Freie Universität Berlin
Institut für Informatik

BACHELORARBEIT

A Beehive Monitoring System Incorporating Optical Flow as a Source of Information

Maximilian Michels

23.03.2011

Gutachter

Prof. Dr. Raúl Rojas

Betreuer

Tim Landgraf

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst habe. Ich habe dazu keine weiteren als die angegebenen Hilfsmittel benutzt und die aus anderen Quellen entnommenen Stellen als solche gekennzeichnet.

Maximilian Michels

Abstract

The goal of this thesis is to design, implement, and evaluate a Beehive Monitoring System. Such a system acquires data about the beehive and its surrounding. It does that by reading out sensors for measuring temperature, weight, weather, and movement. Unlike other systems that have been developed in the past, it incorporates the concept of optical flow to gather information about the movement of the bees within the hive. This new source of information can prove to be helpful for scientists in their research about bees.

»If the bee disappeared off the surface of the globe then man would only have four years of life left.«[1]

[1] Although often attributed to Albert Einstein even by serious media, there is no source proving him to be the author of this quote.

Contents

1 Introduction

1.1 Motivation

Bees are of fundamental importance for nature. They pollinate flowering plants of all kind, including agricultural crops. Bees as one of the main »[...] pollinators are an environmental resource as critical to the long-term survival of a population as are light, moisture, etc.« [1]. Since in nature everything is linked the impact of a demise of bees would be very significant to life on earth.

What at first seems like an unlikely event could become reality soon. Scientists around the world puzzle about the continuing high losses in bee colonies that get worse every year [2]. Those losses have become known under the name of Colony Collapse Disorder, where high losses in bee colonies occur that eventually lead to the obliteration of the beehive [3]. In most cases beekeepers are surprised when high losses in bee colonies occur and it's too late to take actions against the diseases. Although scientists have found out about what causes the bees to die, they have yet to find out what causes those diseases to break out [3].

In fact, that's not the only phenomenon bee scientists try to understand. The degree of understanding of bees is far from complete: »Honey bees are the most complex organism to have evolved along an evolutionary trajectory that began more than a billion years ago.« [4].

The goal of scientists doing research about bees should be to develop techniques which provide new aspects of bee behavior and communication. One example for such a new technique is a robot called »Robobee«[2] which should ultimately imitate the bee dance. It is being developed at Freie Universität Berlin and, just like this thesis, a corporation between the neurobiology and the computer science department.

To explore the outbreak of a disease like Colony Collapse Disorder and to possibly find out about its cause, an automated system that collects information about the beehive can be very meaningful. Such a system could not only be used to track down diseases in bee colonies but also for other types of research. Behavior research, for example, would benefit from automation of time costly and tedious tasks, some of them being impossible to do by hand. Moreover, beekeepers could use an automated system in a daily routine.

An automated system monitors certain values of the beehive and its surrounding. It does that by reading out several sensors. The data can be used to judge about the current health status of the bee colony. Also, the automated system archives the acquired data so that later on conclusions can be drawn from the data. We call such a system a Beehive Monitoring System. In the course of this thesis we will design and implement such a system.

[2] http://robobiene.mi.fu-berlin.de/

1.2 Beehive Monitoring System

A good measure for monitoring a beehive could be the temperature within the hive, the weight of the beehive, and the weather conditions. However, there's much more information available.

When beekeepers open up their beehive they can form an opinion pretty fast of what's going on in the hive. They're doing that by using their sense of sight and the computational power of their brain. Why not include this additional source of information in a monitoring system?

Optical information is crucial when the movement of the bees is of interest. The high computational power of modern computers and the computer vision algorithms developed in the last decades will provide the monitoring system with a sense of sight. Other sources of information could also be desirable, for instance, acoustic information. It's not monitored in this thesis but currently investigated by the team of Prof. Menzel at the institute of neurobiology at Freie Universität Berlin.

There have been many commercial approaches for beehive monitoring systems as in [5], [6], [7], or even a patent for a commercial system [8]. Even so, this paper couldn't find any system based on scientific research. In the course of this thesis we won't build a handy commercial beehive monitoring system. In fact, the system will be quite large, unsuitable for commercial sales, and use an observation beehive rather than a normal beehive.

Then what makes this observation system of particular interest? All of the above monitoring systems use conventional sensors like temperature and weight sensors. We will make use of conventional sensors but also capture the optical flow within the beehive, and prove it to be a new source of information in a beehive monitoring system. We will start off by determining the measures to use in a beehive monitoring system (2.1). Also, we will constitute the conditions for such a system (2.2). According to what measures we determined we will then choose the hardware (3.1). Afterwards we will link the hardware to create a beehive monitoring system (3.2). We will look at how the data captured by the system can be evaluated and prove the optical flow data to be a new source of information (4). Finally, we provide an outlook of how the system could be applied to generate a warning in case of Colony Collapse Disorder (5).

2 System Design

2.1 Subject to Monitoring

2.1.1 Beehive

The beehive will be monitored by observing the following variables:

Temperature Inside the Beehive The bee organism gives off heat. Observing the temperature inside the beehive thus is a good indicator for the health of the bees. The temperature will be measured inside the combs so that the measuring point is close to the bees.

Weight of the Beehive The weight supplies information about the development of a beehive. An incline of the weight shows that the beehive is active, meaning it produces brood or honey. It should be evident that any activities that require the hive to be opened or touched (e.g. feeding or exchanging combs) will change the weight.

Weather of the Surrounding Area One can only understand the measurements inside the beehive correctly, if information about the weather is available. A fall in temperature might be because of an outbreaking disease within the beehive, or it might be because of a cold front. That's why the weather provides a basis to further classify other data.

Images of the Beehive A picture can be worth a thousand words. It can help the beekeeper to get an impression of the beehive development. Also, later on, it can help to see how the hive developed over time. For this purpose we will provide a live stream as well as old pictures of the beehive.

2.1.2 Optical Flow within the Hive

Measuring the optical flow provides a source of information about the motion in a sequence of images. When capturing the optical flow within a beehive that means monitoring the movement of the bees. We will show later, that this information can then be used to calculate the activity of the bees.

Optical flow is defined to be the pattern of motion in an image that transforms this image as seen by an observer into the next image as seen by the same observer at the same angle. Optical flow is the information that tells us everything about the motion of objects in a sequence of images. We generally consider an object moving when its patterns of brightness move.

Granted we think of motion in a picture as motion of brightness patterns in a small region of the image without the change of brightness in the overall image (i.e. change of lighting condition), we can obtain a constraint for the motion in the image[9]:

The 2D Motion Constraint Let the function $I(x, y, t)$ be the brightness at Point (x, y) at time t. Consider a brightness pattern moving in a nxn window to another. Thereby the brightness of the pattern doesn't change. It has only been translated. Thus, the following equation holds:

$$I(x, y, t) = I(x + \delta_x, y + \delta_y, t + \delta_t)$$

Applying the first expansion of the Taylor series about the right side of the equation, we can continue:

$$I(x+\delta_x, y+\delta_y, t+\delta_t) = I(x, y, t)+I(\delta_x, \delta_y, \delta_t) \Longleftrightarrow I(x, y, t) = I(x, y, t)+\frac{\partial I}{\partial x}\delta_x+\frac{\partial I}{\partial y}\delta_y+\frac{\partial I}{\partial t}\delta_t+r$$

The symbol r resembles the higher order terms which are needed to satisfy the equation. Now we subtract $I(x, y, t)$ from both sides:

$$0 = \frac{\partial I}{\partial x}\delta_x + \frac{\partial I}{\partial y}\delta_y + \frac{\partial I}{\partial t}\delta_t + r$$

Afterwards we divide through δ_t:

$$0 = \frac{\partial I}{\partial x}\frac{\delta_x}{\delta_t} + \frac{\partial I}{\partial y}\frac{\delta_y}{\delta_t} + \frac{\partial I}{\partial t}\frac{\delta_t}{\delta_t} + \frac{r}{\delta_t} \Longleftrightarrow 0 = \frac{\partial I}{\partial x}\frac{\delta_x}{\delta_t} + \frac{\partial I}{\partial y}\frac{\delta_y}{\delta_t} + \frac{\partial I}{\partial t} + \frac{r}{\delta_t}$$

If we now take the limit $\lim_{\delta_t \to 0} \frac{\partial I}{\partial x}\frac{\delta_x}{\delta_t} + \frac{\partial I}{\partial y}\frac{\delta_y}{\delta_t} + \frac{\partial I}{\partial t} + \frac{r}{\delta_t}$, this becomes:

$$\frac{\partial I}{\partial x}\frac{\delta_x}{\delta_t} + \frac{\partial I}{\partial y}\frac{\delta_y}{\delta_t} + \frac{\partial I}{\partial t} = 0$$

Let $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $I_t = \frac{\partial I}{\partial t}$, $v_x = \frac{\delta_x}{\delta_t}$, and $v_y = \frac{\delta_y}{\delta_t}$:

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \end{bmatrix} = -I_t$$

The above equation is called the 2D Motion Constraint[9], but also known as the Brightness Constancy Constraint [10] because it can also be derived by applying the chain rule about $\frac{dI}{dt} = 0$, meaning the brightness of a pattern is constant over time [10].

What does the Brightness Constancy Equation tell us about the optical flow components v_x and v_y? We can't solve the equation because of these two unknowns. However, we know a little bit more about them because the solution for two variables that depend on each other always lies on a line (see figure 1).

The Aperture Problem The reason why we can't solve this equation is also called the Aperture Problem[10]. It says that it's not possible to determine the optical flow correctly when

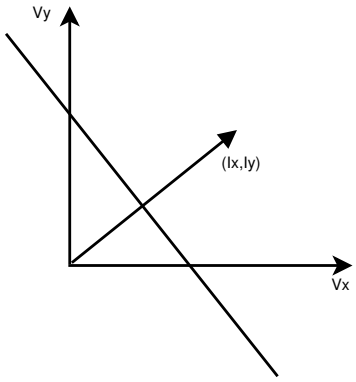


Figure 1: The Constraint, Aperture Problem

a moving object is only partial visible. Consider a moving rotating sphere with constant lighting. If the sphere's texture is plain, an observer won't see any motion.

Additional Contraint To solve the equation and to determine optical flow, it's necessary to introduce an additional constraint. Up till now we only determined optical flow locally. That means we observe a single pixel and not its neighbors.

For this purpose, several algorithms have been developed. The most prominent being the Horn-Schunk algorithm [10] and the Lucas-Kanade algorithm [11]. The Horn-Schunk algorithm assumes the flow is more or less even without any extreme values. That means it is more likely to calculate an optical flow for a pixel in an image that is similar to other values in the image [10]. On the other hand, the Lucas-Kanade algorithm assumes that the optical flow in the area around a pixel is similar [11].

Both algorithms achieve additional variables by making assumptions about other parts of the image. To finally calculate the optical flow, all equations need to be minimized in some way because of too many variables resulting in too much information. So it becomes clear, calculating the optical flow essentially means estimating it. We saw earlier that this is because of the Aperture Problem. For this thesis, the Lucas-Kanade algorithm will be used because it proved to be better for this application.

Limitations There is a limited amount of features one can track with the above method. This is because of three reasons. First of all, it requires too much computational power to track a large amount of features. Secondly, the more bees we track, the denser the features. It's then very likely that features get lost or mixed up. Thirdly, each honeycomb has two sides and thus many bees will leave the image.

It's also questionable whether the features actually will be bees. Clearly, the algorithm doesn't care about what it tracks. But we can infer the selection of bees as features by using the »Good Features To Track«[12] algorithm to choose features with a good contrast. We can then ensure with the right lightning and camera position that mainly bees are being tracked.

However, if other objects like the honeycomb or the frame are being chosen as features it won't distort the quality of the captured data because they are still objects and thus have an optical flow of zero.

2.2 Experiment Environment

Types of Bees All types of bees belong to the family group taxon. Bees that were monitored in the course of this thesis are of the type Apis Mellifera, also known as European honey bee. The European honey bee made its way from Asia and adapted well because of less heat in the summer [4]. The Downside is that the winters are stronger which leads to more honey storage for the bees to survive [4]. It's the most commonly cultivated bee by humans. The race common in Germany is Apis Mellifera Carnica [4].

Modern Bee Keeping In the year 1853, Lorenzo Lorraine Langstroth (1810-1895) published »Langstroth's Hive and the Honey Bee« [13] which brought bee keeping to the next level. Nowadays, the vast majority of beekeepers use the type of beehive he developed. Before Langstroth invented the modern beehive, beekeepers struggled with gaps between the framed honeycombs. The bee colony would fill those spaces with comb or propolis. Thus, defeating the idea of movable honeycomb frames within the beehive and making it very hard to remove the frames to look after the bees. Langstroth found out, that one centimeter space around the honeycomb frames is enough for the bees to move around and prevent them from filling the gaps [13]. Modern beehives are thus also called Langstroth beehives. As said, it's the most common beehive for human cultivation of bees. This thesis, however, uses a bee observation hive which basically is a Langstroth hive with only one column and two rows for removable comb frames. It has a window on each side for observation (see figure 2).

Setting The beehive was put in a room which grants access to the outside through a tube. During winter and spring a heater is used to keep the room warm. The bees can exit the beehive, and therefore the room, trough a tube that leads outside. The room was darkened by applying foil to the window (see section 3.1.5).

Natural Behavior Unnatural conditions might disturb the bees. It defeats the purpose of behavioral science when bees don't behave like they normally do. So it's most important for monitoring that the environment is (or seems) natural to the bees. No sources about a change in behavior due to the type of beehive could be found for this thesis. Apart from this, many scientists have experimented with different light intensities. Generally speaking, the more intense and the more area is covered with light, the more bees will be turning towards the light [14]. So it can be assumed, bees accept the Langstroth beehive when there is no or hardly any visible light for them.

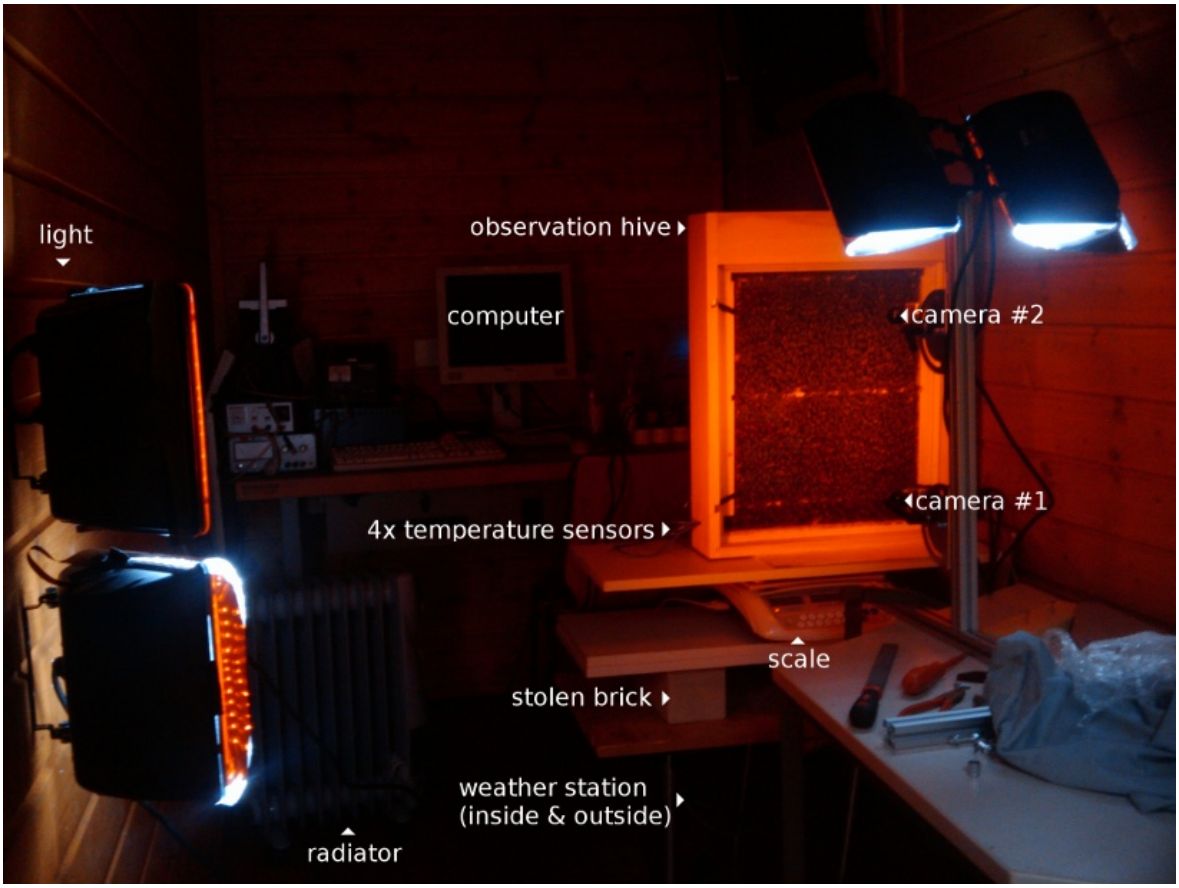


Figure 2: Setting for monitoring the observation hive

Lighting Bees hardly need any light in their beehive, but for using a camera to monitor the optical flow of the bees, light is needed. Karl von Frisch prove, bees can see many colors [15]. Von Frisch showed, bees can even see ultraviolet light which can't be seen by humans. However, bees can't see infrared light [15]. Although often asserted differently, bees can see red light up till a light wave length of 550 - 650 nm (pure red light has 650 - 750 nm) [16]. Thus, in order to film inside the beehive there are too possibilities. Either use an infrared camera or a regular camera combined with pure red light. Due to limited budget the latter was chosen. Of course, there's a tradeoff because we don't know whether the bees will behave differently because of the of the light spectrum they can still see. For this thesis, there was no other way of doing it.

3 Implementation

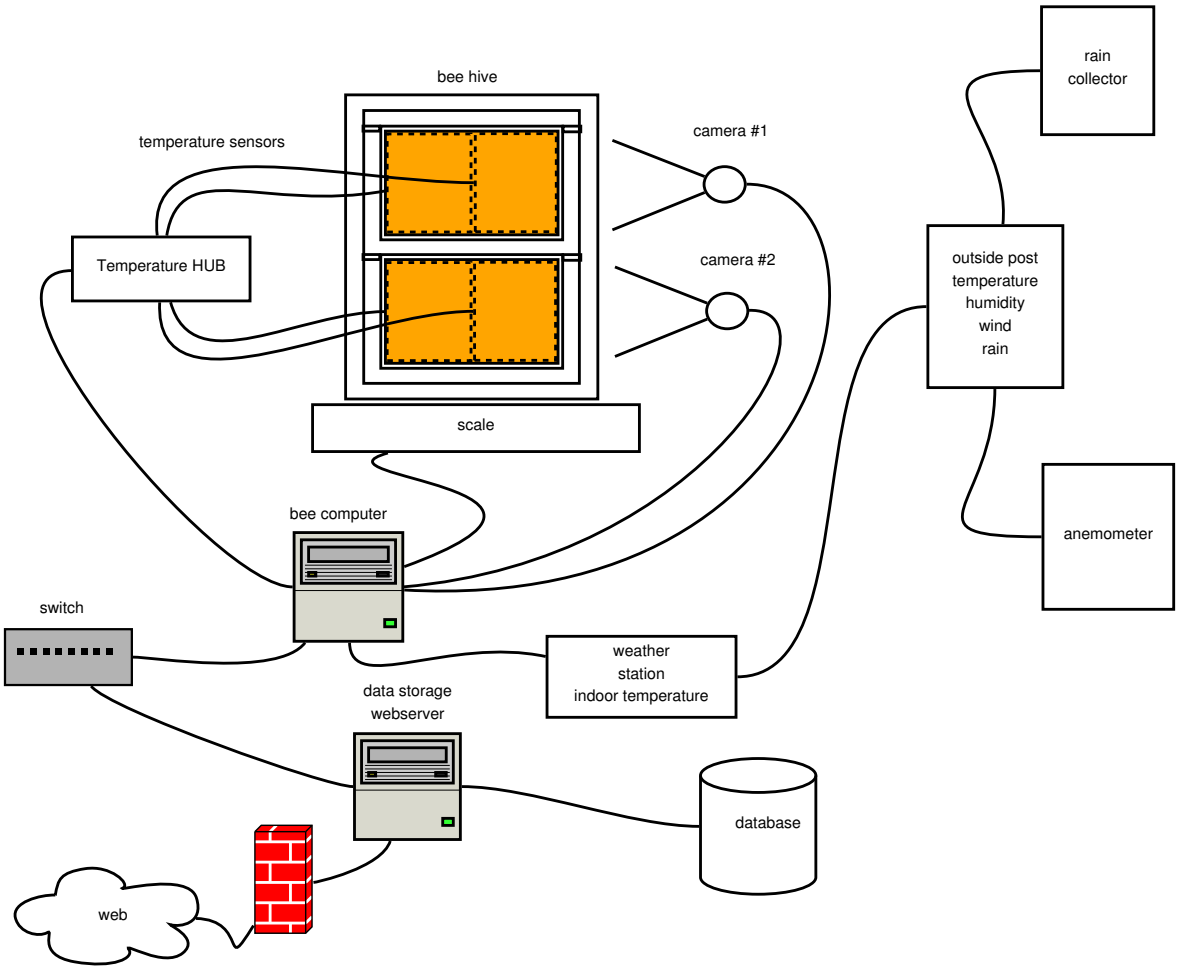
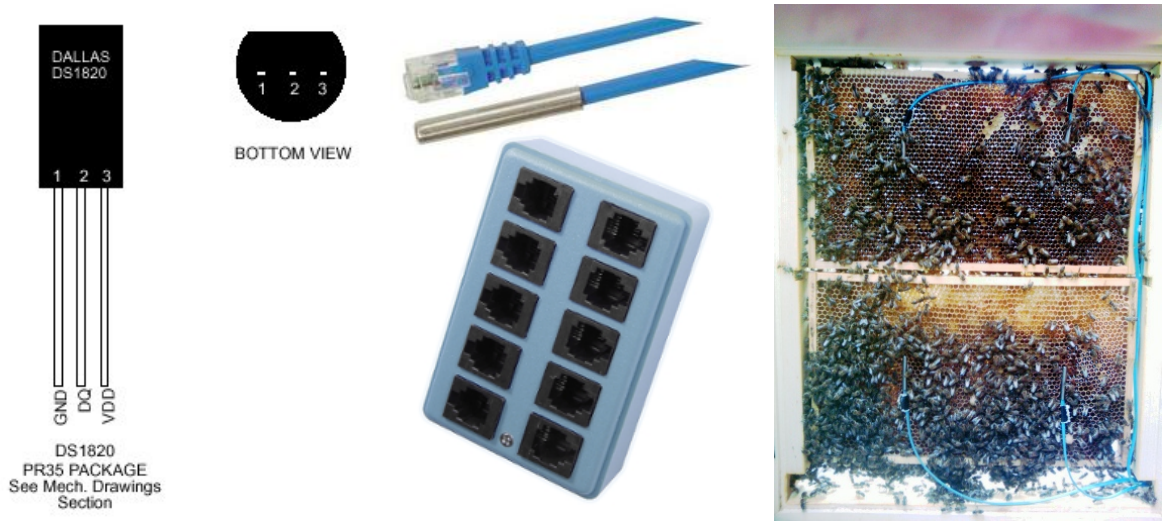


Figure 3: The observation hive sits on a scale. The temperature sensors are put in the combs. Weather conditions indoor and outdoor are measured by the weather station. Two cameras capture images of the combs. All sensors are read out by bee computer (A). If available, it transfers data to the data storage computer and web server (B). Computer B hosts tools for evaluation purposes. A web site can be accessed from the outside via the web, too.

3.1 Hardware

Except for the temperature sensors inside the beehive, all hardware was supplied by Freie Universität Berlin and was not explicitly chosen for the experiment. The main task to be solved in this chapter was to choose hardware and then make it accessible for the computer so that later on it could be read out by the software. Figure 3 shows what hardware is used and how it is linked.



(a) (b) (c)

Figure 4: DS1820 temperature sensor's connections (a), sensor's cover & hub for parallel connection (b), and beehive equipped with sensors (c)

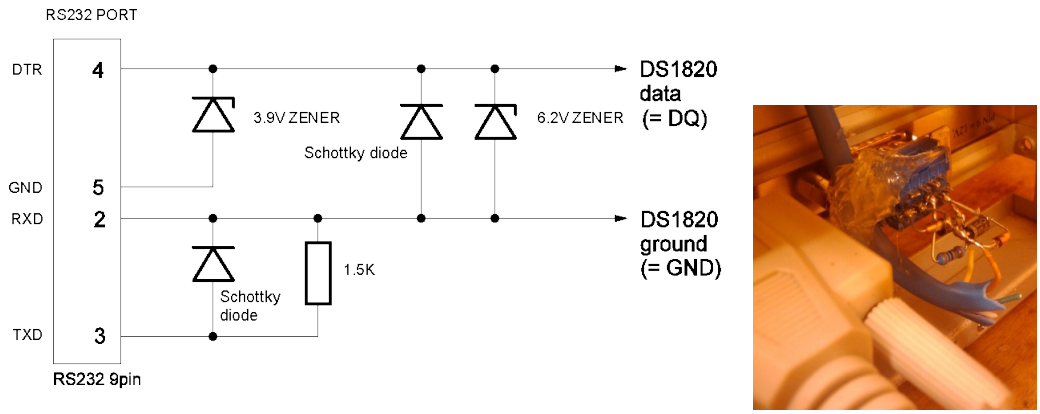
3.1.1 Temperature Sensors

For measuring the temperature inside of the beehive, the »DS1820 Digital Thermometer« (see figure 4) by Dallas Semiconductor was used. The DS1820 has a one-wire interface, which means only one port is necessary to send or receive data from the sensor. That makes it very easy to use. It can be read out via the serial port. Also, it can be powered through the data line which is very convenient because no external power supply is necessary. Such power supply is also referred to as parasitic power. The parasitic power supply is made possible by using the RDX signal of the serial port as a feed back (see figure 5).

The temperature is read out by an open source software called »digitemp« [18]. It runs on both, Windows and Linux. Multiple temperatures can be read out via the serial port when the temperature sensors are connected parallel using a hub (see figure 4). Each sensor has a unique 64-bit serial code which can be used for identification. According to the data sheet the sensor can measure temperatures in the range of -55°C to 125°C. The accuracy is 0.5°C when measuring temperature between -10°C and 85°C. For this thesis this seems to be a sufficient error rate. Actually, the placement of the sensors within the beehive is of much greater importance for the accuracy. The sensors were placed in the combs so that each sensor's radius covered half of the comb frame.

3.1.2 The Weather Station

The weather station WS-2310 by HeavyWeather supplied information about the weather. The weather station comes with a temperature and humidity sensor for inside, and an outpost for



(a) (b)

Figure 5: The circuit diagram for the adapter to read out the DS1820 via serial port (a), the adapter built accordingly plugged into the serial port of computer A (b) [17]

(a) (b) (c) (d)

Figure 6: weather station: central unit (a), outpost (b), anemometer (c), and rain collector (d)

measuring temperature, humidity, wind speed and direction, and rain fall outside. All sensors are connected to a central unit inside (for all items see figure 6).

Since the data protocol for the serial port connection of the weather station is not given details of, the only way to communicate with the weather station is to use the supplied proprietary software. It reads out the data from the sensors once a minute and writes them down in a text file that can be read out. This is not an elegant way of reading out the weather measures but in practice it proved to be reliable.

The data sheet of the weather station provides information about the resolution of the sensors but not about the accuracy. However, there are many advices for the right placement of the sensors to achieve accurate measures. That leads us to the conclusion that the placements of the sensors have a greater influence on accuracy than the actual error rates. The sensors were placed in best position possible. It should be taken into considerance that the sheltered test area is not the optimal place for weather measurements. However, we can back up the data with data from a weather service.

3.1.3 The Scale

Figure 7: »PCE-PCS 30« scale below the yellow beehive

The weight was measured by a »PCE-PCS 30« (see figure 7). It has a serial port connection that can be configured either to send out the weight when it changes or to send it every second. For the sake of simplicity the latter option was chosen. The scale was put under the beehive. It's worth saying that without a level surface the measured values from the scale won't be accurate.

The scale can measure weight up till 30 kg in steps of 0.5 g. The error rate is 0.5 g. Worker bees weigh about 90 mg. So the error rate is roughly equivalent to the weight of six bees. Six bees are a relatively small amount compared to the hundreds of bees which are in the beehive. That makes the error rate bearable.

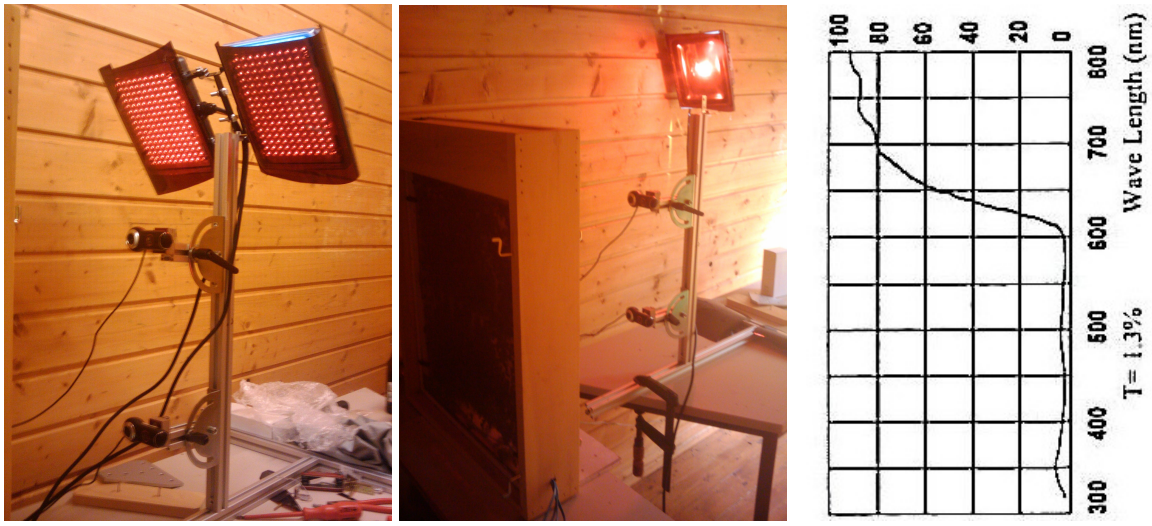
3.1.4 Cameras

Both cameras are Logitech Pro 9000. They have a maximum resolution of 2 mega pixel and are connected to the computer via USB. For the sake of performance when processing images, a lower resolution (640 x 480) was used. The cameras were mounted on a custom-built socket (see figure 8).

3.1.5 Lighting

Cheap spotlights originally manufactured for construction work were used at first. Those spotlights proved not to be good for the long term run because the LEDs were getting darker over time. The new spotlight was more expensive but stayed as bright as on the first day (compare figure 8).

As mentioned early (see section 2.2), bees can see many colors but they can't see pure red (650 - 750 nm). To avoid disturbing the bees, the front of the spotlight was covered with red foil. The foil used doesn't eliminate all other frequencies (as in figure 8) which, as we



(a) (b) (c)

Figure 8: socket with cameras and old spotlights mounted (a), socket with cameras and new spotlight mounted to the wall (b), and the foil's light spectrum (c)

showed in 2.2, does have an influence on bees. LEDs don't emit much heat so the foil on the spotlight doesn't heat up much.

3.2 Software

3.2.1 Software Architecture

Figure 3 represents the set-up for the beehive monitoring system. Two computer systems are deployed. One computer next to the beehive (hereafter A) is used to read out the sensors and process the video images. The second computer (hereafter B) at a different place is used to store all data acquired and provide the means to access this data, i.e. via a web server. The reasoning behind deploying two computers is to store data at a safe and dry place to maximize performance and reliability. Reliability can't be assured for computer A situated near the bees where weather conditions vary from time to time. Just as a high load on the web server running on computer B should not slow down or stop the tracking on computer A. Thus, it makes sense to have two computer to separate important tasks. The goal is also to minimize dependability of computer A on computer B. If, for example, computer B is no available for storing sensor data, computer A should continue capturing data as usual and store the data until computer B is available again.

3.2.2 Reading out sensors

All sensors are read out once a minute. The resulting data is then stored in a buffer. Another process sends out the buffer data to computer B and clears the buffer afterwards (see figure

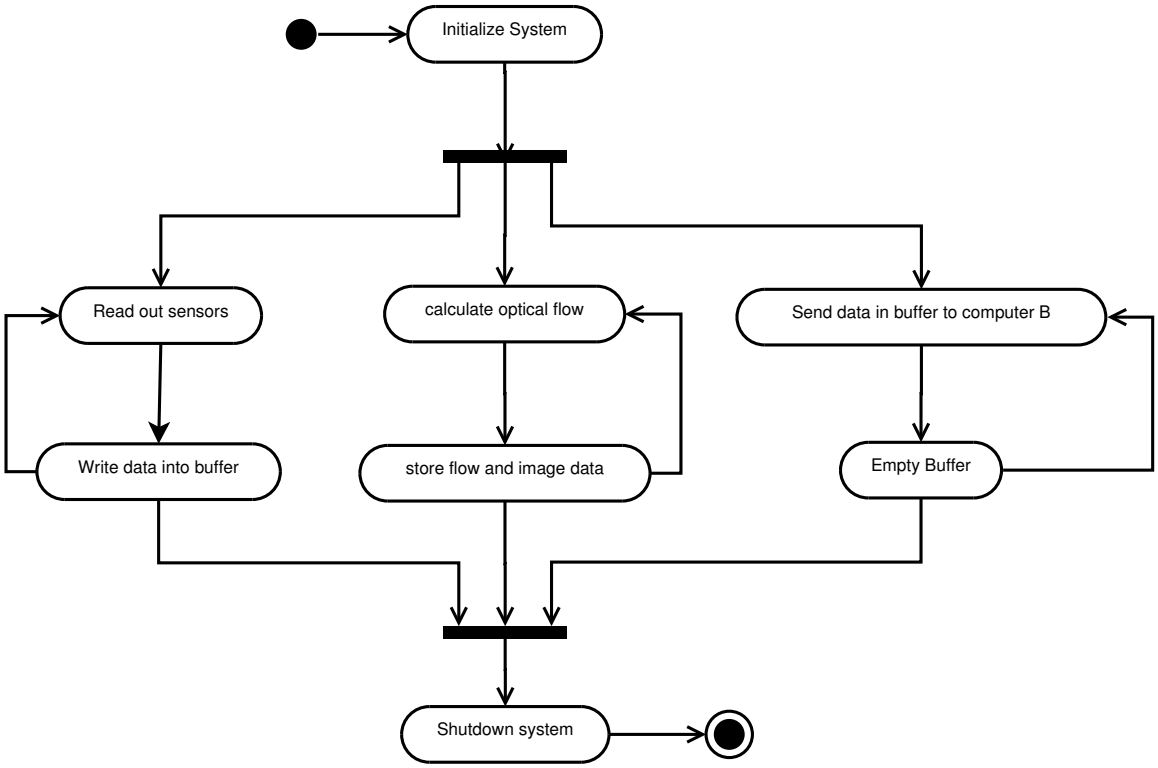


Figure 9: Activity diagram of the program running on computer A. It involves three concurrent tasks: Reading out the sensors, calculating the optical flow, and sending the buffered data to computer B.

9). If computer B is not available, the buffered data will be held until computer B is available again. It's important to take into consideration that concurrent read/write operations on the buffer have to be atomic or use mutual exclusion to prevent incoherence of the data. Semaphores were used to assure mutual exclusion.

3.2.3 Optical Flow

Optical flow is approximated using the »Good Features to Track« algorithm [12] and the optical flow calculation algorithm by Lucas & Kanade [11]. These algorithms are implemented in »Open CV« [19], an open source library »of programming functions for real time computer vision« [19]. Using the library functions, a program was built to process the image data of both web cams. We will refer to such a program as a tracking program or tracker.

It should be evident that, due to limited processing time and power, the tracker can only approximate the optical flow between a sequence of images. When measuring optical flow, it's important to find a balance between time of processing (i.e. frame rate) and the number of features tracked. For this set-up, 150 features for each camera and a frame rate of 15 frames per second proved to be the best balance between speed and quality of the optical flow measurement.

Taking into account that changes in optical flows will not be monitored correctly, because of loss of features, it's best to find new features from time to time. Again, the optimal setting depends on the environment of the experiment. For this experiment 100 frames proved to be the best setting for minimizing error rate and measuring optical flow with a set of features big enough to get an impression of the movement.

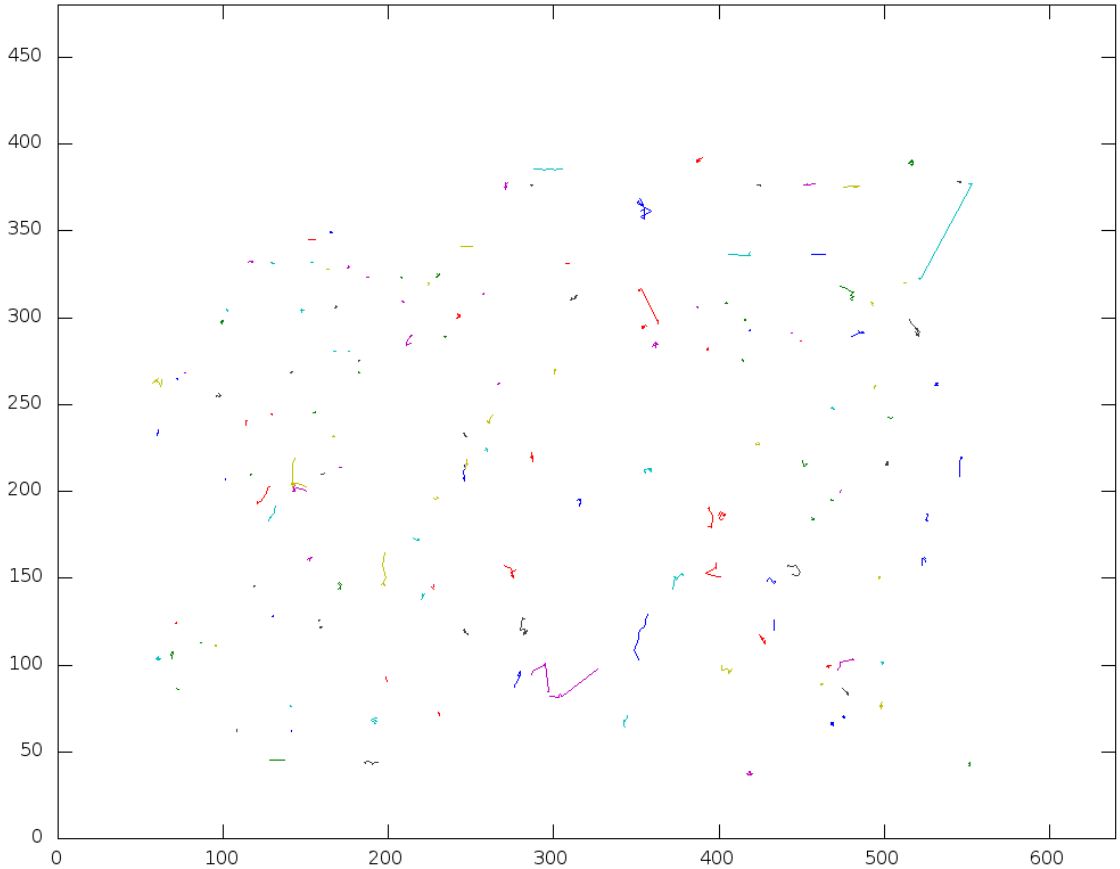


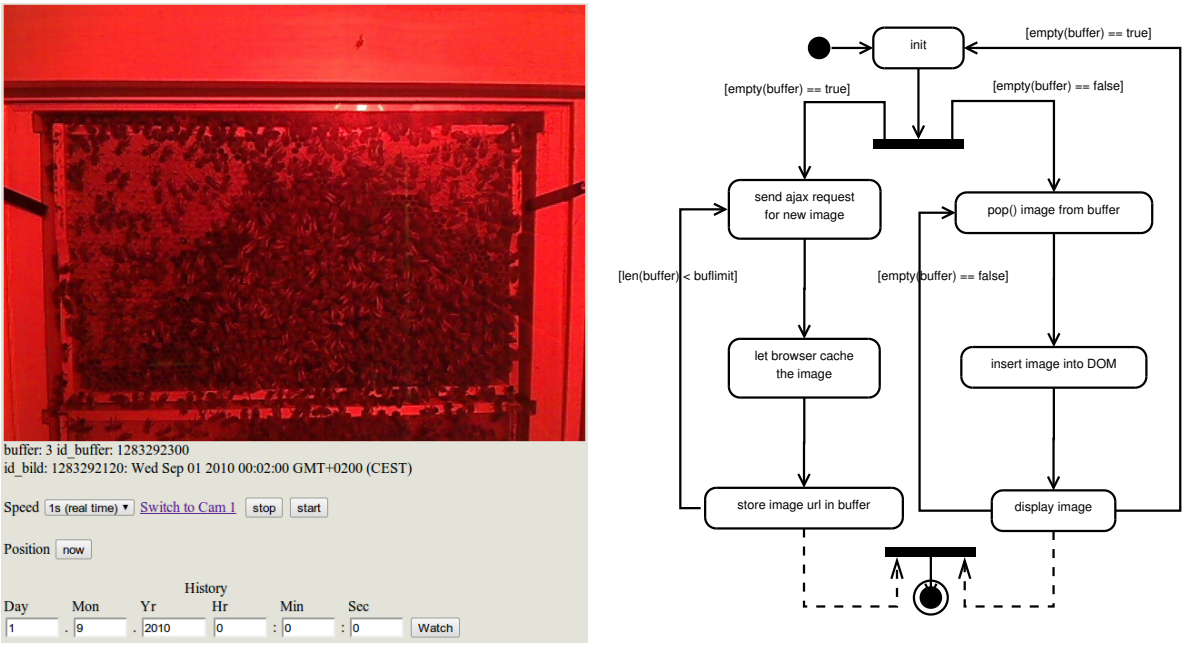
Figure 10: tracked features over a sequence of 100 frames

3.2.4 Broadcasting Images of the Beehive

A picture of the beehive is made every second. This picture is then send over the network together with the sensor and flow data. A web server is running on computer B. When the user enters the broadcasting website (see figure 11), he sees the the newest picture of the beehive which is then replaced every second (smaller intervals are also possible) by a newer picture of the beehive. The replace method uses Ajax to achieve this (see figure 11). For reasons of simplicity and browser platform independence this is implemented by using »JQuery«[20], a JavaScript library. Using JavaScript over reloading a page every x seconds has several advantages. First of all, it uses less bandwidth because the whole page doesn't have to be transferred again each time. Secondly, it's not possible to buffer a picture before

showing it. Thirdly, reloading a page or even a frame doesn't perform when the refresh interval is smaller than one second.

It's possible to change the speed of the playback as well as pausing it. A history feature was built in to watch older pictures of the beehive. Figure 11 shows the streaming website and how the JavaScript streaming works.



(a) (b)

Figure 11: Website for broadcasting the beehive images (a). It enables the users to change the playback speed, pause the broadcasting, change the speed of the playback, and view old images of the beehive. The activity diagram (b) demonstrates the technique used for streaming the images. One process caches up to buflimit images. The other process displays cached images to the user.

3.2.5 Data Transmission and Archiving

Data Transmission In order to transfer all sensor data from computer A to computer B a rudimentary transfer protocol was developed (see figure 12). The protocol uses the TCP/IP protocol suite as a basis. It assures a reliable bidirectional stream-orientated connection. All packets have a sequence number and thus arrive in the same order as sent. TCP checks the integrity of all data and requests a packet again if it was corrupted. No TCP connection routines (i.e. three way handshake, SYN/ACK, etc.) are included in the figure. Same applies for protocols underneath the TCP/IP protocol like the Ethernet protocol.

If B wants to establish a connection with A, A needs to be reachable for B. Also B has to know A's IP address. If A has a static IP address, it's no problem to hardcode A's address into B's program code. However, depending on which network A is connected, the IP might vary. Best

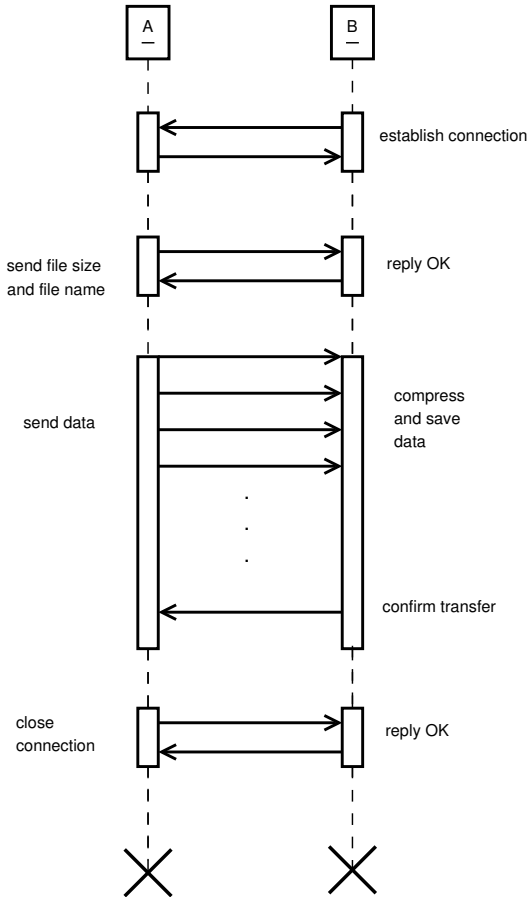


Figure 12: System sequence diagram for transferring all sensor data from computer A to B.

is a domain name pointing to A's IP address. This can be achieved by letting A find out about its IP address and then update a dynamic domain name. We can then safely hard-code A's dynamic host name which has to be resolved every time we want to establish a connection with A.

For transferring the sensor data that have been stored into an array we use a Python's serialization class cPickle. With cPickle we can convert ordinary data types (i.e. lists, tuples, arrays, etc.) into a byte stream. The stream can then, after being sent over the network, be turned into structured data again and be stored in a database. The beehive images and the optical flow sensor data are transferred directly over the network and then compressed right before they are being saved by computer B.

3.2.6 Website

On computer B an Apache web server was set up which hosts a website created for the thesis. The website provides information about the project. The visitor can access and evaluate all monitored data and watch the live stream of the the cameras. The website is available at http://beestream.de.

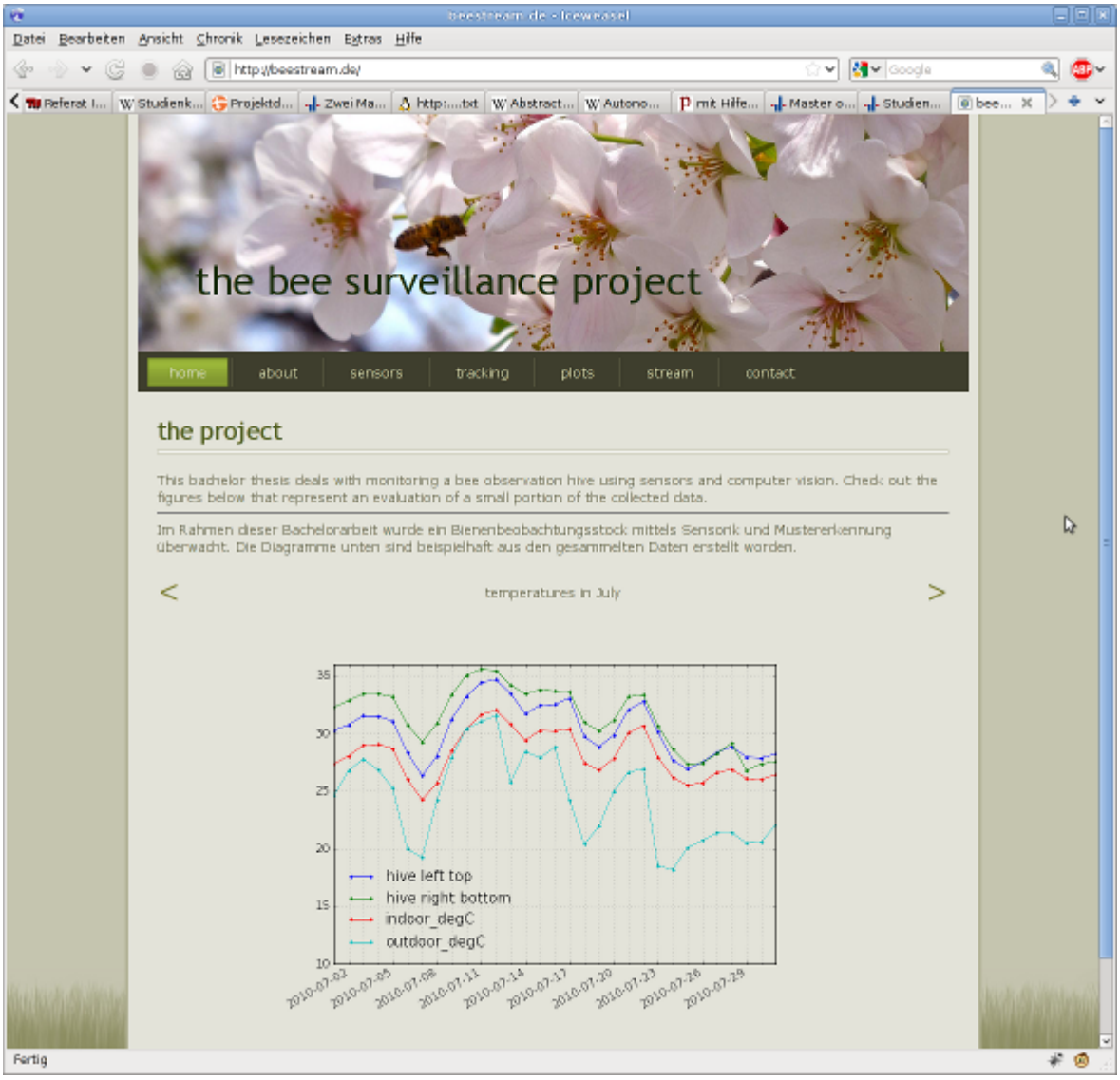


Figure 13: Homepage of the website dedicated to this thesis on http://beestream.de

4 Evaluation

The beehive monitoring system was operating from Mar 6th till Sep 20th 2010. The data it acquired can be divided into two parts: the sensor data that was read out once a minute (temperature within the hive, weight of the hive, and the weather) and the optical flow data that was captured with a frame rate of about 15 frames per second. These differences are reflected in terms of size. About 20 Gigabyte of optical flow data was stored on computer B while the database of the rest of the sensors only amounts to 30 Megabytes. The optical flow sensor thus acquired about 700 times as much data as the other sensors did.

Although all sensors have a reliable measure precision errors in measurement occurred due to changes in the set-up. The optical flow sensor data was affected by the change in lighting due to broken spotlights. The temperature sensors were placed in the combs and had to

be readjusted twice because they fell out. The scale's accuracy wasn't ensured after the beekeeper had to replace a comb frame and thus had to lift up the beehive off the scale and put in back on. The anemometer didn't provide very accurate data because it wasn't placed high up enough. However, accurate weather data is available from the institute of meteorology at Freie Universität Berlin[3] and can be used to back up the data. Considering the scale was put in a similar position afterwards, the weight inaccuracy of the scale measurement is also negligible.

When evaluating any kind of sensor data it has to be clear according to which criteria and conditions the data resulted. Otherwise, the evaluation is subject to ambiguity. Data doesn't say much on its own. It has to be put in relation to data that has been generated using the same (or similar) criteria as its own.

The evaluation techniques represent a first step. All collected data can be analyzed in many different and also more sophisticated ways.

4.1 Non-optical Sensor Data

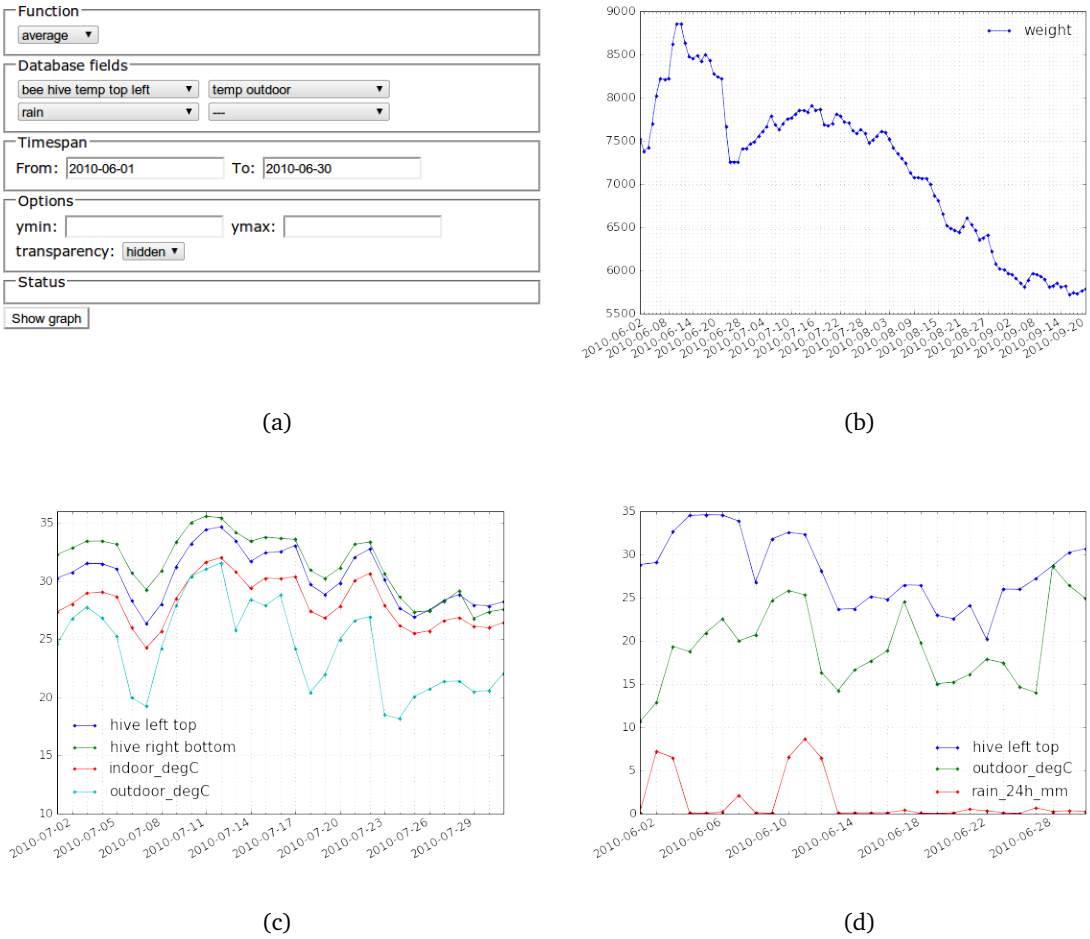
On the website dedicated to this thesis one can create plots of all sensor data collected during this experiment. The plots that can be seen in figure 14 were created with this plot generator. The plot generator was developed in the course of this thesis using Python and Matplotlib. It allows to create diagrams with up to four variables over arbitrary periods of time. For each day, the plotter generates one value per sensor variable, so that the output is a function. It's up to the user to decide whether this value will be the minimum, maximum, or average of a day.

4.2 Optical Flow Sensor Data

4.2.1 Assumptions and Limitations

In section 2.1.2 we discussed the limitations of the tracking algorithm. It uses the »Good Features to Track« algorithm [12] which provides the tracker with well-traceable features. The problem is that these features don't have to be bees. They can be anything: combs, frames, or the beehive. Earlier, we assumed that for the optical flow it doesn't matter because we can leave out features that don't move. In fact, for some questions on the resulting data it doesn't matter, but for some it does. It wouldn't be a problem to ask for motion data in general. But it would indeed be a problem to ask for the motion data of bees. We don't know whether a motion of zero is from a bee or from the comb frame. The underlying problem is the lack of normalization of optical flow data. The number of features is constant but the number of bees are unknown. Thus, the overall motion of a couple of bees moving heavily could be equivalent to many bees moving gradually. The heavy movement of a couple of

[3] http://www.met.fu-berlin.de/de/wetter/



(a) (b)

(c) (d)

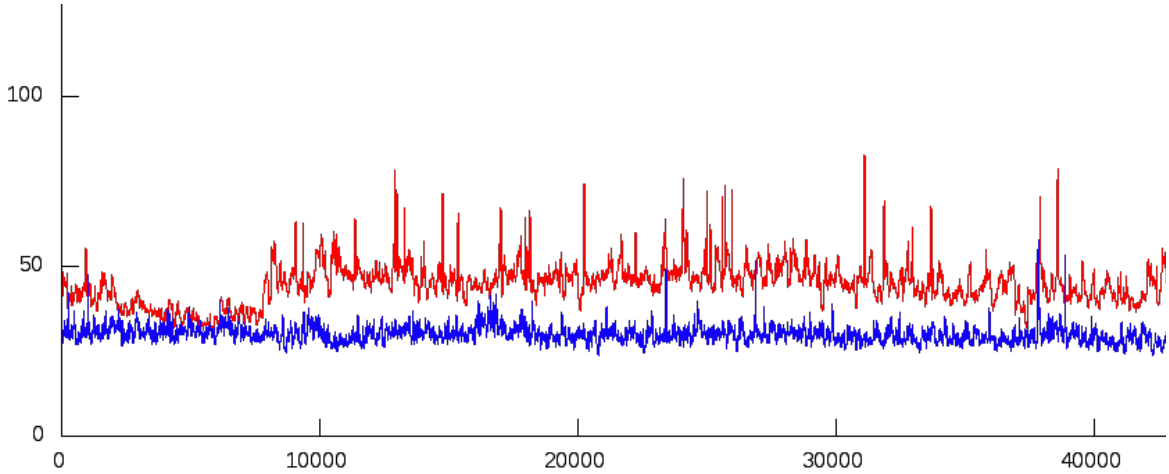
Figure 14: Web GUI for sensor evaluation (a), plot of total weight over time for the whole experiment (b), plot of temperatures over days in July (c), plot of temperatures and rain fall over days in June (d).

bees would be canceled out by very many features that don't move at all and thus making it equivalent to tracking many bees moving gradually. A solution to the problem would be to count the number of bees and reduce the features accordingly.

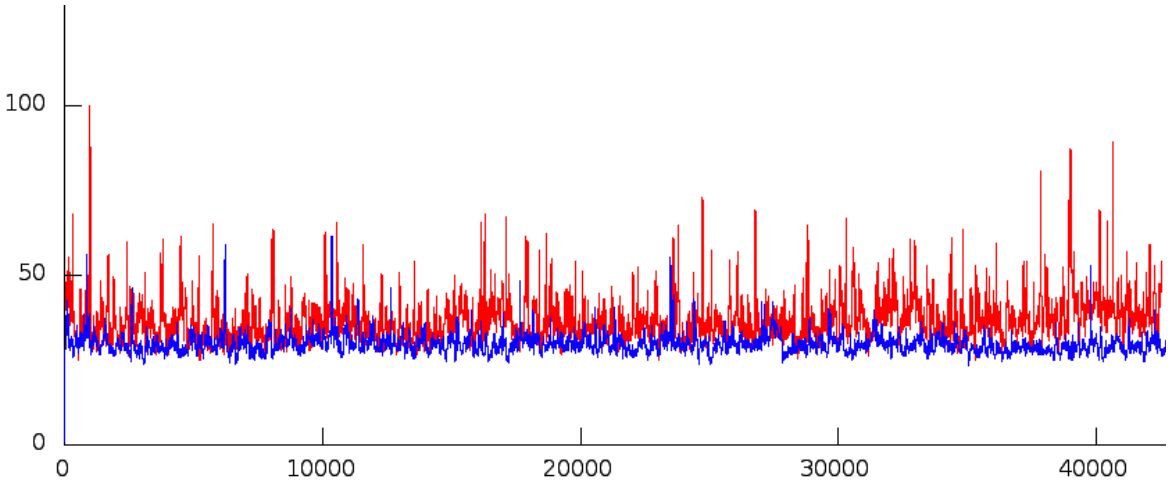
For the following evaluating of the optical flow data it's assumed all features are distributed equally. However, the »Good Features to Track« algorithm searches for high-contrast features but does not care about the position of these features. Therefore, features aren't distributed equally. We have to bear that in mind if we look at optical flow or activity diagrams.

4.2.2 Mean Vector as a Means to Compute Global Activity

For each feature the tracker writes down the position of the feature at a given time. The resulting data is a set of vectors. One approach to evaluate this data is to calculate the mean of all vectors at a given time. We calculate the mean vector by averaging the vectors at a



(a)



(b)

Figure 15: Activity on July 20th (red) and Aug 20th (blue) at 3 pm (a) and 3 am (b)

given time resulting in one vector, which we call the mean vector at a given time. We can then create a histogram with time as the x axis and activity as the y axis (see figure 15).

The two plots in figure 15 show different stages of the beehive development at two times of the day. While the beehive colony seems to be more active in July on both (a) and (b), the difference in activity is not much greater at night in (b). This complies with what could be experienced by looking directly at the beehive.

4.2.3 Mean Vector as a Means to Compute Local Activity

Using the same technique as the above, we can evaluate the optical flow data in a much more interesting way. What if we look at regions instead of the full honeycomb? Doing this we calculate the mean regional vector. We can do this by dividing the honeycomb in even parts. Figure 16 shows a 14x11 grid that is drawn over an image that was taken before the activity

was measured. It helps the observer to get an impression of the regions on the comb that were of interest for the bees at that day. The colors represent activity. The more dark and red the color are, the more activity was monitored.

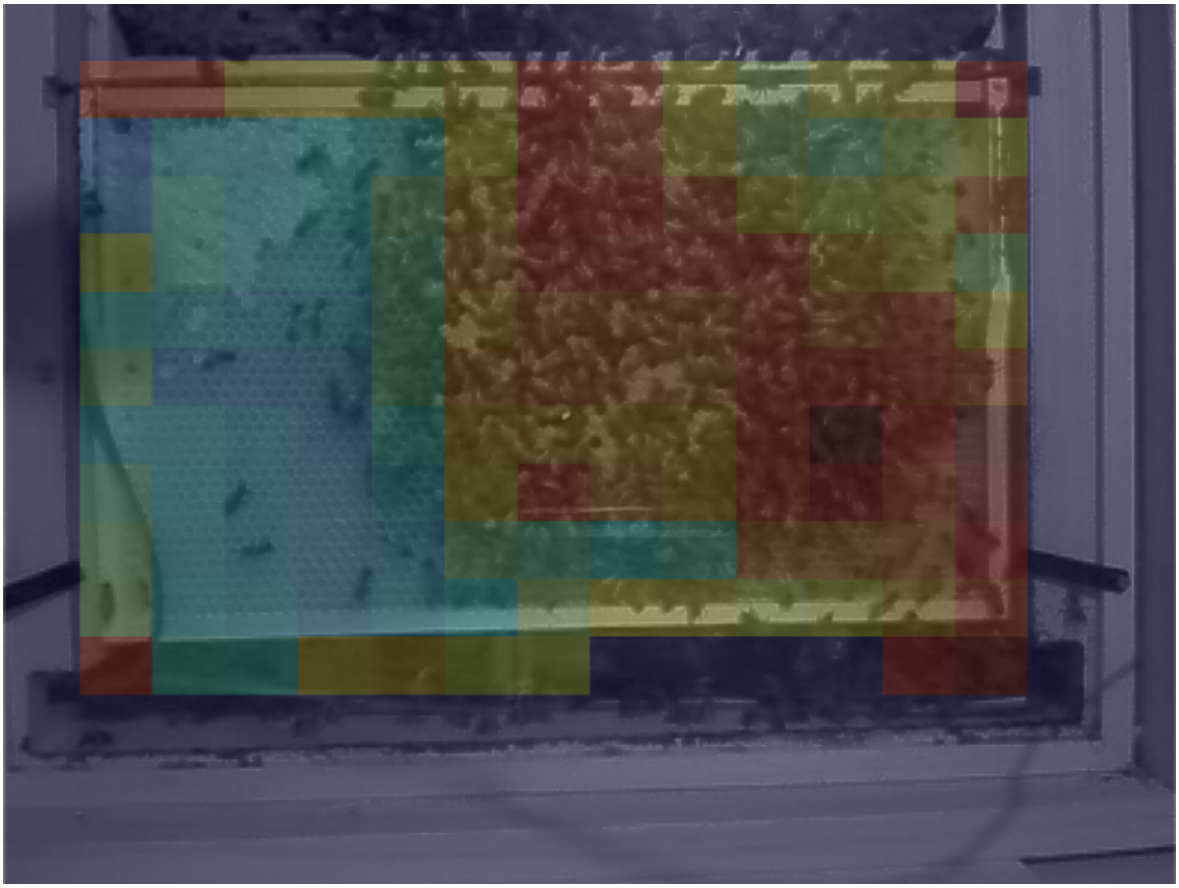


Figure 16: activity in regions of a honeycomb

Worth mentioning is the high activity path that goes from the bottom right (beehive entry/exit) to the top middle of the honeycomb. Clearly, it indicates a movement of the bees up left once they have entered the beehive (and also vice versa). Whereas the left part of the honeycomb indicates a low activity.

Although only a snapshot, these results can be verified by looking at the underlying picture. The low activity on the left matches the left part of the picture where hardly any bees, brood, or honey can be found. The high activity on the right matches the group of bees on the right.

5 Outlook

Losses of industrial honey bees were a known problem in the past. Even in 1869 high losses in bee colonies were observed [3]. However, losses in bee colonies have become much more of a problem since the winter 2006/2007 where they have reached an alarming extent [3]. »In the absence of a known cause, this syndrome was named Colony Collapse Disorder (CCD) [...]« [3]. In a nut shell, what's happening to an infected beehive is the sudden loss of adult worker bees. All the brood, the stored food, and the queen remain in the hive, with not much hope left. Scientists are unsure about what causes this behavior [3].

No doubt, it would be very desirable for beekeepers to receive a warning when there are signs of diseases. Though it's not an easy decision whether a beehive is healthy or not, this beehive monitoring system can provide the data foundation to generate such a warning. A warning system would generate a warning once one of the following values diverges:

- ▷ Temperature within the beehive
- ▷ Weight of the beehive
- ▷ Global activity of the beehive
- ▷ Activity in relation to the weather

Still, creating an algorithm to raise a warning in an intelligent way, is not a trivial task. One could dedicate a whole thesis to developing an artificial intelligence for raising a warning in the correct moment. Certainly, to find out more about how sensor values diverge one would have to generate data of both, healthy and infected beehives. The beehive monitoring system would serve the purpose of generating the data. Currently, the group of Prof. Menzel investigates indicators of diseases in beehives.

6 Conclusion

In the course of this thesis we built a beehive monitoring system. In the beginning, we determined what would be subject to monitoring. We understood what optical flow is and how it can be applied for beehive monitoring. According to the data we wanted to acquire, we chose the hardware. We found out how to communicate with the hardware. Then, we developed an architecture to read out the hardware, process the output, and then store it appropriately. Finally, we took a look at how to evaluate and present the data. We also presented an outlook of what could be done to further automate the system to help to inform the beekeepers about an outbreaking disease in a beehive.

The beehive monitoring system was in operation for five months and proved to be stable in operation. It's capable of supplying beekeepers or scientists with sensor data and a live broadcast. As of now, the system is not suitable for the consumer, though, remains a useful tool for scientists. The drawback to commercial use is not only its size but also the fact that it requires an observation beehive. In an every day environment one would use a regular beehive, eliminating the possibility to use cameras to measure the optical flow within the beehive. If mostly interested in activity of the bees, one could use acoustic information from a microphone as an alternative. Currently, this is being investigated by the team of Prof. Menzel at the institute of neurobiology at Freie Universität Berlin.

The evaluation showed that the technique of capturing the optical flow within a beehive proved to be a new source of information which might be of interest for scientists in the future. We used the data acquired to generate activity diagrams. On such we discovered a path on a comb. Although more analysis would be needed to prove this discovery to be a biological fact, the possibility for scientific evaluation became clear.

One of the downsides of the optical flow data is the lack of normalization. It limits the comparability of two sets of data. However, to normalize the data one would have to estimate the number of bees in a beehive which is not a trivial task. Another downside is that features are chosen with respect to their traceability. They should be chosen also with respect to equal distribution on the comb area to quantify the optical flow data correctly.

7 References

[1] D.A. Levin. The origin of reproductive isolating mechanisms in flowering plants. *Taxon*, 20(1):91–113, 1971.

[2] T. Stankus. A review and bibliography of the literature of honey bee Colony Collapse Disorder: a poorly understood epidemic that clearly threatens the successful pollination of billions of dollars of crops in America. *Journal of Agricultural & Food Information*, 9(2):115–143, 2008.

[3] D. VanEngelsdorp, J.D. Evans, C. Saegerman, C. Mullin, E. Haubruge, B.K. Nguyen, M. Frazier, J. Frazier, D. Cox-Foster, Y. Chen, et al. Colony collapse disorder: a descriptive study. *PLoS One*, 4(8):e6481, 2009.

[4] James Gould. *The honey bee*. Scientific American Library ;Distributed by W.H. Freeman, New York, 1995.

[5] BeeWise Monitoring System. `http://www.beewise.eu`. last accessed Oct 12th. 2010.

[6] Beekeepers dream - PIC32 based bee hive monitor. `http://www.embedds.com/beekeepers-dream-pic32-based-bee-hive-monitor/`. last accessed Oct 12th. 2010.

[7] Automatic weight-trasnmitting bee hive scale. `http://www.bienenwaage.de/html/behive_scales.html`. last accessed Oct 12th. 2010.

[8] Honey bee monitoring system for monitoring bee colonies in a hive. `http://www.freepatentsonline.com/y2004/0077290.html`. last accessed Oct 12th. 2010.

[9] JL Barron and NA Thacker. Tutorial: Computing 2d and 3d optical flow. *Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester*, 2005.

[10] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[11] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International joint conference on artificial intelligence*, volume 3, pages 674–679. Citeseer, 1981.

[12] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 2002.

[13] L. L. Langstroth. *Langstroth's Hive and the Honey-Bee: the Classic Beekeeper's Manual*. Courier Dover Publications, 2004.

[14] E. Wolf and G. Zerrahn-Wolf. The effect of light intensity, area, and flicker frequency on the visual reactions of the honey bee. *The Journal of General Physiology*, 18(6):853, 1935.

[15] K. Von Frisch. *Der Farbensinn und Formensinn der Biene*. Fischer, 1914.

[16] L. Chittka and NM Waser. Why red flowers are not invisible to bees. *Israel Journal of Plant Sciences*, 45(2):169–183, 1997.

[17] Richard Lippmann. Temperaturmessung mit Linux. `http://lena.franken.de/hardware/temperaturmessung.html`. last accessed Oct 12th. 2010.

[18] Brian Lane. Digitemp. `http://digitemp.com`. last accessed Oct 12th. 2010.

[19] Open CV. `http://opencv.willowgarage.com/wiki/`. last accessed Oct 12th. 2010.

[20] JQuery. `http://jquery.com`. last accessed Oct 12th. 2010.

8 List of Figures