. The full solution was finally utilized at the Noorus Winter Cup 2019 event. The competition was held on three consecutive days from 2019-02-15 till 2019-02-17, each day lasting more than 10 hours. The system worked throughout all three days. Figure 17 shows an example of a recorded routine from that event.

To demonstrate the difference between a human operator and the created system, two cameras were placed side by side (Figure 16) to record routines at a different event. The left camera was operated by a human, and the right camera was handled by the software. Figure 18 shows frames taken from both videos side by side. Most of the shots are indistinguishable, and the difference can only be seen during fast movement because the system does not have enough predicting power. Proper evaluation method with well-defined metrics will be used in Section 6.1.

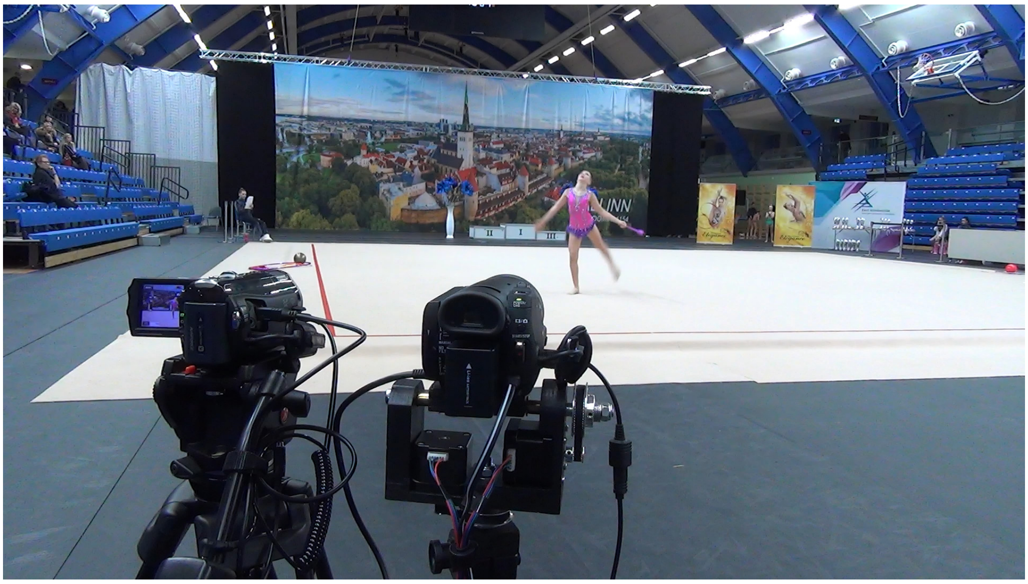


Figure 16. Photo of the setup used for performance evaluation. Left camera is operated manually, right camera is automated.

The developed hardware was also used in another Master's thesis [22] where an end-to-end deep learning approach was used. In that work, 15 routines recorded using the hardware developed in this thesis were evaluated to measure the performance of the system as a whole. A third of videos were recorded perfectly with no tracking loss. Most of the cases when the tracking was lost seem to be caused by deficiencies of the dataset used for training and not the hardware limits. This means that the hardware is capable of satisfying required needs.

The full system I developed was used to record performances for commercial purposes during several other rhythmic gymnastics events. During these events two cameras were used, each placed at a different corner. One camera was operated by a human, and the other camera was controlled by the software I developed. Best camera angles were used in final videos. This has proved that the system is a useful addition to a multi-camera setup, especially when recorded videos from a human-operated camera had less favorable angles.

The cost of a single full prototype was less than 70 € in parts, where roughly 25 % of the cost was in the PCB and electronic components, 40 % in stepper motors and drivers, and the remaining 35 % in other hardware (pulleys, belts, shafts, nuts, etc.). The cost can be decreased significantly if parts are ordered in bulk.

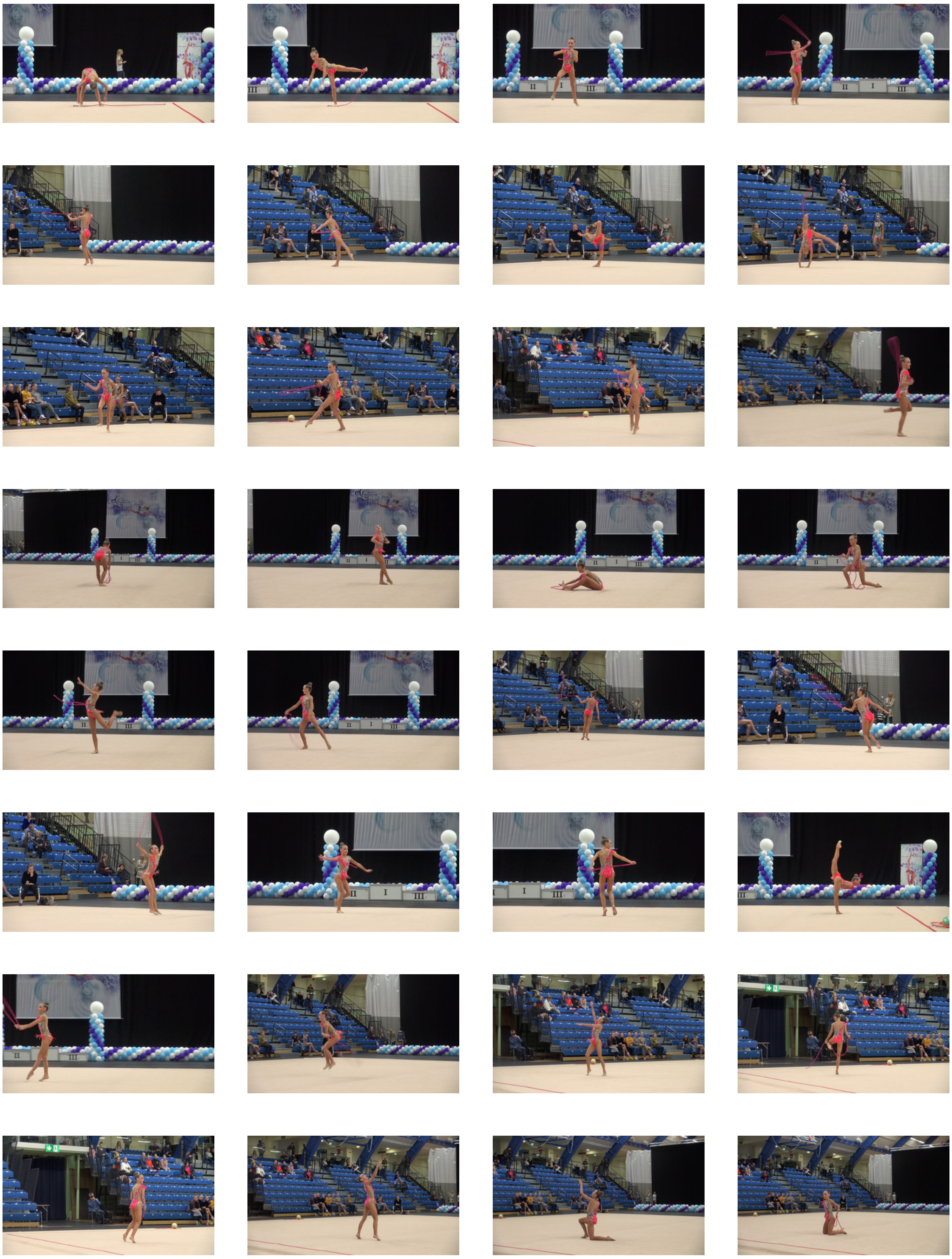


Figure 17. Example of the whole recorded routine, one image every 3 seconds.

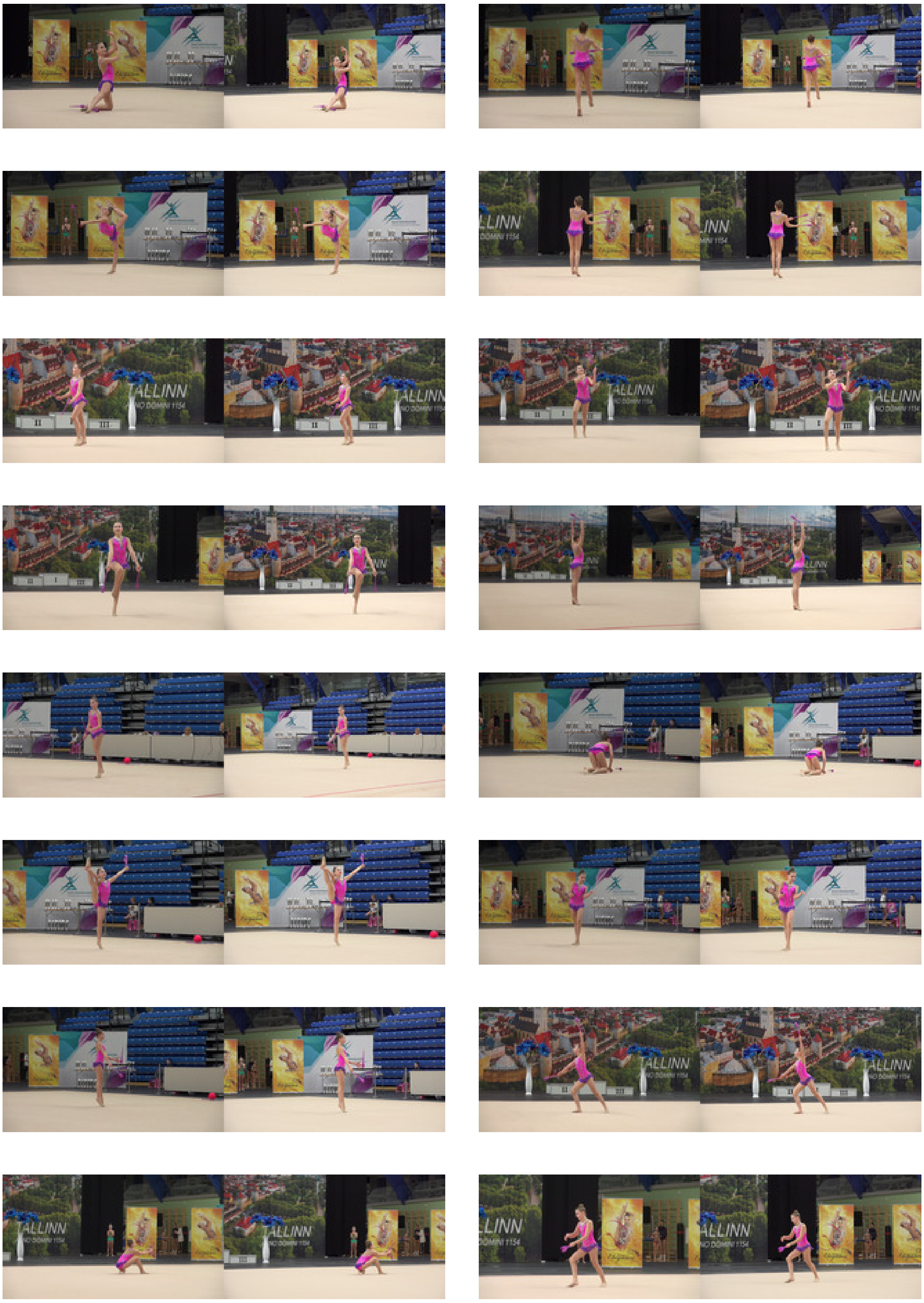


Figure 18. Comparison to a human operator, one image every 6 seconds. Right side shows frames from the developed system, left side shows frames taken from a video recorded by a human operator.

6.1 Evaluation of autonomy

To evaluate the autonomy of the system, a video of 15 subsequent performances was analyzed. Each instance of manual intervention was recorded. The autonomy of the system was calculated using Equation (5):

$$Autonomy = \frac{Total\ time - Manual\ operation\ time}{Total\ time} \tag{5}$$

The results of the evaluation are presented in Table 4. In the table, every case of manual intervention is shown on a separate row. Additionally, other events are listed to give the reader an understanding of what was happening.

Table 4. Evaluation of autonomy.

Timestamp	Routine	Action
00:00:03	1 (ball)	Gymnast is detected
00:00:10	1 (ball)	Start of the performance
00:01:40	1 (ball)	End of the performance
00:01:51	–	Automatic dead sequence[1] initiated
00:01:59	–	Manual intervention
00:02:02	2 (rope)	Gymnast is detected
00:02:15	2 (rope)	Start of the performance
00:03:45	2 (rope)	End of the performance
00:03:58	–	Automatic dead sequence initiated
00:04:04	3 (rope)	Gymnast is detected
00:04:11	3 (rope)	Start of the performance
00:05:41	3 (rope)	End of the performance
00:05:55	–	Manual intervention
00:05:59	4 (rope)	Gymnast is detected
00:06:10	4 (rope)	Start of the performance
00:07:40	4 (rope)	End of the performance
00:07:55	–	Automatic dead sequence initiated
00:07:59	5 (rope)	Gymnast is detected
00:08:11	5 (rope)	Start of the performance
00:09:39	5 (rope)	End of the performance
00:09:53	–	Automatic dead sequence initiated
00:09:56	6 (rope)	Gymnast is detected
00:10:08	6 (rope)	Start of the performance
00:11:39	6 (rope)	End of the performance
00:11:51	–	Automatic dead sequence initiated
00:11:55	7 (rope)	Gymnast is detected

[1] The system starts panning left-right in order to find a gymnast

Table 4. Evaluation of autonomy (continued).

00:12:06	7 (rope)	Start of the performance
00:13:35	7 (rope)	End of the performance
00:13:50	–	Manual intervention
00:13:52	8 (rope)	Gymnast is detected
00:14:00	8 (rope)	Start of the performance
00:15:30	8 (rope)	End of the performance
00:15:41	–	Automatic dead sequence initiated
00:15:57	9 (rope)	Gymnast is detected
00:16:07	9 (rope)	Start of the performance
00:17:39	9 (rope)	End of the performance
00:17:54	10 (rope)	Gymnast is detected
00:18:01	10 (rope)	Start of the performance
00:19:31	10 (rope)	End of the performance
00:19:43	–	Automatic dead sequence initiated
00:19:58	11 (rope)	Manual intervention
00:20:00	11 (rope)	Gymnast is detected
00:20:02	11 (rope)	Start of the performance
00:21:34	11 (rope)	End of the performance
00:21:50	–	Automatic dead sequence initiated
00:21:57	12 (rope)	Gymnast is detected
00:22:05	12 (rope)	Start of the performance
00:23:35	12 (rope)	End of the performance
00:23:51	–	Automatic dead sequence initiated
00:23:54	13 (rope)	Gymnast is detected
00:24:05	13 (rope)	Start of the performance
00:25:35	13 (rope)	End of the performance
00:25:52	–	Automatic dead sequence initiated
00:25:57	–	Manual intervention
00:26:00	14 (rope)	Gymnast is detected
00:26:08	14 (rope)	Start of the performance
00:27:38	14 (rope)	End of the performance
00:27:50	–	Automatic dead sequence initiated
00:27:53	15 (clubs)	Gymnast is detected
00:28:00	15 (clubs)	Start of the performance
00:29:28	15 (clubs)	End of the performance

The calculated autonomy is 99.2 %. Events of lost tracking were usually associated with long periods of time when there was no gymnast on the floor (between two performances). The system detected that there was no gymnast and started panning left-right in an attempt to restore tracking. This behavior is referred to as automatic dead sequence. However,

Table 5. Evaluation of motion and framing quality. b – Better than human, s – Similar to human, w – Worse than human.

	Section	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Video	Eye-pleasing	s	s	s	s	s	b	s	s	w	s	s	s	w	s	s	s	s	s
1	Fluid	s	s	s	s	w	s	s	b	s	s	s	s	s	s	w	s	s	s
Video	Eye-pleasing	w	s	w	s	s	s	s	s	s	s	b	b	w	s	s	s	s	s
2	Fluid	s	s	w	s	s	s	s	s	s	w	s	s	w	s	s	s	s	s
Video	Eye-pleasing	s	b	s	s	w	s	w	s	s	s	b	s	s	s	s	w	s	s
3	Fluid	s	s	s	s	w	w	s	s	s	s	s	s	s	s	s	w	s	s
Video	Eye-pleasing	s	s	s	s	w	s	s	s	s	s	s	s	s	s	s	s	s	s
4	Fluid	s	s	s	s	s	w	s	b	s	s	s	s	s	s	s	s	s	s
Video	Eye-pleasing	s	s	s	w	w	s	s	s	s	s	w	w	s	s	s	s	s	s
5	Fluid	s	w	s	s	w	s	w	w	s	s	s	s	s	s	s	s	s	s

when the next gymnast actually comes to the floor, the system may not be oriented the right way. For that reason, the human operator performed corresponding corrections.

It is fair to conclude that the system can autonomously record individual rhythmic gymnastics performances as long as the gymnast is on the competition area. There were mixed results when the system was recording between performances, and this aspect of the system will be improved in the future.

6.2 Evaluation of perceived framing and camera motion

To evaluate if the achieved camera motion and framing are eye-pleasing, pairs of videos were compared. Each pair consists of a video recorded by a human operator and a video recorded by the system. This setup was already shown in Figure 16. Each video was split into video sections of 5 seconds. Each section pair was then evaluated by an expert (former rhythmic gymnastics coach) based on two metrics. First metric is how eye-pleasing the framing is, meaning how well the gymnast is positioned in the camera view. Second metric is how fluid the motion is during the section. There are three options for each metric: better than human, worse than human, and similar to human. Results of the evaluation are presented in Table 5.

In total, 5.6 % of sections are better than human, 80.0 % are similar to human and 14.4 % are worse than human. In case of camera motion fluidity metric, 2.2 % are better than human, 83.3 % are similar to human and 14.4 % are worse than human.

It is worth mentioning that the human operator was not fatigued, meaning that their performance was close to ideal. Unfortunately, I was unable to find a human operator who would agree to compete against the automated system after 10 hours of work.

According to the results, the system produces output that is comparable in quality to human performance.

6.3 Conclusion and future work

Results of the evaluation showed that the developed system can be successfully used for recording rhythmic gymnastics competitions. Not only the autonomy of the system is very high (99.2 %), but also the produced output is eye-pleasing. The evaluation showed that 85.6 % of videos recorded by the system had the same or better framing than the videos recorded by a human operator. In case of camera motion fluidity metric, the result was also 85.6 %. The biggest differences in framing and fluidity can be seen during fast movement because the system is not predictive enough. Thus, when a gymnast starts moving rapidly, the system reacts with a delay and compensates later by applying higher speed. As a result, at the beginning of a fast motion, framing is not perfect and fluidity is also affected when the system attempts not to lose tracking.

Although the system has already been used for commercial purposes, there are still several aspects to work on.

Currently, the system works well as long as the gymnast is on the floor area. However, at the end of a performance, gymnasts leave the area and there may be a period of time when nothing is detected. Normally it is not a problem because the next gymnast will typically come from the same direction, so the system will seamlessly switch to tracking them. When this does not happen, it is interpreted as if the tracking was lost accidentally, so the system starts moving left-right in order to restore it. While functionally it is not a bad behavior, it does not result in aesthetic videos and the exact moment when the gymnast enters the area is sometimes not captured.

Group performances include more than one gymnast performing on the floor area (5 in Rhythmic Group Gymnastics and 6-10 in Aesthetic Group Gymnastics). In such cases it is important to detect all gymnasts and keep all of them in frame. The system can already keep multiple gymnasts centered, but it does not know if there is anyone outside of the frame.

Other types of moving cameras can also be considered. Dollies are often used for rhythmic gymnastics, and allow angles that would not be achievable otherwise.

Even though this work focused on rhythmic gymnastics, the same hardware can be used for many other sports. The software would need to be changed, although most principles used will apply to all sports that include a single athlete performing on a standard floor area.

7 Summary

The aim of the thesis was to figure out if a low-cost solution for automated filming of local rhythmic gymnastics events can be created. As a result, a complete system was developed consisting of a mechanical design, PCB design, firmware, middleware, image processing software and a control algorithm.

The designed system was assembled and used to record several competitions. To demonstrate the results, comparison to a human operator was made. It showed that the produced camera motion is good enough for the intended purpose and is comparable to human performance. The developed hardware was also successfully used in another thesis where deep learning was utilized instead of image processing.

Autonomy evaluation showed that the tracking works perfectly as long as a gymnast is on the competition area. All performances used to evaluate autonomy of the system were recorded with no tracking loss. Events of lost tracking were usually associated with long periods of time when there was no gymnast on the floor. This aspect of the system will be improved in the future. Even with the events of lost tracking between two performances, the calculated autonomy of the system was measured to be 99.2 %. In addition, the system was able to produce eye-pleasing results. More specifically, framing and fluidity of motion were as good or better than the performance of a well-rested human operator in 85.6 % of cases.

All source files are published under a free/open-source license.

References

[1] G. Thomas, R. Gade, T. B. Moeslund, P. Carr, and A. Hilton, “Computer vision for sports: Current applications and research topics”, en, *Computer Vision and Image Understanding*, vol. 159, pp. 3–18, 2017-06, ISSN: 10773142. DOI: 10.1016/j.cviu.2017.04.011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1077314217300711 (visited on 2019-04-25).

[2] J. Chen and P. Carr, “Autonomous camera systems : A survey”, 2014.

[3] T. Haraguchi, T. Taki, and J. Hasegawa, “Automated tracking of a figure skater by using PTZ cameras”, F. Remondino, M. R. Shortis, and S. F. El-Hakim, Eds., San Diego, CA, 2009-08, 74470F. DOI: 10.1117/12.825463. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.825463 (visited on 2019-04-25).

[4] H. Pidaparthy and J. Elder, “Keep Your Eye on the Puck: Automatic Hockey Videography”, in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa Village, HI, USA: IEEE, 2019-01, pp. 1636–1644, ISBN: 9781728119755. DOI: 10.1109/WACV.2019.00179. [Online]. Available: https://ieeexplore.ieee.org/document/8658666/ (visited on 2019-04-25).

[5] Upendra Rao M. and U. C. Pati, “A novel algorithm for detection of soccer ball and player”, in *2015 International Conference on Communications and Signal Processing (ICCSP)*, Melmaruvathur, India: IEEE, 2015-04, pp. 0344–0348, ISBN: 9781479980819. DOI: 10.1109/ICCSP.2015.7322903. [Online]. Available: http://ieeexplore.ieee.org/document/7322903/ (visited on 2019-04-25).

[6] Y. Ariki, S. Kubota, and M. Kumano, “Automatic Production System of Soccer Sports Video by Digital Camera Work Based on Situation Recognition”, *(:unav)*, 2006. DOI: 10.1109/ism.2006.37. [Online]. Available: http://ieeexplore.ieee.org/document/4061270/ (visited on 2019-04-25).

[7] S. Daigo and S. Ozawa, “Automatic pan control system for broadcasting ball games based on audience’s face direction”, en, in *Proceedings of the 12th annual ACM international conference on Multimedia - MULTIMEDIA ’04*, New York, NY, USA: ACM Press, 2004, p. 444, ISBN: 9781581138931. DOI: 10.1145/1027527.1027634. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1027527.1027634 (visited on 2019-04-25).

[8] Z. Wu and R. J. Radke, “Keeping a Pan-Tilt-Zoom Camera Calibrated”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1994–2007, 2013-08, ISSN: 0162-8828, 2160-9292. DOI: 10.1109/TPAMI.2012.250. [Online]. Available: http://ieeexplore.ieee.org/document/6361403/ (visited on 2019-04-25).

[9] *OpenCV: Camera Calibration*. [Online]. Available: https://docs.opencv.org/3.4.3/dc/dbb/tutorial_py_calibration.html (visited on 2019-05-11).

[10] P. Carr, M. Mistry, and I. Matthews, “Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording”, en, in *Proceedings of the 21st ACM international conference on Multimedia - MM ’13*, Barcelona, Spain: ACM Press, 2013, pp. 193–202, ISBN: 9781450324045. DOI: 10.1145/2502081.2502086. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2502081.2502086 (visited on 2019-04-25).

[11] J. Chen and P. Carr, “Mimicking Human Camera Operators”, in *2015 IEEE Winter Conference on Applications of Computer Vision*, Waikoloa, HI, USA: IEEE, 2015-01, pp. 215–222, ISBN: 9781479966837. DOI: 10.1109/WACV.2015.36. [Online]. Available: http://ieeexplore.ieee.org/document/7045890/ (visited on 2019-04-25).

[12] V. R. Gaddam, R. Eg, R. Langseth, C. Griwodz, and P. Halvorsen, “The Cameraman Operating My Virtual Camera is Artificial: Can the Machine Be as Good as a Human?”, en, *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 11, no. 4, pp. 1–20, 2015-06, ISSN: 15516857. DOI: 10.1145/2744411. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2788342.2744411 (visited on 2019-04-25).

[13] D. Kato, T. Katsuura, and H. Koyama, “Automatic Control of a Robot Camera for Broadcasting Based on Cameramen’s Techniques and Subjective Evaluation and Analysis of Reproduced Images.”, en, *Journal of PHYSIOLOGICAL ANTHROPOLOGY and Applied Human Science*, vol. 19, no. 2, pp. 61–71, 2000, ISSN: 1345-3475, 1347-5355. DOI: 10.2114/jpa.19.61. [Online]. Available: http://joi.jlc.jst.go.jp/JST.JSTAGE/jpa/19.61?from=CrossRef (visited on 2019-04-25).

[14] E. Foote, P. Carr, P. Lucey, Y. Sheikh, and I. Matthews, “One-man-band: A touch screen interface for producing live multi-camera sports broadcasts”, en, in *Proceedings of the 21st ACM international conference on Multimedia - MM ’13*, Barcelona, Spain: ACM Press, 2013, pp. 163–172, ISBN: 9781450324045. DOI: 10.1145/2502081.2502092. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2502081.2502092 (visited on 2019-04-25).

[15] R. Newbury, A. Cosgun, M. Koseoglu, and T. Drummond, “Learning to Take Good Pictures of People with a Robot Photographer”, en, 2019-04. [Online]. Available: https://arxiv.org/abs/1904.05688v1 (visited on 2019-04-25).

[16] M. P. Díaz-Pereira, I. Gómez-Conde, M. Escalona, and D. N. Olivieri, “Automatic recognition and scoring of olympic rhythmic gymnastic movements”, en, *Human Movement Science*, vol. 34, pp. 63–80, 2014-04, ISSN: 01679457. DOI: 10.1016/j.humov.2014.01.001. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167945714000025 (visited on 2019-05-14).

[17] *PTZ Cameras*. [Online]. Available: https://pro.sony/ue_US/products/ptz-network-cameras (visited on 2019-05-13).

[18] *Panasonic Professional PTZ Cameras*. [Online]. Available: https://na.panasonic.com/us/panasonic-professional-ptz-cameras (visited on 2019-05-13).

[19] *JVC Pro Product Overview Page*. [Online]. Available: http://pro.jvc.com/prof/attributes/features.jsp?model_id=MDL102404 (visited on 2019-05-13).

[20] *SMT (SportsMEDIA Technology)*. [Online]. Available: https://www.smt.com/ (visited on 2019-05-14).

[21] K. Zamanian, *Cinematography Tip: Creating the Illusion of Speed*, 2015-11. [Online]. Available: https://www.premiumbeat.com/blog/creating-the-illusion-of-speed/ (visited on 2019-05-17).

[22] M. Voronina, “Automated Camera Motion Control for Rhythmic Gymnastics Using Deep Learning”, 2019, [Online]. Available: https://github.com/RGVID-EU/RoboRG-Docs/releases/tag/1.0.0.

[23] *2017–2020 Code of Points*. [Online]. Available: https://www.gymnastics.sport/publicdir/rules/files/en_RG%20CoP%202017-2020%20with%20Errata%20Dec.%2017.pdf (visited on 2019-05-13).

[24] *OpenMoCo | The Open-Source Photographic Motion-Control Community*, 2013-04. [Online]. Available: https://web.archive.org/web/20130405055622/http://www.openmoco.org/ (visited on 2019-05-14).

[25] *Dynamic Perception*. [Online]. Available: https://www.dynamicperception.com/ (visited on 2019-05-14).

[26] *MultiCamzilla - A Camera Crew at Your Fingertips*. [Online]. Available: https://multicamzilla.com/ (visited on 2019-05-14).

[27] *BescorMP-101 Motorized Pan & Tilt Head*. [Online]. Available: https://www.bhphotovideo.com/c/product/64399-REG/Bescor_MP101_MP_101_Motorized_Pan_Head.html (visited on 2019-05-14).

[28] M. Boehmel, *The SONY LANC protocol*, 2018. [Online]. Available: http://www.boehmel.de/lanc.htm (visited on 2019-04-23).

[29] *gPhoto - gPhoto Home*. [Online]. Available: http://www.gphoto.org/ (visited on 2019-05-09).

[30] *ISO 15740:2013*, en. [Online]. Available: http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/36/63602.html (visited on 2019-05-14).

[31] *Calculating the Angle of View: When Theory Meets Practice • Points in Focus Photography*, 2018-01. [Online]. Available: https://www.pointsinfocus.com/learning/cameras-lenses/calculating-angle-view-theory-meets-practice/ (visited on 2019-05-17).

[32] *Technical Information: Ball bearing types, selection factors, and bearing load data from SDP/SI*. [Online]. Available: https://www.sdp-si.com/Design-Data/Ball-Bearings.php (visited on 2019-05-10).

[33] *Build A RepRap - RepRap*. [Online]. Available: https://reprap.org/wiki/Build_A_RepRap (visited on 2019-05-16).

[34] *ODrive*. [Online]. Available: https://odriverobotics.com/ (visited on 2019-05-18).

[35] *Microstepping Tutorial - ZaberWiki*. [Online]. Available: https://www.zaber.com/wiki/Tutorials/Microstepping (visited on 2019-05-16).

[36] *How Accurate Is Microstepping Really?*, 2016. [Online]. Available: https://hackaday.com/2016/08/29/how-accurate-is-microstepping-really/ (visited on 2019-04-29).

[37] *FFmpeg*. [Online]. Available: https://ffmpeg.org/ (visited on 2019-05-15).

[38] *PartDesign Workbench - FreeCAD Documentation*. [Online]. Available: https://www.freecadweb.org/wiki/PartDesign_Workbench (visited on 2019-04-29).

[39] *Sketcher Workbench - FreeCAD Documentation*. [Online]. Available: https://www.freecadweb.org/wiki/Sketcher_Workbench (visited on 2019-04-29).

[40] *TMC2130-LA DATASHEET*. [Online]. Available: https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC2130_datasheet.pdf (visited on 2019-05-13).

[41] *EZ-USB® FX2LP™ USB Microcontroller High-Speed USB Peripheral Controller*. [Online]. Available: https://www.cypress.com/file/138911/download (visited on 2019-05-13).

[42] *Sigrok*. [Online]. Available: https://sigrok.org/ (visited on 2019-05-04).

[43] *Fx2grok-tiny: An 11x11mm FX2 based Open Hardware logic analyzer for use with sigrok | The ever-expanding world of sigrok*. [Online]. Available: https://www.sigrok.org/blog/fx2grok-tiny-11x11mm-fx2-based-open-hardware-logic-analyzer-use-sigrok (visited on 2019-05-04).

[44] *PCB Design Guidelines For Reduced EMI*. [Online]. Available: http://www.ti.com/lit/an/szza009/szza009.pdf (visited on 2019-05-04).

[45] *KiCad EDA*. [Online]. Available: http://kicad-pcb.org/ (visited on 2019-05-09).

[46] *Open source ARM Cortex-M microcontroller library*, original-date: 2012-04-18T16:51:38Z, 2019-05. [Online]. Available: https://github.com/libopencm3/libopencm3 (visited on 2019-05-13).

[47] texane, *Stm32 discovery line linux programmer*, original-date: 2011-01-14T09:19:12Z, 2019-05. [Online]. Available: https://github.com/texane/stlink (visited on 2019-05-10).

[48] *stm32flash*, en. [Online]. Available: https://sourceforge.net/projects/stm32flash/ (visited on 2019-05-10).

[49] *Open On-Chip Debugger*. [Online]. Available: http://openocd.org/ (visited on 2019-05-10).

[50] B. Sencer, Y. Altintas, and E. Croft, "Feed optimization for five-axis CNC machine tools with drive constraints", en, *International Journal of Machine Tools and Manufacture*, vol. 48, no. 7-8, pp. 733–745, 2008-06, ISSN: 08906955. DOI: 10.1016/j.ijmachtools.2008.01.002. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0890695508000059 (visited on 2019-05-04).

[51] *Affordable Industrial Grade Motion Control*, original-date: 2010-07-12T03:00:43Z, 2019-04. [Online]. Available: https://github.com/synthetos/TinyG (visited on 2019-05-04).

[52] *Marlin, 6th-order jerk-controlled motion planning in real-time*. [Online]. Available: https://github.com/MarlinFirmware/Marlin/pull/10337/commits/a29adde5c0706a572261c40a02c61e1c966928a4#diff-a7c736c674fcde8ddf7bf0e86181e4c0 (visited on 2019-05-04).

[53] STMicroelectronics, *RM0008 Reference manual*. [Online]. Available: https://www.st.com/resource/en/reference_manual/cd00171190.pdf (visited on 2019-05-13).

[54] *Class definitions for Communication Devices 1.2 | USB-IF*. [Online]. Available: https://www.usb.org/document-library/class-definitions-communication-devices-12 (visited on 2019-05-19).

[55] *Getting a Vendor ID | USB-IF*. [Online]. Available: https://www.usb.org/getting-vendor-id (visited on 2019-05-19).

[56] *USB Product IDs - Openmoko*. [Online]. Available: http://wiki.openmoko.org/wiki/USB_Product_IDs (visited on 2019-05-10).

[57] *Zero Configuration Networking (Zeroconf)*. [Online]. Available: http://www.zeroconf.org/ (visited on 2019-05-09).

[58] *Distributed Messaging - zeromq*. [Online]. Available: http://zeromq.org/ (visited on 2019-05-09).

[59] *ROS.org | Powering the world's robots*. [Online]. Available: https://www.ros.org/ (visited on 2019-05-09).

[60] *Why ROS 2.0?* [Online]. Available: https://design.ros2.org/articles/why_ros2.html (visited on 2019-05-09).

[61] *Matplotlib: Python plotting — Matplotlib 3.1.0 documentation*. [Online]. Available: https://matplotlib.org/ (visited on 2019-05-19).

[62] *Video for Linux Two API Specification*. [Online]. Available: https://www.linuxtv.org/downloads/legacy/video4linux/API/V4L2_API/spec-single/v4l2.html (visited on 2019-05-18).

[63] *OpenCV*. [Online]. Available: https://opencv.org/ (visited on 2019-05-19).

[64] C. J. v. d. Branden Lambrecht, Ed., *Vision models and applications to image and video processing*. Boston: Kluwer Academic, 2001, ISBN: 9780792374220.

[65] M. Livingstone, *Vision and art: the biology of seeing*. New York, N.Y: Harry N. Abrams, 2002, ISBN: 9780810904064.

[66] B. V. K. Amanpreet Kaur, "Comparison between YCbCr Color Space and CIELab Color Space for Skin Color Segmentation", [Online]. Available: https://www.ijais.org/archives/volume3/number4/222-0521 (visited on 2019-05-16).

[67] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A Line Segment Detector", *Image Processing On Line*, vol. 2, pp. 35–55, 2012-03, ISSN: 2105-1232. DOI: 10.5201/ipol.2012.gjmr-lsd. [Online]. Available: http://www.ipol.im/pub/art/2012/gjmr-lsd/?utm_source=doi (visited on 2019-05-20).

[68] S.-B. Hung, C.-C. Yu, and Y.-C. Cheng, "A Smith Predictor Enhanced PID Controller", *IFAC Proceedings Volumes*, 7th IFAC Symposium on Dynamics and Control of Process Systems 2004 (DYCOPS -7), Cambridge, USA, 5-7 July, 2004, vol. 37, no. 9, pp. 661–666, 2004-07, ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)31885-2. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667017318852 (visited on 2019-05-16).

[69] J. Owens, *Television sports production*, 4th ed. Amsterdam ; Boston: Focal Press, an imprint of Elsevier, 2007, ISBN: 9780240809168.

Appendix 1 – Source files

All published source files can be found on this page:

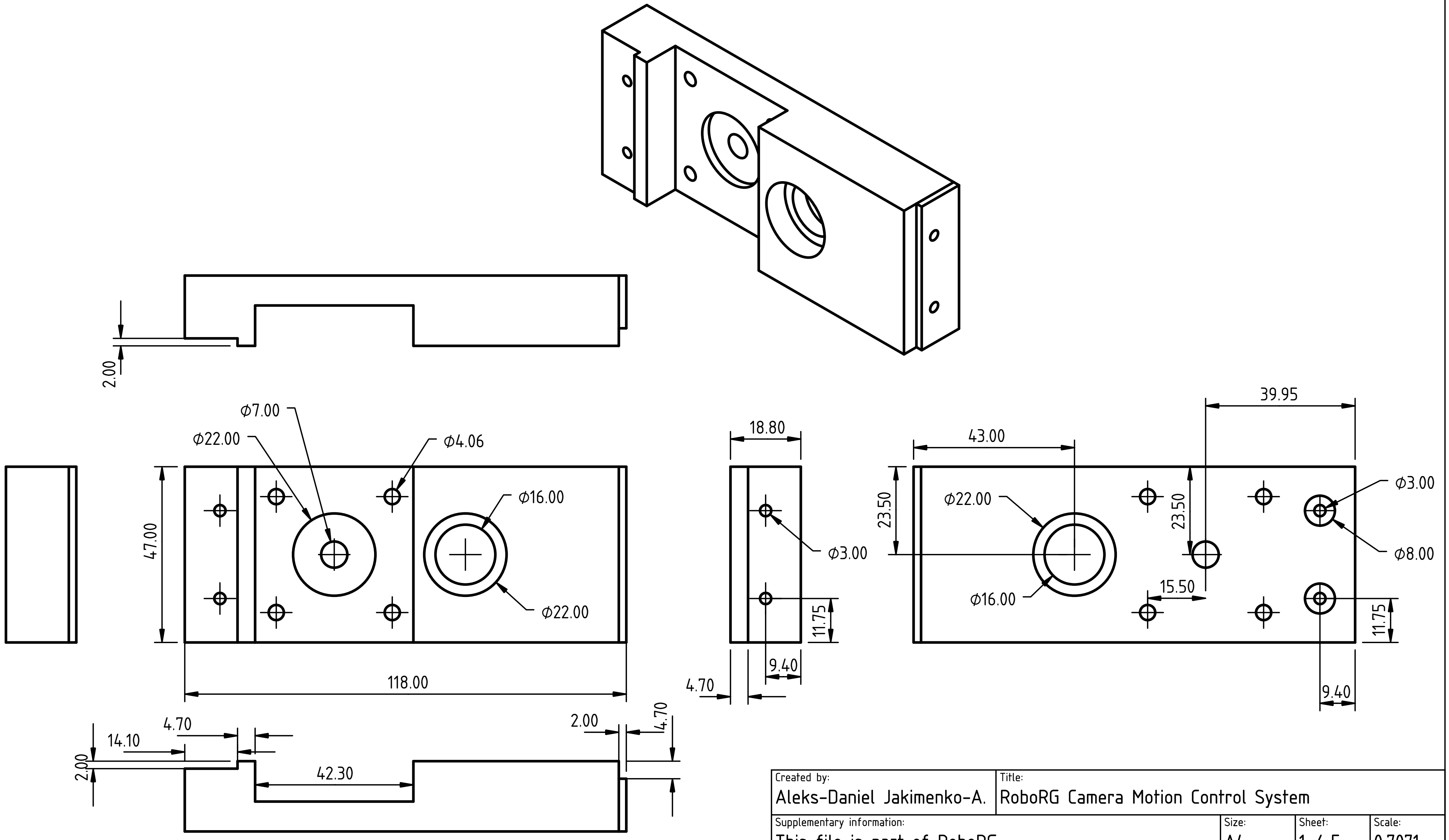
- `https://github.com/RGVID-EU`

More specifically:

- `https://github.com/RGVID-EU/RoboRG-CAD` – mechanical design as described in chapter 3.1
- `https://github.com/RGVID-EU/RoboRG-EDA` – schematics and PCB design as described in chapter 3.2
- `https://github.com/RGVID-EU/RoboRG-Firmware` – firmware as described in chapter 4.1
- `https://github.com/RGVID-EU/RoboRG-Middleware` – middleware and control loop as described in chapters 4.2 and 5.2
- `https://github.com/RGVID-EU/RoboRG-CV` – image processing software as described in chapter 5.1

Appendix 2 – CAD Plans

The next 5 pages show CAD drawings of the mechanical design. However, it is recommended to use computer-aided manufacturing (CAM) instead of relying on these drawings. See Appendix 1 for instructions on how to get the actual source files.



Created by:
Aleks-Daniel Jakimenko-A.

Title:
RoboRG Camera Motion Control System

Supplementary information:
This file is part of RoboRG

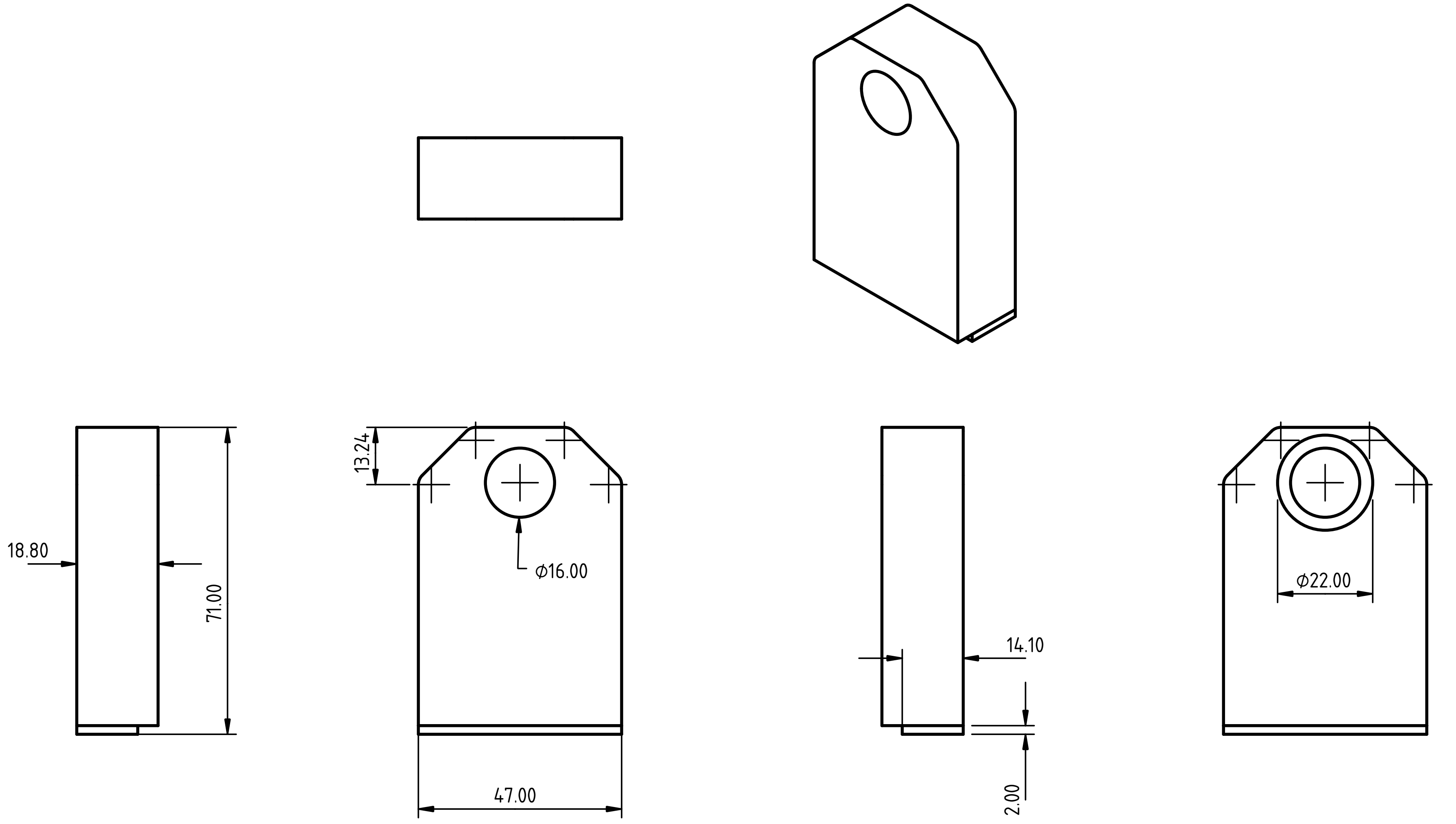
RGVID.EU

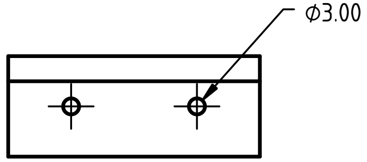
Size:	Sheet:	Scale:
A4	1 / 5	0.7071

Part number:
RoboRG-BottomPlate

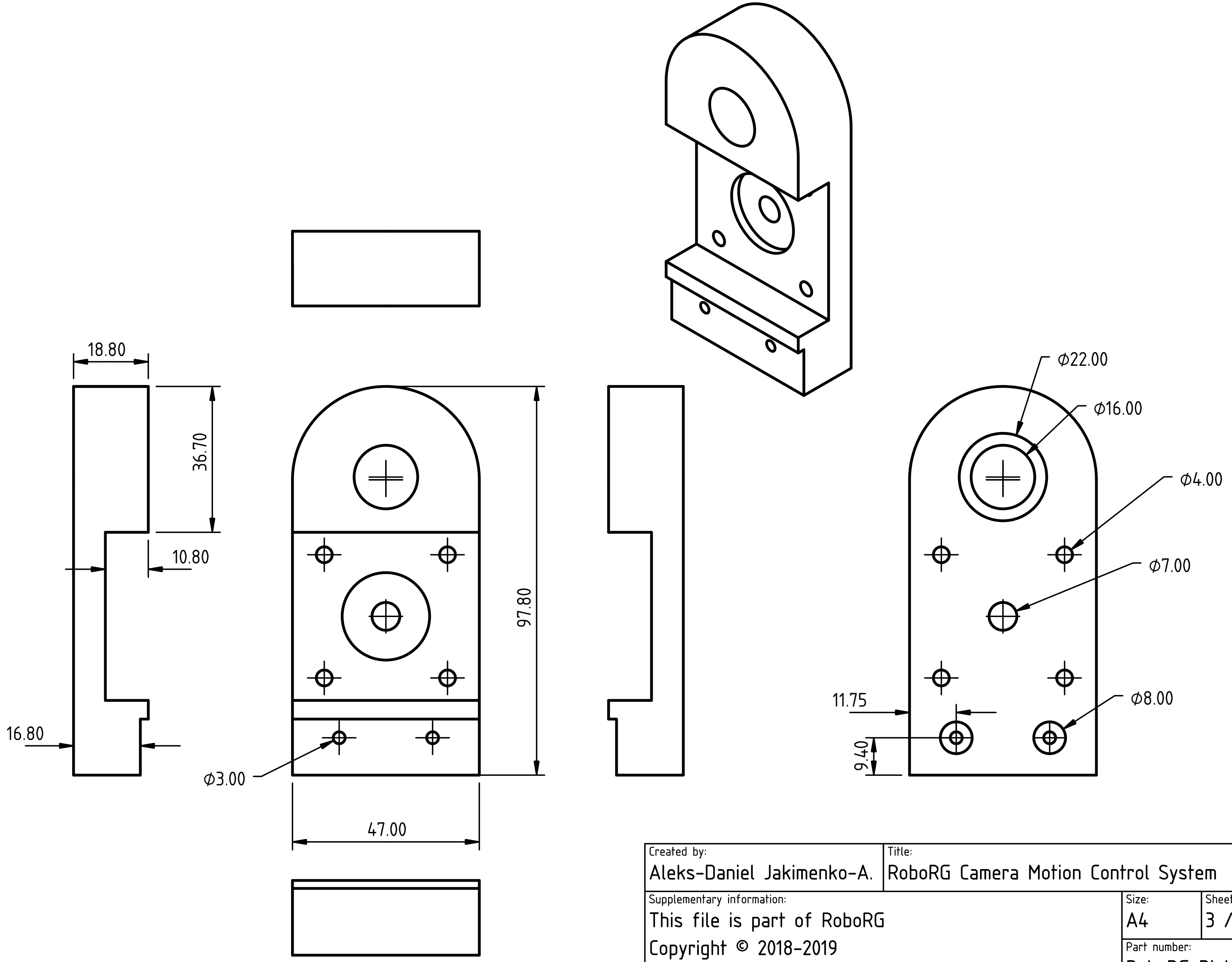
Drawing number:
1

Date:	Revision:
2019-05-13	1.0.0

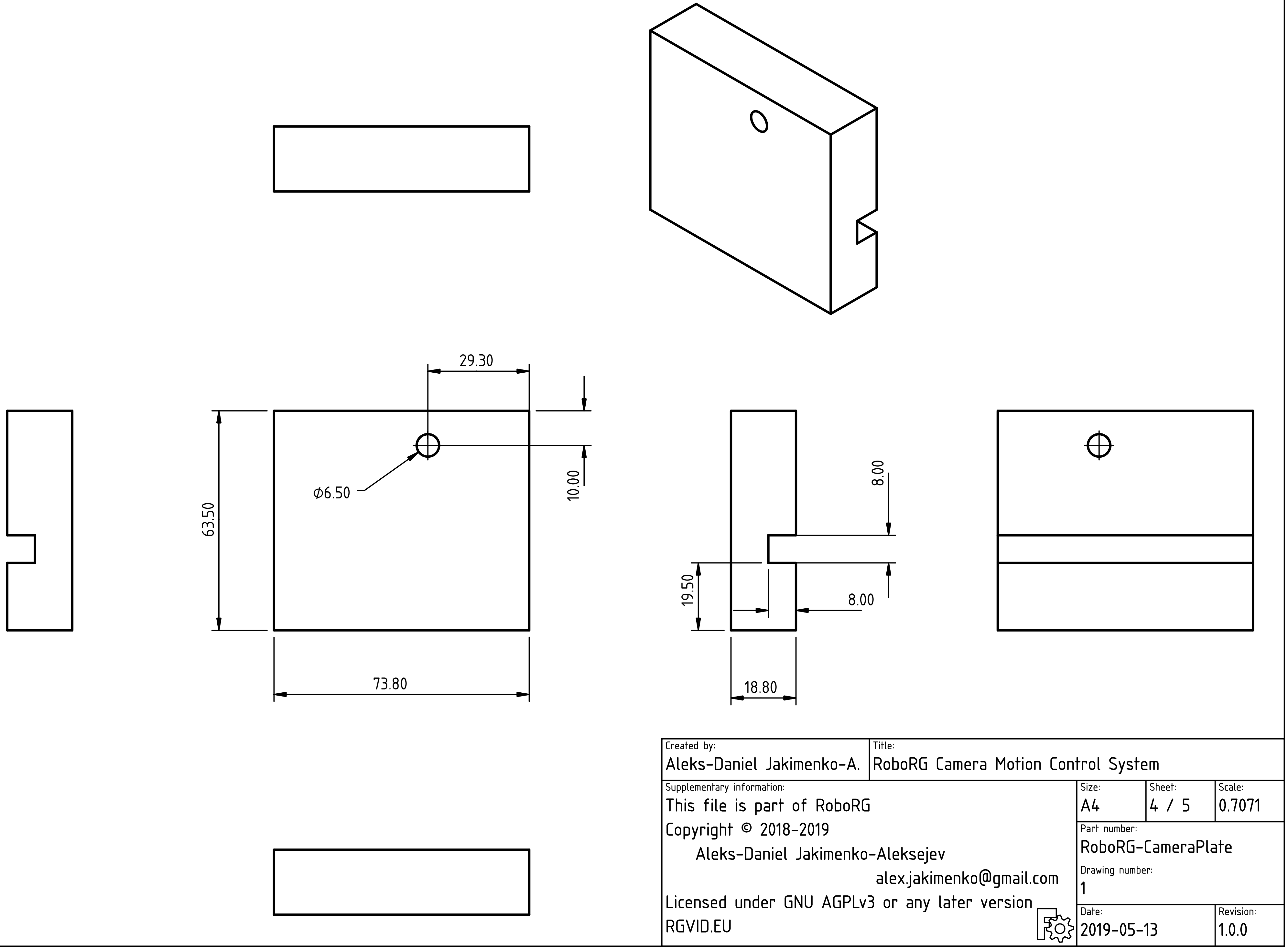




Created by: Aleks-Daniel Jakimenko-A.	Title: RoboRG Camera Motion Control System		
Supplementary information: This file is part of RoboRG	Size: A4	Sheet: 2 / 5	Scale: 0.7071
Copyright © 2018-2019 Aleks-Daniel Jakimenko-Aleksejev alex.jakimenko@gmail.com	Part number: RoboRG-LeftPlate		
Licensed under GNU AGPLv3 or any later version	Drawing number: 1		
RGVID.EU	Date: 2019-05-13		Revision: 1.0.0



Created by: Aleks-Daniel Jakimenko-A.	Title: RoboRG Camera Motion Control System		
Supplementary information: This file is part of RoboRG Copyright © 2018-2019 Aleks-Daniel Jakimenko-Aleksejev alex.jakimenko@gmail.com Licensed under GNU AGPLv3 or any later version RGVID.EU	Size: A4	Sheet: 3 / 5	Scale: 0.7071
	Part number: RoboRG-RightPlate		
	Drawing number: 1		
	Date: 2019-05-13		Revision: 1.0.0



Created by: Aleks-Daniel Jakimenko-A.

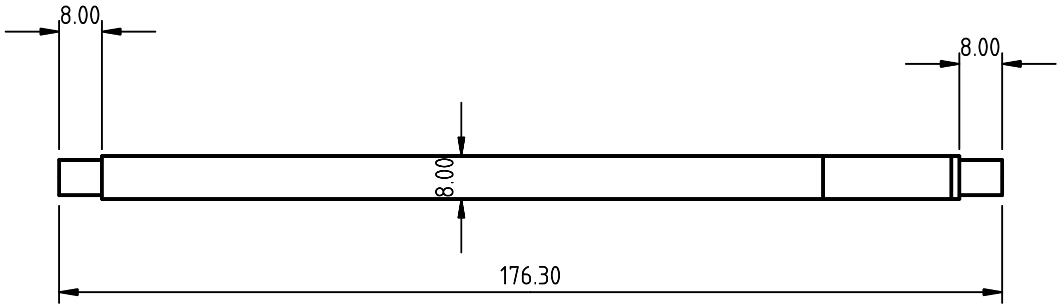
Title: RoboRG Camera Motion Control System

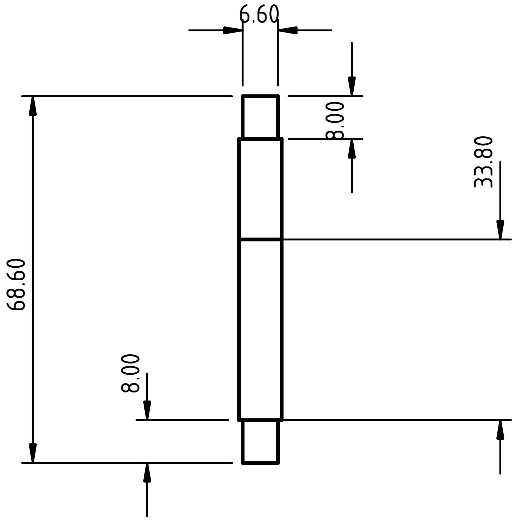
Supplementary information:

This file is part of RoboRG

RGVID.EU

Size: A4	Sheet: 4 / 5	Scale: 0.7071
Part number: RoboRG-CameraPlate		
Drawing number: 1		
Date: 2019-05-13		Revision: 1.0.0



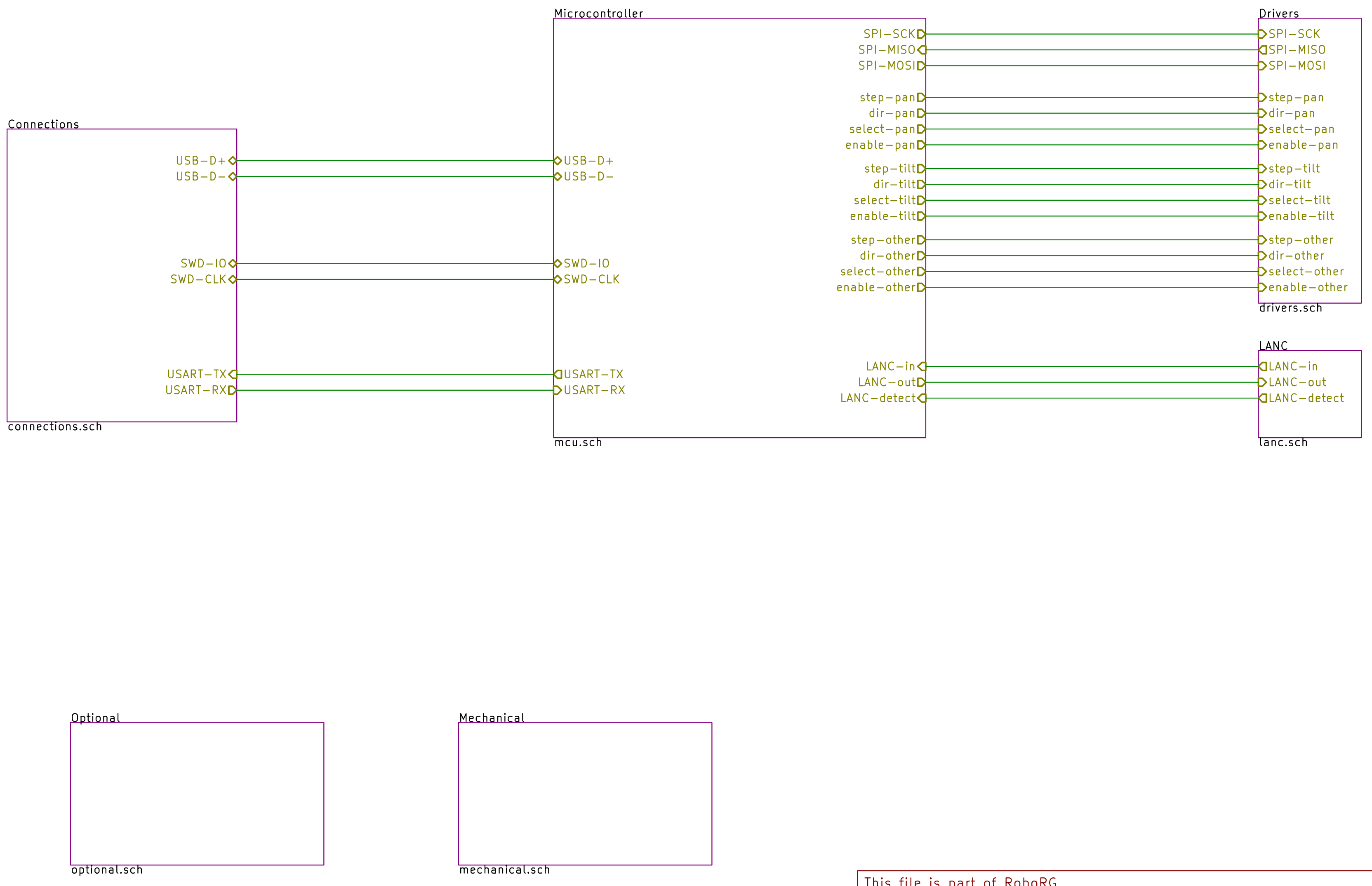


Cut external thread on shaft ends

Created by: Aleks-Daniel Jakimenko-A.	Title: RoboRG Camera Motion Control System		
Supplementary information: This file is part of RoboRG Copyright © 2018-2019 Aleks-Daniel Jakimenko-Aleksejev alex.jakimenko@gmail.com Licensed under GNU AGPLv3 or any later version RGVID.EU	Size: A4	Sheet: 5 / 5	Scale: 0.7071
	Part number: RoboRG-PanShafts Drawing number: 1		
	Date: 2019-05-13		Revision: 1.0.0

Appendix 3 – Schematics

The next 7 pages contain schematic diagrams of the control board.

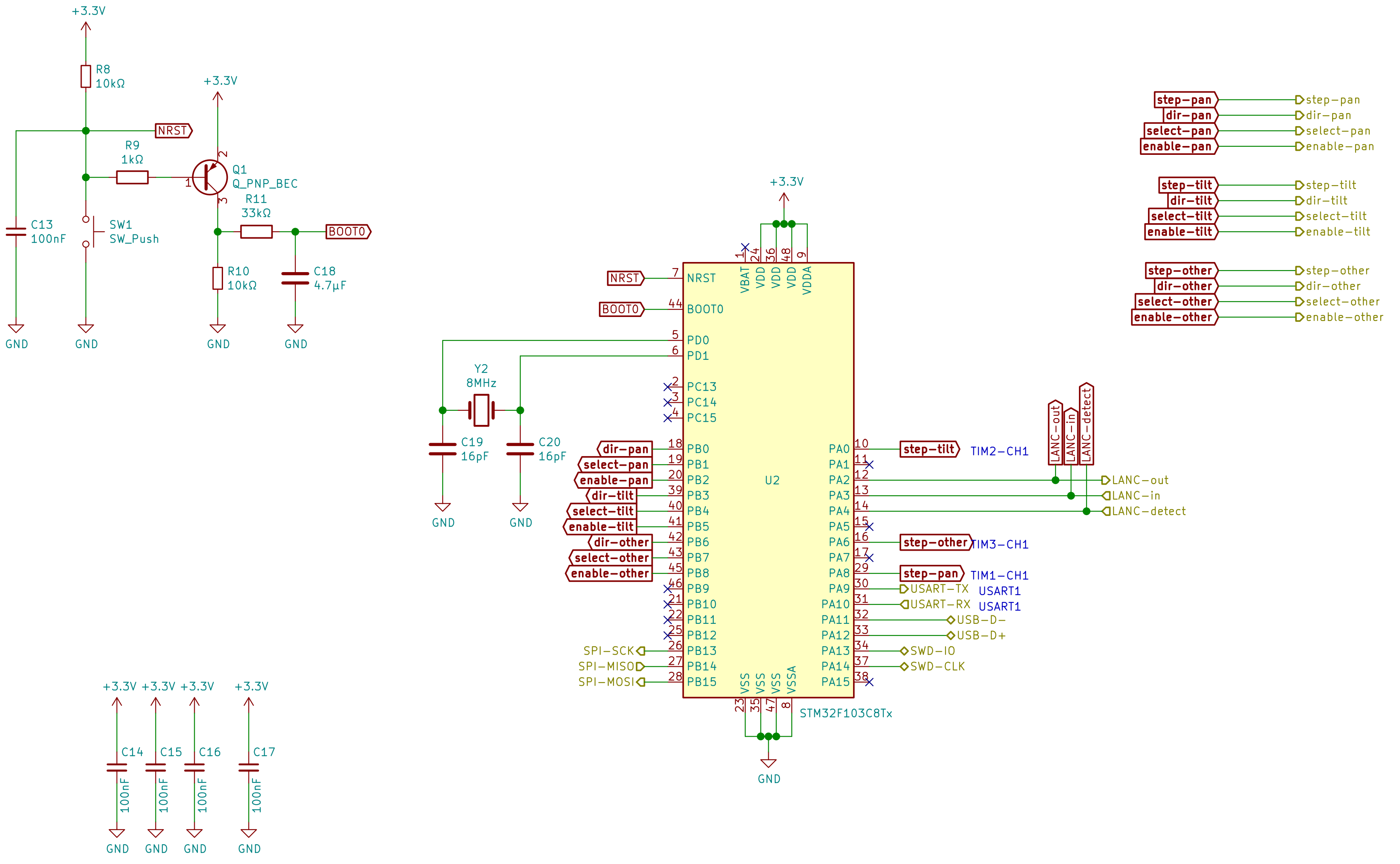


RGVID.EU

Sheet: /
File: RoboRG−eda.sch

Title: RoboRG Camera Motion Control System

Size: A4 Date: 2019−05−16 **Rev: 1.0.2**
KiCad E.D.A. kicad 5.1.0+dfsg1−1 Id: 1/7

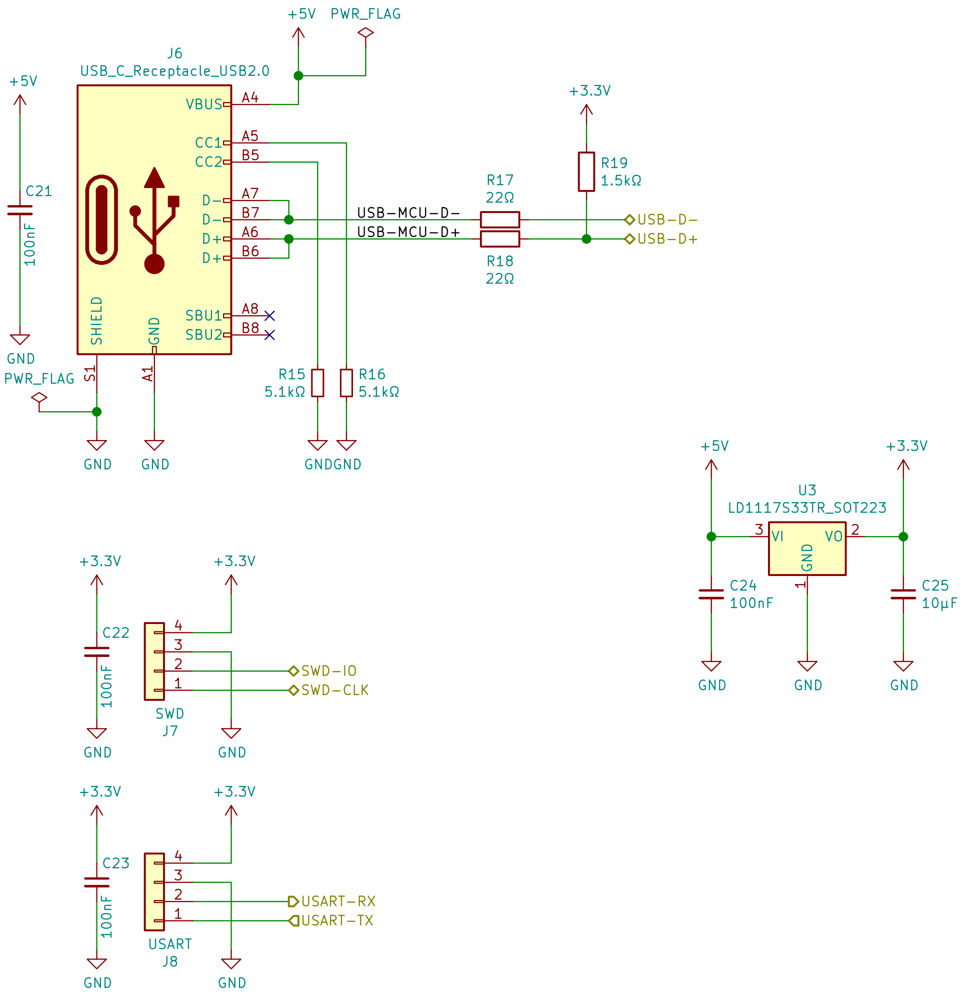


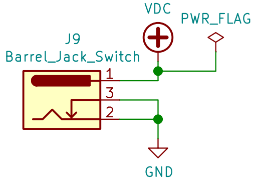
This file is part of RoboRG
Copyright © 2018–2019
Aleks–Daniel Jakimenko–Aleksejev <alex.jakimenko@gmail.com>
Licensed under GNU AGPLv3 (or any later version)
RGVID.EU

Sheet: /Microcontroller/
File: mcu.sch

Title: RoboRG Camera Motion Control System

Size: A4	Date: 2019–05–16	**Rev: 1.0.2**
KiCad E.D.A. kicad 5.1.0+dfsg1–1		Id: 2/7





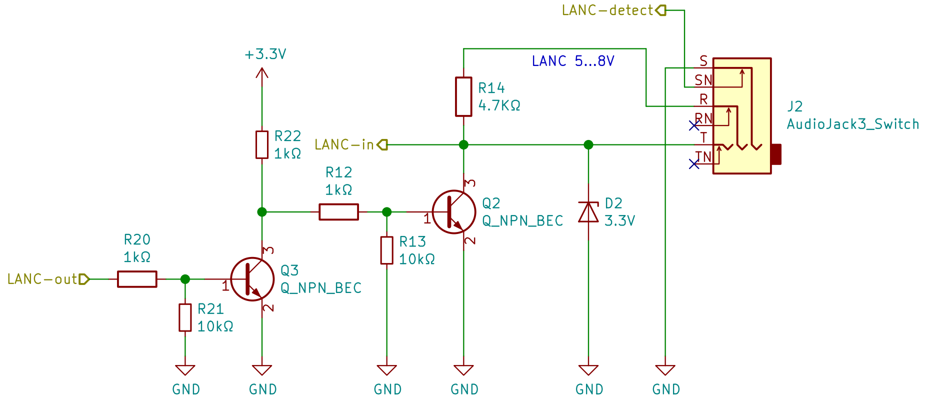
This file is part of RoboRG

RGVID.EU

Sheet: /Connections/
File: connections.sch

Title: RoboRG Camera Motion Control System

Size: A4	Date: 2019−05−16	**Rev: 1.0.2**
KiCad E.D.A. kicad 5.1.0+dfsg1−1		Id: 3/7



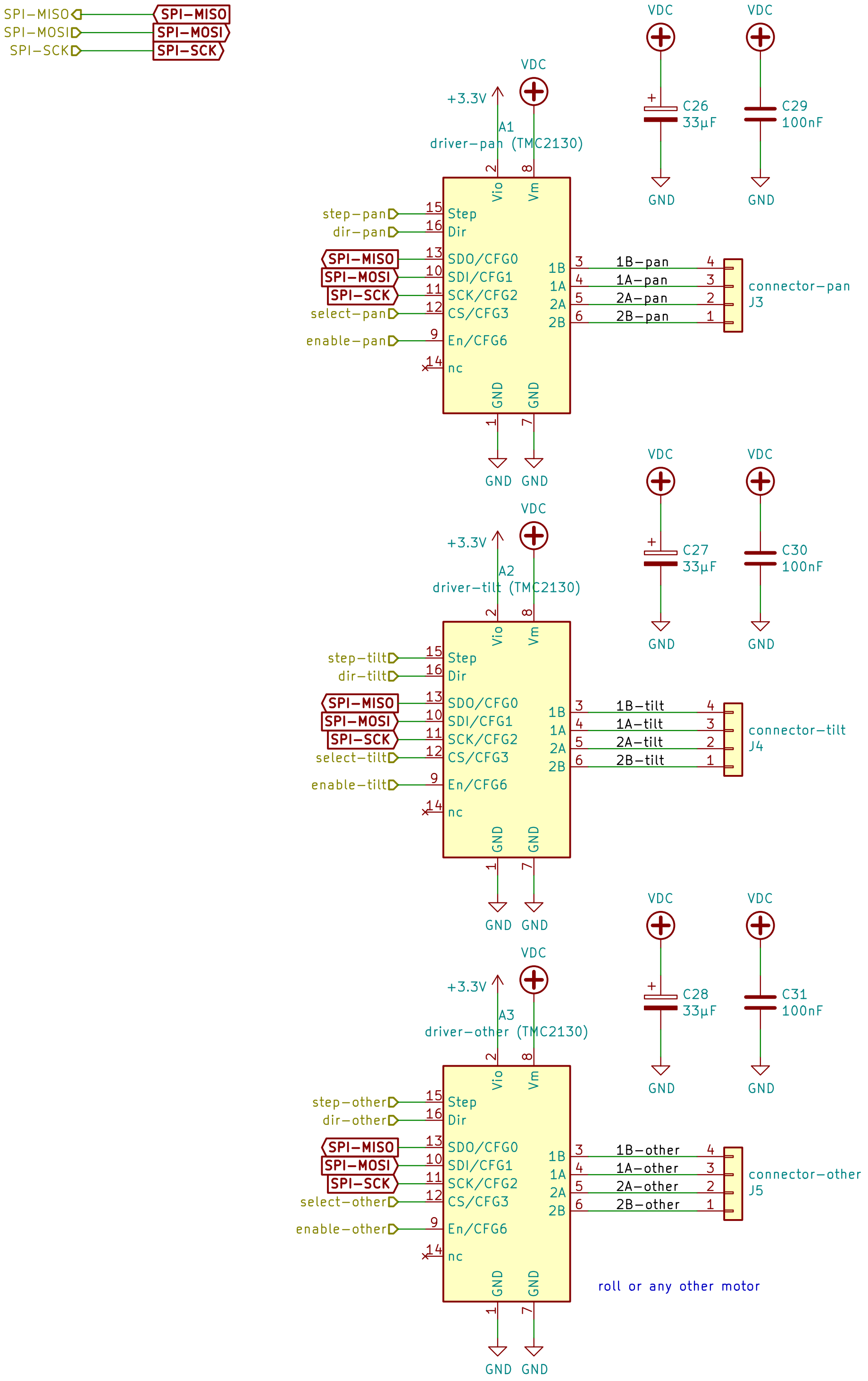
This file is part of RoboRG

RGVID.EU

Sheet: /LANC/
File: lanc.sch

Title: RoboRG Camera Motion Control System

Size: A4	Date: 2019-05-16	**Rev: 1.0.2**
KiCad E.D.A. kicad 5.1.0+dfsg1-1		Id: 4/7

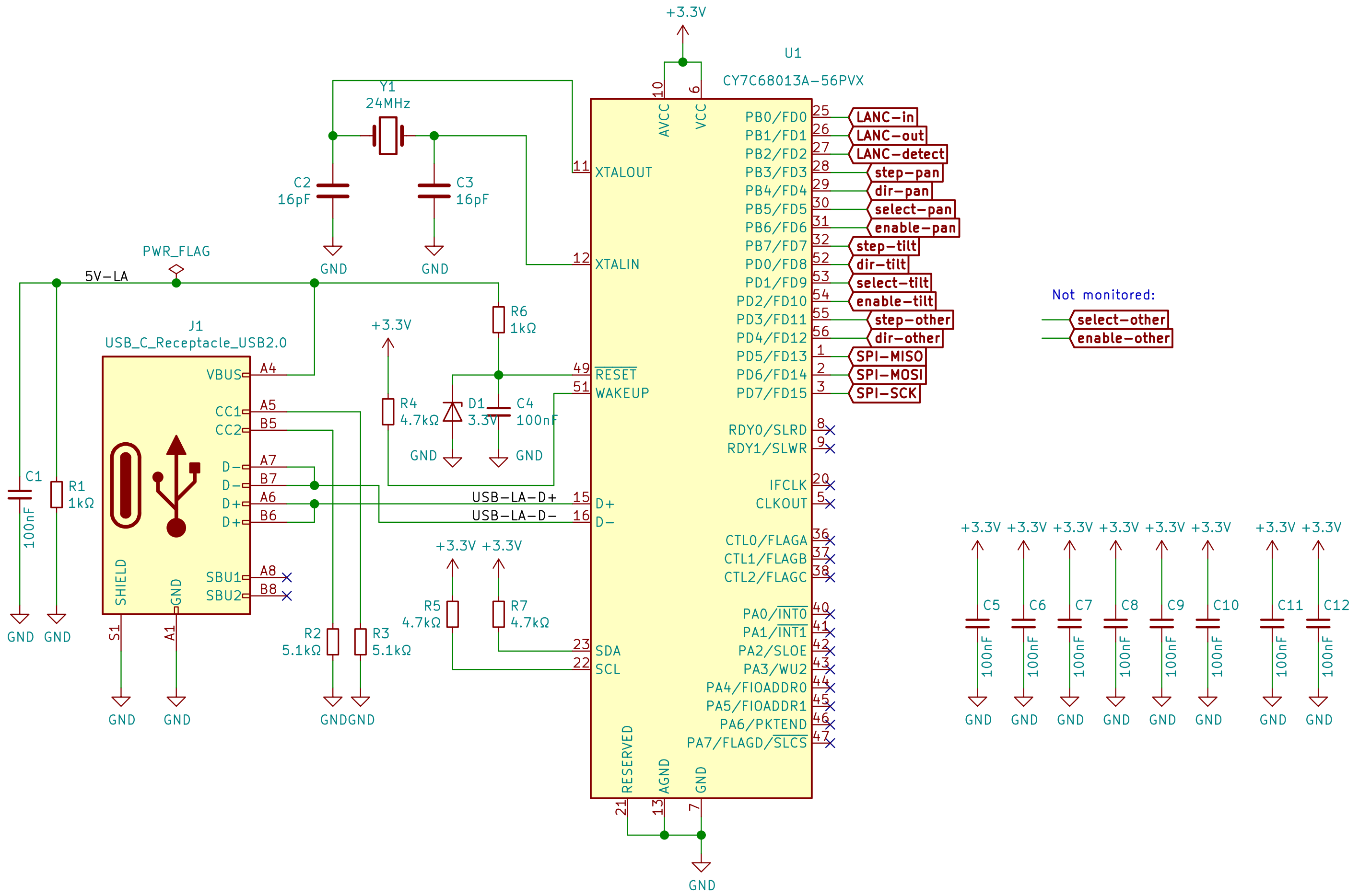


This file is part of RoboRG
Copyright © 2018–2019
Aleks–Daniel Jakimenko–Aleksejev <alex.jakimenko@gmail.com>
Licensed under GNU AGPLv3 (or any later version)
RGVID.EU

Sheet: /Drivers/
File: drivers.sch

Title: RoboRG Camera Motion Control System

Size: A4	Date: 2019–05–16	**Rev: 1.0.2**
KiCad E.D.A. kicad 5.1.0+dfsg1–1		Id: 5/7

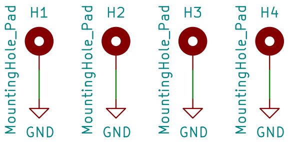


This file is part of RoboRG
Copyright © 2018–2019
Aleks–Daniel Jakimenko–Aleksejev <alex.jakimenko@gmail.com>
Licensed under GNU AGPLv3 (or any later version)
RGVID.EU

Sheet: /Optional/
File: optional.sch

Title: RoboRG Camera Motion Control System

Size: A4	Date: 2019–05–16	**Rev: 1.0.2**
KiCad E.D.A. kicad 5.1.0+dfsg1–1		Id: 6/7



This file is part of RoboRG

RGVID.EU

Sheet: /Mechanical/
File: mechanical.sch

Title: RoboRG Camera Motion Control System

Size: A4	Date: 2019–05–16	**Rev: 1.0.2**
KiCad E.D.A. kicad 5.1.0+dfsg1–1		Id: 7/7

Appendix 4 – Gerber files

The next 3 pages show gerber files. Each fabrication house typically has its own rules, meaning that gerber files may need to be generated in a special way. See Appendix 1 for instructions on how to get the actual source files.

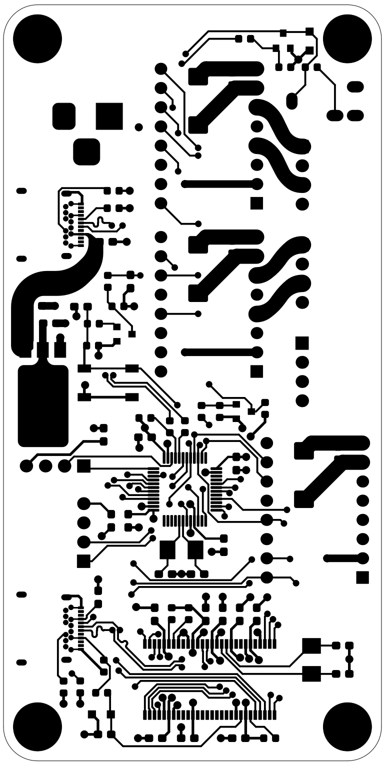


Figure 1. F_Cu layer.

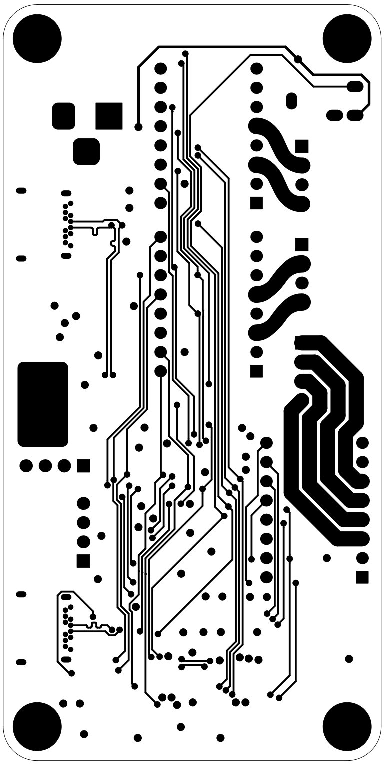


Figure 2. B_Cu layer.

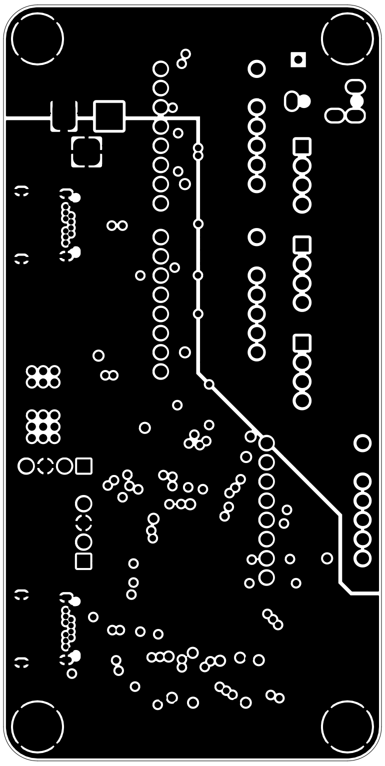


Figure 3. In1_Cu layer.

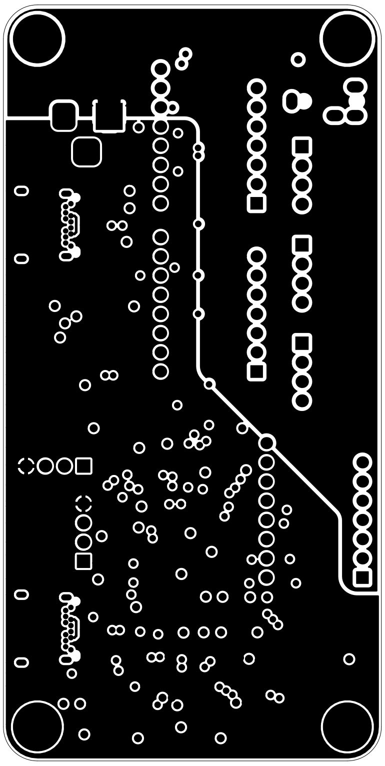


Figure 4. In2_Cu layer.

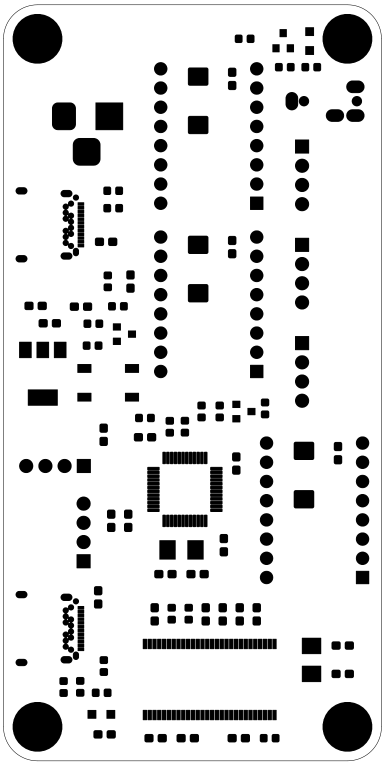


Figure 5. F_Mask layer.

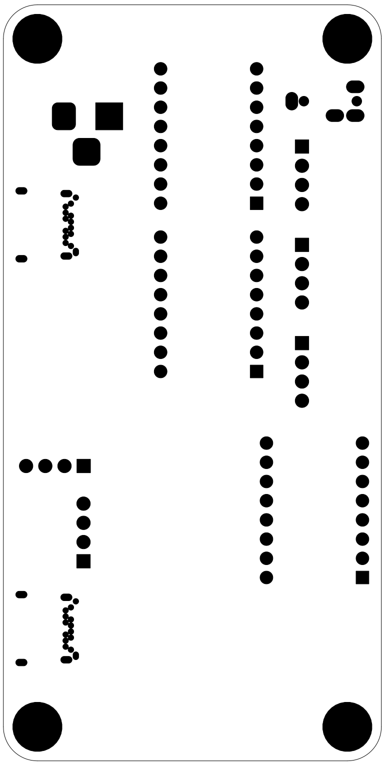


Figure 6. B_Mask layer.

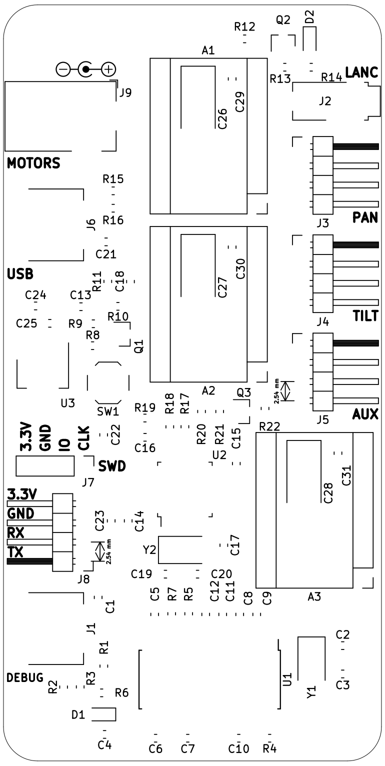


Figure 7. F_SilkS layer.

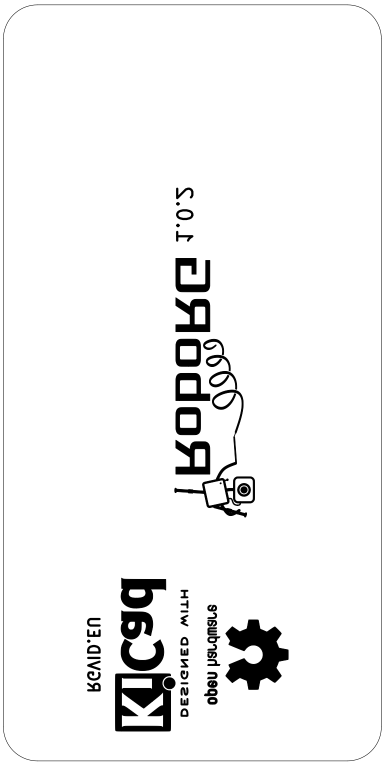


Figure 8. B_SilkS layer.

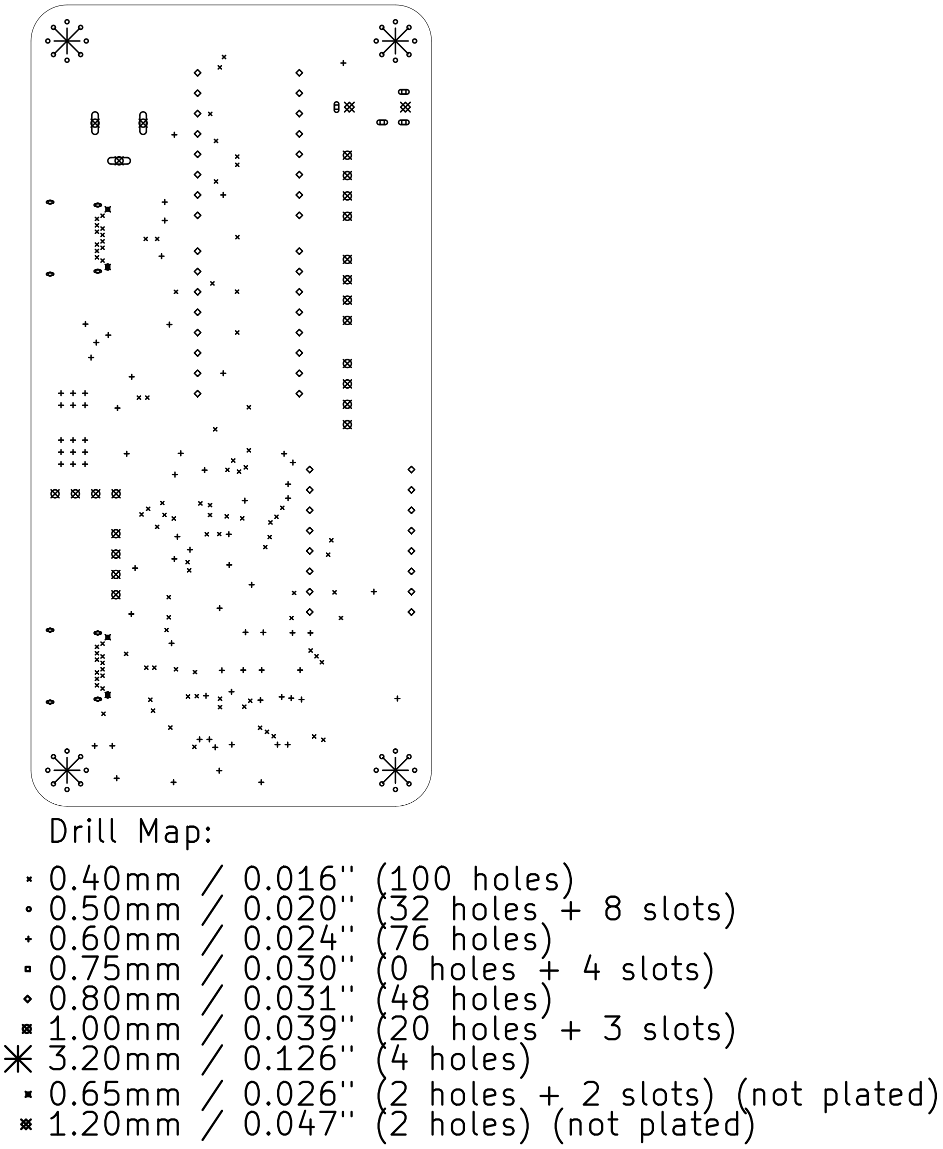

Drill Map:
× 0.40mm / 0.016" (100 holes)
○ 0.50mm / 0.020" (32 holes + 8 slots)
+ 0.60mm / 0.024" (76 holes)
□ 0.75mm / 0.030" (0 holes + 4 slots)
◇ 0.80mm / 0.031" (48 holes)
⊠ 1.00mm / 0.039" (20 holes + 3 slots)
✳ 3.20mm / 0.126" (4 holes)
× 0.65mm / 0.026" (2 holes + 2 slots) (not plated)
⊠ 1.20mm / 0.047" (2 holes) (not plated)

Figure 9. Drill map.