



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE



EnVision - a Vision System for Envirobot

Hardware Aspects

Spring Semester 2018

Author: Elias Klauser

Adressee: Prof. Auke Ijspeert

Advisors: Behzad Bayat, Simon Hauser, Mehmet Mutlu

Contents

1	Introduction	2
1.1	Foregoing studies	2
1.2	Analysis of simulation	2
1.3	Project goal	4
2	Hardware	4
2.1	Camera	4
2.2	Lens	4
2.3	Mechanical parts	6
2.4	Assembly of head module	8
3	Image Processing	10
3.1	OpenCv	10
3.2	Calibration and undistortion	10
3.3	Color recognition	11
4	Future work	14
5	Conclusion	14

Abstract

EnVision is a computer vision system based on a spiking neural network (SNN) for the Envirobot. The Envirobot was developed at the Biorobotics Laboratory at EPFL Lausanne [1]. It is a snake-like swimming robot with an anguilliform type of locomotion. This project was inspired by a study of the Department of Computational Biology at KTH Royal Institute of Technology in Stockholm [2] which aims to establish a computational model of the behavior based on visual stimuli of a lamprey. Three basic behaviors escape, avoidance and approach were considered in this work. For each of those behaviors, one specific color is used as visual stimuli which triggers a certain behavior when this color is detected. The work to carry out by the Biorobotics Laboratory was firstly the development of a webots based simulation in order to observe the behavior of the lamprey following this computational model [3]. The next goal is the implementation of this simulation on the hardware Envirobot. The present paper describes the hardware aspects which are necessary to enable the integration of EnVision in the Envirobot.

1 Introduction

1.1 Foregoing studies

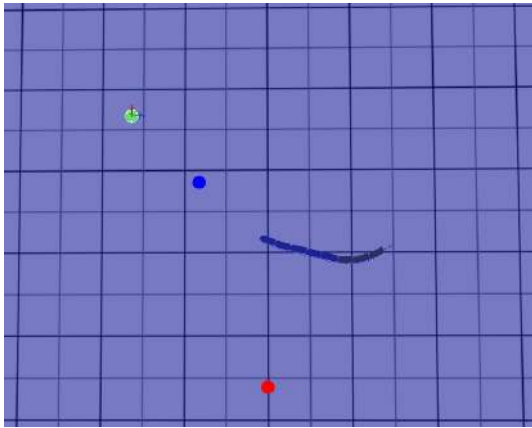
The study carried out by KTH [2] is searching to simulate the visual behavior of a lamprey. A visuomotor center is represented by a spiking neural network (SNN) which receives input signals from a simplified retina as sensor. The study is considering three basic behaviors present in a certain form in all vertebrates which are escape, avoidance and approach. Visual stimuli are used to trigger these behavior patterns. The visual stimuli are all represented by a certain color code. Predators are simulated by red color, obstacles by blue color and prey by green color. The behavior initiated by a simulated predator is an immediate turn to the opposite direction of the detected predator which is followed by a rapid forward movement. This characterizes the escape behavior. The behavior for avoidance is a slight adjustment in the travel direction in order to avoid an obstacle. Prey is approached by an immediate turn to the direction of the detected prey combined by a forward movement. These three visual stimuli are simulated in a three-dimensional space. In an initial stage, all visual stimuli are supposed to be in a two-dimensional space for simplification and due to the fact that Envirobot is not capable of diving.

A simulation of this model was implemented by a project student at the Biorobotics Laboratory [3]. It was implemented with the Webots robot simulator which is a professional software used for educational and research purposes [4]. This software allows to simulate the water environment and its physical properties as well as the hardware of Envirobot.

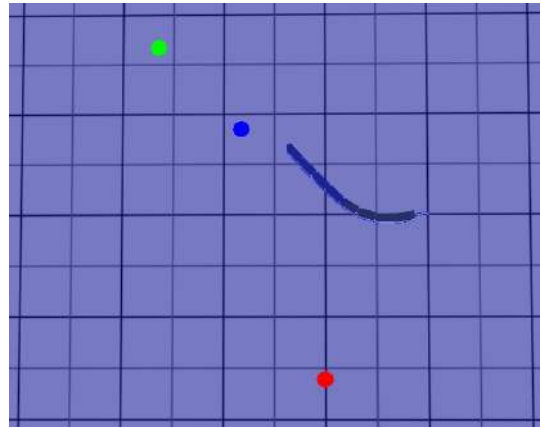
1.2 Analysis of simulation

In order to get a better understanding of the desired final behavior of Envirobot with the Envision system, different situations were simulated and analyzed with the Webots simulation. The situations differed in terms of number and position of stimuli.

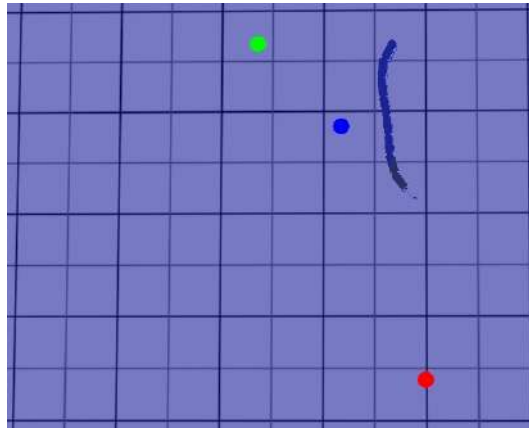
A specific aspect which was analyzed is the detection and behavior of the robot to stimuli which are located close to each other. On Figure 1a, the position setting of the stimuli can be seen with the robot close to its initial position. An initial turn of the robot away from the predator stimulus is already initialized. If there was no predator on the left side of the robot, it would not turn to the right as the prey can visually not be detected behind the obstacle. Figure 1b shows that the obstacle is correctly avoided. The prey is still not detected though, the turn is still mainly due to the predator. Once the obstacle is passed, the head module of the robot is already turned away from the prey due to the escape behavior (See Figure 1c). Hence, the prey is still not detected and the direction opposite of the predator is kept. It would be interesting to set up a similar test situation for the real Envirobot once EnVision is fully implemented. As the resolution of the simulated retina is limited to 64 receptive fields, the maximal detection range especially for stimuli hidden behind others could be limited (see also section 3.3).



(a) Initial turning to prey, mainly due to avoidance of predator



(b) Obstacle avoidance



(c) Prey can not be detected due to orientation of head module

Figure 1: Testing of behavior when two stimuli are close together

A similar case can be seen on Figure 2. Here the distance between obstacle and prey stimulus has been elongated. The behavior of the robot until passing the obstacle is quite similar. In this case, the robot is able to detect and reach the prey even if the prey is further away. It can be detected as the angle between the camera axis and the prey stimulus is smaller compared to the situation of Figure 1.

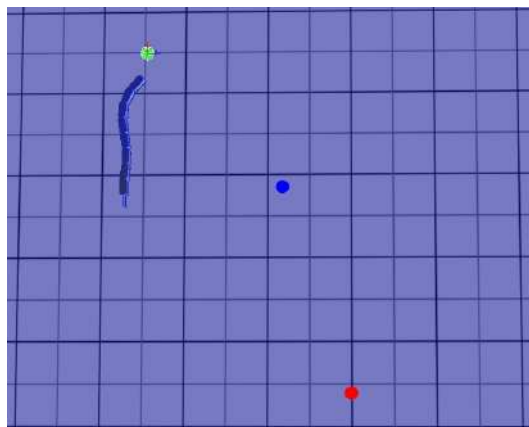


Figure 2: Similar setup with increased distance between prey and obstacle

Further tests were carried out to study the influence of the decay rate of the SNN. Several test settings were analyzed both with a decay rate of 0.00 and 0.05 (for more detailed information see [3]). A change in decay rate doesn't influence overall results in terms of behavior and travel path. It seems as if the anguilliform motion of the simulated Envirobot is more regular with a zero decay rate. Nevertheless, it is possible that this is due to the limited performance of the computer used for the simulation. It should apparently be retested whether the same observation can be made on the hardware Envirobot as well.

1.3 Project goal

The goal of this work is to create a basis in terms of hardware and image processing to enable an autonomous locomotion by computer vision on Envirobot. The hardware aspects cover a new bio-inspired head design which allows camera integration, integration of a NanoPi embedded computer and a IMU unit. For logistical reasons, all the cameras have to be removable and therefore they can hardly be encapsulated to protect them from water leakage. Therefore, absolute watertightness of the head module is a design-critical element. As far as image processing is concerned, the main aspects are camera calibration, image undistortion and color correction. A color detection program will be implemented and tested in order to visually detect the stimuli.

2 Hardware

2.1 Camera

The first tests with the camera which was initially intended to be used for EnVision were not satisfying. The pictures taken with this camera were not equilibrated in terms of color. Sometimes they appeared to be too greenish or blueish which made the color recognition quite difficult. This camera didn't allow any color adjustments. This is why it was decided to use another camera for this project. The decision was taken for the xiMU camera (MU9PC-MH) produced by the German company Ximea [5]. This miniature camera allows quite sophisticated color adjustments which is very beneficial for a color recognition application.



Figure 3: The xiMU camera produced by the company Ximea is as small as a coin. [5]

2.2 Lens

The fisheye lens which was originally intended to be used for this project has quite a large field of view (122°). On the other hand, it is quite limited in terms of detector size ($1/4''$). Therefore, the image had to be cropped in order to obtain only the central part of the image and to delete the black zones (see Figure 5a). This is not optimal as quite a lot of the image resolution is lost. An additional aspect is the fact that fisheye lenses add in general quite a lot of distortion. Several available camera lenses were tested in order to compare their fields of view (see Table 1).

In the study of KTH [2], a total field of view of 360° (180° per eye) is used for the model. In order to get as close as possible to this theoretical value, the decision was taken for the fisheye lens with a focal length of 1.56 mm. A image of this lens mounted on the xiMU camera can be seen on Figure 4. The measured field of view of this lens is 172° (see Figure 5b). The tradeoff of this choice is the higher distortion which is typical for fisheye lenses. Hence, the images taken with this lens will have to be undistorted. For comparison, Figure 5c shows an image taken with a standard lens with a focal length of 2.8 mm. The distortion of the image is considerably smaller compared with the fisheye lenses. However, the field of view of this lens is smaller too.

Lens type	Focal length [mm]	Sensor size ["]	Horizontal field of view [°]	Remarks
Fisheye	1.05	1/4	122	Originally in use
Fisheye	1.56	1/2.5	172	Selected for EnVision
Standard	2.8	1/2.5	85	
Standard	3.6	1/2.5	74	
Standard	6	1/2.5	42	
Standard	8	1/2.5	34	

Table 1: Angles of view for different camera lenses



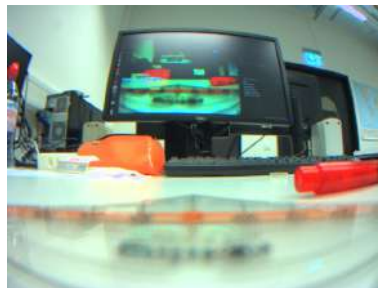
Figure 4: Image of the xiMU camera with the 1.56 mm fisheye lens



(a) Fisheye lens with a focal length of 1.05 mm which was originally intended to be used for EnVision



(b) Fisheye lens with a focal length of 1.56 mm which will be used for EnVision



(c) Standard lens with a focal length of 2.8 mm

Figure 5: Measurement of field of view for three different type of lenses

2.3 Mechanical parts

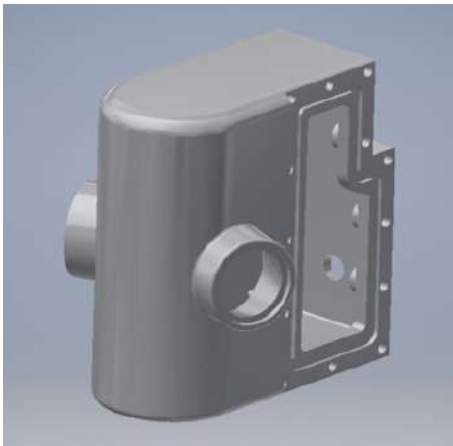
The Envirobot platform is a segmented swimming robot based on the previous robot models developed by Biorobotics laboratory such as Amphibot.[1] Its shape and locomotion are similar to a snake. Thanks to its modular construction, the length and configuration of the robot can be changed with respect to the desired application. The main objective of Envirobot is the monitoring of aquatic environment. Figure 6 shows the current development state of the Envirobot.



Figure 6: Envirobot with the current head module [6]

The head unit currently in use houses a RF antenna for GPS signaling, low-level controller for the anguilliform locomotion and a CAN bus communication module. For the integration of the EnVision system, a new hardware design of the head unit has to be developed. The new head unit accommodates three cameras, a NanoPi embedded computer and a IMU unit. Figure 7a shows the design of the modified head unit. For EnVision, only the two cameras in front of the head (two circular openings visible on Figure 7a) will be used. The third camera is located on the bottom of the head and is used for a visual localization project.

In order to protect the cameras from damage by the water environment, glass domes will be glued on top of the camera openings in the head unit. To facilitate camera mounting, there is a large opening in the back part of the head which can be closed by an aluminum lid. Figure 7b shows the assembly of the head unit with the three domes and the lid.



(a) CAD model of the new head unit for EnVision



(b) CAD model of the assembled head unit including eye domes and lid

Figure 7: Head module before and after assembly

The lid is mounted with a large number of screws (13 screws in total). The head module, which is 3D printed, contains holes around the lid opening which can house thread inserts. These inserts have to be glued inside the holes to make sure that they cannot be pulled out by the screws while mounting the lid. On top of that, a rubber band is glued around the lid opening on the head

unit for the purpose of water impermeability. It also houses the NanoPi which can be mounted with screws on the lid part inside the head unit. Figure 8 shows the whole lid with the inner part dedicated to NanoPi housing. Due to its thermal conduction properties, aluminum has been chosen as material for the lid which will act as a water cooling for the NanoPi. The outer part of the lid will necessarily be in contact with the water environment and therefore be cooled down by thermal conduction.

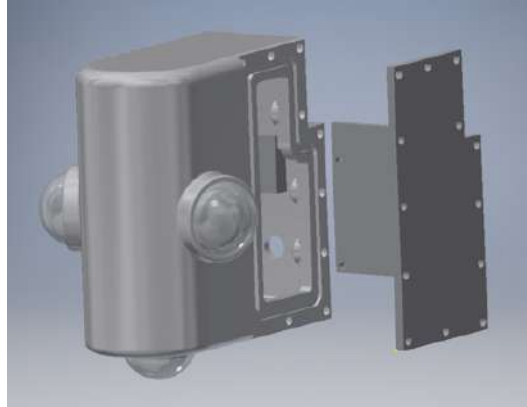
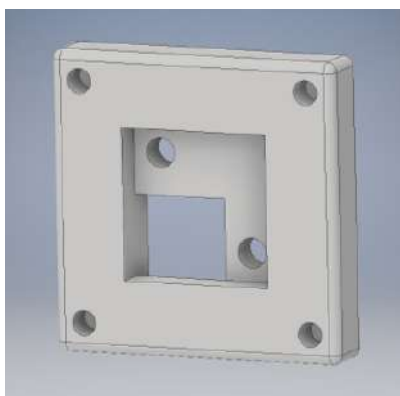
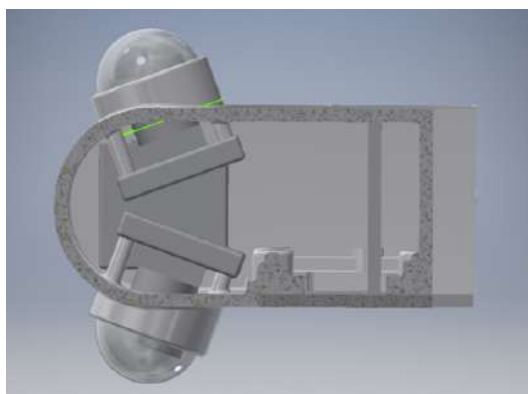


Figure 8: CAD model of the aluminum lid while assembling

One of the mechanical requirements is the removability of the three cameras inside the head module. This is mainly due to the fact that the tests will be carried out with only two Ximea cameras. To fulfill this requirement, a camera holder plate was designed. Figure 9a shows the CAD model of this plate. The Ximea camera can be mounted inside a hole with the precise dimensions of the camera casing and is fixed with screws. The camera holder itself can be assembled inside the camera on top of poles which are integrated in the head module. Figure 9b shows the arrangement of the three cameras inside the cameras when all pieces are assembled. The head was designed such that the camera axis makes an angle of 20° with the back surface of the head module. Thus, the cameras are oriented slightly towards the front which introduces a zone of binocular vision in front of the robot. The risk of missing a stimulus directly in front can be strongly reduced thanks to this arrangement. Furthermore, stereovision could this way be analyzed within the scope of a robot-inspired biological study.



(a) CAD model of the camera holder for the Ximea camera



(b) Half section view of the inner part of the head unit with the three cameras mounted

Figure 9: Visualization of camera mounting system

A critical technical point is the exposedness of the glass domes. When the Envirobot collides with an object as for example the edge of the swimming pool, the glass domes are likely to be damaged. A possible solution for this issue is a mechanical protection with four poles around the camera openings on the head module (see Figure 10). These poles would be relatively rigid as they are part of the 3D printed head module. First tests with the assembled model showed that the glass domes resist quite well to mechanical shocks such that this solution remains only a concept and has not been realized so far.

Another possible improvement to prevent the cameras from damage after water leakage would be to accommodate them in closed watertight chambers. Such a system would make it much more difficult to remove the cameras from the head module.

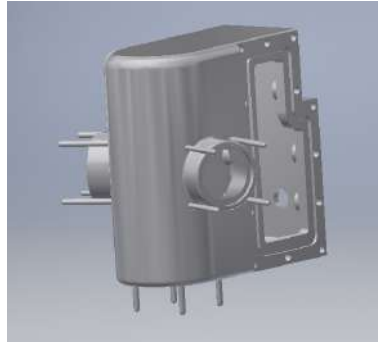


Figure 10: CAD model of the head module with added poles for protection of glass domes

2.4 Assembly of head module

The 3D printing of the head module was done externally with a laser stereolithography technique (SLA) [7]. This method works with a photopolymer resin which changes its properties when exposed to light. The laser system is used to photochemically solidify parts of the resin in order to obtain a 3D printed version of a CAD model. This technique was used mainly because the printed model is watertight which is important for protection of the cameras and electronics. A standard 3D printing procedure which uses ABS filaments does not result in a completely watertight model. The 3D printed head module after assembly of the glass domes, thread inserts and rubber tube can be seen on Figure 11a. The glass domes were glued with two component epoxy resin adhesive. Afterwards, a silicon sealant was applied to guarantee waterproofness of the adhesive area. For fixation of the rubber tube, an elastic silicon sealant was used.

After 3D printing, the pieces had to be manually machined in order to smoothen the surfaces and to make the poles fit into the holes. This step is necessary as the precision of the 3D printed model is not as high as the precision of a machined piece. This procedure can take some time as the assembly should be done with some tensioning. A comparison of the old with the new head module is shown on Figure 11b.

A second module, which is the piece connecting the head module with the main part of Envirobot, was 3D printed with the same technique. This module is used for housing the control electronics for locomotion, communication and power supply of the Envirobot. It can be attached to the head module with a hollow circular connecting piece. This piece is glued on the head module and acts as the only mechanical connection between head and control module. Some data and power cables will be guided inside the connecting piece. The two front modules assembled can be seen on Figure 12a.

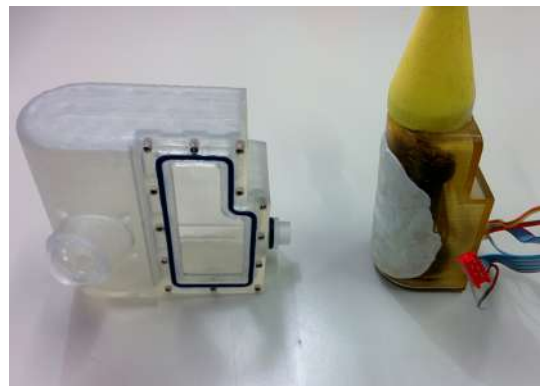
Figure 12b shows the outer piece of the aluminum lid. Initially, the inner piece for NanoPi housing was supposed to be glued on the outer part. This solution was abandoned as the glue layer is not a good thermal conductor. Hence, welding is considered as assembly solution even if welding could

be quite difficult. Due to the small thickness of the plate, it risks to bend during the welding process.

Once the assembly was finished, waterproofness of the glass domes was tested in a small water basin and some leakage was detected. Hence, another layer of silicon sealant was applied and then again iteratively tested. It can be quite complex to evaluate the exact position of leakage. Therefore, all the sealant layers were reapplied.



(a) Head module after assembly of thread inserts, rubber tube and glass domes



(b) Comparison of new head module (left side) with the one currently in use on Envirobot (right side)

Figure 11: Images of the 3D printed head module



(a)



(b)

Figure 12: Head module with attached module for control electronics (a) and outer piece of the aluminum lid (b)

3 Image Processing

3.1 OpenCv

OpenCV (Open Source Computer Vision Library) is an image processing library [8]. It is written in C/C++ and provides a large amount of functions concerning both acquisition and modification of the image. Some of the more sophisticated algorithms are implemented as well. At the beginning of this project, all the libraries such as for example OpenCv had to be installed to be able to run the codes which have already been coded by Roger Fong [3]. Due to some compatibility issues with Linux Ubuntu 16.04, we were finally not able to install some of the necessary library and therefore we are not able to run parts of the original code. Hence, the decision was taken not to use the concerned libraries for the further work and to use other compatible tools instead.

3.2 Calibration and undistortion

A fisheye lens adds a considerable amount of distortion to an image (see also Section 2.2). Hence, it is necessary to undistort all images taken with fisheye lenses. The combination of the camera in use with a specific fisheye lens has to be calibrated. The calibration of the camera lens is performed by taking pictures of a checkerboard at different angles and positions with respect to the camera (see Figure 13). Around 30 different images should be used for calibration to get a satisfying result.



Figure 13: Sample image for calibration with the checkerboard

There are standard OpenCV codes for camera calibration. Specific calibration functions for fish-eye lenses exist. To determine the set of calibration parameters, a open source python code (*calibration.py*) was used [9]. This calibration code determines the values for two undistortion matrices commonly called K and D. K is the intrinsic camera matrix with dimensions 3x3. It contains the focal length along x and y-axes (f_x , f_y) and coordinates of the principle point (c_x , c_y) [10]. The structure of the matrix K is defined as follows:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

D is the a four dimensional input vector of distortion coefficients.

$$D = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{bmatrix}$$

The matrix K and vector D are used for undistortion of all images taken with this specific combination of camera and fisheye lens. Figure 14a shows a sample image with a considerable amount of distortion. On Figure 14b, one can see the undistorted version using the values of K and D evaluated by calibration.

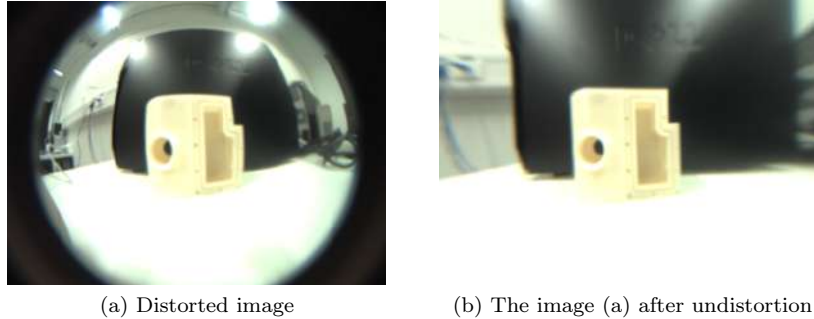


Figure 14: Undistortion of a image taken with a fisheye lens

3.3 Color recognition

As a first task, a code was written to detect circles in the colors red, green and blue printed on a paper sheet. The code first starts a video stream with HSV (hue, saturation, value) color representation. A mouse callback loop was implemented which prints out the corresponding HSV values of the pixel which is selected by a mouse click. For each color, the values were determined for several, different positions and angles including different light situations as the color code for one color-wise homogeneous object can be different according to the specific setting. This allowed to establish a certain range for the hue, saturation and pixel value for each color (mainly hue varies). In a second step, a filter was implemented which lets pass pixels only when all of their HSV values are inside this range.

In OpenCV, the value ranges for HSV representation are defined as follows: $H = [0, 179]$, $S = [0, 255]$, $V = [0, 255]$. The determined color range for the colored printed circles are shown in Table2:

Color	Hue (H)	Saturation (S)	Value (V)
red	$[0, 10] \cup [175, 179]$	$[100, 255]$	$[60, 255]$
green	$[55, 65]$	$[170, 190]$	$[60, 110]$
blue	$[100, 110]$	$[170, 230]$	$[60, 110]$

Table 2: HSV value ranges for the color detection of the printed circles

In the beginning, for all colors the range $[100, 255]$ was used for the S value and the range $[60, 255]$ for the V value. In the file *RGB_detection.cpp*, these values are represented by the variables $s_{min} = 100$, $s_{max} = 255$, $v_{min} = 60$ and $v_{max} = 255$. During the first tests with these values, the result was not satisfying especially for the green and blue color. Often, there was a lot of noise present in the filtered image as there were too many objects which are color-wise close to the color of the circles. Hence, the range for the S and V values was iteratively modified until a better result was obtained.

The next step is to analyze the horizontal axis (x -axis) of the image only. If at any vertical position (y -axis) of the image the searched color is detected, it is printed out in a 64 pixel one dimensional image. There are 64 pixels as the locomotion model for EnVision will use as input a simulated retina with 64 receptive fields for its neuronal network. With the values mentioned above, it is depending on the test setting still hard to reliably detect the circles. Figure 15 shows a situation where the green circles are hardly detected. The blue circles cannot be detected at all.

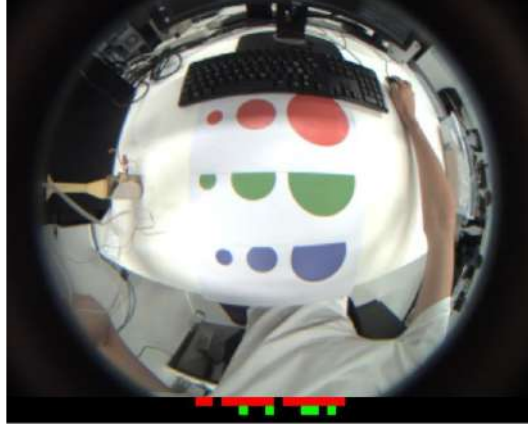


Figure 15: Sample image for color detection with printed circles on top and the corresponding one dimensional RGB color analysis at the bottom

For the final experiments in water environment, waterproof LED lamps will be used which can produce light in different colors (see Figure 16). The HSV value ranges were also determined for the detection of those lamps and are listed in Table 3:

Color	Hue (H)	Saturation (S)	Value (V)
red	[5, 15]	[160, 220]	[200, 255]
green	[65, 75]	[170, 255]	[230, 255]
blue	[95, 105]	[250, 255]	[250, 255]

Table 3: HSV value ranges for the color detection of the LED lamps



Figure 16: Waterproof LED lamps which can produce light in RGB colors

Figure 17 shows a test situation for the detection of the LED lamps. This result is a lot better compared to the detection of the printed circles. Nevertheless, depending on light conditions and environment of the lamps, there is still some noise which cannot be eliminated. For underwater testing, a new color calibration will be necessary either way. Therefore, no further effort was put into noise reduction of the color detection in the atmospheric environment.

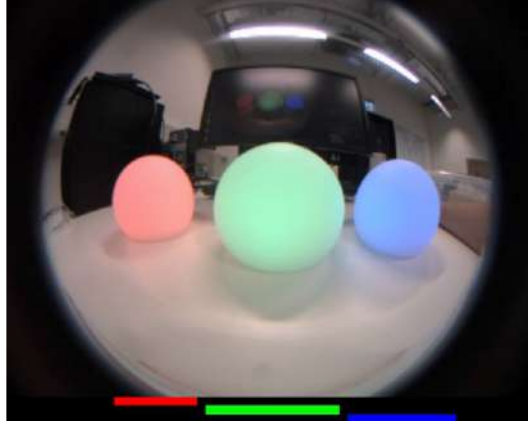


Figure 17: Sample image for color detection with LED lamps on top and the corresponding one dimensional RGB color analysis at the bottom

Some additional parameters can improve the quality of color detection such as exposure time, gain and color correction matrix (see below). These parameters were adjusted by hand with the Ximea CamTool [12] and fine-tuned by observing visually the changes in the image. Finally, a gain equal to zero and a exposure time of 0.1 seconds led to the best result. Obviously, these parameters have to be readjusted according to the test environment.

The color correction matrix allows to adjust the appearance of images in terms of color. The color correction matrix converts the colors of the actual captured image to a color-wise corrected image. It is a 4 by 4 matrix which can be multiplied with a RGBA vector. A (alpha) is a value between 0 and 1 and defines the transparency. The color correction matrix M is defined as follows:

$$\begin{bmatrix} R_{corrected} \\ G_{corrected} \\ B_{corrected} \\ A_{corrected} \end{bmatrix} = \begin{bmatrix} M_{00} & M_{01} & M_{02} & M_{03} \\ M_{10} & M_{11} & M_{12} & M_{13} \\ M_{20} & M_{21} & M_{22} & M_{23} \\ M_{30} & M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} R_{original} \\ G_{original} \\ B_{original} \\ A_{original} \end{bmatrix}$$

The default values for M are:

$$M_{default} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{bmatrix}$$

The values of M used for the color detection in EnVision were experimentally determined. Following values are applied:

$$M_{EnVision} = \begin{bmatrix} 2.0 & -0.5 & -0.5 & 0.0 \\ -0.5 & 2.0 & -0.5 & 0.0 \\ -0.5 & -0.5 & 2.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{bmatrix}$$

4 Future work

The purpose of the current project is to develop all the hardware elements for EnVision in order to enable testing in water environment. For the experiments in water environment, LED lamps will be used which are able to produce light of RGB color spectrum (see Figure 16). Recalibration of the camera has to be done underwater. Outdoor testing as for instance tests in the lake can be quite hard to handle. The water quality, light conditions and objects in the environment of Envirobot would require a quite sophisticated software which allows real time calibration in order to be able to achieve a satisfying color recognition.

Once the color detection system works reliably in water environment, the closed-loop behavior shall be implemented on Envirobot. The webots simulation [3] might be helpful again for this task. It enables to get an idea of the desired behavior of Envirobot.

EnVision could be perfectly used for demo purposes of Envirobot as the EnVision system allows a completely autonomous locomotion without the necessity of remote controlling. Furthermore, EnVision could be adapted for the use on other robots for instance to study vision-based models on robots capable to dive.

5 Conclusion

This semester project enabled to cover many different fields such as mechanical conception, hardware assembly and testing, programming of embedded software, image processing and optics. Throughout the project, several tasks related to different technical disciplines had to be covered in parallel. Therefore, a good organization was mandatory in order to keep track of the upcoming work to be done.

I liked the fact that the outcome of this project was quite unknown in the beginning as it is the case with most research projects. The work actually carried out was much more related to mechanical conception than expected. This is due to the interdependency of all the technical aspects in such a project. As the camera type used for EnVision had to be changed, all the mechanical design had to be adapted accordingly.

During the project work, I used software like Webots and Autodesk Inventor which I didn't know before. Even if some additional time was needed to get a understanding of the software, it was a good opportunity which allows me to realize more complex tasks with these programs in the scope of another project.

It took quite a lot of time to install software and libraries on the student computer. The encountered issues were often related to missing user access rights or to compatibility issues with the Linux distribution Ubuntu 16.04. Nevertheless, I was able to gain usefull skills in terms of installation of software and libraries on Linux-based machines.

I appreciated the work with OpenCV as it is a very complete library for computer vision and image processing. The image processing tools I worked with before this project don't allow a simple implementation of complex algorithms which is possible with OpenCV. I will therefore certainly work with OpenCV again in the future.

An interesting task was the assembly of the 3D printed parts. As the precision of fabrication was not very high, the pieces had to be machined manually. Furthermore, a new aspect was the waterproofing of all the assembled pieces. It is a surprisingly difficult and critical step as already a very narrow opening can be enough to let a considerable amount of water enter the protected area. Hence, the assembly of the head module took a lot more time than originally expected.

When it comes to organizational aspects, I found the weekly meetings with my main supervisor Mehmet Mutlu very beneficial and important. A frequent discussion helps to agree on decisions to take and avoids to get stuck during the project work. Appearing issues could often be resolved quite quickly with the help of my supervisors. In terms of things to improve, a detailed weekly report would have facilitated the writing of the final report. A list of weekly tasks with keywords was the only document which was constantly updated during the work process.

Acknowledgements

I would like to thank my advisors Behzad Bayat, Simon Hauser and Mehmet Mutlu for their support during this project. A special thanks also to François Longchamp for the fabrication of the aluminum lid and the help with assembly of the head module.

References

- [1] Behzad Bayat, Alessandro Crespi, Auke Ijspeert
Envirobot: A Bio-Inspired Environmental Monitoring Platform, Biorobotics Laboratory (BioRob), 2016
- [2] Iman Kamali Sarvestani, Alexander Kozlov, Nalin Harischandra, Sten Grillner, Örjan Ekeberg
A computational model of visually guided locomotion in lamprey, Department of Computational Biology, School of Computer Science and Communication, KTH Royal Institute of Technology, 2012
- [3] Roger Fong
EnVision, a Vision System for Envirobot, Semester Project, Biorobotics Laboratory (BioRob), 2018
- [4] *Webots robot simulator by cyberbotics*, Information, tutorials and download
<https://www.cyberbotics.com/>,
Accessed: 04.06.2018
- [5] *Ximea xiMU camera*, Product information and manual,
<https://www.ximea.com/en/products/subminiature-cameras>,
Accessed: 03.06.2018
- [6] Alisa Alering
Article "Envirobot cleans up", 2017
<https://sciencenode.org/feature/envirobot-cleans-up.php>
Accessed: 05.06.2018
- [7] *3D-printer Form 2 by formlabs*, description and specification
<https://formlabs.com/3d-printers/form-2/>
Accessed: 06.06.2018
- [8] *OpenCV (Open Source Computer Vision Library)*, Information, documentation and downloads,
<https://opencv.org/>,
Accessed: 03.06.2018
- [9] Kenneth Jiang
Tutorial for calibration with fisheye lens, Python code, 2017,
<https://medium.com/@kennethjiang/calibrate-fisheye-lens-using-opencv-333b05afa0b0>,
Accessed: 03.06.2018
- [10] *Documentation of fisheye camera model by OpenCV*,
https://docs.opencv.org/3.4.1/db/d58/group__calib3d__fisheye.html,
Accessed: 05.06.2018
- [11] *XiApi Manual*, Description of color adjustment matrix
https://www.ximea.com/support/wiki/apis/XiAPI_Manual
Accessed: 07.06.2018
- [12] *Ximea CamTool*, Description and download,
https://www.ximea.com/support/wiki/allprod/XIMEA_CamTool,
Accessed: 03.06.2018