Linköping University | Department of Computer and Information Science Master's thesis, 30 ECTS | Master of Science Thesis in Media Technology LiTH-ISY-EX--23/5601--SE

Interactive wide-angle view camera for a virtual watch tower

- A part of the Ngulia Project

Afra Farkhooy Victoria Stråberg

Supervisor : Magnus Malmström Examiner : Fredrik Gustafsson



Upphovsrätt

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår. Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art. Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart. För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years starting from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Afra Farkhooy Victoria Stråberg

Abstract

The declining population of black rhinoceroses in Tsavo West national park, Kenya, has served as the driving force behind Project Ngulia, with Ngulia serving as an enclosed area within the park. As of now, the area is equipped with multiple cameras connected to a system that automatically classify animals and humans. This thesis aims to investigate the suitability of the Insta360 One X2 camera acting as a virtual watch tower for capturing and streaming 360° images. This will work in real-time, providing a remote surveillance experience for the park rangers thereby optimizing their work.

A system was implemented to create a efficient workflow, which includes stitching of the 360° images, file transfer protocol for image transmission and storage, as well as socket programming to facilitate port monitoring and communication. Additionally, the compatibility of two single board computers, LattePanda and Rock 4 SE, with the implemented system was evaluated. User experience methods as field studies, workshops and a user interview were also performed. The work has been developed in Sweden, resulting in limited availability for testing at the target location during the initial months.

The outcome was a both locally and remotely working system, together with LattePanda, capturing images of the waterhole in Ngulia. However, because of the conclusions drawn regarding the power supply and the lack of essential functions in the 360° camera, the system was taken home for further research. Propositions is presented regarding future work, some being that the projects within Ngulia team may collaborate to enhance hardware efficiency and explore the utilization of 360° images in educational and entertainment contexts.

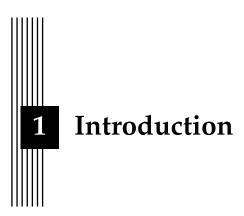
Acknowledgments

We would like to extend a big thank you to our examiner Fredrik Gustafsson and supervisor Magnus Malmström for your support and our valuable discussions which have moved our work forward. Thanks to Carlos Vidal at Linköping University for all the help during the Kenya trip and the deployment regarding hardware and network. Thank you Martin Stenmarck at HiQ for always being there when we were lost and scratched our heads and for helping us make valuable contacts at HiQ. We would also like to express our gratitude to HiQ Norrköping and Linköping for their warm hospitality, generous provision of office spaces, and for embracing us as part of their company culture. Thank you for making us feel welcome and valued. Thanks to Kolmården Zoo who made it possible for us to carry out field studies and test our camera technology. Thanks to the Ngulia team and Moses who answered our questions about our work and gave us wise words. Thanks to the University students in Nairobi for great discussions and questions about our work. Thanks to P-O Hansson, for making our trip to Kenya unforgettable. Finally, we would like to thank our master thesis colleagues, friends and family. We appreciate that you have been able to put up with our chatter about our thesis while at the same time being a great support and showing great interest in our work.

> Afra Farkhooy & Victoria Stråberg Norrköping, June 2023

Contents

Contents			vi
1 Introduction			1
1.1 Background			1
1.2 Motivation			2
1.3 Aim			2
1.4 Research questions			2
1.5 Limitations			2
2 Theory			4
2.1 Project Ngulia			4
2.2 Remote digital towers			5
2.3 Insta360 One X2			6
			6
2.4 Single Board Computer 2.5 Wide-angle camera			9
U			
2.6 Object detection in 360° images			10
2.7 File Transfer Protocol			12
2.8 Socket programming			13
2.9 Secure Shell SSH			13
2.10 User Experience	 •	•	14
3 Method			15
3.1 Implementation			15
3.2 User Experience			26
4 Result			29
4.1 Field studies at Kolmården Zoo			 29
4.2 Final set up and dashboard			31
4.3 End user meeting			31
5 Discussion			34
5.1 Result			34 34
5.2 Method			34 34
			_
5.3 Deployment	 •	•	36
6 Conclusion			38
6.1 Research questions			38
6.2 Future work			38
Bibliography			40



The use of surveillance technology is a method of monitoring the digital and physical actions and communications of people. Common forms include applications on smartphones that gather data and facial recognition software in smart security cameras. By the help of advanced algorithms for object detection and classification with machine learning, a camera surveillance system automatically provides valuable information to the user. Besides the algorithms, the hardware also has a great impact. Deciding on how many cameras are needed for a surveillance system is dependent upon which areas that will be monitored and what characteristic the given area has. Other question is how many angles one wants to cover, how the camera performs under certain circumstances and how much resolution or detailing is required. Camera surveillance is mainly used by citizens and urban environments for protecting their neighborhoods and governments for intelligence gathering, like espionage. However, poaching is fairly new area where camera surveillance is discussed. It is an use that would benefit from the smart remote systems since the work of protecting these threatened animals comes with a risk, one being threatened by poachers or the animals themselves. There is therefore a need for a system that can generate warning alarms, requires minimal manual work and has low latency in order to help rangers.

1.1 Background

Tsavo West National Park is the home to the Ngulia rhino sanctuary, where around 100 black rhinoceros in a 100-square-kilometer reserve are being protected against illegal hunting. Rhinos are being hunted and killed in pursuit of the coveted horns, which is worth more than gold on the black market. In some cultures, the horns are even believed to be able to cure diseases. During the 1970s, the Ngulia reservation was inhabited by approximately 10,000 black rhinos, accounting for half of Kenya's rhino population [1]. Roughly 50 years later, the population has decreased to one percent of the earlier population, now representing about 13 percent of Kenya's rhino population. This drives Project Ngulia, a public-private partnership to combat this problem [2]. Some of the most valuable partners in this project are Smart Savannahs, Linköping University, HiQ and Kolmården Zoo. Linköping University's main role is to provide the park rangers with new, high-end technical solutions which facilitates the daily work of protecting the rhinos. The solutions are based on conducting research on how different technologies can be developed and what hardware is most appropriate for each

application. Over the past years, students from Linköping University has been a part of this project, as a part of their master thesis ([3], [4], [5]). Project Ngulia is, to this date, provided with databases containing thousands images of savannah animals. The earlier master thesis students has contributed with technical knowledge and developed machine learning based sensor systems to automatically classify animals in camera images. They have built up the entire chain of training models on computer clusters, for implementation in both frontend and backend. These techniques are still in need of continued development to help the park rangers to be connected to the park without being physically present. The main purpose is to achieve an effective reporting and a secure communication system.

1.2 Motivation

Previous projects within Project Ngulia have used camera traps powered by solar cells that detects moving objects in an environment. This year's project introduces another type of camera that has an even wider range, the 360° camera which will act as an virtual watchtower. Within Project Ngulia, the virtual watchtower is the first major study to explore this topic. The camera is still meant to capture images in real time, but this results in that the end user getting a wider field of view and control over the environment. Before the start of the project, the camera Insta360 One X2 has been chosen by the Project Ngulia team. In this master thesis, it will be investigated whether it is a suitable camera to output image streams and display them. In addition to the fact that 360° camera are new within Project Ngulia, the associated dashboard is also a new solution, which were the image will be displayed. The end users are the Ngulia team and the park rangers. The motivation to use a 360° camera and present the images in a virtual watch tower is to be provided with a full width remotely surveillance system. The 360° images will also be sent through the detection algorithms to test if the animals could be registered.

1.3 Aim

The aim is to create a virtual watch tower that contributes to a remote user having a similar experience to a physical person in the same position. The work will roughly be divided into two fundamental tasks. The first task is to establish a seamless integration with the Insta360 One X2, eliminating the need for physical interaction to capture an image. It will be investigated whether the camera of choice is suitable for the virtual watchtower. Additionally, the compatibility of two single board computers, LattePanda and Rock 4 SE, will be evaluated regarding its performance on how it streams the image to the backend. The second task is to display the 360° image to a dashboard. One tentative place for the hardware's is on a high altitude near the tree lodge located at Ngulia, where both power and Wi-Fi are provided.

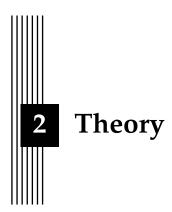
1.4 Research questions

- 1. Is the Insta360 One X2 a suitable camera for the purpose of functioning in a surveillance system that should provide the park rangers with a remote experience while at the same time being functional?
- 2. Which of LattePanda and Rock 4 SE is best compatible with the system implemented to interact with the camera remote and why?

1.5 Limitations

During the project, the virtual watchtower will be reviewed and evaluated in the test location Kolmården Zoo in Sweden. This means that Kenya, as the target location, will not be available

during most of the project. Also, the hardware is not tested in the target location. Therefore, it is not known how it is affected by the different climate factors that prevail in Kenya compared to Sweden. Power to the camera and access to Wi-Fi is assumed to exist and is left outside the project.



2.1 Project Ngulia

Developing, deploying and teaching tech for a safer wild life

-Project Ngulia

The initiative for Project Ngulia was conceived and initiated by the Stimson Center in 2013. However, it was not until seven years later when master thesis students, studying within physics and media technology, begun to contribute to the project solving major issues and improving the smart surveillance system. After each master thesis project, the students presented the new developed solutions and what conclusions were drawn to give the next year's students insights into what new problems have been created and that should be further studied.

The first students that got involved in the project were Amanda Tydén and Sara Olsson in year 2020 [3]. The issue that were addressed then was to develop the first smart surveillance system. That included developing a detector that analyses the video and image streams in real-time from camera traps that were supposed to be placed around the park. The aim of the work was to identify rhinoceros, humans and a set of six common large animals in the African savannah by motion detection. The detector should also be able to automatically send interesting observations in real-time to a server for persons of interest to observe. Different machine learning methods were compared and performed. Also, the authors created their own training data with images collected from different data sources. The development of an object detection which can identify a set of African animal species was fulfilled and with good results from both image and video streams. However, due to the pandemic, Tydén and Olsson were not able to visit the target location in Kenya and thus could not gather further diversified training data from the Ngulia reservation. It is also discussed in future work how it would be interesting to further investigate how the system could be combined with sensors such as sound, lidar or IR to trigger to start the system when a living creature is near since it would reduce the use of energy. They also discussed the interesting topic of tackling the challenge of identifying individuals.

The year after, Johan Forslund and Pontus Arnesson joined Project Ngulia and continued to contribute to the earlier work [4]. Forslund and Arnesson introduced a camera-equipped microprocessor to capture the images. The focus was mainly to detect and classify humans in

the production environment. They, as Tydén and Olsson, also faced the challenge of not being able to have physical access to the production site due to the pandemic. This makes it harder to train a network. Another point that was discussed was that the authors were aware that in the real production, the microprocessor will not be connected to a computer thus not being able to be monitored. It can therefore be difficult to troubleshoot the reason why no images are taken nor uploaded from the camera trap. One solution to this could be to implement regular status updates that is sent to a user or server as a text file containing information such as battery life and Wi-Fi strength. That would, in addition to giving valuable information, also assure that the system is active and running.

Concluding the last three years, Johan Lindér and Oscar Olsson were the two final master thesis students that contributed to the project during 2022 [5]. A contribution of their work was an end-to-end smart surveillance system that can use different camera sources to produce valuable information to stakeholders. This time the students were able to take the system all the way to Kenya for deployment and is to this date still running. They emphasized the need of still needing more training data for the classification model to be more reliable which would improve the end result as well. Lindér and Olsson also discussed the chosen camera for the work and the limitations with the quite poor performance from the sensor when it comes to image quality, but also the sensor memory which makes it hard to hold several images of higher quality during run time.

All the insights and conclusions that the previous master thesis student draw during their work is beneficial for generations after that continues improving the system within Project Ngulia.

2.2 Remote digital towers

Digital towers - The future of air traffic control.

That is how the software company Searidge Technologies describes the innovative technology [6]. This is not an exclusive opinion. Thanks to super-fast fibre networks and high definition cameras, air traffic controllers are now being offered remote sites hundreds, or even thousands, of miles away from the airport, still keeping them in the loop by the help of visual control rooms by electronic means. The benefits are air traffic controllers situational awareness, efficiency, flexibility and safety to name a few.

In 2016, Norbert Fürstenau published the book called *Virtual and Remote Control Tower - Research, Design, Development, Validation, and Implementation* [7]]. Here, Fürstenau writes a personal account regarding the origins of the virtual air traffic control tower. Fürstenau starts by highlighting the benefit of eliminating the need for direct visual observation and consequently the requirement for a costly tower building at an exposed location in visual distance from the runway. He further continues with the history of this technology which goes back about 20 years, where ICAO (International Civil Aviation Organization) contributed with one major technological aspect of remote watch towers based on the their regulations for a aerodrome traffic control. Citing ICAO document from 2001:

...Aerodrome controllers shall maintain a continuous watch on all flight operations on and in the vicinity of an aerodrome as well as vehicles and personnel on the manoeuvring area. Watch shall be maintained by visual observation, augmented in low visibility conditions by radar when available.

Since then the technology has evolved. Today, remote digital towers operations are being implemented globally, and the world is seeing the many benefits these services can offer airports. However, air traffic control are still the only area using this technique and there is a need for further research finding new possible areas of use, one of them being surveillance systems for animal poaching.

Aperture	F2.0
35mm Equivalent Focal Length	7.2mm
Exposure value	±4EV
White balance	Auto, 2700K, 4000K, 5000K, 6500K, 7500K
Photo Resolution	360: 6080x3040 (2:1). Pano: 4320×1440 (3:1)
Photo Format	insp and RAW
Gyroscope	6-axis gyroscope
Bluetooth	BLE 4.2
Wi-Fi	802.11a/b/g/n/ac
USB	Туре-С
Micro SD Card	UHS-I V30 speed class
Battery Capacity	1630mAh
Charging Time	85 minutes
Use Environment	-20°C to 40°C
Weight	149g
Dimensions (W x H x D)	4.62 x 11.30 x 2.98 cm
Run Time	5.7K@30fps – 80 minutes

Table 2.1: Specifications about Insta360 One X2

2.3 Insta360 One X2

Insta360 One X2 is a pocket 360° camera created by the company Insta360 [8]. The company was founded in 2015 and makes action cameras, mobile/desktop applications and Software Development Kit (SDK) for developers. The Insta360 One X2 camera produces both image and video streams with its dual fish eye lenses. The available formats are *insp*, a panoramic image captured by an Insta360 camera, and *RAW*, an uncompressed image. The camera is compatible with iOS and Android devices. Specifications about the camera can be seen in Table 2.1

This kind of camera is used for different 360° footage. The paper 360° *Image Orientation and Reconstruction with Camera Positions Constrained by GNSS Measurements* presents a solution for coupling 360° images and high-precision GNSS systems (Global Navigation Satellite Systems) for direct georeferencing of outdoor projects without the need for manually measuring Ground Control Points (GCPs) [9]. The system consists of a geodetic pole with an GNSS antenna and an Insta360 One X2. The system enabled the simultaneous capture of GNSS data and geotagged images. The authors conducted tests in a courtyard to evaluate the performance of the proposed approaches. The tests involved acquiring geotagged images, 360° videos and GCP measurements. The results showed that the proposed approaches could achieve metric accuracy comparable to using manually measured GCPs. The authors highlighted the advantages of using 360° camera, such as a larger FOV, allowing capturing the entire scene around the user in a short time which speeds up the data acquisition process, and that they are considered as low-cost instruments - making them more accessible for various applications.

2.4 Single Board Computer

Single Board Computer (SBC) is a complete computer integrated onto a single circuit board, including memory, Input/Output (IO), microprocessor, and all other necessary functions [10]. It it a common topic of conversation when discussing internet of things. Jonathan Á. Ariza and Heyson Baez, professor's within technology in electronics, discusses this subject in their work *Understanding the role of single-board computers in engineering and computer science education: a systematic literature review* [11]. The authors drew the attention to that the implications,

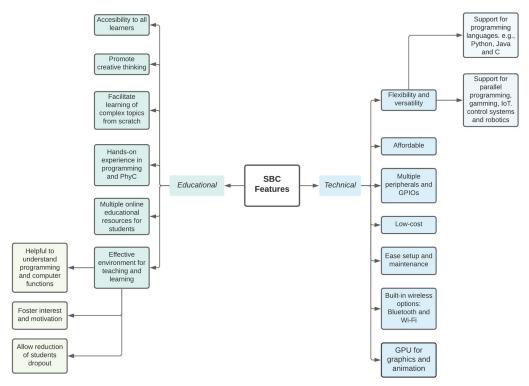


Figure 2.1: Technical and educational single-board computer (SBC) features described by the authors. IoT, Internet of Things

possibilities and constraints of these devices in engineering have not been explored in detail. They therefore used systematic literature reviews search to answer a set of questions to identify and reveal current gaps, but also to find out how SBCs are employed in higher education and what learning outcomes are derived from their usage. The results showed the positive aspects about SBCs, some of them being the low cost, ease of maintenance, support for multiple programming languages and the online resources to experiment and learn (Figure 2.1). Students that were involved in the study highlighted the interaction with the SBCs as an important factor in their learning process by increasing their motivation, creativity and curiosity. Ariza and Baez emphasized the interest of exploring how SBCs can help to promote digital inclusion in developing countries or in communities with problems in access to education and socioeconomic difficulties.

2.4.1 Rock 4 SE

Radxa designed the Rock 4 SE SBC based on the Rockship RK3399 series SoC [12]. It runs Android, Linux and Berkeley Software distributions. There are currently seven various models of the Rock 4 on the market, of which the Rock 4 SE is one of them. The hardware has the following specifications as seen in Table [2.2]:

As the specification says, Rock 4 SE is equipped with ARM big.LITTLE technology that uses a dual processor: ARM Cortex-A72 Dualcore 1.5Ghz and ARM Cortex-A53 Quadcore 1Ghz. ARM (abbreviation for Advanced RISC Machine) is currently the most widely used Central Processing Unit (CPU) architecture in the world [13]. In contrast to traditional CPUs, big.LITTLE CPUs employ a unique approach. Instead of relying solely on multiple identical cores, big.LITTLE CPUs consist of two distinct core clusters designed for different purposes. One cluster comprises high-performance cores intended for demanding tasks, while the other cluster consists of power-efficient cores suitable for conventional tasks. This design

Processor	big.LITTLE ARM Technology	
	(Dual Cortex - A72@1.5GHz, Quad Cortex - A53@1.0GH)	
Ram	4GB LPDDR	
Storage capability	Micro SD, M.2 (M key, eMM)	
USB	3.0*2, 2.0*2	
Wi-Fi	802.11 ac	
Bluetooth	5.0	
I/O	40 expansion header	
Power	9-12 V	

Table 2.2: Specifications about Rock 4 SE

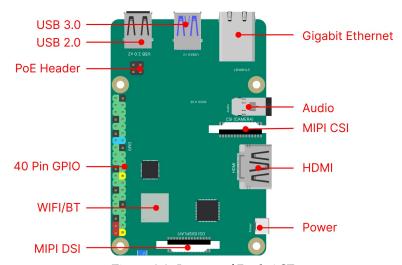


Figure 2.2: Layout of Rock 4 SE

allows big.LITTLE CPUs to efficiently allocate and distribute workloads, ensuring optimal performance while minimizing power consumption. High-performance cores offer impressive speed and power but consume more energy, while power-efficient cores are less powerful but consume significantly less power.

2.4.2 LattePanda

LattePanda themselves describe that they provide with innovative and reliable SBCs with latest technology for makers and developers [14]. They continue by saying that LattePanda has expanded from makers to industry scenarios, becoming the key parts of various industry project. LattePanda is the worlds first SBC with Windows 10 operating system but can also separately run Atmega32u4 chip on the same board which runs Arduino [15]. Its small size makes it suitable for a wide range of of tasks typically performed on a standard PC. The LattePanda SBC that is provided in this work has the following technical specification seen in Table [2.3]:

LattePanda is based on Intel processor developed by Intel Corporation [16]. Intel are a highly regarded and widely used computer processors known for their exceptional performance and technological advancements. Intel processors have become synonymous with reliability, industry-leading performance, and cutting-edge technology, solidifying their position as a key player in the global market.

Processor	Atom X5-Z8300 Intel 1.8
Co-Processor	ATmega32U4 (Arduino)
Ram	2GB DDR3L
Storage capability	32 GB
USB	3.0*1, 2.0*2
Wi-Fi	4G 802.11n
Bluetooth	4.0
Ethernet	LAN co
I/O	GPIOs for Intel chip, 20 GPIOs for Ardui
Power	5V/2A

Table 2.3: Specifications about LattePanda

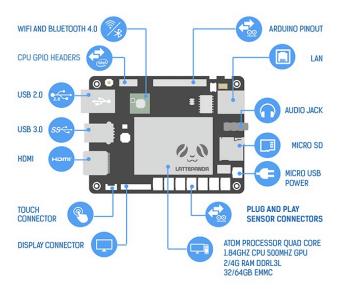


Figure 2.3: Layout of LattePanda

2.5 Wide-angle camera

Wide-angle cameras have a wide-angle lens or a broader FOV compared to standard lenses. They capture a wider perspective, allowing for the inclusion of more elements in the frame and a greater sense of expansiveness. The FOV determines how much of the surrounding area will be included in the photograph. A wider FOV will capture a larger area, while a narrower FOV will capture a smaller, more zoomed-in portion [17]. In Figure 2.4, different FOV are shown in a camera. A fisheye camera is one kind of wide angle camera which will be presented in this subsection.

2.5.1 Fisheye camera

A fisheye camera lens has an wide FOV close to 180° or even beyond, resulting in a highly distorted image with a characteristic bulging effect called Barrel distorion. This effect is created by the curved shape of the wide-angle lens. In the center of the frame is usually less distorted, with objects appearing relatively normal or slightly stretched. [18]

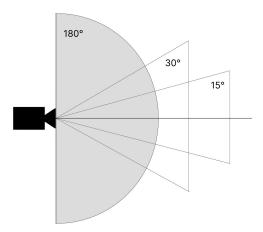


Figure 2.4: Field of View. FOV of wide angle camera is shown in grey area.

2.5.2 Image stitching

Image stitching, or photo stitching, is a technique for combining multiple photographic images to create a high-resolution image or segmented panorama. A fisheye image stitching method often includes three stages [19]:

- 1. Fisheye unwarping
- 2. Image alignment
- 3. Blending

Two fisheye lenses in a camera, called a dual-fisheye, can give images that can be stitched to produce a full spherical 360x180 panorama [20], which are seen in the Figures [2.5] and [2.6]. Dual-fisheye cameras can provide panoramic photos of a high caliber while being compact, light, and affordable. But since the images' intersection points are so small, merging photos taken with dual-fisheye cameras is more challenging. Roberto and Perazzo describes that methods design for regular images are not able to stitch dual-fisheye with good quality [19]

In the work, *Dual-fisheye lens stitching for 360-degree imaging*, Tuan Ho and Madhukar Budagavi developed a method to stitch dual-fisheye images. [21] They performed an equirect-angular projection of each fisheye image during the unwarping stage. When a fisheye image is projected into an equirectangular format, it captures the entire spherical or hemispherical view and enables the viewer to explore the scene as if they were present at the location where the image was captured. Equirectangular projection is a representation of the sphere which maps longitude directly to the horizontal coordinate, and latitude to the vertical coordinate. An equirectangular image cover 360° horizontally and 180° vertically, which means it has a proper aspect ratio of 2:1. This projection is often used for the source images in panoramic [22]. Figure [2.7] shows the process from fisheye coordinates to equirectangular projection. Roberto together with other authors have in the work 360 Stitching from Dual-Fisheye Cameras Based on Feature Cluster Matching [23], visualized how the equirectangular projections compose the full 360° panorama and its respective overlap regions, see Figure [2.8]

2.6 Object detection in 360° images

Object detection is a computer vision technique that involves identifying and localizing objects within images or videos. To generate useful results, object detection algorithms frequently use machine learning or deep learning. Humans can quickly identify and pinpoint



Figure 2.5: Dual fisheye image. The front camera is seen to right and the back camera to the left.



Figure 2.6: Equirectangular image

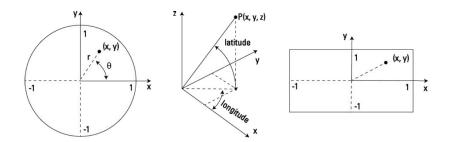


Figure 2.7: Convert from fisheye coordinate (left) to 3D vector (center) and then to equirect-angular image (right).

objects of interest when viewing photos or videos. Using a computer, object detection aims to simulate this intelligence.

The issue aimed to be solved will determine the optimal method for object detection. When deciding between machine learning and deep learning, the essential factor to keep in mind is if you have a strong GPU and a large number of labeled training images. If either of these questions has a negative response, a machine learning strategy might be the preferable option. More photos tend to make deep learning techniques more effective, while GPUs shorten the model's training time [24].



Figure 2.8: Equirectangular projection with markings for the overlap regions between the lenses. The centered image represents the front lense, while the dark parts at each side represents the back lense

The authors, Jesus Benito-Picazo and Enrique Domínguez et al. of the work, *Deep Learning-Based Security System Powered by Low Cost Hardware and Panoramic Cameras*, proposed a surveillance system for the detection of moving abnormal items that is based on deep learning [25]. Both microcontrollers and 360° cameras power this system, making it low-energy and a low cost system. The authors mentioned that Pan-Tilt-Zoom (PTZ) cameras are commonly used in surveillance system, but that there is a "lack of a comprehensive theory which sets the foundations for the development of practical systems." Also, they continue that PTZ and other conventional cameras have a limited FOV - which make them focus on the 360° cameras in their paper.

For the object detection and object characterization, the system uses convolutional neural network in a scene - which is a network architecture for deep learning. A mathematical model was suggested to provide a certain number of potential detections in the frame, which will be utilized to find and follow abnormal objects. In their paper's conclusion, the authors state that their experimental results support the low-cost solution's successful performance.

2.7 File Transfer Protocol

File Transfer Protocol (FTP) is a communication protocol for transferring files (programs, documents, etc.) between computers. [26]. It is based on TCP/IP (Transmission Control Protocol/Internet Protocol), which is the standard for communication between computers and the basis of the internet [27]. FTP can be used, for example, when a file is to be downloaded from the internet to its own computer, or, if the user has permission, from its own computer to the internet. The protocol works on a client/server model. The server listens for FTP requests from remote clients and sets up a connection if a request is received. A. Sathyanarayanan presents different advantages of an FTP server [28]. These are:

- Enables the transfer of big files and folders between two computers, regardless of their size, which is otherwise not possible with other software.
- Guarantees complete file transfer, even in the event of a break-in connection with resumption facility.
- Allows for scheduling file transfers and adding items to a waiting list for uploads and downloads.
- It offers a scripting option in the command line to control file transfers of several files.
- Is widely supported on all hosts and speeds up data transfers.

2.8 Socket programming

A socket serves as a connection point or endpoint that can be named and addressed within a network [29]. Socket programming demonstrates the utilization of socket Application Programming Interfaces (APIs) to establish communication links between remote and local processes. The processes that use a socket may run on the same system or may be spread over various networks on various computers. Both standalone and network applications can benefit from sockets. The use of sockets makes it possible to distribute work to the most productive machine, transmit information between processes running on the same machine or across a network, and gain quick access to centralized data. The network standard for TCP/IP is socket application program interfaces (APIs). Socket APIs are supported by many operating systems. The i5/OS sockets can accommodate a variety of transport and networking protocols. Thread safety is present in both socket system and socket network functions.

2.9 Secure Shell SSH

A network communication protocol called SSH, Secure Shell, enables two computers to talk to one another and share information [30]. SSH is suited for usage over insecure networks due to an inherent feature that encrypts communication between the two computers. SSH is frequently used to "login" to remote computers and carry out tasks, but it can also be used to transfer data. Client/server architecture governs SSH, as seen in Figure [2.9] [31]. A system administrator often installs and manages an SSH server program that accepts or denies inbound connections to its host computer. Then, users use SSH client software, often installed on other computers, to send the SSH server requests such as "Please send me a file". All communications between clients and servers are encrypted securely and guarded from modification.

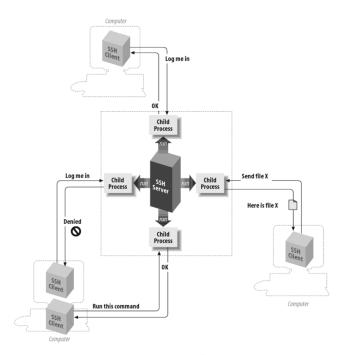


Figure 2.9: SSH architecture

2.10 User Experience

No product is an island. A product is more than the product. It is a cohesive, integrated set of experiences. Think through all of the stages of a product or service – from initial intentions through final reflections, from the first usage to help, service, and maintenance. Make them all work together seamlessly [32].

-Don Norman, inventor of the term "User Experience".

Norman joined Apple Computer 1993 and defined the term user experience as he became a self-selected title of User Experience (UX) architect. However, Norman himself, in an interview with Peter Merholz, admits that his first definition was overused at the point that it has been misinterpreted [33]. Biggest mistake, according to Norman, is to consider UX just as the designed product and not as a whole system. ISO (International Organization for Standardization) has defined a more formal definition of user experience:

Person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service [34].

In the definition, UX includes a user's perceptions and response and the use of a product. The book *Interaktionsdesign och UX* written by Mattias Arvola, treats the aspects with the design from requirement specification to finished product [35]. Arvola describes the design process through three different phases: the concept phase, the processing phase and the detailing phase. Each phase is carried out one after the other and builds on the insights and results from the previous phase. Several components in these phases are presented below.

Field study

Field studies involves observation, analysis and investigation of the real-world contexts to gather information that can enhance the further development of the product. The notes can advantageously be divided into two columns, one for the observations and the second for own interpretations and reflections. There is four distinct notes that can be made during the field study:

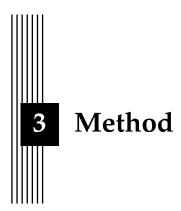
- 1. Observations and facts
- 2. Aspects to modify
- 3. Aspects to keep and enhance
- 4. Aspects to take into account

Workshop

Workshop is a interactive session that bring together stakeholders, users and designers to collaborate, generate ideas and solve problems by knowledge sharing. The purpose is to gather inputs from stakeholders and continue to further develop the product.

User interview

User interviews can be partly a face-to-face conversation, but also a opportunity for a stakeholder to evaluate the product while answering questions regarding it. During a user interview, it is important to listen to the participants and ask open-ended questions and aim to get the them to tell and explain the problem and their opinions. To start a question with "why" can be good to go in depth into how people reason and act. Arvola emphasizes the importance to supplement interviews with observations because sometimes people say things that do not match reality and how it actually works [35].



3.1 Implementation

In total, the camera system is divided into various parts such as hardware and software. In the following section, the system will be described in details how the parts were implemented and planned for mounting at the target place.

3.1.1 System

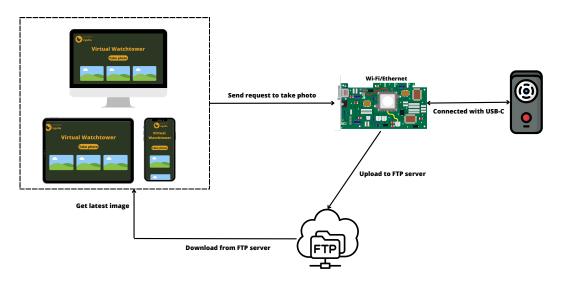


Figure 3.1: Flowchart of the intended system

Figure 3.1 shows the main idea of how the final system is supposed to be implemented and the interaction between each essential parts. The system can be divided into two fundamental components: hardware, being the SBC and camera, and software, being all the developed code. Furthermore, a three-fold division can be applied to describe the system's components based on their respective locations. Firstly, the client side of the system was designed as a

web-based interface accessible to users from any location, allowing them to interact with the system, in this particular case being that the user sends a request to take a photo in real-time. The most recent images is also displayed on the page, thereby providing users with the opportunity to actively engage with them in a 360° view. The SBC and camera is physically mounted and connected through USB-C at the designated target location, Ngulia in Kenya. Lastly, the FTP Server, responsible for data storage, is situated at HiQ in Linköping.

3.1.2 Hardware

The hardware part of the system includes three different objects, the 360° camera Insta360 One X2 and the two SBCs - Rock 4 SE and LattePanda. In this section all of the hardware will be presented in detail regarding the set up.

3.1.2.1 Insta360 One X2

The camera is used to photograph and deliver 360° view images to the end user. The camera consists of dual lenses that make up 57K 360° coverage. In addition to the camera lenses, there is a round touch screen for viewing images and selecting various settings, as well as two buttons used for taking pictures and on/off. In the final environment these features will not be used. The camera is connected with USB-C. This helps to introduce continuous stream, which gives that the battery is not needed. The camera is protected by being in a custom case made by a 3D printer. The design and functionality of the case was discussed together with the Ngulia team. The case will fit both the camera and the USB-C cable. It is also equipped with a roof, protecting against fair weather and rain. The case contains of three different parts, see Figure [3.2]

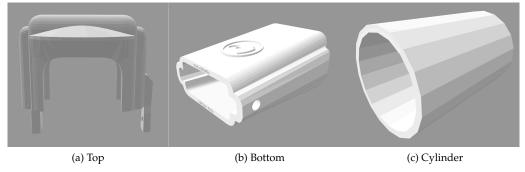


Figure 3.2: Three different parts of camera case

3.1.2.2 Rock 4 SE

As the Rock replaces the computer, the code was transferred to the hardware. Firstly, basic system setup was made and also installation of the operating system Ubuntu Server 20.04, which is based on Linux kernel. The tech company OKdo provides a step-by-step guide [36] to do this. To begin with, the Ubuntu server operating system was downloaded specifically for Rock 4 SE.

The next step was to flash the SD card that is to be inserted in the Rock. This is done by the open source BalenaEtcher. This step will physically write the software image file onto the SD Card so it will be ready for use.

When the SD card was flashed it was time to connect the Rock. Other necessities are CAT5 ethernet cable and USB-C for power supply to connect the Rock to any laptop host. Once the power supply is plugged into the Rock it starts to boot for 2-3 minutes. After the boot, a terminal could be open on any host PC to check if the Rock can be reached over the network

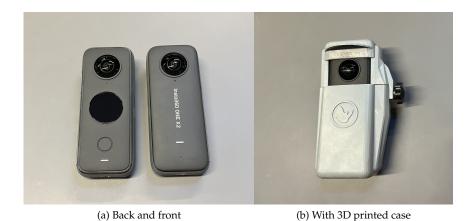


Figure 3.3: Insta360 One x2

by sending a ping request: *ping rockpi-4b*. The terminal should respond with showing the IP adress of the Rock on your network and other details.

Once connectivity with the board is confirmed, a remote session can be started with SSH to connect securely to the Rock from a terminal on you host PC. When doing this for the first time, it is recommended to update the system with the commands:

```
sudo apt update
sudo apt upgrade
```

Some other optional settings can also be made like setting the timezone, keyboard, Wi-Fi and hostname. This can be done at any time. The Rock is now setup and ready for some programming. The Figure 3.4 shows the connection set up with the SD card, an ethernet and power cable.



Figure 3.4: The start up - Rock 4 SE

3.1.2.3 LattePanda

The LattePanda gets started with connection to a monitor, keyboard, mouse and power adapter. To power on the LattePanda, the power button was pressed and hold for approximately 1 second. The onboard blue LED indicator then lights up, and LattePanda will boot into OS. The default OS is Windows 10. In this project an USB stick was flashed with Ubuntu 22.04 and installed the LP in its *BIOS GUI*. The ethernet was connected to internet according to the commands below. The Figure 3.5 shows the connection set up with the ethernet cable, power cable, mouse, keyboard and Wi-Fi-antenna.

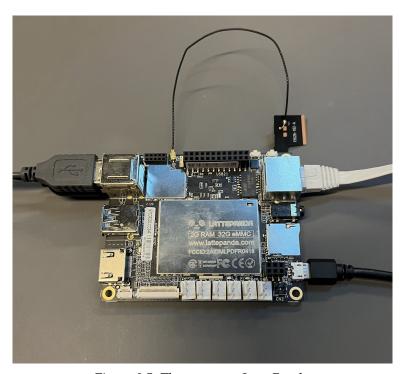


Figure 3.5: The start up - LattePanda

3.1.3 Software

The software is divided into backend and frontend. The backend refers to the underlying software that handles the data processing of the system, which includes server-side programming, APIs and FTP handling. The frontend corresponds to the user interface part of the soft-

ware and involves design, presentation and interaction elements that users directly interact with. This includes user input handling and rendering of data received from the backend.

3.1.3.1 Insta360 SDK

Insta360 provides multiple Software Development Kits (SDK) for developers to tailor your app for any Insta360 camera. Documentation of the SDKs can be found on Github [37]. To get full access to the SDK, an application process and approval of the application were required, which were done through HiQ. The SDKs that are relevant for this project are:

- CameraSDK-Cpp a C++ library to control Insta360 camera.
- MediaSDK-Cpp a C++ library to handle stitching, editing of media from Insta360 camera.

Both of the SDKs require the camera to be connected to a computer via USB-C. The first mentioned SDK offers a menu, presenting users with multiple options from which they are expected to make a selection, see code below.

```
std::cout << "Succeed to open camera!\n" << std::endl;</pre>
  std::cout << "Usage" << std::endl;</pre>
   std::cout << "1: take photo" << std::endl;</pre>
   std::cout << "2: get serial number" << std::endl;</pre>
   std::cout << "3: get file list(only video and photo)" << std::endl;</pre>
   std::cout << "4: delete file" << std::endl;</pre>
   std::cout << "5: download file" << std::endl;</pre>
   std::cout << "6: start recording" << std::endl;</pre>
   std::cout << "7: stop recording" << std::endl;</pre>
   std::cout << "8: test set exposure settings:" << std::endl;</pre>
   std::cout << "9: test set capture settings:" << std::endl;</pre>
   std::cout << "10: start preview live streaming:" << std::endl;</pre>
   std::cout << "11: stop preview live streaming:" << std::endl;</pre>
   std::cout << "16: get uuid " << std::endl;</pre>
   std::cout << "17: test take photo and download " << std::endl;</pre>
   std::cout << "18: get current capture status " << std::endl;</pre>
   std::cout << "19: start timelapse " << std::endl;</pre>
   std::cout << "20: stop timelapse " << std::endl;</pre>
   std::cout << "21: get batty " << std::endl;</pre>
   std::cout << "22: get storage info " << std::endl;</pre>
   std::cout << "0: exit" << std::endl;</pre>
```

The image that is provided from taking a photo is in dual fisheye, see Figure 2.5, and the stitched image in equirectangular, Figure 2.6.

This type of image is irrelevant for the final purpose of this project and there is therefore a need to stitch the image and transform it to a panoramic image, which is the purpose of the second SDK. Unlike the previous SDK, which has a menu of options for the user to pick from, this SDK stitches a preselected image. The SDK is provided with three different stitch types:

```
enum class STITCH_TYPE {
   TEMPLATE, //very fast but not good stitching effect
   OPTFLOW, //slow but best stitching effect
   DYNAMICSTITCH, //fast and keeps a relatively good stitching effect
};
```

How these work are and what mathematical models are used is not included in the documentation due to confidential reasons. However, when asking Insta360 support team regarding the recommended stitch type the following response was provided:

Generally speaking, the effect of optical flow will be better. Slow means that there are more sampling points, and it is also a way to improve the stitching effect

Since video streaming is out of scope for this master thesis, all the embedded functions regarding video was removed. The image stitcher is as follows:

```
auto imageStitcher = std::make_shared<ins_media::ImageStitcher>();
//set params
imageStitcher->SetInputPath(input_paths);
imageStitcher->SetStitchType(stitch_type);
imageStitcher->SetHDRType(hdr_type);
imageStitcher->SetOutputPath(output_path);
imageStitcher->SetOutputSize(output_width, output_height);
imageStitcher->EnableFlowState(bEnableFlowState);
//perform stitch
imageStitcher->Stitch();
```

These two SDKs was modified based on what capabilities the end users want. Primarily, the SDK for stitching was embedded in the first mentioned SDK, where option 1 (*Take photo*) will include stitching. This is to minimize number of iterations for the user to get the final result.

3.1.3.2 FTP - Upload image

When the image is taken and stitched, the output should finally be uploaded to a FTP server in the directory /360camera. There are several FTPs provided within this work but the one that is mainly used is within Kolmården Zoo. To be able to access and connect to the FTP server, one must initialize the following variables:

- Host
- User
- Password

The solution contains two simple FTP clients implemented in the SDK, one for Windows and the second for Linux. The following code are presented in pseudo code. See A.1.1 for the original code.

Windows:

```
function upload(image) {
    ftpDirectory = "/360camera/"
    fullFtpPath = ftpDirectory + image

    openInternetConnection() //With HINTERNET
    connectToFtpServer("FTP_HOST", "FTP_USER", "FTP_PASSWORD")
    uploadFileToFtpServer(image, fullFtpPath)
    closeFtpSession()
    closeInternetConnection()
}
```

HINTERNET, created by Microsoft, handles are organized in a hierarchical structure, where the handle obtained from *internetOpen* function serves as the root node. The subsequent level is occupied by handles obtained from the *InternetConnect* function. Finally, the leaf

nodes in the hierarchy consist of handles acquired from the *FtpOpenFile*, *FtpFindFirstFile*, and *HttpOpenRequest* functions. **Linux**:

```
function upload(image) {
    ftpPath = "/360camera/"
    fullFtpPath = ftpPath + image

ftpClient = CFTPClient()
    ftpClient.InitSession(FTP_HOST, PORT, FTP_USER, FTP_PASSWORD)

if ftpClient.UploadFile(image, fullFtpPath):
        print "File uploaded!"

else:
        print "Error."

ftpClient.CleanupSession()
    remove(image)
}
```

The above code is a simple FTP client wrapping library *libcurl* for FTP requests. *CFTP-Client*, an extension for the FTP, uses gzip and tar compression to reduce the file sizes during transfer and consolidates them into a single tar.gz archive. This archive is temporarily stored in the server and client's respective temporary directories, and it automatically extracts itself on the server, wich enhances the speed of file transfer through FTP.

3.1.3.3 FTP - Download image

The end user will view the 360° images from a dashboard, which are fetched from the backend written in ExpressJS, a framwork based on NodeJS for building web-applications. To get started, the express application generator tool was used to quickly create a skeleton. The application generator is ran by

```
$ npx express-generator
```

This generates following structure:

```
Express App

app.js

bin

www

package.json

public

images

javascripts

stylesheets

routes

indexj.js

users.js

Views

error.jade

index.jade

layout.jade
```

The two files in the *routes* folder can be removed and replaced with alternate files requires modification. In this case, a new file, *ftpAPI.js*, was created whose purpose is to connect to the FTP server. *ftp* is a FTP client module for Nodejs that provides an asynchronous interface for communicating with a FTP server. To connect to the FTP server, the four FTP related variables are initialized again.

The aim with the application is to get the three latest images from the /360camera directory. In order to transmit images from the backend server to the client, a series of steps need to be performed. Firstly, a connection is established with the FTP server, followed by accessing the relevant directory. Next, a list containing the names of all the images is retrieved and reversed to get them in the right order. A loop goes through the list and pushes the three recent images into an array. Lastly, this array is sent to the client. The code snippet below shows this process in pseudo (A.1.3):

```
router.get('/', (req, res) => {
    connectToFtpServer()
    if not connected:
        closeConnection()
        sendErrorResponse()
    if connected:
        listAllFiles('/360camera/') => {
            if (err):
                closeConnection()
                sendErrorResponse()
            else:
                result = []
                for each index of the 3 latest images:
                     result.push(list[index])
                sendResponse(result)
                closeConnection()
        }
});
```

Since the above code only retrieved the name of the images, we had to retrieve the images themselves as well, one at a time:

3.1.3.4 Socket

There are two distinct code versions for the socket programming tailored for Windows and Linux that opens a port that React can send its request to. Since both of the code versions are based on the same logic, only the Linux version is presented in pseudo (A.1.4):

```
declare server_fd, acceptedClient_socket, valread as integers
declare address as sockaddr_in #includes sin_family, sin_addr.s_addr, sin_port
declare addrlen as integers
declare buffer as array of characters with size 1024

server_fd = createSocket()
setSocketOptions(server_fd, address)
bindSocket(server_fd, address)
startListening(server_fd)
acceptedClient_socket = acceptConnection(server_fd, address, addrlen)
```

The presented code is a common way of socket programming and there are multiple examples and documentation on internet. The SDK then listens for incoming connections from React, which will send a request which is a number corresponding to one of the options from the menu and execute it.

```
valread = readData(new_socket, buffer)

create stringstream ss with buffer
create string line

getline(ss, line, '/')
option = convertToInt(line)

clear buffer
closeSocket(new_socket)
shutdownSocket(server_fd)
```

3.1.4 Frontend

There are two fundamental parts in the React dashboard, one being to show the three recent images from the FTP server and second being the opportunity for the user to take a real-time photo which will also be displayed in the dashboard after it is taken. A prototype of the dashboard is shown in Figure [3.6].



Figure 3.6: Prototype of the dashboard

The render function, containing the HTML code and that defines the structure of the content is presented as follows:

```
</div>
```

To always show the recent images on the dashboard, React Hook *useEffect* was used to fetch data and directly updating the DOM (Document Object Model). At first, the backend got fetched and the name of each image was retrieved. All original code is seen in A.2.

The render function calls *ShowImage*. This takes in each image name which is currently saved as a *blob* (binary large object), which stores binary data as a single entity, and present it as a image by using *URL.createObjectURL(data)*:

When the user presses the button "Take photo", the function *handleClick* is called, which is seen in Appendix. The function fetches the local IP adress and port of the SBC running the Insta360 SDK. It sends a constant variable with a value of 1, corresponding to a specific option in the SDK menu that triggers its execution. After the button is pressed, a timer counts to 15 seconds before it reloads the page and displays the three images again, now with the most recent one taken by the user.

```
function handleClick() {
    fetch(SDK_ipAdress + SDK_port + option) //option=1
    jsonToText(option)
    errorMessage()
    reloadWindow(15000 ms)
}
```

3.1.5 Hardware mounting

The camera and SBC is supposedly to be mounted in the tree lodge in front of the water hole at the target location connected together through USB-C. The SBC will be connected to power and Wi-Fi and supply the camera. The 3D case will also be tested and evaluated regarding if the model is visible in the periphery of the image. The Ngulia team will take photos through the React dashboard from another location to analyze the set up further, test the connection between the dashboard and the SBC containing the SDK and make adjustments. The final set up will consist of:

- Camera
- 3D case
- SBC
- USB-C
- Adjustable stand for the camera

3.2 User Experience

User Experience (UX) encompasses various methods that are used to understand and improve the user's experience with a product or service. In this section, three different methods field studies, workshops and user interview will be presented.

3.2.1 Field study - set ups at Kolmården Zoo

The first set up of the camera system was tested in the park, Kolmården Zoo. The system consisted of the camera, USB-C cable, Windows computer and a camera stand. The camera was mounted 5 meters above ground level. The tree lodge in Kenya is at about the same height, which gives an approximate picture of how the result will be in Kenya. Alternative "1" was typed in from the SDK menu which resulted in a captured photo that was stitched and sent up to the FTP server.



Figure 3.7: First set up in Kolmården Zoo

At the second and last set up before Kenya, the camera system consisted of the same type of hardware as the first set up in Kolmården Zoo. This time the camera case was tested and the React dashboard was used to capture the image. Every captured image was directly sent to the FTP server. The system consisted of the camera, 3D printed camera case, USB-C cable and Windows computer. The images were captured through the React dashboard as seen in "Take photo" button was clicked which resulted in a captured photo that was stitched and sent up to the FTP server.

3.2.2 Workshops

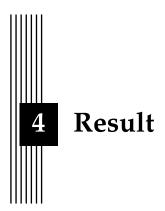
During this semester, two workshops have taken place. The first one, in March, was together with the other master thesis students, Donald Bunge, responsible for project Ngulia in Kenya, and James Mwenda, a member of the rhino unit team at Ol Pejeta Conservancy in central Kenya. During this workshop, the project was showcased in its current state along with an overview of the remaining work that needed to be completed. Bunge and Mwenda had the opportunity to share their knowledge and how the project could be implemented at the target location. They mentioned that it is important to have the primary view towards the water hole due to the fact that the rhinos, and other animals, are mostly there.

The second workshop occurred in may, involving the Ngulia team and biology students from the University of Nairobi, specifically hosted at the aforementioned university. The project was showcased here as well, now in its nearly fully developed state. Some of the given feedback from the biology student were to analyze the maximum focal length, which refers to the farthest distance at which the camera lens can effectively focus and capture clear details of a object, in this case rhinos. There was also discussions regarding whether the camera should be triggered by an automatic sensor and the cameras capabilities for capturing images during nighttime or low-light conditions.

3.2.3 User interview

There are two fundamental end users that will integrate with the dashboard. One of them being the Ngulia team which will use the dashboard presumably from Sweden. The park rangers at the target location is also a valuable end user. Given the absence of communication with the park rangers until the field trip, the questions were formulated based on the final product and the discussions held during the two workshops. The answers will be carefully considered when drawing the conclusions. The following questions will be asked to a park ranger:

- 1. What do you think about the position of the camera? Should it be mounted at another spot and capture from another angle?
- 2. Which image view is preferred and why? (360° viewer or equirectangular)
- 3. How will the dashboard be used by the park rangers?
- 4. Which approach is preferable for capturing images: manual or automatic? What are the reasons for favoring one option over the other? If the automatic mode is chosen, at what frequency should the camera take pictures?
- 5. Is night vision considered a desire, and if so, what are the reasons behind it?
- 6. Is there a requirement for additional 360° cameras at this particular waterhole or any other waterhole in the Ngulia reservation and why?
- 7. What is the recommended method of powering the camera? Which type of power source should be utilized?
- 8. Is quality important in the images? Why is it so?
- 9. Do you have any inputs regarding this camera system?



4.1 Field studies at Kolmården Zoo

Doing a field study contributes to greater understanding of how the system works as it analyzes how it works in the real environment instead of reading information and related work about the surveillance system. As previously written in [3.2.1] there were field studies at Kolmården Zoo. On the first occasion, the camera system was set up to test and see how the images turned out with the camera mounted at a height of 5 meters. The result is seen in Figure [4.1] On the second occasion, the 3D printed camera case seen in Figure [4.1] b) was used, showing an equirectangular image. In a 3D viewer, the case is not visible, as it is otherwise in an equirectangular image. In the final result in the dashboard, the user will see the equirectangular images. The rhinos that were out in the park were also photographed. Images taken of the rhino were sent to the detection team to see if the individual could be detected by the algorithms. The result yielded 76 percent rhino detection from the Insta360 One X2 image. The detection algorithm could only handle cropped images, which was the case here, see Figure [4.2]

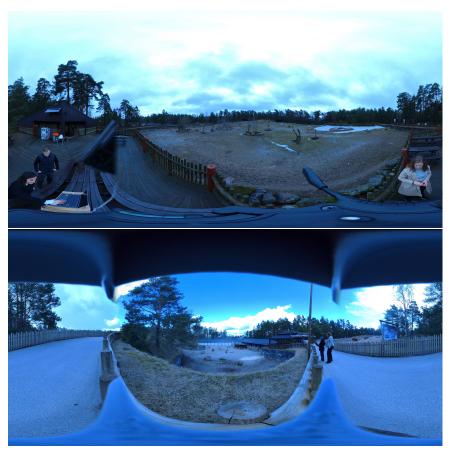


Figure 4.1: Image above is captured at first field study in Kolmården Zoo. Second image is captured at the last field study.

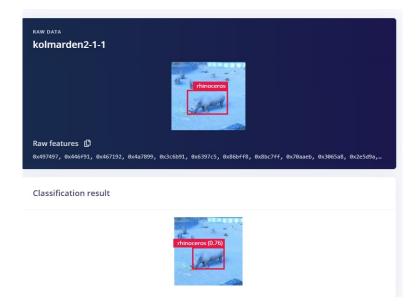


Figure 4.2: A detected rhino from a cropped 360° image captured in Kolmården Zoo

4.2 Final set up and dashboard

The final camera system included the LattePanda, Insta360 One X2 with camera case, USB-C cable and power bank. The camera system was connected to the router of the house. The camera was mounted in the corner of the house against the water hole, see Figure [4.3]. As seen, the camera has a blue case which was an extra one. The result of the dashboard is seen in Figure [4.5]. It was created to display the latest images, horizontal order, from the FTP server.



Figure 4.3: Close up at the camera, camera system connected to the router and SSH into LattePanda.

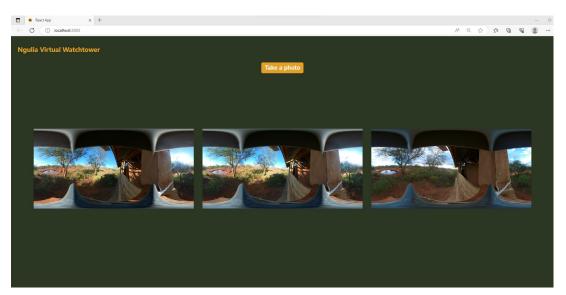


Figure 4.5: Dashboard displays three images from FTP server

4.3 End user meeting

Once at Ngulia HQ, a conversation was held with the park ranger Moses who answered questions about the 360° camera view. Moses gave feedback about the camera system as well as he navigated through a 360° image viewer, as seen in Figure [4.4]

1. What do you think about the position of the camera? Should it be mounted at another spot and capture from another angle?



Figure 4.4: First image above is captured 15 May 07:00:13 in tree lodge. Second image is instead shown in a 3D viewer

It is good and you can see the waterhole up close and clearly. The rhinos come from the other side of the water hole so it would also be good to have it at that position to get a closer view.

2. Which approach is preferable for capturing images: manual or automatic? What are the reasons for favoring one option over the other? If the automatic mode is chosen, at what frequency should the camera take pictures?

It would be great if this 360° camera system would cooperate with the other projects. For example, when the detection team recognizes an animal, it should alarm to the 360 camera to take pictures. It would be great if the camera could take a picture every 15-30 minutes. This is because of that the rhinos often moves every 15-30 minitues when it is at the water hole.

3. Which image view is preferred and why? (360° viewer or equirectangular) It is better to integrate with a 360° image than just an equirectangular image. It is great to navigate, zoom and look around in the whole area instead of a fixed image.

4. How will the dashboard be used by the park rangers?

We will check the images on the dashboard in our office at Ngulia HQ. The mobile phone should also be a good way to check the images for the patrols who can check the images directly from there. VR glasses also sound fun for the future.

5. **Is night vision considered a desire, and if so, what are the reasons behind it?**The rhinos come at night so a requirement is to have night vision. They don't have

lights at the water holes because that would scare the rhinos away. One possibility is to have infrared light.

6. Is there a requirement for additional 360° cameras at this particular waterhole or any other waterhole in the Ngulia reservation and why?

The water hole 2 should also have the 360° cameras, this is due to the fact that this waterhole is much larger and is often visited by many rhinos. The downside is that the water hole does not have a tree house like the one you visit.

7. What is the recommended method of powering the camera? Which type of power source should be utilized?

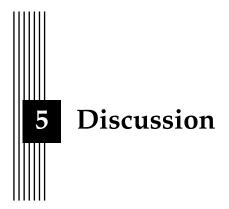
Better if the camera runs via solar panels and not power. The placement of the camera at the moment may not be good as elephants may disturb it. We could put the camera in a metal box and protect the sides but that also makes it harder for the elephants to break. Elephants and rhinos can scratch the poles we install the camera on, causing it to fall.

8. Is quality important in the images? Why is it so?

Yes. If the quality is better, it can help park rangers identify each individual. This is because every rhino has an unique marking in the ears. Every marking represent a specific rhino.

9. Do you have any inputs regarding this camera system?

It would be great if the 360° cameras is mounted along the border of the fence. This can be good for not only rhinos but poachers, to keep an eye on the boundary of the reserve.



5.1 Result

The result was a locally functioning camera system with the LattePanda and the associated dashboard created in React. The images from the camera could be taken through SSH to the SBC which helped the camera SDK to upload the images to the FTP server which were then retrieved from the dashboard where they were displayed. The SDK was also adjusted to take photos immediately when it was running, precisely to avoid many commands to run the program. On the last day, the camera was up all night and took pictures early in the morning at 7 AM through SSH remotely.

Different stitching methods were tested when the camera was assembled. We chose to use the best but slowest stitching method *OPTFLOW* to see the quality when uploading to the FTP server. We recognize that this method gave a slightly better result than the other two methods. The overlap regions between the dual fisheye images were more transparent. Also, the stitching time did not take noticeably much longer time than the other methods.

The file size was also changed to 4K image, i.e. 4000x2000 pixels to be able to retrieve more information in the images. The intention was to test this, as the high-resolution images had the potential to be viewed through VR-glasses.

The stitched images captured by the Insta360 One X2 were sent to the machine learning team to check if an animal could be detected by the algorithm. In Kolmården Zoo, a rhino was captured by the camera and the same image got successfully detected by 70 percent. Today, the algorithm is adapted to only handle the cameras that is implemented in the system with its images and not the wide-angle cameras.

5.2 Method

5.2.1 LattePanda and Rock 4 SE

Initially, the Rock was the only available SBC, as it was the one present at HiQ. The intention was to use a different SBC initially, Raspberry Pi because of its popularity and prior utilization within Project Ngulia, but due to logistical challenges and material shortages being a outcome of the COVID-19 pandemic, the availability of numerous SBCs has been severely limited, thereby impacting the available options for this master thesis. When it came time

to transfer the Insta360 SDK to the Rock, it was realized that the the part responsible for stitching was not supported by an ARM processor, only Intel processor, as the technical specifications presented (2.2 and 2.3). An alternative option was to develop a custom stitching solution, as this is a relatively common issue with access to numerous documentation available online. However, due to lack of time but also the late occurrence of this problem, the only option was to require a new SBC with the right processor. Moreover, the resources were limited, both in terms of SBC availability and SBCs with Intel processors, as it does not appear to be as commonly. LattePanda was in fact one of few available SBCs to order. However, the LattePanda was still a good replacement. The project team encountered into a challenge related to connecting the SBC to ethernet or Wi-Fi, since a monitor, keyboard and mouse was required, which was limited when we reached the target location. The Rock could simply at first be used with ethernet to later connect it to an availbe Wi-Fi. An advantage with LattePanda is its compatibility with working with Windows as well as Linux. When considering the price of both SBCs, the Rock is nearly half the prize of the LattePanda. This could serve as an argument for choosing the the former one, if their is a constrained project budget and where quantity comes before quality.

Rock 4 SE:

- + 1000 SEK
- + Not in need of monitor at start up
- + RAM 4 GB, better prestanda
- ARM processor

LattePanda:

- + Intel processor
- + Windows compatible
- 2000 SEK
- Needs monitor at start up
- RAM 2 GB

5.2.2 Insta360 One X2

The Insta360 camera proves to be a compatible device that has potential to fit into the system and capture reasonably good images from the site. However, there are certain aspects that should be carefully considered when before implementing the camera into the current system.

The Insta360 SDK greatly facilitated our work and provided us with easy ways to interact with the camera remote. One important aspect that Moses mentioned was the desire of having the pictures taken automatically with a 15 minutes interval. That was quickly implemented to see how the camera would perform which was well accomplished. However, the interaction was somewhat limited, and some essential functions were missing. One of the most crucial functions that was absent was the ability to trigger the camera to turn on, without having to physically press the camera button which is the case as of today. Although there was an "auto power turn off" setting on the camera, it did not prevent the camera from turning off. There were several instances during testing where the camera turned off without any apparent reason. On the last day in the field, the camera managed to stay on throughout the night but turned off somewhere between 7 AM and 8 AM for an unknown cause.

During the workshops that took place, the availability of night vision was discussed. It turns out, the camera does have night vision capability. Unfortunately, its performance was not tested nor evaluated due to lack of time. However, the bigger problem is that a function would have been needed that could automatically switch between night vision and daylight mode, which were not provided by the Insta360 team. This results in a significant issue as the park ranger Moses himself confirmed that the camera is most needed during night time, since there is most animal activity during late hours.

As the SBC, the price could also be a topic of discussion. The camera is quite expensive, compared to the other cameras used in Project Ngulia. A possibility would be to compare other 360° cameras, some of them being cheaper, and evaluate their performance to see if one could get the same results as from the Insta360 camera. If that is the case, the Ngulia team could aquire several cameras within budget.

The downside of switching camera is that the developed code during the semester cannot be fully utilized to another camera since the SDK is very restricted to Insta360 own devices and their development of the code. In that way it would be beneficial to explore other options to be self independent.

5.2.3 System

The project team initially possessed a Windows computer, hence it was a natural choice to start developing the FTP client and socket for that operative system. However, when the SDK was fully developed and ready to be migrated into a Linux system in the Rock, we run into one major obstacle. Due to lack of knowledge, the team did not initially fully comprehend the extent to which the FTP client and socket relied on Windows and its underlying libraries. Although the logic stays the same regardless of operative system, it was necessary for us to implement the FTP client and socket again but now to also be compatible with a Linux system. When the team were introduced to the LattePanda, which is compatible with Windows, the team chose to continue to interact with a Linux system in the SBC. This was because the team came to a conclusion the Linux system was easier to work with as it allows direct execution of programs through the terminal, particularly when working remote on SBCs. Nevertheless, the code for Windows remains to provide future developers with the option to choose.

Regarding the frontend, unfortunately the team did not have the time to work with the images in React nonetheless than just displaying them. The initial idea was to implement a 360° viewer to give the user the opportunity to interact with the image. This was also something that was confirmed of Moses to be a essential part and therefore something that should be implemented in the future.

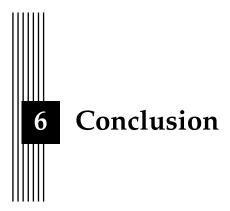
5.3 Deployment

The installation of the camera was placed on the corner of the tree house facing the water hole. The position of the camera system was easy to reach if any adjustments would have been made, such as rotating the camera in different directions or turning the camera on/off. In the planning stage, a cylinder was made in the 3D box to be attached to the stand. The stand was planned to be hollow because of the cables would be placed inside, which should protect the cables. The pile was ordered and picked up on site at Ngulia HQ. The problem was that the stand was not hollow, which meant that the camera had to be mounted directly on a flat surface. In this case, was the cylinder part not used in the final test.

At first, we ran the command SSH to the router and then executed the same command but now into the LattePanda. The router situated at the tree lodge is equipped with a Remote Management System (RMS) that enables SSH access to the router from any location. Consequently we were able to successfully establish a SSH connection to the LattePanda from anywhere. Carlos Vidal, head of technology in Ngulia team, has access to the RMS, which

includes numerous routers from various projects. If another device would try to SSH to the LattePanda, it must be on the same local area network. To access remotely, access to the RMS is a requirement. In this case, Carlos only has exclusive access to the router because he possessed additional information about the router and its log in credentials could not be shared due to confidentiality. Taking photos remote through the dashboard which connects to the LattePanda and the camera was not feasible. This was due to that the IP of the LattePanda could not not be exposed to the public network, because it is not possible to execute port forwarding with the router located at Ngulia, which is a LTE router.

Unfortunately, the camera system had to be dismantled on the last day as the camera drew too much power. Since the camera did not have a sleep function or a power on/off function, this meant that the camera had to be on at all times. There is a limited power supply in the tree house in Ngulia, which contributed to the 360° camera not being able to continue for a while longer.



6.1 Research questions

1. Is the Insta360 One X2 a suitable camera for the purpose of functioning in a surveillance system that should provide the park rangers with a remote experience while at the same time being functional?

The camera proves to be efficient in capturing, stitching and transmitting images within a relatively short period of time. The images also shows in good quality and have the potential to be used in the existing machine learning models implemented in the larger system today. However, there are some significant weaknesses. Night vision was not tested, and there is an issue with transitioning from night vision to daylight. Additionally, the camera lacks a feature that can trigger its power on or off. Being dependent on Insta360 and their updates to the SDK is also considered a drawback. These factors weigh heavily when considering whether Insta360 One X2 is a suitable camera to implement in the surveillance system and therefore the answer to this question is no.

2. Which of LattePanda and Rock 4 SE is best compatible with the system implemented to interact with the camera remote and why is it so?

For short-term, if the aim is to get the system up and running as quick as possible, the answer is to choose LattePanda, since Insta360s SDK for stitching is supported by an Intel processor. It demonstrated its ability to remain operational for an extended period. Even though the camera eventually switched off, it was still possible to connect to the LattePanda by SSH, something that was not tested with the Rock. Nonetheless, if the opportunity arises to develop stitching method, not dependent on Insta360, Rock serves as an viable choice due to its affordability and ease of remote work.

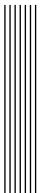
6.2 Future work

The 360° camera was new for this year, which meant that the machine learning team with the detection algorithms are not adapted to 360° images. If 360° cameras are to be introduced into the system in the future, an algorithm should be implemented that can detect those kind of images directly without having to crop them, as was the case now. It will facilitate the workflow for this project and also increase the cooperation between the different camera systems.

In addition to the images being detected effectively, another positive aspect would be if the 360° camera could be triggered by alarms from other systems with sensors. One thing that would be beneficial is to collaborate with an another Ngulia project, the master thesis *Detection and Tracking of Elephants using Seismic Direction of Arrival Estimates* [38], who were working on elephant detection through ground vibrations. If there had been a possibility to turn the camera on and off remote, it could have been triggered when ground vibrations are detected at the location. This would help the other students obtaining images that could complement their data while at the same time effectively conserve power, which is currently consumed by the camera continuously. This cooperation between the different project was even desired from Moses. Also, sending camera information often, less than every 15 minutes, such as battery level, could help the camera not shutting off and keeping it active.

Virtual reality is becoming an increasingly popular technology to use. 360° cameras deliver images that can be used in a VR context, helping the user feel present in the environment where the image was taken. It could be used within entertainment or even as an education tool. Images from Ngulia can be experienced remotely in, for example, Sweden, which can create knowledge, arouse interest and commitment to the world's endangered animals. Kolmårdens Zoo can also offer this type of VR experience from home or on site, which can attract more people to become aware of the threats to animal life.

In the future work with 360° cameras, there are several aspects to take into account. The night vision is an important factor that has to be tested and evaluate regarding the performance of object detection at night. It is also therefore important to take external lights into account and if that is a requirement in order for night vision to work. It is limited with electricity consumption in Ngulia. Therefore it is important to research about the hardware and its requirements for being active. One alternative is to introduce the solar panels to the 360° camera, such as the other master thesis projects did. Also, it is important to think of the placement of the 360° camera. If the camera is set up at a higher height, on the roof of the tree house, it will take pictures that include the entire surrounding environment.



Bibliography

- [1] Project Ngulia. Project Ngulia. https://projectngulia.org Last accessed 31 may 2023.
- [2] Kolmården. Noshörning Projekt Ngulia | Bevarandeprojekt på Kolmården. https://www.kolmarden.com/vart-uppdrag/bevarandeprojekt/noshorning-ngulia. Last accessed 19 january 2023.
- [3] Amanda Tydén and Sara Olsson. "Edge Machine Learning for Animal Detection, Classification and Tracking" (2020). Master Thesis, Linköping University.
- [4] Johan Forslund and Pontus Arnesson. "Edge Machine Learning for Wildlife Conservation Detection of Poachers Using Camera Traps" (2021). Master Thesis, Linköping University.
- [5] Johan Lindér and Oscar Olsson. "A Smart Surveillance System Using Edge-Devices for Wildlife Preservation in Animal Sanctuaries" (2022). Master Thesis, Linköping University.
- [6] Searidge Technology. Digital Towers: The Future of Air Traffic Control. https://www.codot.gov/programs/remotetower/PDF_Files/SR_Tower_Brochure. Last accessed 22 january 2023.
- [7] Norbert Fürstenau. Virtual and Remote Control Tower Research, Design, Development, Validation, and Implementation. 2nd ed. Springer Cham, 2022.
- [8] Insta360. Insta360 ONE X2. https://www.insta360.com/product/insta360-onex2. Last accessed 2 february 2023.
- [9] Previtali M., Barazzetti L., Roncoroni F., Cao Y., and Scaioni. "360° image orientation and reconstruction with positions constrained by GNSS measurements". XLVIII-1/W1 (2023). Last accessed 10 june 2023, pp. 411–416. DOI: https://doi.org/10.5194/isprs-archives-XLVIII-1-W1-2023-411-2023
- [10] LattePanda. What does single board computer (SBC) stand for? https://www.lattepanda.com/blog-308783.html Last accessed 1 june 2023.
- [11] Jonathan Á. Ariza and Heyson Baez. "Understanding the role of single-board computers in engineering and computer science education: A systematic literature review". Computer Applications in Engineering Education 30.1 (2021). Last accessed 5 june 2023, pp. 304–329. DOI: https://doi.org/10.48550/arXiv.2203.16604.

- [12] Radxa. ROCK 4 SE. https://wiki.radxa.com/Rock4/4se Last accessed 1 june 2023.
- [13] Arol Wright. What Is the big.LITTLE Architecture? https://wiki.radxa.com/ Rock4/4se Last accessed 2 june 2023. 2021.
- [14] LattePanda. About us. https://www.lattepanda.com/aboutus. Last accessed 2 june 2023.
- [15] Sam. What is LattePanda? https://core-electronics.com.au/guides/lattepanda/what-is-lattepanda/Last accessed 2 june 2023. 2017.
- [16] Andy Patrizio. Intel. https://www.techtarget.com/whatis/definition/ Intel. Last accessed 2 june 2023. 2017.
- [17] Rahul Awati. Definition: Field of View (FOV). https://www.techtarget.com/whatis/definition/field-of-view-FOV. Last accessed 5 june 2023. 2022.
- [18] Lorex Technology. Why are the corners of the camera image curved? https://help.lorextechnology.com/link/portal/57356/57366/Article/3598/why-are-corners-camera-image-curved. Last accessed 5 june 2023. 2021.
- [19] R. Roberto, D. Perazzo, J.P Lima, V. Teichrieb, J.P Quintino, F.Q.B da Silva, A.L.M Santos, and H. Pinho. "Using Local Refinements on 360 Stitching from Dual-fisheye Cameras". Vol. 5. 3. Last accessed 5 june 2023. SciTePress, 2020, pp. 17–26. DOI: https://doi.org/10.5220/0008874100170026.
- [20] D. Schneider, E. Schwalbe, and H.-G. Maas. "Validation of geometric models for fisheye lenses". *ISPRS Journal of Photogrammetry and Remote Sensing* 64.3 (2009). Last accessed 5 june 2023, pp. 259–266. DOI: https://doi.org/10.1016/j.isprsjprs.2009.
- [21] Tuan Ho and Madhukar Budagavi. "Dual-fisheye lens stitching for 360-degree imaging" (2017). Last accessed 6 june 2023, pp. 2172–2176. DOI: https://doi.org/10. 1109/ICASSP.2017.7952541.
- [22] Yingbin Wang, Zhenzhong Chen, and Shan Liu. "Equirectangular Projection Oriented Intra Prediction for 360-Degree Video Coding". 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP) (2020). Last accessed 5 june 2023, pp. 483–486. DOI: https://doi.org/10.1109/VCIP49819.2020.9301871
- [23] T. Souza, R. Roberto, V. Teichrieb, J..P Silva do Monte Lima, J. Peixtot Quintino, F.Q.B da Silva, A.L.M Santos, and H. Pinho. "360 Stitching from Dual-Fisheye Cameras Based on Feature Cluster Matching" (2018). Last accessed 6 june 2023, pp. 313–320. DOI: 1109/SIBGRAPI.2018.00047
- [24] MathWorks. Object Detection. https://www.mathworks.com/discovery/object-detection.html Last accessed 12 june 2023. n.d.
- [25] J. Benito-Picazo, E. Domíngues, E. J Palomo, and E. López-Rubio. "Deep Learning-Based Security System Powered by Low Cost Hardware and Panoramic Cameras". From Bioinspired Systems and Biomedical Applications to Machine Learning. IWINAC. Lecture Notes in Computer Science(). 11487 (2019). Last accessed 12 june 2023, pp. 317–326. DOI: https://doi.org/10.1007/978-3-030-19651-6_31.
- [26] The National Encyclopedia. FTP. https://www.ne.se/uppslagsverk/encyklopedi/lÃěng/ftp. Last accessed 6 june 2023. 2023.
- [27] The National Encyclopedia. TCP/IP. https://www.ne.se/uppslagsverk/encyklopedi/lÃěng/tcpip. Last accessed 6 june 2023. n.d.
- [28] A. Sathyanarayanan. What is FTP Server? https://www.educba.com/what-is-ftp-server/. Last accessed 6 june 2023. n.d.

- [29] IBM Documentation. Socket programming. https://www.ibm.com/docs/en/i/7.1?topic=communications-socket-programming. Last accessed 6 june 2023. 2021.
- [30] University Collage London. What is SSH and how do I use it? https://www.ucl.ac.uk/isd/what-ssh-and-how-do-i-use-it. Last accessed 6 june 2023. 2023.
- [31] Robert G. Byrnes Daniel J. Barrett Richard E. Silverman. "SSH, The Secure Shell: The Definitive Guide" (2005). Last accessed 6 june 2023, pp. 1–2.
- [32] Colin Pace. What Is UX (User Experience) Design A Quick Guide. https://uxcel.com/blog/what-is-user-experience-ux-design. Last accessed 4 june 2023.
- [33] Peter Merholz. Peter in Conversation with Don Norman About UX Innovation. https: //web.archive.org/web/20131207190602/http://www.adaptivepath. com/ideas/e000862/ Last accessed 24 january 2023. 2007.
- [34] ISO. Ergonomics of human-system interaction Part 210: Human-centred design for interactive systems. https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en. Last accessed 5 june 2023. 2019.
- [35] Mattias Arvola. *Interaktionsdesign och UX*. 2:2. Last accessed 6 june 2023. Studentlitteratur, 2021.
- [36] OKdo. https://www.okdo.com/getting-started/get-started-with-rock-4-se-and-debian/ Last accessed 14 april 2023.
- [37] Insta360. https://github.com/Insta360Develop. Last accessed 23 march 2023.
- [38] Daniel Goderik and Albin Westlund. "Detection and Tracking of Elephants using Seismic Direction of Arrival Estimates" (2023). Master Thesis, Linköping University.



Appendix A

A.1 Backend

A.1.1 FTP - Upload image for Windows

```
void upload(std::string& param)
{
   std::string ftpDirectory = "/360camera/";
   std::string fullFtpPath = ftpDirectory + param;

HINTERNET hInternet = InternetOpen(NULL, INTERNET_OPEN_TYPE_DIRECT,
        NULL, NULL, 0);

HINTERNET hFtpSession = InternetConnect(hInternet, "FTP_HOST",
        INTERNET_DEFAULT_FTP_PORT, "FTP_USER", "FTP_PASSWORD",
        INTERNET_SERVICE_FTP, INTERNET_FLAG_PASSIVE, 0);

FtpPutFile(hFtpSession, param.c_str(), fullFtpPath.c_str(),
        FTP_TRANSFER_TYPE_BINARY, 0);

std::cout << "File Uploaded = " << fullFtpPath.c_str() << std::endl;
InternetCloseHandle(hFtpSession);
InternetCloseHandle(hInternet);
}</pre>
```

A.1.2 FTP - Upload image for Linux

```
void upload(std::string& param) {
  std::string ftpPath = "/360camera/";
  std::string fullFtpPath = ftpPath + param.c_str();
  // CFTPClient is allocated on the main function's stack
  CFTPClient ftpClient;
  // Initialize a session
  ftpClient.InitSession(FTP_HOST, 21, FTP_USER, FTP_PASSWORD);
    // Upload image (relative path to the executable) to FTP server
    if (ftpClient.UploadFile(param.c_str(), fullFtpPath))
      std::cout << "File uploaded!\n";</pre>
    }
    else
      std::cout << "Error.\n";</pre>
  // Explicit session cleanup
  ftpClient.CleanupSession();
  remove(param.c_str());
```

A.1.3 FTP - Download image

```
router.get('/', (req, res) => {
   const c = new Client()
   c.connect(ftpServer)
   c.on('error', (e) => {
      c.end()
      res.status(500).send(e)
   })
   c.on('ready', function () {
      c.list('/360camera/', function (err, list) {
         console.log([err, list]);
         if (err) {
            c.end()
            return res.status(500).send(err)
         list.reverse();
         let result = [];
         for (let i = 0; i < 3 && i < list.length; ++i) {</pre>
            result.push(list[i]);
         res.send(result)
         c.end()
      })
   });
})
```

```
router.get("/:file", (req, res) => {
  const c = new Client();
  c.connect(ftpServer);
  c.on("error", (e) => {
     res.status(500).send(e);
     c.end();
  });
  c.on("ready", function () {
      c.get("/360camera/" + req.params.file, function (err, content) {
         if (err) {
           c.end();
           return res.status(500).send(err);
         }
        res.header("content-type", mime.lookup(req.params.file));
         content.pipe(res).on("end", () => {
           c.end();
        });
     });
  });
});
```

A.1.4 Socket for Linux

```
int server_fd, new_socket, valread;
  struct sockaddr_in address;
  int opt = 1;
  int addrlen = sizeof(address);
  char buffer[1024] = { 0 };
  // Creating socket file descriptor
  if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {</pre>
    perror("socket failed");
    exit(EXIT_FAILURE);
  // Forcefully attaching socket to the port 5555
  if (setsockopt(server_fd, SOL_SOCKET,
    SO_REUSEADDR | SO_REUSEPORT, &opt,
    sizeof(opt))) {
    perror("setsockopt");
    exit(EXIT_FAILURE);
  address.sin_family = AF_INET;
  address.sin_addr.s_addr = INADDR_ANY;
  address.sin_port = htons(PORT);
  // Forcefully attaching socket to the port 5555
  if (bind(server_fd, (struct sockaddr*)&address,
    sizeof(address))
    < 0) {
    perror("bind failed");
    exit (EXIT_FAILURE);
  if (listen(server_fd, 3) < 0) {</pre>
    perror("listen");
    exit(EXIT_FAILURE);
  if ((new_socket
    = accept(server_fd, (struct sockaddr*)&address,
       (socklen_t*) &addrlen))
    < 0) {
    perror("accept");
    exit(EXIT_FAILURE);
  }
```

```
valread = read(new_socket, buffer, 1024);
    printf("%s\n", buffer);

std::stringstream ss{ buffer };
    std::string line;
    std::getline(ss, line, '/');
    std::getline(ss, line, ' ');
    option = stoi(line);
    memset(buffer, 0, sizeof(buffer));

// closing the connected socket
    close(new_socket);
    // closing the listening socket
    shutdown(server_fd, SHUT_RDWR);
    std::cout << "Client disconnected.\n";</pre>
```

A.2 Frontend

```
function ShowImage(imageObject) {
   const renderAfterCalled = useRef(false);
   const [image, setImage] = useState([]);
   useEffect(() => {
      if (!renderAfterCalled.current) {
         fetch("http://localhost:8001/ftpAPI/" +
             imageObject.image).then(res => {
            if (res.status !== 200) {
               console.log("Something went wrong...");
               return;
            }
            res.blob().then(data => {
               setImage(URL.createObjectURL(data));
            });
         })
         renderAfterCalled.current = true;
   }, []);
   return image ? (
     <Box image={image} />
   ) : null
```

```
const handleClick = (e) => {
     e.preventDefault();
     console.log("Loading...");
     const option = 1;
     fetch("http://" + ipAdress + ":5555/" + option, {
        method: 'post',
        headers: {
           'Accept': 'application/json',
            'Content-Type': 'application/json'
        }
     })
        .then(JSON.parse(option)) \ensuremath{//} It parses the output
        .catch(function (error) {
           console.log("error---", error)
        })
     setTimeout("window.location.reload(false)", 15000);
  }
```