A METHODOLOGY FOR VISIBLE LIGHT COMMUNICATION IN SWARM ROBOTIC SYSTEMS

by

Alexander Maxseiner
A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Master of Science
Electrical Engineering

Committee:

____________________	Dr. Daniel M. Lofaro, Thesis Director
____________________	Dr. Cameron Nowzari, Committee Member
____________________	Dr. Brian Mark, Committee Member
____________________	Dr. Monson Hayes, Department Chair
____________________	Dr. Kenneth S. Ball, Dean, College of Engineering and Computing
Date: ________________	Fall Semester 2021 George Mason University Fairfax, VA

A Methodology for Visible Light Communication in Swarm Robotic Systems

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

By

Alexander Maxseiner
Bachelor of Science
George Mason University, 2019

Director: Dr. Daniel M. Lofaro, Professor
Department of Electrical and Computer engineering

Fall Semester 2021
George Mason University
Fairfax, VA

Dedication

I dedicate this thesis to Mom, Dad, Sister and family that has supported me over the years.

Acknowledgments

I would like to thank Dr. Daniel Lofaro for his guidance and support throughout this research. I would like to also thank Dr. Cameron Nowzari and Dr. Brian Mark for being part of my Thesis Defense committee. I would like to thank Donald Sofge for supporting me through the collaboration with NRL.

Table of Contents

List of Tables

List of Figures

Abstract

A METHODOLOGY FOR VISIBLE LIGHT COMMUNICATION IN SWARM ROBOTIC SYSTEMS

Alexander Maxseiner

George Mason University, 2021

Thesis Director: Dr. Daniel M. Lofaro

Robotic swarms can benefit from means of communications between agents to perform most collaborative tasks. A communication method that scales linearly with the swarm size is required as the number of agents increases significantly. We propose a method that communicates inherently with proximity, as opposed to traditional fully connected communications. Also knowing swarm state information can improve the efficiency of the swarm. Visible Light Communication (VLC) using a camera and LEDs solves two problems for swarms at once, communication and the location of the sending agent. Information from the frequency of a blinking LED can be inferred over time. Parsing data from many agents in a single agent's frame simultaneously is possible due to using a camera and segmenting the frame. The location of the sending agent can be extracted in the local frame of the receiving agent with the camera frame and the centroid of the light source. In this method of communication, the receiver's communication rate is scaled with the number of agents in "frame". When the camera is not being used to communicate, it can be used to collect information from the environment or usage for other tasks. Methods to perform communication and localization using LEDs paired with Digital and event-based cameras are presented. Methods to improve the data rate using Frequency Shift Keying(FSK) and

Multiple Frequency Shift Keying(MFSK) were presented and tested for their effectiveness. Results showed a low bit-error rate using a camera while inferring spatial locality of the sending agent, showing the feasibility for VLC for robotic swarms.

Chapter 1: Introduction

Swarm robotics is an important research topic due to the potential of many simple robots to accomplish complicated goals. In our research, swarm robotics takes inspiration from nature where social insects work together to accomplish an overall goal. In nature, many simple insects are able to accomplish complex tasks out of the reach of an individual insect. Each species has a unique way of communicating and working together to complete their tasks. Swarm robotics aims to study methods and individual agents to mimic these behaviors in order to accomplish complicated tasks. An issue with swarm robotics that appears consistently is scaling the agents into large numbers. Scalability needs to be addressed for every aspect of swarm robotics, cost, perception, communication.

A recent example of swarm robotics is the Kilobot swarm[1]. Using this platform thousands of agents can collectively form shapes. Each individual agent costs about $14 and takes 5 minutes to set up. Thousands of these agents can be deployed at a time and instructed into different formations.

A key segment of swarm robotics is communication. A majority of current swarm research assumes communication to be ideal. During my research I have found that this assumption is inaccurate. Robotic swarm communication is particularly challenging because the number of agents in the swarm is not fixed and in most cases, like a bee hive or ant farm, there are hundreds or thousands of individual agents. In the computer world, it's common to have all agents or computers to talk to and communicate information with each other. In the natural world, each agent cannot communicate with every other agent. Nature has shown that fully connected communication is not needed and many times not desired. As the number of agents increases, the number of communication links required to have a fully connected swarm increases exponentially with the number of agents. This effect and the need for scalability is shown in Figure 1.1. This affects the bandwidth, latency, complexity,

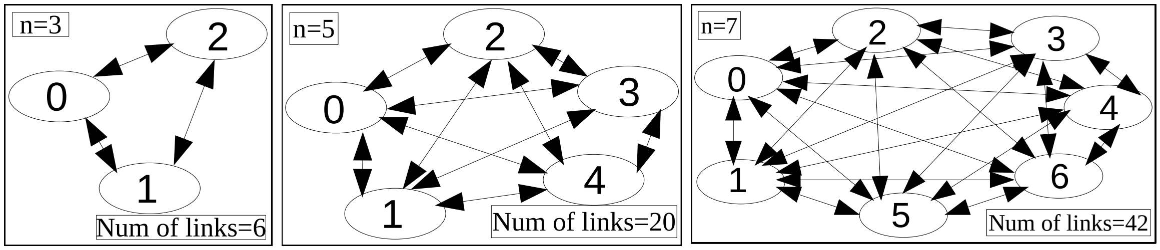

Figure 1.1: A demonstration of how communication links exponentially increase($n(n-1)$ links), where n is the number of agents. Additionally, each agent must receive information from $n-1$ agents in a fully connected communication network. This demonstrates how with an expanding swarm size communication between every agent is not ideal from a communication coordination perspective.

and other key aspects of the network and swarm system. This highlights the need for swarm communication to scale. Traditional forms of communication over structured networks over radio, such as Wi-Fi, do not transfer well to swarms because of the exponential increase in communication links.

Key Questions:

- (Q1) Can we design a swarm communication system such that the communication links increase linearly with swarm size?
- (Q2) Can we gain swarm state information from a limited or non-ideal swarm communication network?

Technical Challenges:

- (TC1) Designing a communication setup such that each individual agent doesn't rely on a centralized hub (Q1)
- (TC2) Design a communication setup such that an individual agent can receive state information utilizing the properties of the communication system (Q2)

- (TC3) Design a communication setup such that communication is only with agents of most immediate importance (Q1, Q2)

The focus of this thesis is on communication methods that scale linearly with the number of agents (Q1) while simultaneously providing state information to the agents (Q2). Visible Light Communication (VLC) is utilized to answer Q1, see Section 3.3. To answer Q2 the inherent spatial data collected via VLC is used to naturally provide state information about each of the agents being communicated with, see Section 3.3.6. The work presented in this paper directly builds upon my previous work done on VLC for swarm robotics, Maxseiner et. al. [2, 3].

1.1 Importance of Communication for Swarms

Sharing information between agents is important for swarms to operate efficiently. Majority of swarms deployed today rely on centralized communication systems to allow the agents to communicate. For example, Intel's drone light show[1] uses a centralized communication system that assigns movements based on agent state information, such as GPS coordinates, and sends them directly to the agents from the centralized communication system. However, a centralized communication system is not a realistic way to implement a swarm for real-world situations because as swarms grow in size, bandwidth limitations will be surpassed. Additionally, it is common to assume a perfect connection with the centralized communication system. This however is not a valid assumption when implementing a real-world swarm. Real world limitations that may cause problems include bandwidth limitation, channel limitations, range limitation, etc.

1.2 Bio-inspired Visible Light Communication

Swarms in the natural world communicate and operate without a centralized communication system. For example colonies of ants, flocks of birds, and schools of fish perform complex

[1] Intel's Drone Light Show (2021-10-15): `https://inteldronelightshows.com/`

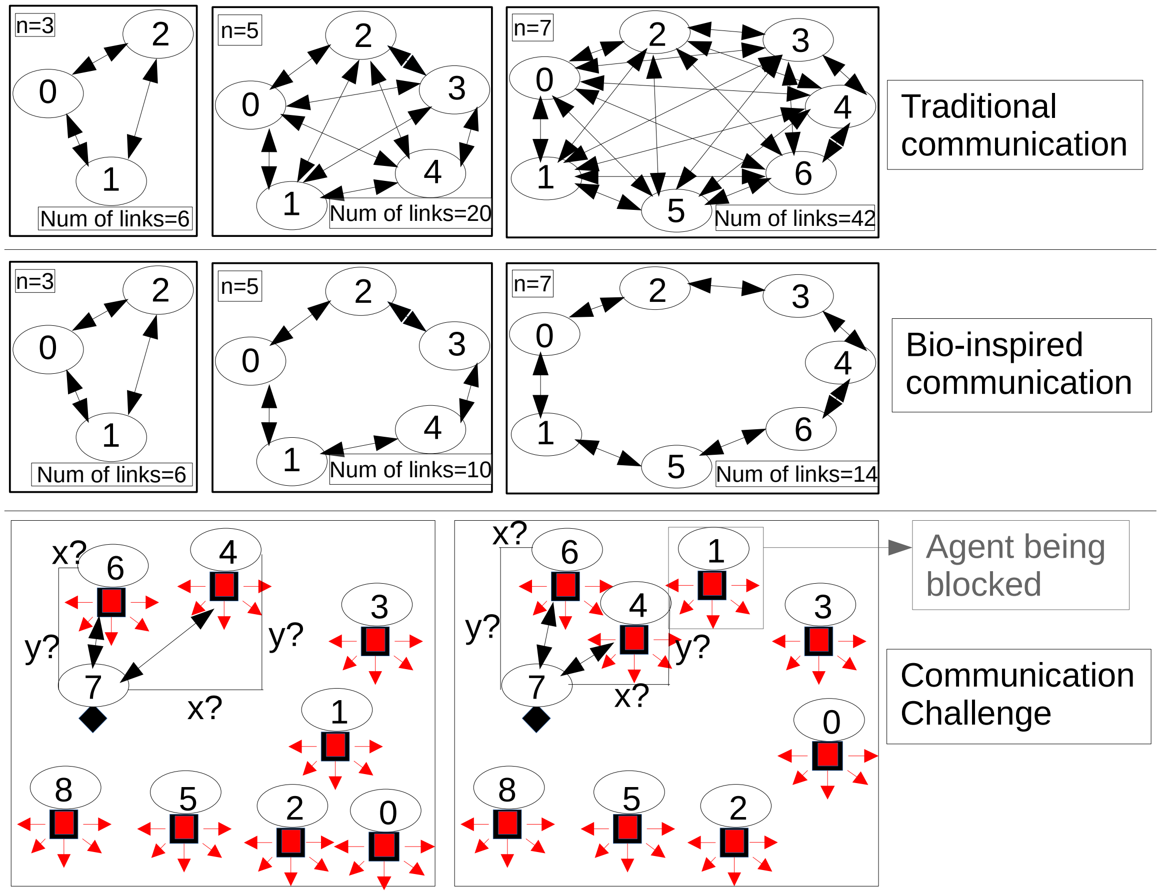

Figure 1.2: Various types of communication are outlined in this illustration, the double sided arrows imply two-way communication. (**TOP**) the traditional fully connected communication, this form of communication does not scale linearly with the number of agents in the swarm, making this form of communication undesirable. The number of communication links is $n(n-1)$ with n being the number of links. (**Middle**) The communication used in nature involving communicating with only the agents near you, here the agents only communication with the agents next to them, this form of communication does scale linearly, number of links is $2n$. (**Bottom**) The communication system desired in (Q1-Q2) and outlined in (TC1-TC3). As seen, the receiving agents are communicating with the agents nearby, solving the problem of having too many communication links(TC1 and TC3). Additionally, in the (Bottom Right) illustration we see that agent 4 is blocking agent 1 from sending information to agent 7. Also we would like to know where the sending agent is in respect to the receiving agent. This is shown with the x and y offset(TC2). This is showing how the communication links should be inherently filtered so that irrelevant information isn't received.

swarm tasks without the need for a centralized communication system. This is done via local communication between agents via smell, sounds, visual perception, etc. In the modern world, we can communicate visually using what is known as Visible Light Communication (VLC). VLC was partially inspired by communication between agents in nature such as fireflies' and deep sea fishes' use of bioluminescent glow to communicate. See Section 2.4 for the latest work in VLC.

Communication between agents using light-based methods was implemented by using modified lighter-than-air autonomous agent (LTA^3) platforms [4, 5], which build on the Georgia Tech Miniature Autonomous Blimp (GT-MAB) [6] platform design. An example of how light-based communication would look on these agents is shown in Figure 3.1.

Chapter 2: Background

2.1 Communication

Communication in engineering systems has existed for over one hundred years now. Radio signals were transmitted over the Atlantic in 1901 and the first radio broadcast happened in 1920. Since then lots of progress has been made to regulate, modernize, and efficiently use the spectrum[7]. Communication in engineering systems can take many forms including Radio, Audio, Light, and non-verbal. In each case shared information is transmitted over a medium to constitute communication.

2.1.1 Swarm Communication

There has been extensive work in developing swarm algorithms where seamless communication between agents is assumed or that there is a centralized communication hub. For testing purposes a centralized node can be used to communicate[8] or a mesh network[9]. However when testing swarm behavior in unknown environments there may not be a communication system that can be available beforehand. In this situation, communication among the agents with no middle communication hub is required.

There are two types of communication for swarms, implicit and explicit. Implicit and explicit communication have varied definitions based on the context of the system. In general, explicit communication is when you intend a message for a specific receiver. Implicit communication is an agent's behavior that is observed as communication. Variations in the definitions and systems are explored below.

Work comparing these methods of communication in simulation has been done[10]. Three communication methods were compared, IR, Vision, and mechanical interaction.

IR is explicit communication, while vision and mechanical interaction is implicit communication. IR won in overall performance metrics for a stick pulling experiment.

Explicit Communication

Explicit (or direct) communication is exhibited when agents directly talk to each other; this includes one agent talking to another agent via message passing or one agent broadcasting a message to other agents in the swarm [11, 12]. Breazeal et. al.[13] define explicit communication as sharing specific information with a receiver.

Implicit Communication

Implicit (or indirect) communication is exhibited when an agent leaves clues or signs as it moves in the environment. Additionally, Breazeal et. al.[13] define implicit communication as "conveying information that is inherent in behavior but which is not deliberately communicated".

Where implicit and explicit communication differ in definition is with behavioral implicit communication. Some define conveying information without directly stating information, such as a head nod, behavioral implicit communication[14].

Many existing swarms from nature communicate implicitly. For systems that scale agents as high as swarms do, implicit communication on some level is necessary as relying solely on explicit communication between agents would be cumbersome. Robotic swarms are a natural match for implicit communication because implicit communication is robust and scalable compared to explicit communication[15].

Implicit communication using robotic swarms has been demonstrated by Wang et. al.[16]. In this work, simulations demonstrated the ability to transport large objects using many small agents without any explicit communication among the agents. The BlueSwarm project uses implicit communication to determine an individual agent's behavior using LEDs and cameras. Determining if there are other agents to either side of the individual agent allows the individual agent to make decisions on its own behavior[17].

Pheromone communication is a bio-inspired form of implicit communication that is a popular communication method for large scale systems. PheromoneSimBot, a simulator to develop a model of individual agent behavior in a swarm setting using pheromones to communicate[18]. Additionally "virtual pheromones" have been explored as a method to control robotic swarms[19, 20].

Comparing explicit and implicit communication

Comparisons between Implicit and Explicit communication have been done. McPartland et. al. [21] compared implicit communication to explicit communication by having two teams compete against each other.

Combining explicit and implicit communication

There has been some work outlining the need to combine explicit and implicit communication in robotic systems. The benefits of combining both is the ability to directly communicate information and infer information at the same time. For certain systems this will not be ideal. This is a good option for communication inside a swarm, as you can exchange direct information and infer position or behaviour about other agents at the same time. The downside is that the explicit information rate may decrease depending on the system. There are also benefits from a human-robot interaction(HRI) and a human-swarm interaction standpoint[12]. Humans are good at making inferences, we do them all the time. However the human may need to be familiar with the robots possible behavior before being able to infer much about the robot.

2.2 Radio Communication

Radio communication has been around for many years, it is the most developed and reliable form of communication on land. We use it everyday in our phones, cars, and homes. However there are limitations on the spectrum as the number of users increase, therefore the utilization of the spectrum must be better [22]. Radio communication comes in many sizes and shapes from a small XBee module up to a radio antenna to communicate with a satellite.

2.2.1 Radio Communication in swarms

Wireless network radio communication is a popular approach for swarm communication [23–30] and communication systems in general. Wireless network radio communication is the easiest way to get a message from a single agent to many other agents from an engineering perspective. However, using wireless radio communication poses bottleneck issues when expanding the swarm [11] and many use a centralized node, which is not practical for most applications of swarms. Additionally, some downsides in radio communication are that jamming techniques are well established [31–33] for wireless radio communication. Another downside is that many wireless radio communication relies on a centralized hub for routing, for example a Wi-Fi router or cell tower. For swarms aimed to aid during disaster scenarios this would be a serious hindrance. Search and Rescue(SAR) swarms, which has a rich literature [34–36], would be obsolete in situations where the swarm depended on a centralized hub in disaster situations. Additionally, moving a centralized messaging system to be set up in an area during a disaster situation is prohibitive due to potential lack of resources and human accessibility.

Radio communication works well for small swarms or multi-agent systems, where a small swarm or multi-agent system is defined as having less than 30 agents. As the density of the swarm increases, there would be some level of parsing of radio messages required for each agent. With each additional agent communicating takes away from swarm intelligence, by adding additional compute time for each agent.

2.2.2 Cognitive radios

Many of the recent advances in radio communication relate to cognitive and software defined radios. Cognitive radios aim to deal with the increased demand for spectral resources by dynamically allocating and using the spectrum. In the past radios have been statically allocated due to the limitation imposed by the dated hardware[37, 38]. Cognitive radios aim to intelligently decide which channels to use for communication in order to optimally utilize the spectrum [39–41].

One of the researched solutions is to use particle swarm optimization to reduce the computation cost in genetic algorithms being used the solve problems of Cognitive radios [42–44]. Additionally, swarming behavior found in nature has been used to solve some of the problems posed by Cognitive Radios[45, 46].

2.2.3 Cognitive radios and swarms

The problem being solved by cognitive radios is similar to the problems swarm robotics have using radio communication. There are a growing number of users that need to use a limited spectrum, however everyone isn't using the spectrum at the same time. There is a method to detect which frequencies are open, then send your message on the open frequencies. There has been work adapting cognitive radios for multi-agent systems [47]. These advances could be a possible solution to bottleneck issues that would be encountered when scaling swarms to larger numbers. However, many of these techniques require FPGAs and Software Defined Radios(SDR). This advanced hardware will increase the per unit cost of an agent. This would not allow these agents to be scaled on the order of thousands until these products can be miniaturized and the cost is driven down significantly.

2.3 Audio communication

The primary market for acoustic or audio communication, other than humans talking, is underwater communication. Acoustic signals are the primary medium for communication underwater. Some of the first forms of this were in U.S. submarines during WWII. Underwater acoustics is often regarded as the most difficult medium for communication as there are many issues with multipath, attenuation and noise and the Doppler effect to pay attention to[48, 49].

2.3.1 Audio communication in Nature

Many animals and insects communicate using acoustics including birds, frogs, crocodilians, and mammals [50]. Birds have been shown to use acoustics and visuals for mating[51]. A species of Blind cavefish have evolved to use "Sharp Clicks" during certain behaviors for foraging[52]. Treefrogs use vocal or acoustic communication to determine mates, for communication between rivals, or relatives[53]. Lastly, as everyone knows crickets communicate using acoustic communication[54].

2.4 Visible Light Communication

Recent work has shown the feasibility of VLC as a useful communication protocol using LEDs [55–62] for a variety of reasons including the potential for high data rate(500Mb/s bidirectional[63]), low cost of LEDs, and ability to combine with positioning estimation. VLC has a couple of advantages compared to radio communication. LEDs also double as a lighting fixture for a room or area. LEDs cost less money, the energy required for LEDs is much less than that of an antenna [64], and there are many areas where visible lights are already used to communicate with humans such as traffic and automobile lights. VLC being visible to everyone in the immediate area can be seen as an advantage or disadvantage from a security perspective because everyone can see what the agent is displaying. However, the light cannot penetrate most walls making the communication limited to the room or area. From a human robot interaction or human swarm interaction perspective, VLC can add another line of communication that is quick and natural to humans.

Another option for VLC is communicating while the LEDs appear to be turned off. Tian et. al.[65] developed a platform that communicates at a 1.3-m distance with 1.6-Kbps data rate with LEDs emitting extremely low luminance. This work presents interesting results and security implications not permitted by other forms of VLC.

2.4.1 VLC in robotics

The BlueSwarm project[17,66] is a bio-inspired approach to organize swarms of underwater fishing using LEDs and cameras. These LEDs and cameras are used to detect and estimate other robotic fish to decide how to move. VLC has been implemented in a couple of different ways using robots for communication and/or position information. Murai et. al. [67] use VLC as a method to provide GPS and show hazards ahead for delivery robots in a hospital. This was done by replacing the lights in the hospital with LED devices that would be able to give the robots information. The information given to the robot can be a hazard ahead or to map the robot's position in the hospital using sensors and VLC. Sharifi et. al. [68] also used VLC as a method of localization using four LEDs for a single differential drive robot. This was done by using a photodiode and allocating a unique frequency ID to each LED and measuring the received signal strength from each LED. Image sensors and LED lights have been used to perform position estimation within a five centimeter accuracy[69] and centimeter of accuracy [70]. Additionally, On-Off-Keying(OOK) has been used for control of robots indoors[71]. OOK if a simple form to convey information using amplitude modulation. However, this technique is prone to disruptions due to noise. Which led us to use FSK and MFSK instead.

Visible Light Positioning(VLP)

Visible Light has been studied in robotics for communication purposes. An emerging field is using light to gather positioning estimates, or sometimes referred to as Visible- Light-Positioning(VLP) [72–74]. This has been shown feasible indoors, as the sun outside would cause many issues, however indoors there can be many reflections and could lead to multipath. Gu et. al. [75] showed the possibility of multipath in indoor environments, then developed a solution using nonlinear estimation.

Combination of Visible Light Positioning and Communication

The previous studies mentioned using VLC to localize or give information directly to the robot while Rust et. al. [76] used VLC as a method of communication and localization to remotely operate underwater vehicles (ROVs) for nuclear reactor inspection. This is similar to the approach generated in this thesis.

2.4.2 Pure optical Visible Light Communication

Hunter et. al. [77] showed a feasible approach for optical communication using an LED flashlight as the transmitter and digital camera as the receiver without line of sight. VLC for many communicators and receivers as line of sight is a double edged sword for robotics. For high density swarms, limiting the communication to neighbors is better so that the communication spectrum doesn't get clogged with data, adding additional data that doesn't require line of sight could hinder light communication in a dense or large swarm, but could be a large help for other places in robotics. Takai et. al. [78] present a system using an LED emitter and CMOS image sensor with a camera receiver that can achieve a 15Mbps data rate per pixel. This shows promise for much higher data rates using Cameras and LEDs performing VLC. However, these works have been aimed toward the automotive market. Conveniently cars already have bright LEDs required by law, Roberts et. al. [79] propose a positioning system using headlights and taillights.

An open-source platform called OpenVLC [80], has been developed using a BeagleBone black that provides a physical layer, essential medium access primitives, and interaction using Internet protocols. This serves as a great starter kit for people looking to get into VLC, and because it is open source this allows people to build the platform out.

LED-to-LED communication

Communication using LEDs as transmitters and receivers has been studied and shown to be feasible [81, 82]. There are aspects that could be attractive for swarms. Namely being lightweight, low cost, and efficient communication. However for our case, LEDs would need to be placed around the agent to infer the spatial location. This form of localization would be much lower resolution than using a camera.

2.4.3 Infrared

There has been some prior work in using IR sensors for communications in swarm robotics[83–85]. IR communication has an added benefit of performing obstacle detection with the IR emitter and receiver[85]. IR sensors also save power per agent when compared to using radio. However, the data rate for IR sensors is less than the data rate for radio communication. There has been some relevant work in swarm robotics using optical communication, specifically with UUVs. Berlinger et. al.[86] used a photodiode to guide their robotic fish to the goal area. Additionally, the Kilobots swarm mentioned in the Introduction uses IR for coordination[1].

2.4.4 Li-Fi

Li-Fi has been a popular research area and many products have been released in the past couple years[87–89]. Li-Fi is not the ideal usage for swarm robotics, because it inherently adds a centralized communication hub and possibly limits the area in which swarms can be used. However Li-Fi is worth mentioning because many of the technologies used to develop Li-Fi can be possibly carried over into decentralized VLC efforts. Additionally, there is promise of using usb dongles that would be developed for use with an overhead LED router, however can be configured to communicate with each other with no overhead LED required. The possible system would use the usb dongles with the sending and receiving of light hardware to communicate using the usb dongles while placed on agents. However, there is no realized spatial information, the increase in data rate may be able to make up for that.

2.5 Why communication for swarms is different

Communication for swarms is different from communication for other engineering systems. In typical engineering systems there is a defined number of agents and a defined range for the communication to operate. Typically the more information sent and received, the better. However, for swarm robotics the need for large amounts of data is not as necessary. Effective communication is required, however the communication must scale up above thousands of agents in any environment. The requirement for every agent to communicate with every other agent in the swarm is undesirable. Communication between thousands of agents at a time is a logistical and computational nightmare. The information transmitted must be enough to assist the swarms' intelligence, but not enough to slow down information transmission.

2.5.1 Event based cameras

Event-based cameras are a recent advancement in computer vision and cameras. An event-based camera provides asynchronous local changes in brightness[90, 91]. Each pixel in the camera's resolution is independent and continuous. For a great survey of Event-based vision written by the same author as many of the cited papers for event-based vision see[92]. There is no color, but the event based cameras have a fast temporal resolution. The DVXplorer has a temporal resolution of 200 microseconds[1]. Allowing the information rate to be much greater than using a digital camera, while inferring spatial locality. These bio-inspired cameras are expensive but have the possibility to revolutionize robotics. There are datasets[93, 94] and simulators[95] published in order to promote this line of work, without buying the sensor.

[1]DVXplorer Event Based Camera (Date Accessed: 2021-09-01) https://inivation.com/wp-content/uploads/2021/08/2021-08-iniVation-devices-Specifications.pdf

Localization

There has been many algorithms to utilize this high and unique data. Gallego et. al. developed an Extended Kalman Filter(EKF) for localization[96]. Additionally, Gallego et. al.[97] developed an approach to track the pose of an event-based camera in 6-DOF using a Bayesian filter methodology. This was then compared with state of the art methods using frame based cameras and the tracking was accurate. Jin et. al. [98] developed a camera relocalization system using a convolutional neural network (CNN) and long and short-term memory (LSTM). Another 6-DOF localization method was developed by Bryner et. al.[99] using a principled event generation model within a maximum-likelihood framework. Some earlier work using a Condensed Particle filter tracker was used to self localize the camera[100].

Object detection

Computer vision techniques including object detection have also been explored. Particularly, Ceolini et. al. [101] utilizes an event based camera along with other sensors to detect hand gestures. Ghosh et. al. [102] developed a real-time object detection and tracking method using an event-based camera with a CNN and an Field Programmable Gate Array(FPGA) to process in real time. Sokolova et. al.[103] identified human gait using an event-based camera that worked comparable with state-of-the-art methods using digital cameras.

Event-based cameras comparison

The literature on event-based cameras is not as rich as other popular fields and a majority of the papers show that event-based cameras provide similar results. This is promising because event-based cameras and its literature is young.

Chapter 3: Methodology

3.1 Solution setup

Cameras and LEDs will be used to communicate information between agents. By using a camera we are limited by line-of-sight of the agent, thus filtering out many except the agents nearby. Dividing the receiving agents camera frame into many Regions of Interest(ROI), gives a method for many stream of LED information to be received simultaneously. This solves the problems stated in TC1 and TC3. This method of dividing the camera frame is shown in Figure 3.1.

By nature of using a Camera and LED, we can extract the spatial position respect to the agent sending information by calculating the centroid and angle offset. This can be done in each ROI, allowing a receiving agent to receive the position and data from many sending agents nearby simultaneously. This solves the problem of knowing the swarm state information posed in TC2.

3.2 Solution explanation

There are a couple reasons to perform VLC with a camera. First of all, the communication should be between agents with no central hub for information relaying. Using a camera allows for communication and localization, communication by receiving information from LEDs blinking and localization by calculating the centroid and angle offsets of sending agents relative to the receiver. Local agent-to-agent communication is important for swarm communication. As communicating to more than just your neighbors would cause unnecessary inefficiencies throughout the swarm.

Using a camera with filtering we can infer the contour area of the LED's over time. Using

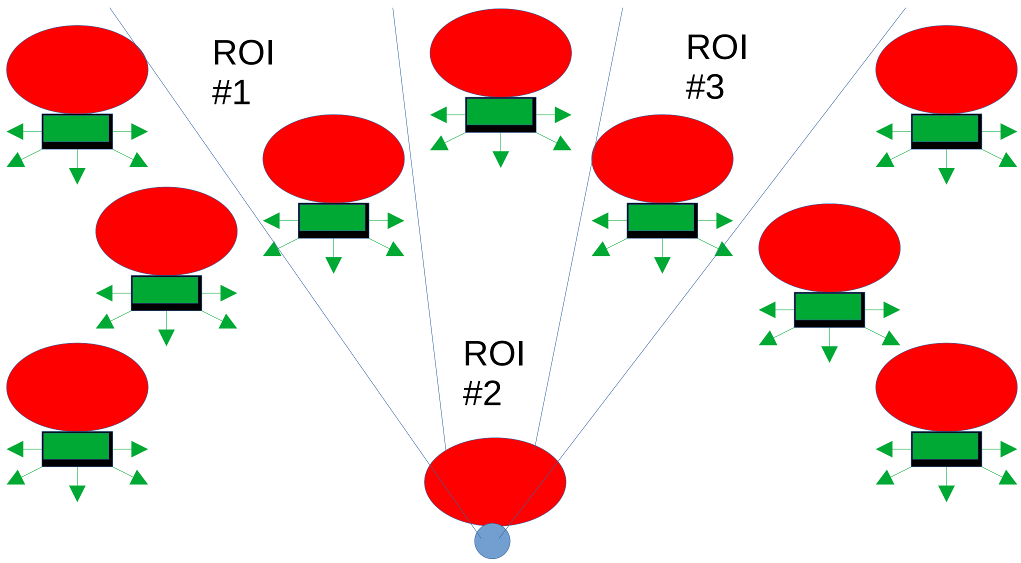

Figure 3.1: Diagram showing how regions of interest would work in a robotic swarm. The circle under the blimp in the center represents a camera. Notice that the other blimps do not have a camera. Eventually blimps would be outfitted with a camera and LED cluster. Showing just the camera or LED cluster signifies the blimp looking for information and possible communicators with LED clusters in a swarm respectively.

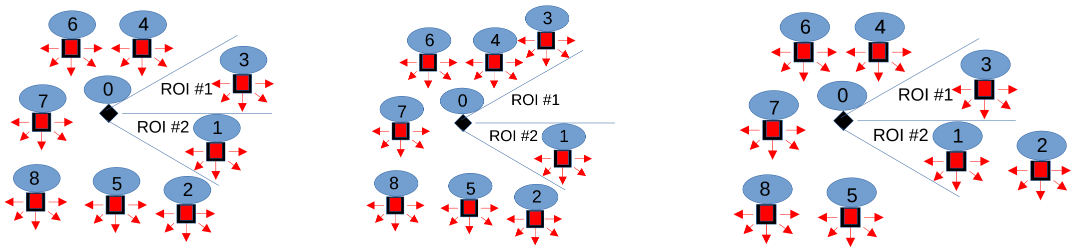

Figure 3.2: Figure showing how ROI would work in a swarm. The black triangle signifies a camera and the red boxes signify LEDs. Every agent would have both, but for the sake of demonstration they are shown to have either in this figure. There are two situations shown, both having two ROIs; the receiving agent would get information from those two agents. In the (**LEFT**) display there is an agent in both ROIs. In the (**MIDDLE**) figure there is no agent in ROI 1, however this still provides information to the agent about the swarm. (**RIGHT**) This is an example, Agent 0 can see and communicate with agent 1 and 3 but not 2. Because Agent 0 could communicate with agent 2 recently this imparts information to Agent 0 that Agent 2 is behind one of the other agents.

this we can use techniques from signal processing to infer the frequency of the LED over time. Then we can alternate this frequency over time to indicate information using Frequency Shift Keying(FSK). Additionally, we will show how we can expand this method to many frequencies to speed up the communication using Multiple Frequency Shift Keying(MFSK).

Blinking used as a communication method has already been done before, however we are proposing a method that allows a receiving agent to infer the spatial locality of the sending agent. Additionally this method allows us to communicate with multiple agents at a time. This is shown in Figure 3.1 and 3.2. This is ideal for robotic swarms.

3.3 Visible Light Communication Algorithm Digital camera

To test the feasibility of VLC for robotic swarms, LEDs will flash on and off at defined frequencies to represent data. The camera will sample the blinking through the video stream to determine the frequency of blinking. This algorithm is shown in Algorithm 1 and defined mathematically below.

To detect the rate at which an LED was blinking, a mask or binary image is made from each image from the video stream, shown in equations (3.12), (3.13), and (3.14). Then the moments and centroid are calculated using equation (3.36). The contour area from the moment calculation is then recorded for performing a Fast-Fourier Transform (FFT) to determine the frequency.

3.3.1 Convert Red-Green-Blue(RGB) image to Hue-Saturation-Value(HSV) image

Consider an image I of resolution $R_x > 0$, $R_y > 0$. Let $P = \{p = (x, y) \in \mathbb{Z}^2 \mid 0 \leq x < R_x, 0 \leq y < R_y\}$ represent the set of pixels in the image. Additionally, $P_{tot} = R_x R_y$ is the total number of pixels in the image frame for any given image. Also images are coming in at F frames per second.

For a given image, R_p, G_p, and B_p represent the RGB values at each pixel location p. Let $0 \leq R_p \leq 255$, $0 \leq G_p \leq 255$, and $0 \leq B_p \leq 255$. The following calculations are done for every pixel in the image.

First we scale down the RGB values

$$R'_p = \frac{R_p}{255},\ G'_p = \frac{G_p}{255},\ B'_p = \frac{B_p}{255}\ \forall p \in P \tag{3.1}$$

$$C_p^{\max} = \max(R'_p, G'_p, B'_p)\ \forall p \in P \tag{3.2}$$

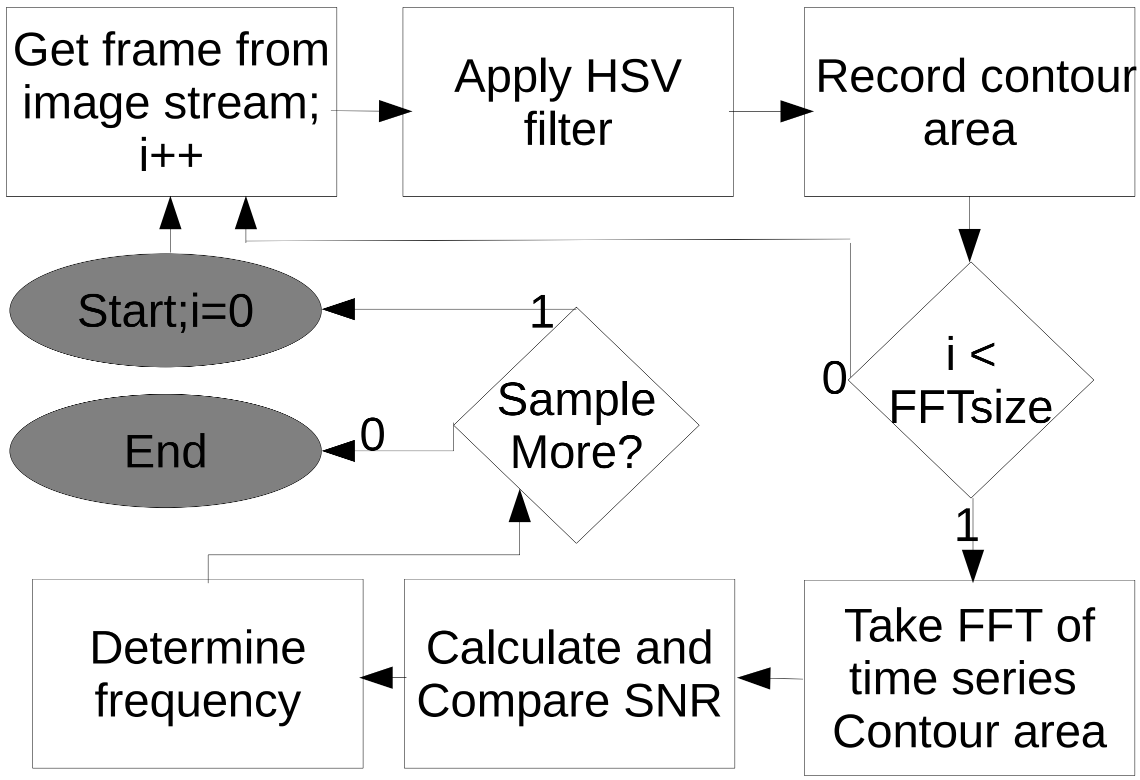

Figure 3.3: Algorithm to parse information from other blimps trying to communication information. Receiving frames from a video stream using a digital camera is how “sampling” is done in this communication system. The image is filtered and the contour area of the binary image is recorded as time series data. The time series data is used to determine the frequency using a FFT. From there on sampling can continue or end. Sampling would continue under FSK or MFSK, however if a singular frequency is to be determined, one FFT bin is required.

$$C_p^{\min} = \min(R'_p, G'_p, B'_p)\ \forall p \in P \tag{3.3}$$

$$\Delta = C_p^{\max} - C_p^{\min}\ \forall p \in P \tag{3.4}$$

Here we calculate the Hue

$$H_p = \begin{cases} 0^\circ & \Delta = 0 \\ \xi(\frac{G'_p - B'}{\Delta} \mod 6) & C_p^{max} = R' \\ \xi(\frac{B'_p - R'_p}{\Delta} + 2) & C_p^{max} = R' \\ \xi(\frac{R'_p - G'_p}{\Delta} + 4) & C_p^{max} = R' \end{cases} \forall p \in P \tag{3.5}$$

where $\xi = 60^\circ$ and mod, refers to the modulo operation.

and saturation

$$S_p = \begin{cases} 0 & C_p^{\max} = 0 \\ \frac{\Delta}{C_p^{\max}} & C_p^{\max} \neq 0 \end{cases} \forall p \in P \tag{3.6}$$

and value.

$$V_p = C_p^{\max}\ \forall p \in P \tag{3.7}$$

now we have HSV for every pixel in the image.

3.3.2 HSV filtering

Now that we have the HSV values for every pixel. A filter can be used to find colors. In the following equations, H_p^f or H_p^{*f}, S_p^f, and V_p^f refer the the HSV filtered binary image.

These values are shown in 3.1.

There are two Hue filtering ranges for a RED led(H_p^f and H_p^{*f}) because the red range for hue loops around as Hue is an angle. Thus mathematical and computer definitions require two filtering ranges.

$$H_p^f = \begin{cases} 1 & H_{pmin} \leq H_p \leq H_{pmax} \\ 0 & \text{else} \end{cases} \quad \forall p \in P \tag{3.8}$$

$$S_p^f = \begin{cases} 1 & S_{pmin} \leq S_p \leq S_{pmax} \\ 0 & \text{else} \end{cases} \quad \forall p \in P \tag{3.9}$$

$$V_p^f = \begin{cases} 1 & V_{pmin} \leq V_p \leq V_{pmax} \\ 0 & \text{else} \end{cases} \quad \forall p \in P \tag{3.10}$$

$$H_p^{*f} = \begin{cases} 1 & H_{pmin}^* \leq H_p \leq H_{pmax}^* \\ 0 & \text{else} \end{cases} \quad \forall p \in P \tag{3.11}$$

Where p is a pixel in an image, I_p is a pixel in an image from the video stream, and the constraints H_{pmin}, H_{pmax}, S_{pmin}, etc. are defined in Table 3.1.

$$\Omega_p = H_p \cap S_p \cap V_p \tag{3.12}$$

$$\Lambda_p = H_p^* \cap S_p \cap V_p \tag{3.13}$$

Where Ω_p represents the first mask, and Λ_p represents the second mask. Ω_p and Λ_p are

Table 3.1: (**TOP**) HSV ranges creating mask for sampling a Red LED. (**BOTTOM**) HSV ranges creating mask for sampling a Blue LED. These values can be adjusted depending on the situation or other colors can be added.

Scale	H_{pmin}	H_{pmax}	S_{pmin}	S_{pmax}	V_{pmin}	V_{pmax}	H^*_{pmin}	H^*_{pmax}
Value	0	10	50	255	175	255	170	180

Scale	H_{pmin}	H_{pmax}	S_{pmin}	S_{pmax}	V_{pmin}	V_{pmax}
Value	110	130	50	255	175	255

calculated for all pixels p in the image.

$$\Gamma_p = \Omega_p \cup \Lambda_p \tag{3.14}$$

The masks are combined using the equation above where Γ_p is the final binary image. This represents the area that our filter determines contains the color in question.

Note, Γ_p, Ω_p and Λ_p contains only 1's and 0's.

3.3.3 Frequency determination

To find the frequency of blinking, we will create a time series function of the contour area. In order to find the contour area from the binary image, a summation is used.

$$C = \sum_{p \in P} \Gamma_p \tag{3.15}$$

C is the contour area for the given binary image.

The contour area is then taken as a "sample" in time and recorded in a contour area time series function c_n.

$$c_n = C \tag{3.16}$$

where $n = 0, 1, 2, \dots, N-1$ and N is the total number of samples.

Algorithm 1 Sequence describing the movement of a binary image sampling to DFT to frequency determination via SNR for FSK.

```
n ← 0
while n < N do
    I ← I_new                                   ▷ Get new data from camera
    Γ_P ← HSV(I)
    for p ∈ P do
        C ← C + Γ_p
    end for
    c_n ← C
    n ← n + 1
end while
C_k = F(c_n)                                    ▷ Using eqn. 3.17
SNR_f0 = SNR(C_k, f_0)                          ▷ Using eqn. 3.19
SNR_f1 = SNR(C_k, f_1)                          ▷ Using eqn. 3.20
if SNR_f0 > SNR_f1 then
    freq ← f_0
else
    freq ← f_1
end if
return freq
```

Once a sufficient amount of data is recorded($n = N - 1$), an Discrete Fourier Transform(DFT) can be taken to deduce information on the frequency of the signal. The spike would show the frequency of blinking. This algorithm is illustrated in Figure 3.3.

$$C_k = \sum_{n=0}^{N-1} c_n e^{-\frac{j2\pi}{N}kn}, k = 0, 1, 2, \dots N-1 \tag{3.17}$$

where C_k is a sequence of complex numbers representing the function of frequency.

With the FFT bins defined as follows

$$\text{FFTbin}_n = n\frac{F}{N},\ n = 1, 2, \dots N \tag{3.18}$$

with each bin representing $\frac{F}{N}$ frequencies, N being the number of samples, and F being the frame rate or sampling rate.

3.3.4 FSK usage

FSK allows for significantly more complicated information to be passed. Using the same communication setup, a digital camera and LED cluster, information can be passed using FSK. Frequencies f_1 and f_0 to represent binary digits 1 and 0 respectively. One sequence of this is shown in Figure 3.4. And sequency of the synchronization combined with information sending is shown in Figure 3.6.

Signal to noise ratio(SNR) calculation for FSK

Given that there will be only two frequencies "blinked" at. Let $f_0 > 0$ and $f_1 > 0$ be the two frequencies to transfer information.

Now the SNR can be calculated.

$$\mathrm{SNR}_{f0} = 10 \log \frac{C_{f_0}}{C_{f_{noise}}} \tag{3.19}$$

$$\mathrm{SNR}_{f1} = 10 \log \frac{C_{f_1}}{C_{f_{noise}}} \tag{3.20}$$

where $f_{noise} = \{f \in F \mid f \neq f_0,\ f \neq f_1\}$. F is the all the possible frequency bins for the computed FFT.

Using the calculated signal to noise ratios, we can determine which frequency is "blinking"

$$F_{blinking} = \begin{cases} f0 & \mathrm{SNR}_{f0} > \mathrm{SNR}_{f1} \\ f1 & \mathrm{SNR}_{f1} > \mathrm{SNR}_{f0} \end{cases} \tag{3.21}$$

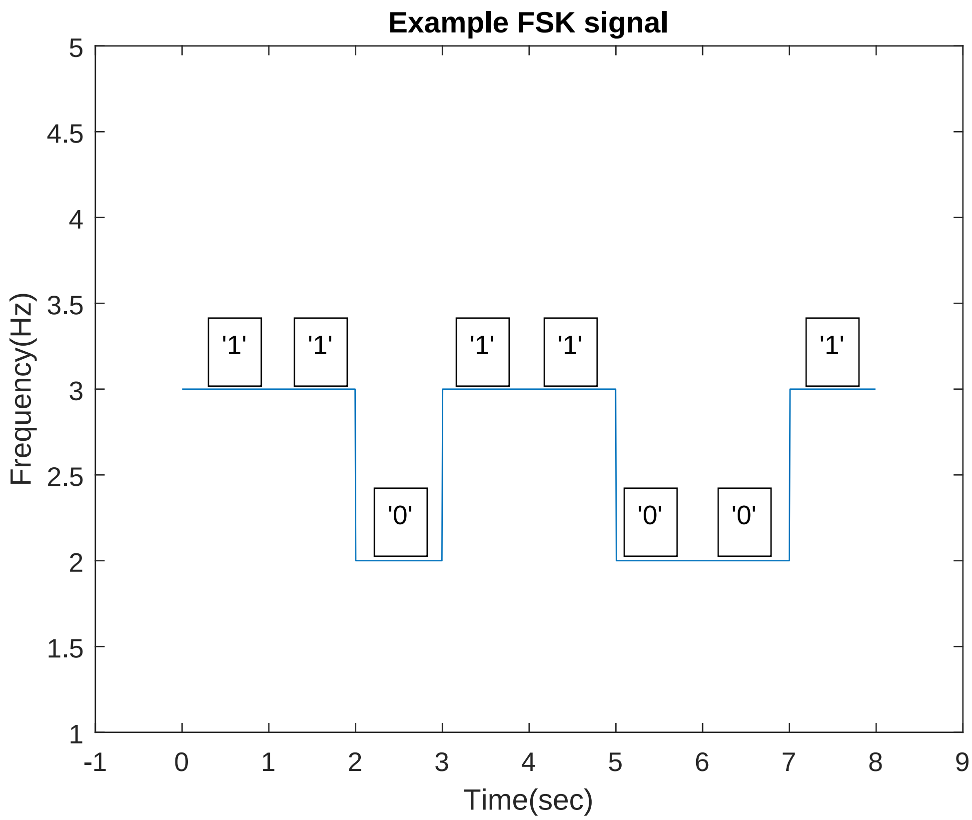

Figure 3.4: Example of ideal communication using FSK without 8-n-1 encoding. Here $f_1 = 3$Hz and $f_0 = 2$Hz. A sampling period for a certain frequency here is one second. When the signal is oscillating at 3Hz a binary '1' is represented and when oscilating at 2Hz a binary '0' is represented. The signal show represents a byte of information containing '11011001'.

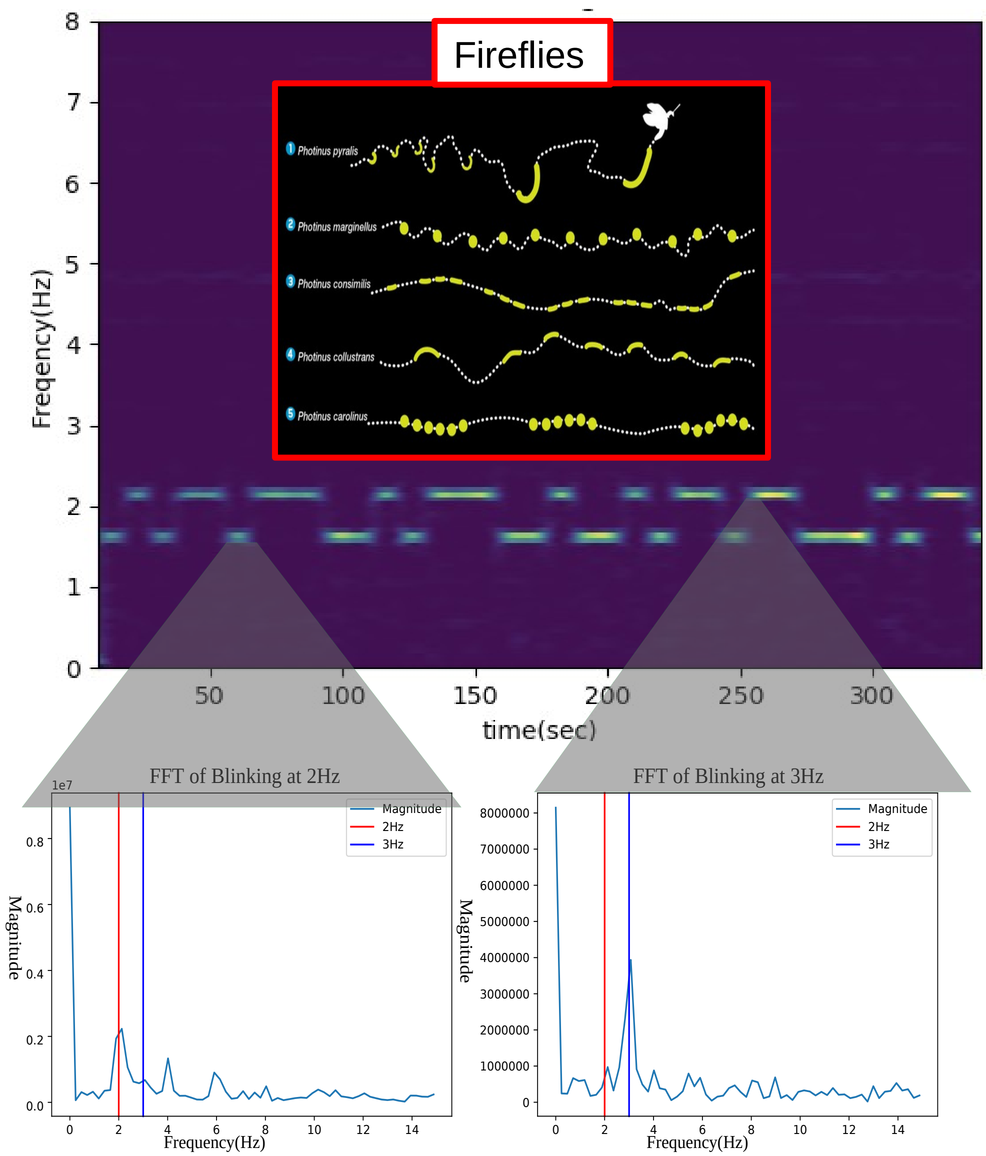

Figure 3.5: The two graphs above show an example of FFT of blinking data with $f_0 = 2$Hz and $f_1 = 3$Hz on the left and right respectively for communicating with FSK. The spikes at 3 and 2Hz describe the rate the LED is blinking. These FFT snapshots are one of the many computed; the SNRs are calculated using the FFT calculations. The SNR would be calculated using eqn 3.19 and 3.20. For the left f_0 would be selected and for the right f_1 would be selected using eqn 3.21. Additionally, the comparison between fireflies and the blinking communication is shown. Fireflies communication in a similar manner using bioluminescent glow, examples of the communication methods of a single firefly is shown.

Baud rate calculation

To calculate the baud rate we must consider the amount of samples required to infer a single bit, which is the number of samples placed into the FFT bin N, referred to earlier. The frame rate of the camera must now be considered, let F be the frames per second of the camera recording.

Single-agent baud rate

The baud rate is how fast the system can change symbols. Using FSK, each sampling period

$$R_i = \frac{F}{N} \tag{3.22}$$

where R_i is the baud rate of the system.

T_i is the seconds it takes to change symbol in the system, which can be calculated many ways. One is inverting the baud rate, another is multiplying the number of samples and the time in between samples.

$$T_i = R_i^{-1}, T_i = t_d N \tag{3.23}$$

Additionally, the data rate(D_i) can be calculated. The Data rate is scaled by the number of bits communicated per symbol, in this case it would be 1.

$$D_i = R_i \tag{3.24}$$

Information encoding

For FSK we used 8-n-1 encoding. An example of this encoding strategy is shown in Figure 3.6. This necessary to synchronize the sender and receiver so that the information can be received properly. In order for the FFT bins to be parsed correctly without overlap some synchronization at the beginning of the data being sent.

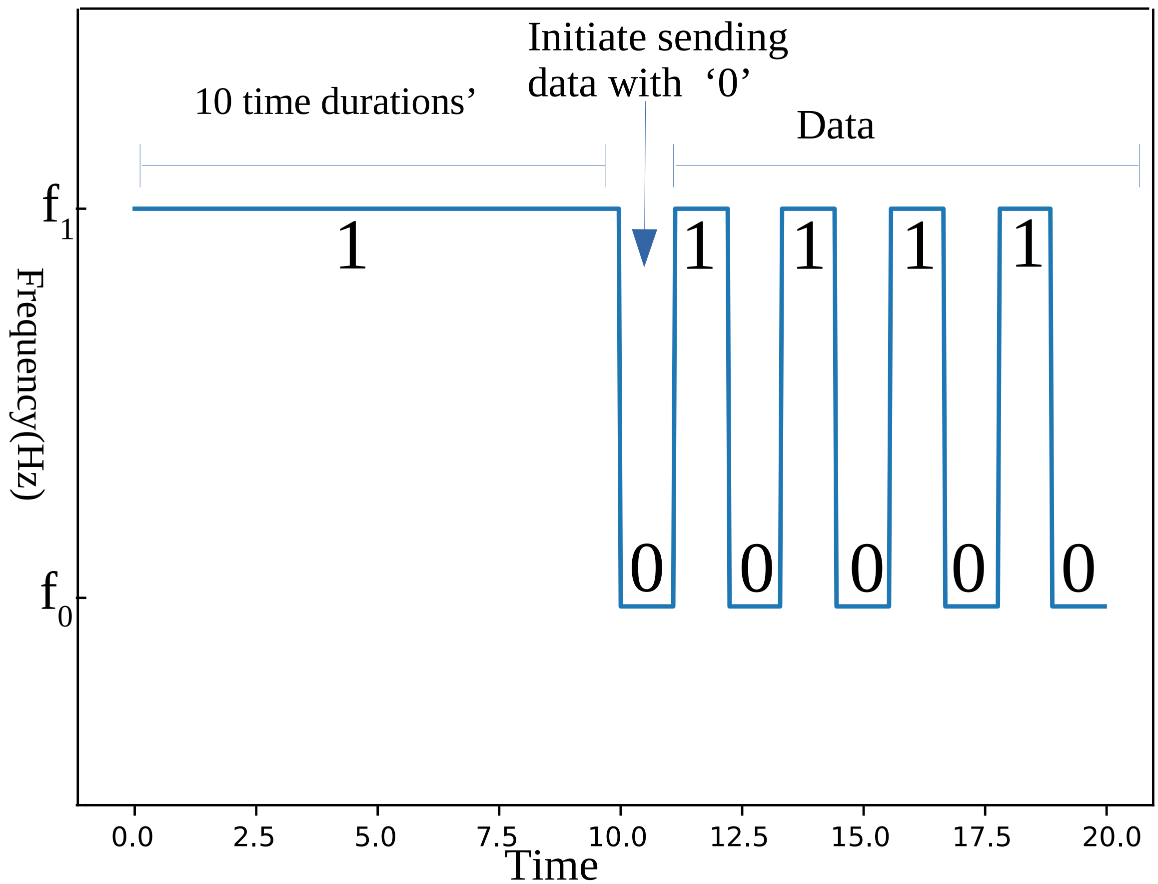

Figure 3.6: A graph describing 8-n-1 encoding for FSK to communicate with VLC. This sequence is repeated every time data is sent. The byte of data received is '10101010' in this example.

3.3.5 MFSK usage

Using MFSK, more data can be packed into the same communication time as using many frequencies. Each frequency can represent more bits in MFSK. Using the same tools discussed in 3.3.4 except with 2^n frequencies. An example of MFSK is shown in Figure 3.7. The range of frequencies for MFSK is defined as:

$$f_{range} = \{f \in \mathbb{Z} \mid f_{min} \leq f \leq f_{max}\} \tag{3.25}$$

Where $f_{min} \geq 2$ is the minimum and f_{max} is the maximum of the possible frequencies.

$$f_{max} = f_{min} + 2^{n_{bits}} - 1 \tag{3.26}$$

Algorithm 2 Sequence describing the movement of a binary image sampling to DFT to frequency determination via SNR for MFSK.

$n \leftarrow 0$
while $n < N$ **do**
    $I \leftarrow I_{new}$ ▷ Get new data from camera
    $\Gamma_P \leftarrow$HSV(I)
    for $p \in P$ **do**
        $C \leftarrow C + \Gamma_p$
    end for
    $c_n \leftarrow C$
    $n \leftarrow n + 1$
end while
$C_k = \mathcal{F}(c_n)$ ▷ Using eqn. 3.17
$F \leftarrow FFT_{bins}$
for $f \in F$ **do**
    $\text{SNR}_f = \text{SNR}(C_k, f)$
end for
$\text{freq} = \text{argmax}_f(\text{SNR}_{f0}, \text{SNR}_{f1}, \ldots, \text{SNR}_{fmax})$
return freq

Signal to noise ratio(SNR) calculation for MFSK

The SNR will be calculated for all frequencies in f_{range} below

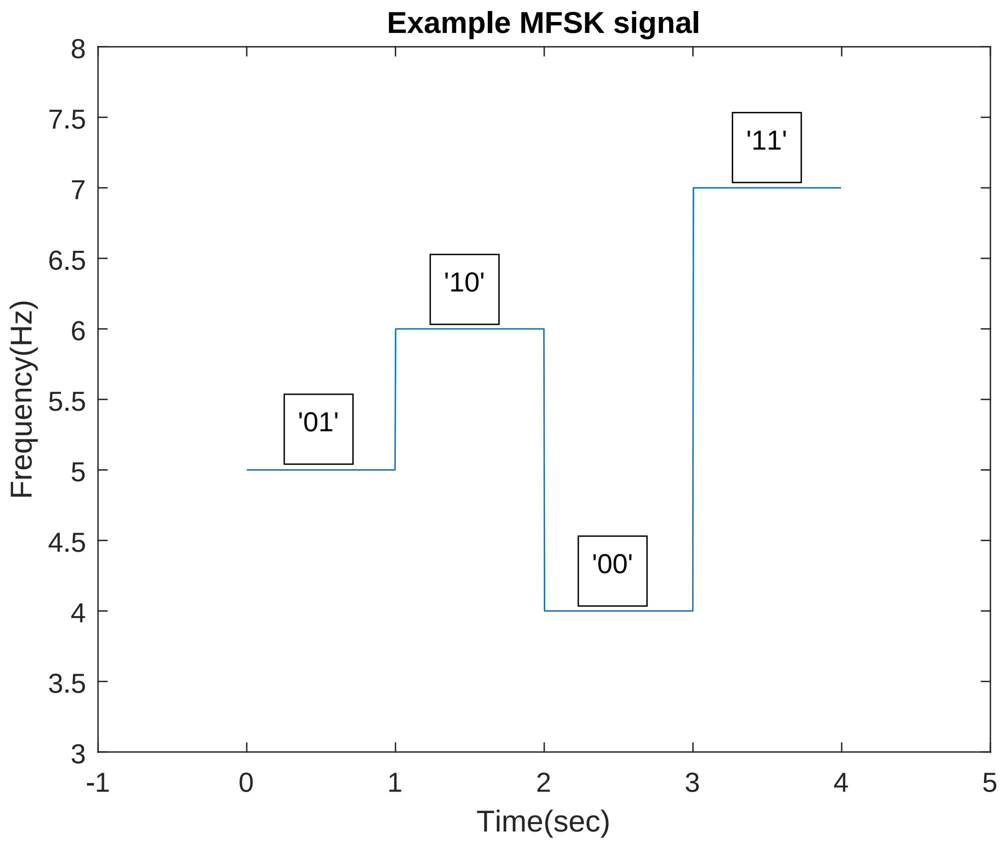

Figure 3.7: Example graph showing four frequency MFSK. The 4 frequncies are $f_0 = 4$Hz, $f_1 = 5$Hz, $f_2 = 6$Hz, and $f_3 = 7$Hz. When the signal is oscillating at 4Hz a binary '00' is represented for that time period. A binary '01' is represented for 5Hz, a binary '10' for 6Hz, a binary '11' for 7Hz. This communication specifically is sending "01110010". When compared to the FSK communication example in Figure 3.4, notice that by doubling the available frequencies, the time to communicate the same amount of information has been cut in half. The sampling period in the above figure is the same as in Figure 3.4, 1 second.

$$
\begin{aligned}
\mathrm{SNR}_{f0} &= 10\log\frac{C(f_0)}{C(f_{noise})} \\
\mathrm{SNR}_{f1} &= 10\log\frac{C(f_1)}{C(f_{noise})} \\
&\vdots \\
\mathrm{SNR}_{fmax} &= 10\log\frac{C(f_{max})}{C(f_{noise})}
\end{aligned}
\tag{3.27}
$$

Using the calculated SNR's we can look for the max SNR using argmax to find the relevant frequency for this sampling period.

$$
F_{blinking} = \mathrm{argmax}_f(\mathrm{SNR}_{f0}, \mathrm{SNR}_{f1}, \ldots, \mathrm{SNR}_{fmax}) \tag{3.28}
$$

Single-agent baud rate

The baud rate is how fast the system can change symbols. The baud rate for MFSK is the same for FSK.

$$
R_i = \frac{F}{N} \tag{3.29}
$$

where R_i is the baud rate of the system. T_i is the seconds it takes to change symbol in the system, which can be calculated many ways. One is inverting the baud rate, another is multiplying the number of samples and the time in between samples.

$$
T_i = R_i^{-1} \tag{3.30}
$$

The data rate is scaled by n_{bits}, the number of bits each frequency represents in the communication setup.

$$D_i = R_i n_{bits} \tag{3.31}$$

3.3.6 Regions of interest(ROI)

The unique part of this communication algorithm is the use of ROI for communication. Using different ROI for communication allows multiple agents to communicate to a single agent at once. The agent receiving messages can determine the spatial location of the sending agents and to receive multiple messages at once. The spatial location can be extracted using equation (3.38). For the calculation of centroid and spatial location we will assume we already have the binary image as described in equation 3.14.

Calculate moments of binary image

The contour area was calculated in 3.15 and defined the variable C for an instantaneous frame. Thus $M_{00} = C$. The sum of the x coordinates is done

$$\text{sum}_x = \sum_{p=(x,y)\in P} x\ \Gamma_p \tag{3.32}$$

and the sum of the y coordinates is done

$$\text{sum}_y = \sum_{p=(x,y)\in P} y\ \Gamma_p \tag{3.33}$$

$$M_{10} = \frac{\text{sum}_x}{M_{00}} \tag{3.34}$$

$$M_{01} = \frac{\text{sum}_y}{M_{00}} \tag{3.35}$$

Calculate centroid and angle offset

The x coordinate of the centroid is calculated

$$C_x = \frac{M_{10}}{M_{00}}, \tag{3.36}$$

and calculate the y coordinate of the centroid.

$$C_y = \frac{M_{01}}{M_{00}} \tag{3.37}$$

where C_x and C_y are the x and y locations of the centroid.

$$X_{deg} = (C_x - \frac{W}{2}) \cdot \frac{FOV_w}{W} \tag{3.38}$$

Where C_x is the x-coordinate of the centroid in the image defined in equation (3.36), W is the width of the image in pixels, FOV_w is the field of view horizontally, and X_{deg} is the angle offset.

$$Y_{deg} = (C_y - \frac{H}{2}) \cdot \frac{FOV_h}{H} \tag{3.39}$$

Where C_y is the y-coordinate of the centroid in the image defined in equation (3.37), H is the height of the image in pixels, FOV_h is the field of view vertically, and Y_{deg} is the angle offset.

Multiple regions of interest

In addition to being able to infer angle offset of the agent, using multiple regions of interest allow the receiving agent to gain information from many agents at a time. However some methods must be employed to distinguish the ROIs. Here we will assume that the image frames are already segmented. Given the original frame I, there are $I_1, I_2, \dots I_l$ which represent the segmented frames, where $l > 0$ is the number of ROIs in frame. Similarly, there are subsets of pixel set P corresponding to each image segmentation, $P_1, P_2, \ldots P_l$.

$$P_i = \{(x_i, y_i) = p_i \in P \mid R_{ixmin} \leq x_i < R_{ixmax}, R_{iymin} \leq y_i < R_{iymax}\} \forall P_i \in \{P_1, P_2, \ldots P_l\} \tag{3.40}$$

where R_{ixmin}, R_{ixmax}, R_{iymin}, and R_{iymax} describe the image segmentation for I_i respectively and may depend of x or y to describe lines rather than only rectangular regions. Now each P_i is segmented, however the full frame isn't represented. Here we will "zero out" the rest of the binary image using the complement P_i^C. Additionally, Γ_{ip} will get the binary image data from the whole frame binary image Γ_p, for each frame segmentation. Let Γ_{ip} represent the binary image of the segmentation I_i.

$$\Gamma_{ip} = \Gamma_p \ \forall p \in P, \ \forall \Gamma_{ip} \in \{\Gamma_{1p}, \Gamma_{2p}, \ldots, \Gamma_{lp}\} \tag{3.41}$$

$$\Gamma_{ip} = 0 \ \forall p \in P_i^C \tag{3.42}$$

Now each Γ_{ip} contains the data exclusively for each particular region of the frame and zero's for the rest of the frame. Having the whole frame represented is important because when performing centroid and angle offset calculations in in eqn 3.32 through eqn 3.39. If the frame is not fully represented the angle offset calculated will not be correct. Now we can replace Γ_{ip} with Γ_p for each image frame for eqn 3.15 to eqn 3.39 to perform communication for each agent in each ROI.

Multiple regions of interest effect on data rate

The data rate of the receiving agent will be scaled by the number of agents in frame. Let $R_{oi} > 0$ be the number of ROI in the image frame.

$$D_{fin} = D_i R_{oi} \tag{3.43}$$

where D_{fin} is the final data rate of the receiving agent and D_i is the data rate when communicating with a single agent. While calculating the theoretical maximum baud rate we must include the resolution of the camera, because every pixel could be a ROI. Thus, $R_{oi} = P_{tot}$ under theoretically conditions. With P_{tot} being the total number of pixels.

3.3.7 Communication Assurance

One of the uses of communicating with a digital camera and LEDs is you can extract the spatial location and degree offset. This was done in eqn 3.38 and 3.39. Using these values control methods can be used to assure communication. For the single agent-to-agent communication method a PID controller can be used.

x_{deg} and y_{deg} are to approach 0 to stay centered on the communicator. These are instantaneous angle offsets, recording these over time allows us to construct a time series function of the degree offsets. Let $x_{deg}(t) = x_{deg}$, $y_{deg}(t) = y_{deg}$ with $0 \leq t < t_f$. Since we are trying to center on the communicator, the error functions $e_x(t)$ and $e_y(t)$ directly correlate to $x_{deg}(t)$ and $y_{deg}(t)$ respectively($e_x(t) = x_{deg}(t)$ and $e_y(t) = y_{deg}(t)$).

$$u_x(t) = K_{px} e_x(t) + K_{ix} \int_0^t e_x(\tau) d\tau + K_{dx} \frac{de_x(t)}{dt} \tag{3.44}$$

where K_{px}, K_{ix}, and K_{dx} are the horizontal proportional, integral, and derivative coefficients of the system.

$$u_y(t) = K_{py} e_y(t) + K_{iy} \int_0^t e_y(\tau) d\tau + K_{dy} \frac{de_y(t)}{dt} \tag{3.45}$$

where K_{py}, K_{iy}, and K_{dy} are the vertical proportional, integral, and derivative coefficients of the system.

Figure 3.8: Two LTA blimps flying and communicating. The LTA Blimp with digital camera is on the left. The LTA blimp with the LED to signal information on the right, a diffuser can be seen to disperse the light information can be seen under the blimp. This test demonstrated the communication algorithm working on prototype agents.

The inputs $u_x(t)$ and $u_y(t)$ can be used to direct the motor outputs of agents to stay aligned with the communicating agent.

When information is detected by an agent in the swarm, an agent can use the centroid of the mask to keep the light in the center of the camera frame. This is to ensure that the camera can get enough data to sample and make the correct calculation. This was done because we found that the wide-angle cameras had issues detecting information at significant distances. This process is shown in Figure 4.3.

3.3.8 Consistent frame rate with digital cameras

One of the necessary components of using a digital camera as a sampling method, is a consistent frame rate. However, this can be difficult given when trying to maximize frame rate and recording video pragmatically. In order for the communication method to be deployed on LTA blimps the frame rate must be reduced to keep a consistent frame rate. However, for a consistent higher frame rate during bit error rate testing we have to use a recording software that encoding using a graphics processing unit(GPU) and Nvidia drivers. The software to record the video using Nvidia drivers was bandicam[1].

3.4 Communication Algorithm using Event-based camera

The math definitions for the communications using an event based camera will change significantly because the event based camera doesn't have a frame rate. Events at pixels stream to the user asynchronously based on moving light intensity. Using a sampling period to consolidate this data will be used to simulate a "frame".

The sampling period is an interesting part of this system because it can be changed on the fly, as it is just a parameter in the algorithm that can be changed, and the effects for resolution of the algorithm are significant.

Again, consider an image I of resolution $R_x > 0$, $R_y > 0$. Let $e = (p_e, t_e, O)$ represent a single pixel "event". With $p_e = (x_e, y_e) \in \mathbb{Z}^2$ be the coordinate of the incoming pixel s.t.

[1]`https://www.bandicam.com/`

$0 \le x_e < R_x$, $0 \le y_e < R_y$, $t_e \in \mathbb{R}$ is the time at which pixel p is received, and $O \in \mathbb{Z}$ is the polarity of the event. Let E be the continuous stream of events coming from the event-based camera. Let $P = \{p_e = (x_e, y_e) \in \mathbb{Z}^2 \mid 0 \le x_e < R_x, 0 \le y_e < R_y\}$ The exact definition for the polarity O is shown below.

$$O = \begin{cases} 1 & \text{pixel on} \\ 0 & \text{pixel off} \end{cases} \tag{3.46}$$

Let t_{ep} represent the time for the initial "sampling period", to simulate a frame rate of the camera. Where the "virtual" frame rate is $F = \frac{1}{t_{ep}}$, 1 frame per t_{ep}.

$$E_{col} = \{e \in E \mid t_{ep0} < t_e < t_{ep0} + t_{ep}\} \tag{3.47}$$

where t_{ep0} is the start of the "sampling period", when $t_e == t_{ep0} + t_{ep}$ then $t_{ep0} = t_{ep0} + t_{ep}$ and the collection of event is sampled in the following equation. This is further defined in Algorithm 3. An interesting attribute of using event based cameras in this manner is that t_{ep} can be changed as needed. Virtually, you can change the "sampling rate" in post or in real time to better accommodate your current solution. Now a "virtual" binary image must be created using the received event in E_{col}. Let Ξ_p be the binary image.

$$\Xi_p = \begin{cases} 1 & p_e \in E_{col} \\ 0 & p_e \notin E_{col} \end{cases} \quad \forall p_e \in P \tag{3.48}$$

Now the contour area can be calculated

$$C_e = \sum_{p_e \in P} \Xi_p \tag{3.49}$$

and then the contour area is sampled

$$z_n = C_e \tag{3.50}$$

where $n = 1, 2, \ldots, N-1$ and N is the number of samples.

Now we have a function of the number of pixels coming through during a set time, this will act like our contour area function in 3.16. This will include pixels that are not necessarily for communication because we do not have a color filter because there is no color, however the speed increase makes up for the lost effectiveness. Using z_n we can calculate the frequency domain Z_k using a discrete Fourier Transform(DFT).

$$Z_k = \sum_{n=0}^{N-1} z_n e^{-\frac{j2\pi}{N}kn}, k = 0, 1, 2, \ldots N-1 \tag{3.51}$$

where Z_k is a sequence of complex numbers representing the function of frequency.

3.4.1 Noise Filter

A noise filter is placed in series before the data is read to determine the communication. The noise filter is a "standard spatiotemporal filter that stores a map of event timestamps. The filter also provides a hot-pixel filter with learning to filter 'broken' (always-active) pixels in the sensor, as well as a refractory period filter, which limits the rate at which single pixels can consecutively generate events."- from the Inivation website[2]. The parameters can be tuned for this filter but the design seems to be proprietary, see the parameters and descriptions in Table 3.2.

3.4.2 FSK using event-based camera

Now that the function of frequency is calculated we can make comparisons using the SNR to determine the frequency of blinking. This is done using the same methods as the digital camera.

[2]From `https://inivation.gitlab.io/dv/dv-docs/docs/noise-filters/`

Table 3.2: A list of parameters and descriptions for the proprietary noise filter used on the event-based camera data.

Noise filter parameter	**Description**
backgroundActivityTime	Maximum time difference(µs) for events to be considered correlated and not be filtered out
backgroundActivityEnable	Enable the background activity filter
refractoryPeriodTime	Minimum time between consecutive events to not be filtered out
refractoryPeriodEnable	Enable the refractory period filter
hotPixelLearn	Learn the position of current hot (abnormally active) pixels
hotPixelEnable	Enable the hot pixel filter

Again, the SNR can be calculated.

$$\mathrm{SNR}_{f0} = 10 \log \frac{Z_{f_0}}{Z_{f_{noise}}} \tag{3.52}$$

$$\mathrm{SNR}_{f1} = 10 \log \frac{Z_{f_1}}{Z_{f_{noise}}} \tag{3.53}$$

where $f_{noise} = \{f \in F \mid f \neq f_0,\, f \neq f_1\}$. F is the all the possible frequency bins for the computed FFT.

$$F_{blinking} = \begin{cases} f0 & \mathrm{SNR}_{f0} > \mathrm{SNR}_{f1} \\ f1 & \mathrm{SNR}_{f1} > \mathrm{SNR}_{f0} \end{cases} \tag{3.54}$$

Now that we have determined the frequency of blinking using an event-based camera, FSK can be performed using SNR the same way that would be done using a digital camera.

Algorithm 3 Sequence describing how the event-based camera receives blinking information using FSK.

```
t_ep0 ← 0
n ← 0
Events ← 0
for e ∈ E do                                ▷ E Represents the continuous stream of events
    (p_e, t_e, O) ← e
    if t_e ≤ t_ep0 + t_ep then                          ▷ t_ep is assumed to be given
        Events ← Events + 1
    else
        z_n ← Events
        Events ← 1
        t_ep0 ← t_ep0 + t_ep
        n ← n + 1
    end if
end for
Z_k = F(z_n)                                                   ▷ Using eqn. 3.51
SNR_f0 = SNR(Z_k, f_0)                                         ▷ Using eqn. 3.52
SNR_f1 = SNR(Z_k, f_1)                                         ▷ Using eqn. 3.53
if SNR_f0 > SNR_f1 then
    freq ← f_0
else
    freq ← f_1
end if
return freq
```

3.4.3 MFSK using event-based camera

MFSK can be done the same way as a digital camera using the function of frequency to determine out of a larger number of frequencies, which frequency is blinking. However HSV filtering is not required as event-based cameras themselves do not receive visible light.

The range of frequencies for MFSK using an event-based camera is defined as:

$$f_{range} = \{f \in \mathbb{Z} \mid f_{min} \leq f \leq f_{max}\} \tag{3.55}$$

Where $f_{min} \geq 2$ is the minimum and f_{max} is the maximum of the possible frequencies.

$$f_{max} = f_{min} + 2^{n_{bits}} - 1 \tag{3.56}$$

Algorithm 4 Sequence describing how the event-based camera receives blinking information using MFSK.

$t_{ep0} \leftarrow 0$
$n \leftarrow 0$
Events $\leftarrow 0$
for $e \in E$ **do** ▷ E Represents the continuous stream of events
  $(p_e, t_e, O) \leftarrow e$
  if $t_e \leq t_{ep0} + t_{ep}$ **then** ▷ t_{ep} is assumed to be given
    Events $\leftarrow$ Events $+ 1$
  else
    $z_n \leftarrow$ Events
    Events $\leftarrow 1$
    $t_{ep0} \leftarrow t_{ep0} + t_{ep}$
    $n \leftarrow n + 1$
  end if
end for
$Z_k = \mathcal{F}(z_n)$ ▷ Using eqn. 3.51
$F \leftarrow FFT_{bins}$
for $f \in F$ **do**
  $\text{SNR}_f = \text{SNR}(C_k, f)$
end for
$\text{freq} = \text{argmax}_f(\text{SNR}_{f0}, \text{SNR}_{f1}, \ldots, \text{SNR}_{fmax})$
return freq

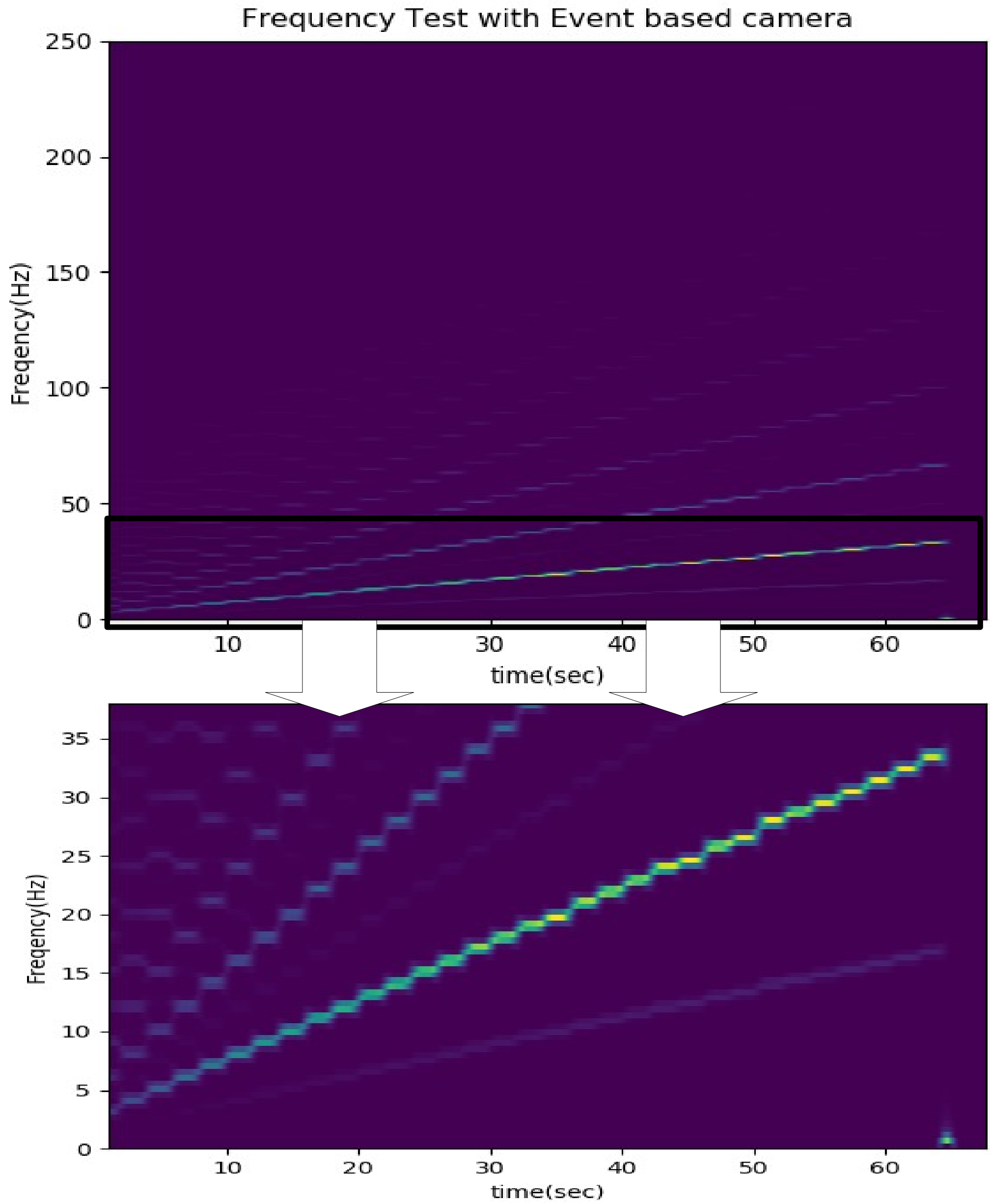

Figure 3.9: A 34 frequency sweep using the event based camera. An example of how the physical layout of the event-based cameras and the LED can be seen in Figure 4.6. Having 34 possible frequencies to pick from allows each frequency to represent 5 bits of information, increasing the baud rate significantly.

Signal to noise ratio(SNR) calculation for MFSK

The SNR will be calculated for all frequencies in F_{range} below

$$
\begin{aligned}
\mathrm{SNR}_{f0} &= 10\log\frac{Z(f_0)}{Z(f_{noise})} \\
\mathrm{SNR}_{f1} &= 10\log\frac{Z(f_1)}{Z(f_{noise})} \\
&\vdots \\
\mathrm{SNR}_{fmax} &= 10\log\frac{Z(f_{max})}{Z(f_{noise})}
\end{aligned}
\tag{3.57}
$$

Using the calculated SNR's we can look for the max SNR using argmax to find the relevant frequency for this sampling period.

$$F_{blinking} = \mathrm{argmax}_f(\mathrm{SNR}_{f0}, \mathrm{SNR}_{f1}, \ldots, \mathrm{SNR}_{fmax}) \tag{3.58}$$

Single-agent baud rate

The baud rate is how fast the system can change symbols. The baud rate for MFSK is scaled by n_{bits}, the number of bits each frequency represents in the communication setup.

$$R_i = (Nt_{ep})^{-1} \tag{3.59}$$

where R_i is the baud rate of the system. T_i is the seconds it takes to change symbol in the system, which can be calculated many ways. One is inverting the baud rate, another is multiplying the number of samples and the time in between samples. Notice the time to record stays the same.

$$T_i = R_i^{-1} \tag{3.60}$$

The data rate is scaled by n_{bits}, the number of bits each frequency represents in the communication setup.

$$D_i = R_i n_{bits} \tag{3.61}$$

3.4.4 Regions of interest(ROI) with Event-based camera

The unique part of this communication algorithm is the use of ROI for communication. Using different ROI for communication allows multiple agents to communicate to a single agent at once. The agent receiving messages can determine the spatial location of the sending agents and to receive multiple messages at once. The spatial location can be extracted using equation (3.38). For the calculation of centroid and spatial location we will assume we already have the binary image as described in equation 3.14.

Calculate moments of binary image for a single agent in frame

The contour area was calculated in 3.49 and defined the variable C_e for an instantaneous frame. Thus $M_{00} = C_e$. The sum of the x coordinates is done

$$\text{sum}_x = \sum_{p_e=(x_e,y_e)\in P} x_e \ \Xi_{p_e} \tag{3.62}$$

and the sum of the y coordinates is done

$$\text{sum}_y = \sum_{p_e=(x_e,y_e)\in P} y_e \ \Xi_{p_e} \tag{3.63}$$

$$M_{10} = \frac{\text{sum}_x}{M_{00}} \tag{3.64}$$

$$M_{01} = \frac{\text{sum}_y}{M_{00}} \tag{3.65}$$

Calculate centroid and angle offset

The x coordinate of the centroid is calculated

$$C_x = \frac{M_{10}}{M_{00}}, \tag{3.66}$$

and calculate the y coordinate of the centroid.

$$C_y = \frac{M_{01}}{M_{00}} \tag{3.67}$$

where C_x and C_y are the x and y locations of the centroid.

$$X_{deg} = (C_x - \frac{W}{2}) \cdot \frac{FOV_w}{W} \tag{3.68}$$

Where C_x is the x-coordinate of the centroid in the image defined in equation (3.66), W is the width of the image in pixels, FOV_w is the field of view horizontally, and X_{deg} is the angle offset.

$$Y_{deg} = (C_y - \frac{H}{2}) \cdot \frac{FOV_h}{H} \tag{3.69}$$

Where C_y is the y-coordinate of the centroid in the image defined in equation (3.67), H is the height of the image in pixels, FOV_h is the field of view vertically, and Y_{deg} is the angle offset.

Multiple regions of interest

In addition to being able to infer angle offset of the agent, using multiple regions of interest allow the receiving agent to gain information from many agents at a time. However some methods must be employed to distinguish the ROIs. Here we will assume that the image frames are already segmented. Given the original frame I, there are $I_1, I_2, \ldots I_l$ which represent the segmented frames, where $l > 0$ is the number of ROIs in frame. Similarly, there are subsets of pixel set P corresponding to each image segmentation, $P_1, P_2, \ldots P_l$.

Additionally, the segmentation's are generalized

$$P_i = \{(x_i, y_i) \in P \mid R_{ixmin} \leq x_i < R_{ixmax}, R_{iymin} \leq y_i < R_{iymax}\} \ \forall P_i \in \{P_1, P_2, \ldots P_l\} \tag{3.70}$$

where R_{ixmin}, R_{ixmax}, R_{iymin}, and R_{iymax} describe the image segmentation for I_i respectively and may depend of x or y to describe lines rather than only rectangular regions.

Now each P_i is segmented, however the full frame isn't represented. Here we will "zero out" the rest of the binary image using the complement P_i^C. Additionally, Ξ_{ip} will get the binary image data from the whole frame binary image Ξ_p, for each frame segmentation. Let Ξ_{ip} represent the binary image of the segmentation I_i. Copying the original image into each segmented image

$$\Xi_{ip} = \Xi_p \ \forall p \in P, \ \forall \Xi_{ip} \in \{\Xi_{1p}, \Xi_{2p}, \ldots, \Xi_{lp}\} \tag{3.71}$$

Then zeroing out the other ROI

$$\Xi_{ip} = 0 \ \forall p \in P_i^C \tag{3.72}$$

Now each Ξ_{ip} contains the data exclusively for each particular region of the frame and zero's for the rest of the frame. Having the whole frame represented is important because when performing centroid and angle offset calculations in in eqn 3.32 through eqn 3.39. If the frame is not fully represented the angle offset calculated will not be correct. Now we can replace Ξ_{ip} with Ξ_p for each image frame for eqn 3.15 to eqn 3.39 to perform communication for each agent in each ROI. The effect of communicating with multiple agents at a time as it relates to the data rate is shown below. Let $R_{oi} > 0$ be the number of ROI in the image frame.

$$D_{fin} = D_i R_{oi} \tag{3.73}$$

where D_{fin} is the final data rate of the receiving agent and D_i is the data rate when communicating with a single agent. While calculating the theoretical maximum data rate we must include the resolution of the camera, because every pixel could be a ROI. Thus, $R_{oi} = P_{tot}$ under theoretically conditions. With P_{tot} being the total number of pixels.

3.5 Algorithm discussion

3.5.1 Benefits of VLC/Line-of-sight Communications in Swarms

VLC and line-of-sight communications methods provide information to each agent in the swarm even if no-communication is received. This is due to the inherent high density of swarming agents. If an agent does not receive any communications from another agent that lack of communications provides information. The information it provides is: 1) The agent is too far away, 2) the agent is broken/has malfunctioned, and/or most importantly 3) there is something between the two agents. The latter piece of information can be gathered in a time series and used as an input for the swarm behavior. It is thought that this piece of information is more useful when the density of agents increases. Further analyses and experimentation need to be done on this.

3.6 Comparison of Digital and Event based cameras

The underling math being done to perform VLC using Event-based and Digital cameras is similar, using the area as a "sample" then inferring frequency over time. Typically, digital cameras have a relatively low frame rate compared to the temporal resolution of event based cameras. This gives the event based cameras an advantage temporally. However digital cameras can see visible light as RGB values giving an advantage to perform CSK or other color filtering to complement the communication.

3.6.1 Communication sensitivity comparison

Digital cameras are more resilient to other distractions in frame during communication due to color filtering when compared to event based cameras due to the lack of filter. However, VLC could be jammed using same color light currently used for communication for a digital camera.

3.6.2 Data rate comparison

In table 3.3 a comparison between the symbolic data rate between the different forms of communication, digital and event-based cameras. The two camera types are evaluated using MFSK and FSK.

Table 3.3: Table comparing the symbolic definitions of the different communication methods, FSK and MFSK using an event-based camera and a digital camera. This allows readers to understand how each of these variable change the effective data rate outcome with changes to the resolution of sampling rate of the system.

	FSK(digital)	**MFSK(digital)**	**FSK(event)**	**MFSK(event)**
Data rate($\frac{\text{bit}}{\text{sec}}$)	$\frac{F}{N}$	$\frac{Fn}{N}$	$(t_{ep}N)^{-1}$	$\frac{n}{t_{ep}N}$

where F represents frame rate, N is the number of samples in the FFTbin, n is the number of bits represented by each Frequency, t_{ep} is the sampling time for the event-based camera to simulate a frame rate.

Next we will evaluate the symbolic data rates at typical values for each variable to give an idea of what is expected for each method of communication.

where $N = 128$(FFTbins), $F = 30$(frames per second), $t_{ep} = 2000(\mu\text{s})$, $n = 4$(bits per frequency). The resolution and accuracy can be increased, however it can be seen that this would decrease the Data rate. Additionally, using more expensive hardware to get a higher frame rate would provide a faster data rate.

Table 3.4: Typical data rate values for the respective hardware and communication setups. Obviously, these values can vary from setup to setup but these should serve as ballpark values and show the trend between the communication strategies.

	FSK(digital)	**MFSK(digital)**	**FSK(event)**	**MFSK(event)**
Data rate($\frac{\text{bit}}{\text{sec}}$)	0.23	0.93	3.9	15.6

3.7 Agent Setup

Lighter-than-air blimps were designed with LED clusters and digital cameras to communicate using VLC. The messages would be transmitted, or displayed, by agents in the swarm then received, or viewed, by other agents in the swarm. Two types of blimps were made, one to transmit messages and one to detect the messages. Both can be seen in action in Figure 3.8. This was done to simplify the design and show a proof of concept. On a fully implemented swarm, all agents would have blinking and camera capabilities. The hardware design for both blimps is shown in Figure 4.2.

The blimp to transmit messages was outfitted with an LED cluster and diffuser. The diffuser can be seen on the right side of Figure 4.2. The diffuser was used to present a bigger area for the other agents to view and in turn allowed the information to be seen from all angles. The blimp also had the ability to move using motors with propellers.

Chapter 4: Analysis/Results

Testing and analysis will show why the described techniques of communicating with a camera and LED are effective for swarm robotics. First the aerial testing of an elementary communication setup will be shown in 4.1 using a digital camera. Next the communication with a digital camera and FSK will be tested using many “agents” and the bit-error rate will be evaluated in 4.2. Finally, the communication with an event-based camera and MFSK will be tested using “agents” and the bit error rate will be evaluated in 4.3.

4.1 Agent-Agent Aerial Communication

The communication setup describing use of a camera and LEDs was implemented on a LTA^3. This testing was to show VLC with a camera could be done on an aerial agent.

The setup to initially test the communication algorithm was two LTA^3 placed in the same area to see if communication was possible using a digital camera as shown in Figure 3.8. This was done successfully, however a controller to keep the receiver agent aligned was necessary. The PID controller described in Equations 3.44 and 3.45. These control inputs were executed by the hardware on the blimp shown in Figure 4.2. The initial tests were to see if VLC was possible between two aerial agents by blinking at certain frequencies. An example of the results are shown in Figure 3.5.

For the LED information to be dispersed effectively, a “diffuser” was made used around the communicating LED. This also dispersed light to all angles around the agent. The diffuser used in initial tests is shown in Figure 4.1.

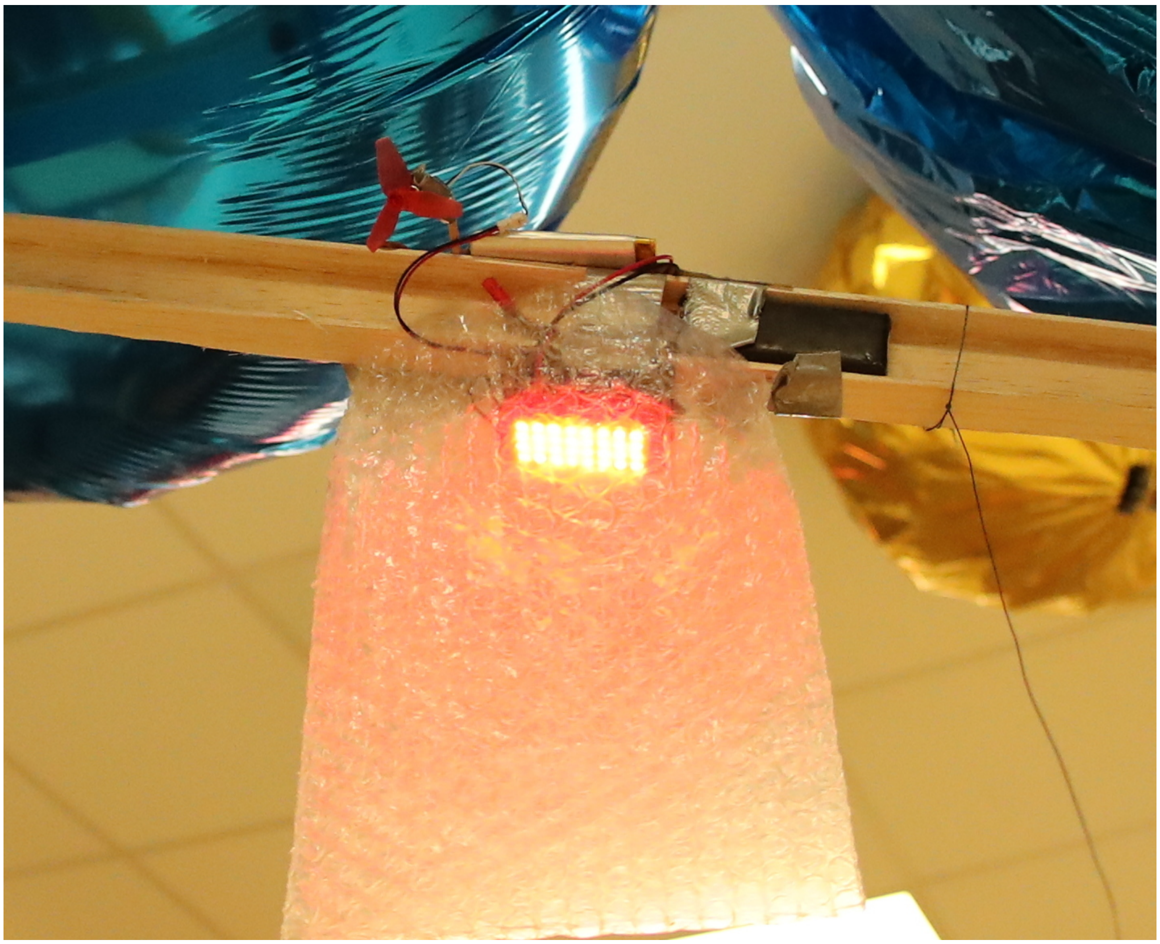

Figure 4.1: Above is the diffuser used under the blimp platform to better visualize the color being shown by the LED cluster.

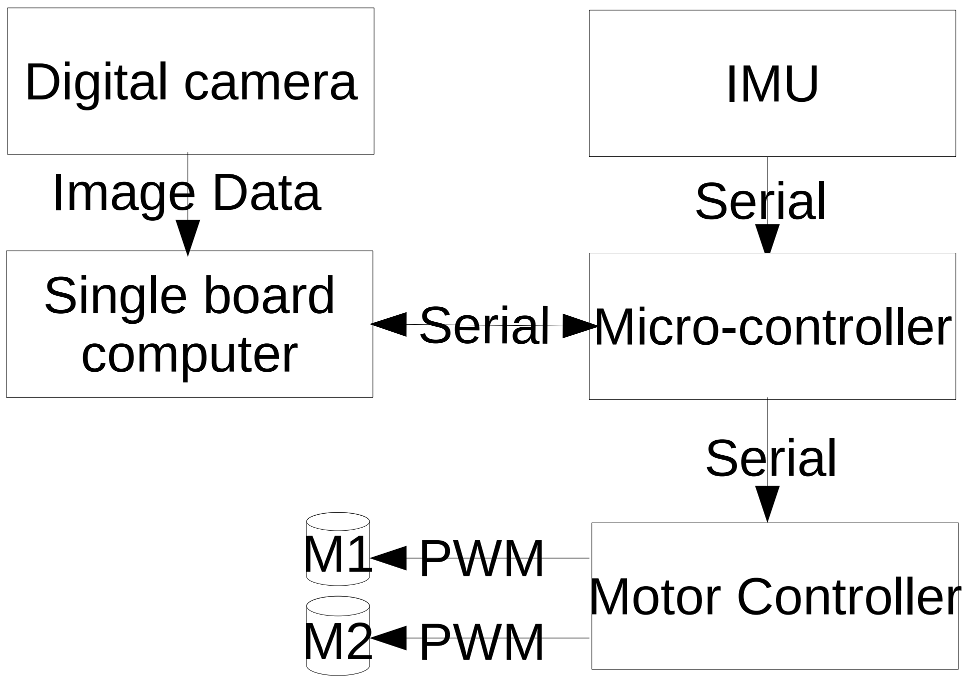

Figure 4.2: Hardware diagram for the sensing blimp. M1 and M2 represent the motors 1 and 2 on the blimp.

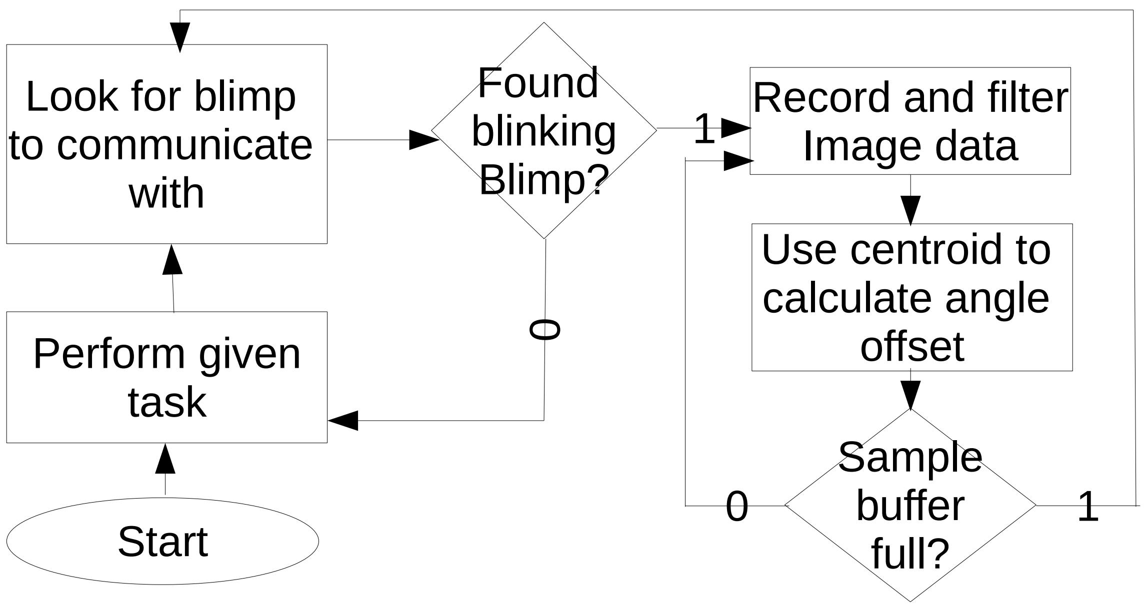

Figure 4.3: Algorithm used to keep the blimp centered on another blimp trying to communicate data. This is required as the LTA blimps can drift during communication. This demonstrates a use of the ROI alongside the communication aspect of light. This demonstrated the usage of both position and communication in use. This flow diagram assumes only one agent in frame.

Table 4.1: Bit error rate test for our swarm communication using a digital camera with many blinking LEDs or agents communicating in-front of it. Bit Error Rate for communications with one, two, and three regions being read and sent is shown below. This was performed in order to evaluate the communication standard.

Bit Error Rate(%)			
Number of Agents	1	2	3
Error rate	4.83%	4.44%	6.32%

4.2 FSK Testing

FSK communication was tested using the same baseline as the previous test to send blinking information with an LED cluster and interpret video stream data. The physical test setup is shown in Figure 4.4. The data flow of the FSK testing is shown in Figure 4.5. A computer would provide a randomly generated character for the sender to send, the computer would record each character generated. The computer with a digital camera connected would interpret the results and compare them with what was intended to be sent. To test the effectiveness of the proposed FSK communication method, bit error rate tests were done without any feedback loop to correct missed bits. Bit error rate test is a standard test in digital communications to evaluate a communication method. The results of the bit error rate testing are shown in Table 4.1.

With 10,192 bits sent or 1274 characters(bytes), there was a bit error rate of 4.83% for one region of interest. 10,048 bits send or 1256 characters(bytes), there was a bit error rate of 4.44% for two regions of interest.

This shows the effectiveness using a digital camera. This setup inherently address TC1, TC2, and TC3. Also This setup addresses many agents communicating at the same time and the effectiveness.

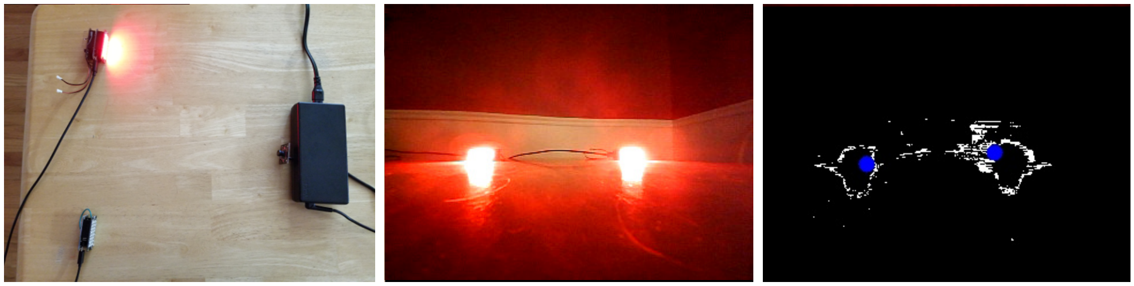

Figure 4.4: A series of pictures describing an example test setup and what the camera sees during the test setup. This test setup simulates 3 agents, two agents communicating simultaneously to the camera posing as an agent. (**LEFT**) overhead view of the example test setup. (**CENTER**) The RGB image the camera sees. (**RIGHT**) The mask generator by HSV filtering. The white area represent the contour area and is what is used over time to infer frequencies. The blue dots represent the calculated centroid for each region of interest. In this picture there are two regions of interest (left and right), divided down the center with the white line.

The heightened error rate when the number of agents is at 3, is suspected to be from bleed-over from each LED cluster into another ROI or segment of the frame. This can be addressed by have a diffuser for the LEDs. Diffusers were used for the aerial agent tests but not in these bit-error rate tests.

4.3 MFSK Testing

Initial testing using an Event-based camera was performed to show the viability of event-based cameras to receive information using LEDs. In these tests there were 8 possible frequencies to "blink" at. Meaning each frequency represented for a sampling period, inferred 3 bits.

An example of the physical test setup is shown in Figure 4.6. The bit-error rate was determined the same method as with the digital camera. The process is shown in Figure 4.7.

Initial evaluation using bit-error rate of the event-based camera communication with

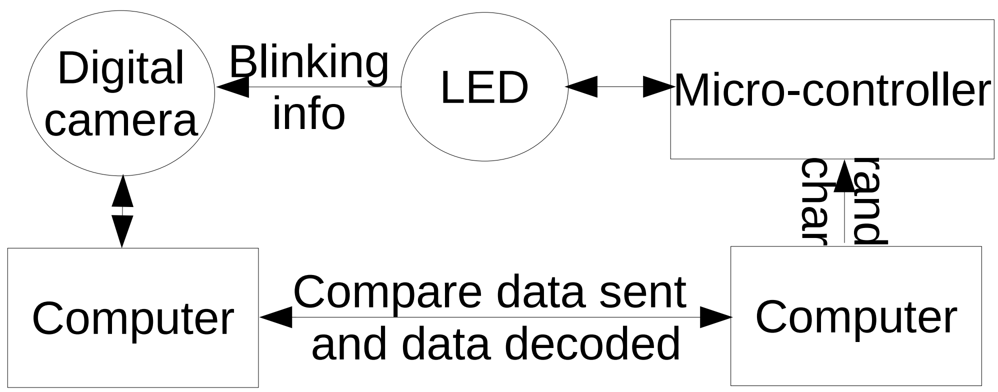

Figure 4.5: A flow chart explaining the testing setup for calculating the bit error rate of FSK with VLC. Computer 1(Right) would provide a random character to the micro-controller, the micro-controller would convert that character to binary using ASCII encoding. Computer 2(Left) records video of the LED blinking with the digital camera. Then the decoding is done on computer 2 and compared with what was generated on computer 1. The many harmonics present in the FFT have no effect on the proposed system. This is due to the algorithm comparing between 2Hz and 3Hz only. The DC component of the system at 0Hz can be ignored.

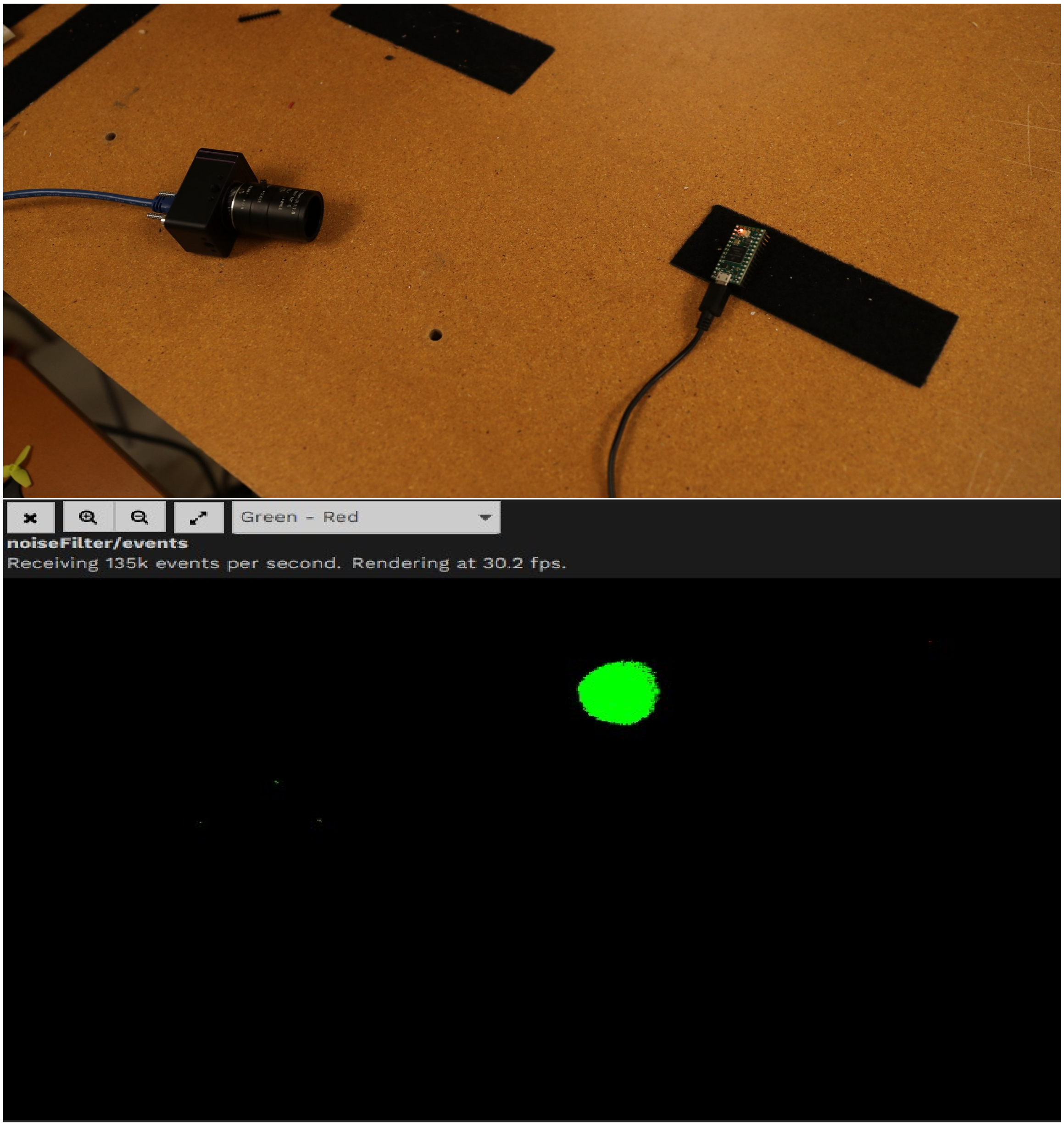

Figure 4.6: **Top**) An overhead picture of a DXplorer event-based camera and a micro-controller communicating information using its on-board LED. (**Bottom**) A visual playback of the data being received by the DXplorer event-based camera. The playback shows the output of the data after been placed through a noise filter built into their software.

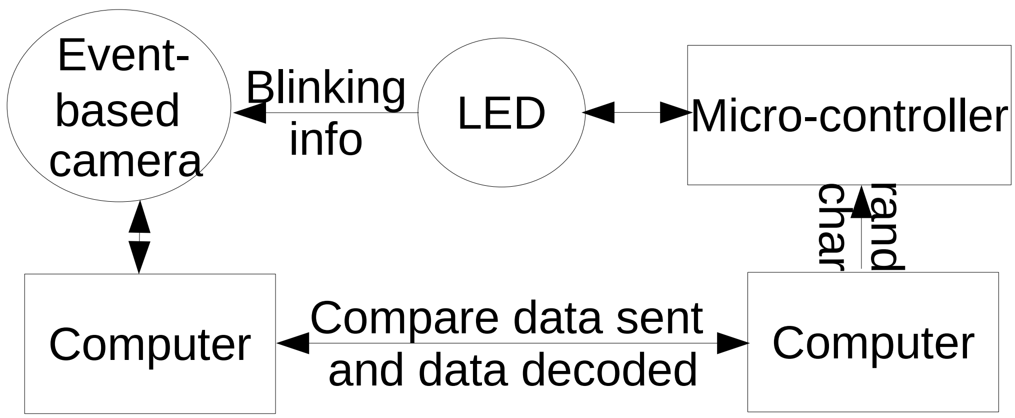

Figure 4.7: Method of evaluating the bit error rate for the communication setup using an event-based camera. Random data was provided to the micro-controller which blinked an on-board LED. This random data was received by the event-based camera and parsed on a computer. Then the data that was sent to the micro-controller and what was parsed by the computer was compared.

and LED with MFSK was performed. A total of 1002 bits were sent. Additionally, only 1 ROI was tested. This is shown in Table 4.2.

This shows the effectiveness using a digital camera. This setup inherently address TC1, TC2, and TC3.

Table 4.2: Bit error rate testing for the swarm communication setup was done using an event-based camera with a single agent communicating. Bit Error Rate for communications with one region is being read and sent is shown below. This was performed in order to evaluate the communication standard for MFSK using event-based cameras.

Bit Error Rate(%)	
Number of Agents	1
Error rate	3.89%

Chapter 5: Conclusion

Swarm robotics has many possibilities for completing complex tasks. This requires hundreds, if not thousands of agents to work together. For complex tasks to be completed, communication and localization to some degree is required. Swarm communication must scale linearly with swarm size, posing a challenge as traditional communication systems typically operate in a fully connected manner. This paper took inspiration from nature where social insects typically only communicate with the most important or nearby agents. In this thesis, methods were proposed and tested to solve communication and swarm state perception using cameras and LEDs. Cameras have the benefit of having a limited sight range, this limits the communication to the agents the receiver can "see", answering TC1 and TC3. Also, LEDs limit the number of communicating agents due to agents blocking each other, solving the challenges posed in TC1 and TC3. Communicating with only essential agents is important as having too many communicating at a time, will lead to latency and bandwidth issues. Using cameras and LEDs, communication was performed by blinking LEDs at different frequencies to represent shared information. Additionally, localization was done using the location of the agent in frame then translating that into an offset. This allows the receiving agent to infer information about the sending agent's position, solving the challenges in TC2. A digital camera was used initially, HSV filtering was used to track contour area of the LEDs blinking over time. However, the sampling rate was dependent on the frame rate of the camera which led to a limitation to the communication rate. To explore faster communication rates event-based cameras were used. A similar methodology was used to interpret the blinking information using an event based camera. The results of these methods were shown, showing the feasibility of swarm communication using cameras and LEDs.

Bibliography

Bibliography

[1] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3293–3298.

[2] A. B. Maxseiner, D. M. Lofaro, and D. A. Sofge, "Visible light communications with inherent agent localization and simultaneous message receiving capabilities for robotic swarms," in *2021 18th International Conference on Ubiquitous Robots (UR)*, 2021, pp. 633–639.

[3] A. Maxseiner, D. Lofaro, and D. Sofge, "Visible light communication for robotic swarms," in *The 4th International Symposium on Swarm Behavior and Bio-Inspired Robotics (SWARM2021)*, 2021.

[4] T. Schuler, D. Lofaro, L. Mcguire, A. Schroer, T. Lin, and D. Sofge, "A study of robotic swarms and emergent behaviors using 25+ real-world lighter-than-air autonomous agents (lta3)," in *SWARM2019*, 2019.

[5] J. Gibson, T. Schuler, L. McGuire, D. M. Lofaro, and D. Sofge, "Swarm and multi-agent time-based a* path planning for lighter-than-air systems," *Unmanned Systems*, vol. 08, no. 03, pp. 253–260, 2020. [Online]. Available: https://doi.org/10.1142/S2301385020500181

[6] S. Cho, V. Mishra, Q. Tao, P. Vamell, M. King-Smith, A. Muni, W. Smallwood, and F. Zhang, "Autopilot design for a class of miniature autonomous blimps," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 2017.

[7] S. Pandit and G. Singh, "An overview of spectrum sharing techniques in cognitive radio communication system," *Wireless Networks*, vol. 23, no. 2, pp. 497–518, 2017.

[8] F. Mondada, G. C. Pettinaro, A. Guignard, I. W. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. M. Gambardella, and M. Dorigo, "Swarm-bot: A new distributed robotic concept," *Autonomous Robots*, 2004.

[9] Ming Li, Kejie Lu, Hua Zhu, Min Chen, Shiwen Mao, and B. Prabhakaran, "Robot swarm communication networks: Architectures, protocols, and applications," in *2008 Third International Conference on Communications and Networking in China*, 2008.

[10] K. I. Easton and A. Martinoli, "Efficiency and optimization of explicit and implicit communication schemes in collaborative robotics experiments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. IEEE, 2002, pp. 2795–2800.

[11] Y. Tan and Z.-y. Zheng, "Research Advance in Swarm Robotics," *Defence Technology*, 2013.

[12] N. Gildert, A. G. Millard, A. Pomfret, and J. Timmis, "The need for combining implicit and explicit communication in cooperative robotic systems," *Frontiers in Robotics and AI*, vol. 5, p. 65, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2018.00065

[13] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin, "Effects of nonverbal communication on efficiency and robustness in human-robot teamwork," in *2005 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2005, pp. 708–713.

[14] C. Castelfranchi, G. Pezzulo, and L. Tummolini, "Behavioral implicit communication (bic): Communicating with smart environments," *IJACI*, vol. 2, pp. 1–12, 01 2010.

[15] Y. Mohan and S. Ponnambalam, "An extensive review of research in swarm robotics," in *2009 world congress on nature & biologically inspired computing (nabic)*. IEEE, 2009, pp. 140–145.

[16] Z. Wang and M. Schwager, "Multi-robot manipulation without communication," in *Distributed autonomous robotic systems*. Springer, 2016, pp. 135–149.

[17] F. Berlinger, M. Gauci, and R. Nagpal, "Implicit coordination for 3d underwater collective behaviors in a fish-inspired robot swarm," *Science Robotics*, vol. 6, no. 50, 2021.

[18] A. H. Purnamadjaja, J. Iskandar, and R. A. Russell, "Pheromone communication simulation for mobile robots using java 3d," in *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)*, 2007, pp. 261–266.

[19] D. W. Payton, M. J. Daily, B. Hoff, M. D. Howard, and C. L. Lee, "Pheromone robotics," in *Mobile Robots XV and Telemanipulator and Telepresence Technologies VII*, M. R. Stein, H. M. Choset, D. W. Gage, and M. R. Stein, Eds., vol. 4195, International Society for Optics and Photonics. SPIE, 2001, pp. 67 – 75. [Online]. Available: https://doi.org/10.1117/12.417331

[20] D. Payton, R. Estkowski, and M. Howard, "Compound behaviors in pheromone robotics," *Robotics and Autonomous Systems*, vol. 44, pp. 229–240, 09 2003.

[21] M. McPartland, S. Nolfi, and H. Abbass, "Emergence of communication in competitive multi-agent systems: a pareto multi-objective approach," in *GECCO '05*, 2005.

[22] J. Winters, "On the capacity of radio communication systems with diversity in a rayleigh fading environment," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 5, pp. 871–878, 1987.

[23] K. Hawick, J. Story, and R. Shepherd, "An architecture for swarm robots," *Technical Report DHPC-121, University of Wales, Bangor, North Wales*, 2002.

[24] F. Mondada, G. C. Pettinaro, A. Guignard, I. W. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. M. Gambardella, and M. Dorigo, "Swarm-Bot: A New Distributed Robotic Concept," *Autonomous Robots*, 2004.

[25] A. Turgut, F. Gökçe, H. Çelikkanat, L. Bayindir, and E. Sahin, "Kobot: A mobile robot designed specifically for swarm robotics research," *Technical Report METU-CENG-TR-2007-05, Middle East Technical University*, 2007.

[26] T. Zeng, M. Mozaffari, O. Semiari, W. Saad, M. Bennis, and M. Debbah, "Wireless Communications and Control for Swarms of Cellular-Connected UAVs," *Physics Letters B*, 2019.

[27] M. Chamanbaz, D. Mateo, B. M. Zoss, G. Tokić, E. Wilhelm, R. Bouffanais, and D. K. P. Yue, "Swarm-Enabling Technology for Multi-Robot Systems," *Frontiers in Robotics and AI*, 2017.

[28] P. Benavidez, K. Nagothu, A. Kumar Ray, T. Shaneyfelt, S. Kota, L. Behera, and Mo Jamshidi, "Multi-domain robotic swarm communication system," in *2008 IEEE International Conference on System of Systems Engineering*. Monterey, CA, USA: IEEE, Jun. 2008.

[29] M. Campion, P. Ranganathan, and S. Faruque, "Uav swarm communication and control architectures: a review," *Journal of Unmanned Vehicle Systems*, vol. 7, no. 2, pp. 93–106, 2018.

[30] ——, "A review and future directions of uav swarm communication architectures." in *EIT*, 2018, pp. 903–908.

[31] K. Grover, A. Lim, and Q. Yang, "Jamming and anti–jamming techniques in wireless networks: a survey," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 17, no. 4, pp. 197–215, 2014.

[32] M. Lichtman, J. D. Poston, S. Amuru, C. Shahriar, T. C. Clancy, R. M. Buehrer, and J. H. Reed, "A communications jamming taxonomy," *IEEE Security & Privacy*, vol. 14, no. 1, pp. 47–54, 2016.

[33] K. Dabcevic, "Intelligent jamming and anti-jamming techniques using cognitive radios," *PhD Programme in Computational Intelligence University of Genoa*, 2015.

[34] L. Ruetten, P. A. Regis, D. Feil-Seifer, and S. Sengupta, "Area-optimized uav swarm network for search and rescue operations," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020, pp. 0613–0618.

[35] W. Zhao, Z. Tang, Y. Yang, L. Wang, and S. Lan, "Cooperative search and rescue with artificial fishes based on fish-swarm algorithm for underwater wireless sensor networks," *The Scientific World Journal*, vol. 2014, 2014.

[36] M. Bakhshipour, M. J. Ghadi, and F. Namdari, "Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach," *Applied Soft Computing*, vol. 57, pp. 708–726, 2017.

[37] E. Hossain, D. Niyato, and Z. Han, *Dynamic Spectrum Access and Management in Cognitive Radio Networks.* Cambridge University Press, 2009.

[38] Q. Zhao and A. Swami, "A Survey of Dynamic Spectrum Access: Signal Processing and Networking Perspectives," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07.* Honolulu, HI: IEEE, Apr. 2007, pp. IV–1349–IV–1352. [Online]. Available: https://ieeexplore.ieee.org/document/4218359/

[39] H. Harada, "Software defined radio prototype toward cognitive radio communication systems," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, 2005, pp. 539–547.

[40] S. Pandit and G. Singh, "An overview of spectrum sharing techniques in cognitive radio communication system," *Wireless Networks*, vol. 23, no. 2, pp. 497–518, Feb. 2017. [Online]. Available: http://link.springer.com/10.1007/s11276-015-1171-1

[41] J. Iii, "An integrated agent architecture for software defined radio," *Ph.D. Dissertation, KTH*, 07 2000.

[42] A. H. Mahdi, J. Mohanan, M. A. Kalil, and A. Mitschele-Thiel, "Adaptive Discrete Particle Swarm Optimization for Cognitive Radios," in *2012 IEEE International Conference on Communications (ICC)*, Jun. 2012, pp. 6550–6554, iSSN: 1938-1883.

[43] Z. Zhao, S. Xu, S. Zheng, and J. Shang, "Cognitive radio adaptation using particle swarm optimization," *Wireless Communications and Mobile Computing*, vol. 9, no. 7, pp. 875–881, 2009, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcm.633. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcm.633

[44] J. Sonnenberg, D. B. Chester, J. Schroeder, and K. Olds, "Quantifying the relative merits of genetic and swarm algorithms for network optimization in cognitive radio networks," in *MILCOM 2012 - 2012 IEEE Military Communications Conference*, 2012, pp. 1–8.

[45] C. Doerr, D. Grunwald, and D. C. Sicker, "Local independent control of cognitive radio networks," in *2008 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, 2008, pp. 1–9.

[46] H. Anandakumar and K. Umamaheswari, "A bio-inspired swarm intelligence technique for social aware cognitive radio handovers," *Computers & Electrical Engineering*, vol. 71, pp. 925–937, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S004579061731193X

[47] N. J. Kaminski, "Social intelligence for cognitive radios," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2014, copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-05-21. [Online]. Available: http://mutex.gmu.edu/login?url=https://www.proquest.com/dissertations-theses/social-intelligence-cognitive-radios/docview/1906256809/se-2?accountid=14541

[48] M. Stojanovic and P.-P. J. Beaujean, "Acoustic communication," in *Springer Handbook of Ocean Engineering*. Springer, 2016, pp. 359–386.

[49] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE communications magazine*, vol. 47, no. 1, pp. 84–89, 2009.

[50] Z. Chen and J. J. Wiens, "The origins of acoustic communication in vertebrates," *Nature communications*, vol. 11, no. 1, pp. 1–8, 2020.

[51] A. Kumar, "Acoustic communication in birds," *Resonance*, vol. 8, no. 6, pp. 44–55, 2003.

[52] C. Hyacinthe, J. Attia, and S. Rétaux, "Evolution of acoustic communication in blind cavefish," *Nature communications*, vol. 10, no. 1, pp. 1–12, 2019.

[53] H. C. Gerhardt, "Acoustic communication in two groups of closely related treefrogs," *Advances in the Study of Behavior*, vol. 30, pp. 99–167, 2001.

[54] T. Forrest, "Acoustic communication and baffling behaviors of crickets," *The Florida Entomologist*, vol. 65, no. 1, pp. 33–44, 1982.

[55] F. A. Umrani, F. Ahmed Dahri, A. W. Umrani, and H. Bux Mangrio, "The Internet of LED: Indoor Visible Light Communication Using LED as Transmitter and Receiver," in *2018 21st Intl. Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2018.

[56] A. Poulose, "Visible light communication," in *Masters of Technology in Communications Systems, Christ Univ., Bengaluru, India*, 2017.

[57] T. Komine and M. Nakagawa, "Fundamental analysis for visible-light communication system using led lights," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 100–107, 2004.

[58] A. Jovicic, J. Li, and T. Richardson, "Visible light communication: opportunities, challenges and the path to market," *IEEE Communications Magazine*, vol. 51, no. 12, pp. 26–32, Dec. 2013, conference Name: IEEE Communications Magazine.

[59] M. Z. Afgani, H. Haas, H. Elgala, and D. Knipp, "Visible light communication using ofdm," in *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006.* IEEE, 2006, pp. 6–pp.

[60] S. Arnon, *Visible light communication.* Cambridge University Press, 2015.

[61] P. H. Pathak, X. Feng, P. Hu, and P. Mohapatra, "Visible light communication, networking, and sensing: A survey, potential and challenges," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2047–2077, 2015.

[62] L. E. M. Matheus, A. B. Vieira, L. F. M. Vieira, M. A. M. Vieira, and O. Gnawali, "Visible light communication: Concepts, applications and challenges," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3204–3237, 2019.

[63] L. Grobe, A. Paraskevopoulos, J. Hilt, D. Schulz, F. Lassak, F. Hartlieb, C. Kottke, V. Jungnickel, and K.-D. Langer, "High-speed visible light communication systems," *IEEE communications magazine*, vol. 51, no. 12, pp. 60–66, 2013.

[64] S. Kadirvelu and V. Baba, "Visible light communication using LED as receiver with the effect of ambient light," *Optical and Quantum Electronics*, 2017.

[65] Z. Tian, K. Wright, and X. Zhou, "The darklight rises: Visible light communication in the dark," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016, pp. 2–15.

[66] F. Berlinger, P. Wulkop, and R. Nagpal, "Self-organized evasive fountain maneuvers with a bio-inspired underwater robot collective," in *Intl. Conference on Automation and Robotics (ICRA)*, 2021, (finalist for best paper award).

[67] R. Murai, T. Sakai, H. Kawano, Y. Matsukawa, Y. Kitano, Y. Honda, and K. C. Campbell, "A novel visible light communication system for enhanced control of autonomous delivery robots in a hospital," in *2012 IEEE/SICE Intl. Symposium on System Integration (SII)*, 2012.

[68] H. Sharifi, A. Kumar, F. Alam, and K. M. Arif, "Indoor localization of mobile robot with visible light communication," in *2016 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2016.

[69] T. Tanaka and S. Haruyama, "New position detection method using image sensor and visible light leds," in *2009 Second International Conference on Machine Vision*, 2009, pp. 150–153.

[70] S. Haruyama, "Visible light communication using sustainable LED lights," in *2013 Proceedings of ITU Kaleidoscope: Building Sustainable Communities*, 2013.

[71] C. E. Yew, M. B. Hidayab, and C. C. Leong, "Visible light communication for robotic control," in *2018 IEEE 4th International Symposium in Robotics and Manufacturing Automation (ROMA)*, 2018, pp. 1–5.

[72] M. Vieira, R. Costa, A. Pereira, and P. Fonseca, "A validation framework for visible light positioning in mobile robotics," in *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 2016, pp. 47–52.

[73] K. Qiu, F. Zhang, and M. Liu, "Visible light communication-based indoor localization using gaussian process," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3125–3130.

[74] J. Armstrong, Y. A. Sekercioglu, and A. Neild, "Visible light positioning: a roadmap for international standardization," *IEEE Communications Magazine*, vol. 51, no. 12, pp. 68–73, 2013.

[75] W. Gu, M. Aminikashani, P. Deng, and M. Kavehrad, "Impact of multipath reflections on the performance of indoor visible light positioning systems," *Journal of Lightwave Technology*, vol. 34, no. 10, pp. 2578–2587, 2016.

[76] I. C. Rust and H. H. Asada, "A dual-use visible light approach to integrated communication and localization of underwater robots with application to non-destructive nuclear reactor inspection," in *2012 IEEE International Conference on Robotics and Automation*, 2012.

[77] K. B. Hunter, J. M. Conrad, and A. R. Willis, "Visible light communication using a digital camera and an led flashlight," in *IEEE SOUTHEASTCON 2014*, 2014.

[78] I. Takai, S. Ito, K. Yasutomi, K. Kagawa, M. Andoh, and S. Kawahito, "LED and CMOS image sensor based optical wireless communication system for automotive applications," *IEEE Photonics Journal*, 2013.

[79] R. Roberts, P. Gopalakrishnan, and S. Rathi, "Visible light positioning: Automotive use case," in *2010 IEEE Vehicular Networking Conference*. IEEE, 2010, pp. 309–314.

[80] Q. Wang, D. Giustiniano, and D. Puccinelli, "Openvlc: Software-defined visible light embedded networks," *VLCS 2014 - Proceedings of the 1st ACM MobiCom Workshop on Visible Light Communication Systems*, 09 2014.

[81] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross, "Led-to-led visible light communication networks," in *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*, 2013, pp. 1–10.

[82] S. Schmid, G. Corbellini, S. Mangold, and T. Gross, "An led-to-led visible light communication system with software-based synchronization," in *2012 IEEE Globecom Workshops*, 12 2012, pp. 1264–1268.

[83] S. Kornienko, O. Kornienko, and P. Levi, "Minimalistic approach towards communication and perception in microrobotic swarms," in *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, 2005.

[84] M. Dwight, "Stupid robot tricks: a behavior-based distributed algorithm library for programming swarms of robots," *Masters in Electrical Engineering and Computer Science, Massachusetts Institute of Technology*, 2005.

[85] R. Islam, I. Mobin, M. N. Shakib, and M. M. Rahman, "An approach of IR-based short-range correspondence systems for swarm robot balanced requisitions and communications," *ArXiv*, 2016.

[86] F. Berlinger, J. Dusek, M. Gauci, and R. Nagpal, "Robust maneuverability of a miniature low-cost underwater robot using multiple fin actuation," *IEEE Robotics and Automation Letters (RA-L)*, 2017.

[87] M. S. Islim and H. Haas, "Modulation techniques for li-fi," *ZTE communications*, vol. 14, no. 2, pp. 29–40, 2019.

[88] J. Rani, P. Chauhan, and R. Tripathi, "Li-fi (light fidelity)-the future technology in wireless communication," *International Journal of Information & Computation Technology*, vol. 7, pp. 1517–1520, 01 2012.

[89] D. Tsonev, S. Videv, and H. Haas, “Light fidelity (li-fi): towards all-optical networking,” in *Broadband Access Communication Technologies VIII*, vol. 9007. International Society for Optics and Photonics, 2014, p. 900702.

[90] P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128-128 120 db 15 -s latency asynchronous temporal contrast vision sensor,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[91] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, “A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor,” *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[92] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis *et al.*, “Event-based vision: A survey,” *arXiv preprint arXiv:1904.08405*, 2019.

[93] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, “The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam,” *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 142–149, 2017.

[94] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, “The multivehicle stereo event camera dataset: An event camera dataset for 3d perception,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.

[95] H. Rebecq, D. Gehrig, and D. Scaramuzza, “Esim: an open event camera simulator,” in *Conference on Robot Learning*. PMLR, 2018, pp. 969–982.

[96] G. Gallego, C. Forster, E. Mueggler, and D. Scaramuzza, “Event-based Camera Pose Tracking using a Generative Event Model,” *arXiv:1510.01972 [cs]*, Oct. 2015, arXiv: 1510.01972. [Online]. Available: http://arxiv.org/abs/1510.01972

[97] G. Gallego, J. E. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, “Event-based, 6-dof camera tracking from photometric depth maps,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2402–2412, 2018.

[98] Y. Jin, L. Yu, G. Li, and S. Fei, “A 6-dofs event-based camera relocalization system by cnn-lstm and image denoising,” *Expert Systems with Applications*, vol. 170, p. 114535, 2021.

[99] S. Bryner, G. Gallego, H. Rebecq, and D. Scaramuzza, “Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 325–331.

[100] D. Weikersdorfer and J. Conradt, “Event-based particle filtering for robot self-localization,” in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2012, pp. 866–870.

[101] E. Ceolini, C. Frenkel, S. B. Shrestha, G. Taverni, L. Khacef, M. Payvand, and E. Donati, "Hand-gesture recognition based on emg and event-based camera sensor fusion: A benchmark in neuromorphic computing," *Frontiers in Neuroscience*, vol. 14, p. 637, 2020.

[102] R. Ghosh, A. Mishra, G. Orchard, and N. V. Thakor, "Real-time object recognition and orientation estimation using an event-based camera and cnn," in *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*. IEEE, 2014, pp. 544–547.

[103] A. Sokolova and A. Konushin, "Human identification by gait from event-based camera," in *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, 2019, pp. 1–6.

Curriculum Vitae

Alexander Maxseiner received his Bachelors in Computer Engineering from George Mason University. During his Master's degree in Electrical Engineering he interned at the United States Naval Research Laboratory. After finishing his Master's degree in Electrical Engineering he will start a full-time position at the United States Naval Research Laboratory.