

Reducing dizziness when using a video-see-through head-mounted display

Segovia Barreales, Richard

1 INTRODUCTION

THIS project has its origins in the internship that I did at the Computer Vision Center (CVC) during the summer of 2017. During my internship I developed a basic video-see-through viewer for a head mounted display (HMD) developed in the CVC. Along with the development we discovered that the main concern that our prototypes faced during our user testing sessions was the feeling of dizziness and motion sickness produced by the headset. Hence the motivation for this project is reduce the adverse physical reactions that it produces to the users [22], [1] and continue improving the viewer adding new modules while keeping its expandability. It has to be said that this research project is focused mainly on the software, the prototypes and other hardware aspects will be out of our concerns and will be developed by CVC researchers. A communication channel though is opened to discuss about the development of the whole project.

The head mounted displays first appeared in 1965 when Ivan Sutherland developed the first HMD called "the sword of Damocles" [20], it established the ground for further development in the field. The development of these technologies has grown in the last years mainly centered in the video-games field. Some examples, are the Oculus [11] or the HTC Vive [5]. These products are virtual reality headsets, therefore they are only capable of showing computer generated scenes. The developed project prototypes differ in this matter because they can also show the real world using two integrated cameras. This kind of HMDs are called video-see-through, the current version of the prototype can be seen in Figure 1.

2 OBJECTIVES

2.1 Initial objectives

First, we will develop a software that will be capable of capture stream from a stereo camera and display it in real time. The software will show one camera stream on each half of the screen, also the HMD will enable only to each eye to see one half of the screen, all of this will simulate the stereo effect (creating a depth effect). Another goal is



Fig. 1: Current version of the HMD video-see-through prototype.

to prepare this software to be easily expandable with new modules in the future.

Second, as is reported in [22], and we discovered in a user testing session in the CVC [1], using head mounted displays can cause a variety of adverse physical reactions, as the end goal for our HMD is to be used it in workplaces, these symptoms can reduce the concentration and the efficiency of the workers. For that reason, reducing these adverse physical reactions is a priority. To achieve this goal, we will try to apply two techniques:

- Accommodation-Vergence: As is reported in the [22], [10] articles the mismatch between accommodation and vergence in head mounted displays causes a conflict on the expected depths increasing the feeling of discomfort and dizziness. One idea to reduce this effect is to dynamically move the position of screen frames as the focus changes from closer to distant objects and vice versa. To achieve this, we will use a neural network that will be able to discern between closer and distant objects, also we will need the depth map obtained via the stereo camera setup.

In addition, a progressive change from one focus to the other may be needed, as a big change in focus can induce sickness to the user.

- Depth of field (DoF) blur: Recent investigations [3] suggest that applying a DoF blur to a scene viewed using a head-mounted display can reduce visual discomfort, the challenge here is that our project has to make this DoF blur in real time in a real world environment, in contrast to the developed in that investigation where they used computer generated scenes. To

• E-mail: richard.segovia@e-campus.uab.cat
 • Menció en Computació
 • Project supervised by: Coen Antens (CVC) and Felipe Lumbreras (Computació)
 • Course 2017/18

achieve this, we are going to use disparity map and the information obtained by a neural network that discerns between closer and distant objects.

Third, as the HMD will be used in workplaces, the idea of adding data to the environment, could improve the workflow and the work efficiency. For this reason, we think, that adding a third camera to the system can add valuable information that can be mixed with the environment. A similar idea can be found here [17] where a third camera is used to add information to the real world. For example, by adding an infrared camera, the user can be warned by highlighting objects too hot to be touched.

Fourth, as we discovered in a user testing done in the CVC [1] last January, using a computer while wearing the HMD is a difficult task, mainly because the screen was unreadable for the brightness saturation of the image. For that reason, a system capable of detecting computer screens and adapt the brightness, contrast and others parameters of the image stream will be developed.

Along all the development, user testing sessions will take place to check the improvements made between the different versions of the developed software and the different prototypes developed. The main goal of this sessions will be to evaluate whether the developed software works and if it reduces the sickness feelings of the users. The procedure used in the user testing will be like the performed in [3], where the Simulator Sickness Questionnaire [14] was used to evaluate the symptoms.

2.2 Progress and changes

After these firsts weeks of development some objectives have changed and other have appeared, as we will explain in section 4, the main issue was that the depth map generation needed some work before integrating it into the system.

The integration of a reliable depth map generation is essential to the project, it is needed to complete both Accommodation-Vergence and DoF goals, and as it give us knowledge of the environment it would probably be key in future developments. For these reasons it was decided to add it as an objective itself. The main goal will be to integrate a fast and reliable depth map generator with all the required components.

As the depth map generation requires a stereo pair of undistorted images to obtain a reliable depth map, the integration of a system capable of reading calibration files or calibrate itself and undistort the output of the cameras is required.

The performance of the depth map generation is also a key issue, as a slow image generation will give the system high latency and that could cap the uses cases of the module.

The current objectives and tasks list can be found in the appendix, this was done to gather all the goals that this project seek to achieve. Future references to the objectives and tasks will use the appendix list numeration. As we will see in 4, not all of this goals will be reachable due to time constraints.

3 METHODOLOGY

Scrum and its variants are one of the most spread work methodologies nowadays. Therefore we believe convenient to use scrum as the foundation of this project, however, as this project will only be done by one person, some changes have to be made.

First as there is only one developer the daily meeting will be substituted with a weekly meeting with the project supervisors, in this case the tutor and the boss of the laboratory department in the CVC. In these meetings we will evaluate the development done, the issues faced that week and the problems solved. In addition to that we will discuss and review future milestones and the progress towards them.

Second, a backlog will be prepared at the beginning of the project and will contain the main goals divided in tasks, these tasks will be organized in groups of sprints easily done in a week. As each sprint has a backlog of task to be done, a tool can be used to organize this backlog, in this case Trello[2] will be used.

Third, each iteration over the documentation and the development will be kept by a version tracker, in this case, Github [12] and its desktop client [13].

Keeping the main scrum methodology, I think it is interesting to grab some ideas from the Lean software development [6]. The idea of removing the so called "Muda", non-important extra features and processes, could be beneficial for this project because it could reach an overwhelming dimension for the limited time that is had.









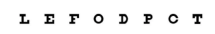
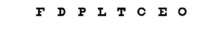
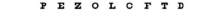
Involving the tools that will need to be used develop the project, first of all, the current viewer version is made in C++ using QT libraries [4], therefore any tools that would be used need to be compatible with the current version of the viewer. For that reason among others, TensorFlow[15] was chosen as the main library for building the neuronal network. Currently, TensorFlow supports Python and C++¹ as development languages. In addition, it is also possible to save a trained model in Tensor and load it using OpenCV both in Python and C++. Other reason to choose Tensor over other alternatives is because is commonly used by the CVC researchers, therefore, if any advice is needed, it can be solved asking one CVC researcher.

The custom scrum methodology already explained has been followed without any change in the development. Every part of the development process has been documented and every change in the code has been uploaded to a private Github repository. It is worth mention that the OpenCV [21] library has been used to develop the calibration and undistort module.

Once the tasks 3, 4a and 4b were finished, a protocol was designed in order to do every user testing session in the same conditions. It has the following considerations:

- To perform the testing, 3 ophthalmological patterns, Figure 2, will be set in 3 different locations, one at reading distance, other at close distance, like the pc screen; and other far away at the wall of the room.
- Each user will do one session per day and the distance calibration done by the user will be done in a random order.

¹Currently, the TensorFlow's C++ runtime is only available for Linux.

Panel Snellen	www.provisu.ch
	1 20/200
	2 20/100
	3 20/70
	4 20/50
	5 20/40
	6 20/30
	7 20/25
	8 20/20
	9
	10
	11

Para realizar esta prueba siga las siguientes instrucciones:

Fig. 2: Ophthalmological pattern used in the user testing sessions from task 4a. The pattern was downloaded from [7].

- Every user will rest for a few seconds between each calibration of the preset.
- Every preset calibration will start in with the camera images in the center of the ROI.
- Once the user calibration is finished, every user will comment its feelings when using a setting preset in the other distances.

4 PLANNING

4.1 Dataset

To be able to develop this project a dataset will be necessary to train the neural networks and to check the depth map. In this case as we have available three hardware prototypes, a data collection procedure will be possible. For that reason, a dataset recording session is planned.

Also, a dataset will be used to make sure our training is correct and to encompass a broader range of environments. For that reason, we selected first the Middlebury stereo dataset [19] because it has a great dataset of stereo images and the groundtruth of their depth map. Additionally a dataset with indoor and outdoor scenes is recommended. Several dataset could be used with that purpose, for example the MIT's places dataset [23] and the INDSECS indoors only dataset [18].

A calibration dataset is required to generate a quality calibration. This dataset will contain a series of images of a chess like pattern located in different parts and depths of the image. A requirement is that the patterns appears completely in both cameras. This dataset will be updated every time the cameras or the HMD gets manipulated, that is because any minor change in the focal length or the position between the cameras will affect the results of the calibration.

To be able to evaluate the depth map generation, a video sequence of the stereo pair must be captured and later processed by the depth map module. It would be convenient the integration of a video capture module to the current program to ease the evaluation task.

4.2 Accommodation-Vergence

In this stage, a neural network will be trained to detect the distance of the focus and the necessary vergence between images to improve the user comfort. This will, alongside with the DoF blur, be the main time-consuming stage of the project and will have the following parts:

1. Preset creation: In this task, the current viewer will be expanded with the creation of multiple and easy to change presets that will allow further user testing.
2. User testing, presets: In this first user testing session after the finalization of task 1, the users will be told to configure their own focus settings and to try to change between them to adapt the focus to closer or distant objects. The goal here is to acknowledge whether the user feels better accommodation with the focus between objects.
3. Progressive preset change: After the development of 1 and 2, if the users report sickness because of the sudden change between presets, a smooth transition between them will be developed to reduce this issue.
4. Disparity map: In parallel with the development of the tasks 1, 2 and 3, a disparity map generator from a stereo pair is required to continue with further development. As this subject is complex per se and is not the main goal of this project, a library that already implements a good stereo matcher, like LIBELAS[8], will be used.
5. Neural network with camera streams: This task will start after the analysis of the data collected in the task 2. The main goal will be to create a neural network capable of returning the value of the vergence. To achieve this several architectures will be tested and compared, with one, two or three streams from the cameras and including the disparity map information when developed.
6. User testing: In this user session, the goal will be to check if the development done in this stage has reduced the sickness feelings of the users, and with which solution the user feel more comfortable.

4.3 Depth of field blur

This stage will be developed after the finishing of the previous Accommodation-Vergence phase 4.2 and shares some development with it.

This problem can be resolved in two ways, using a disparity map only to create the DoF and using Neural Network to aid the creation of it.

1. Mock up blur: to ensure that the results of [3] are applicable to real world scenarios, a mock up blur photo dataset will be made, and a user testing session will take place.
2. Designing de pipeline: this task goal is to design, both pipelines, the disparity map only and the neural network aided.
3. Disparity map version: this version will generate the DoF blur using only the disparity map to obtain the depth and using the center point of the cameras as the focus point.
4. Neural network adaptation: this task will take the neural network developed in the previous phase and will modify it to make it usable in this phase of the project. The idea is that this network will produce an indicative value of the blurriness needed using the camera images to determine whether the scene has closer or distant objects.
5. User testing: Along with the development of 3 and 4, a user testing session will take place to check whether this system has reduced the dizziness and which solution is more comfortable for the user.

4.4 Improving screen quality

In this stage a system to improve the quality when the user is looking a screen will be developed. This system can be split in two tasks:

1. Quality improvements: In this task a system capable of changing contrast, brightness and other image parameters will be developed, this system will not be automatic and will need the interaction of the user.
2. Automatic switching: The goal of this task will be to free the user from the interaction with the system developed in task 1, to achieve this a neural network will be developed to detect when a user is looking a screen.

Part of the development of this objective will be left to future projects due to time constrains.

4.5 Third camera

This stage will have as a main goal the integration of a third camera in the viewer software and highlighting relevant information obtained using the third camera on stream viewed by the user. It can be split in the following tasks.

1. Integration with the viewer: Introducing a third camera in the system will mean making some modifications in part of the viewer code. The goal of this task will be to prepare the different modules of the viewer for the introduction of the third camera.

2. Research mixture: After the development of task 1, a system to allow the mixture of images between visible and the third camera spectrum will be developed, the goal will be to add information to the user from the third camera different spectrum.

Part of the development of this objective will be left to future projects due to time constrains.

4.6 Research, documentation and planning

This stage of the project can be split in three parts:

1. Initial documentation: the documentation process will start at the beginning of the project and it includes, recollect information about the state of the art, do the planning and search about the different technologies involved in this project.
2. Progress documentation: this will be written along all the project and will consist in document every change and issue that can happen. When a part of the project changes its orientation or a milestone is achieve the documentation will be updated with the changes.
3. Closure documentation: When the project will approach the finish, an article, a recompilation of all the work done and a presentation will be made to wrap up the project.

4.7 Planning changes

During the firsts weeks of work some problems arose, the main issue was that in order to integrate a system capable of generating depth map image, it was required to undistort the image pair captured. This is because each optical lens of each camera introduces a distortion to the image captured. In addition, as the lenses are not exactly equal and they are not exactly aligned, a stereo calibration system was required to produce the undistortion. This stage of the project was split in four parts.

- Calibration development: To develop the calibration and undistortion, the OpenCV library [21] was used, it was selected because it has a great variety of calibration and undistortion functions already implemented and also because it was already being used in other modules of the interface. The goal of this task was to develop and integrate a monocular and stereo calibration module.
- Depth map development: As explained in 4.2, LIBELAS library was selected. One of the main reasons for this selection was the performance. That is, because high processing times must be avoided to prevent latency in the DoF or the vergence modules. The goal of this task was integrating the LIBELAS library with calibration and the camera output aiming to obtain the depth map.
- Calibration dataset: Once the calibration and the depth generation systems were integrated, the calibration dataset was captured, following the explained in 4.1. This dataset needs to be updated every time the headset gets manipulated to ensure the calibration, and therefore the undistortion is accurate.

- Depth map evaluation: Once the module is completely working, a evaluation will take place to evaluate the quality and the reliability of the depth map obtained. This task is currently in progress. The goal of this task will be to document, calibrate and modify the depth map to ensure its outputs have the enough quality to continue with further modules.

Before starting the user testing, we realized that several changes have to be done in the way the images move inside the screen of the viewer. These changes were done, first to make more user friendly the setting up of the camera positions and secondly to be able to use most of the screen of the HMD without leaving any black margins like it was happening with the current version. Moreover, these changes allow the user to fit the size of the objects shown on the screen to a closer to real size, therefore reducing the possible inconveniences produced by this size gap between the screen and the reality.

At this point of the project is obvious that some tasks of the project will not be done due to time constrains, these tasks are 7, 6b and 5c. The task of making the change between presets is still undone due to the delay in the user testing.

As some task like the user testing sessions, progressive preset selection and the neural network development were delayed, the planning timeline had to be updated. The current version of the Gantt diagram can be found in the figure 4, in the appendix.

5 DEVELOPMENT PROGRESS

5.1 First development phase

At the start, the first objective that needed to be completed was 1. It was delayed because the main HMD prototype was being revised. Therefore the first objective started was 3, the development of the depth map system. The LIBELAS library was downloaded and some of the headers were updated to make it compatible with the compiler used, Microsoft Visual c++ compiler v15. After the firsts stand-alone test we realized that a calibration and undistort module was necessary, as LIBELAS requires a totally undistorted image with completely rectified epipolar lines².

The implementation of the calibration and undistort parts was difficult mainly due to the lack and the outdated documentation of the calibration library in OpenCV C++. Finally, the module was implemented in two classes, the monocular calibration class and the stereo calibration class, the stereo one uses two monocular calibration objects to store the non-shared parameters. Both can save and load their parameters from YAML³ configuration files.

Once the calibration and undistort modules were integrated, the depth map module was finally tested within the interface. The first thing noticed was the huge latency between each frame, as can be seen in the results in 1. These results are not optimal, since we need a high-speed system that adapts quickly the vergence distance. We found out two possible ways to address this issue:

²In stereo systems, when two points that can be seen in both images and are in the same Y coordinate

³YAML is a human readable serialization language commonly used to save data and for configuration files.

fps	Seconds	Width	Height
0.50	1.98	1600	1200
0.82	1.21	1100	1100
2.16	0.46	800	600
4.67	0.21	600	400
2.97	0.33	600	600

Table 1: Frames per second (fps) and seconds between frames for the generation of the depth map for each image resolution.

- Reduction of the image resolution: In our use case, only the center of the images is necessary, this is because it is where the user will be usually looking at, therefore taking off the outer pixels of the images will not alter the behavior of the vergence system and will improve the speed of the generation of depth map images.
- Faster libraries and implementation: Another option would be using faster or optimized libraries to be able to speed up the performance in the generation of images. As the we are using libraries already implemented our possibilities are limited in this matter, an option would be using a multithreading version of the LIBELAS library, already implemented in [8].

In this case, the first option will be selected, but if needed the second solution can also be used. In Table 1 can be observed the current performance of the depth map generator. As well as the times, in the Figure 3 we can see a full resolution disparity map and the original image. In this example is visible that the disparity is far from perfect, that is because there is still some calibration that can be done in the LIBELAS parameters. Also, is worth mention the extremely high accuracy needed in the calibration phase to obtain a correct depth map, any minor change in the camera setup, can give inaccurate depth maps. At this point, we considered that the only remaining task to be done in 3 is 3d.

The point 2b was also developed in these firsts weeks of work, the implementation of this module has been done with OpenCV. It has two options: The first save the images in memory and at the end of the recording session saves it to the main storage. This option allows the recording at high frame rates but due to memory size limitations can only record a couple of seconds. The second option let the user to record and save directly to the main storage this allow for longer footage but at slower frame rates.

The point 1 was also done as the preset creation and saving module were developed.

Related with objective 6, the third camera has arrived and some tests have been done to prepare its integration.

5.2 Second development phase

As its explained in the planning changes, we realized that the visualization of the streams inside the screen was not perfect and that the size of the objects was not customizable. To achieve a better experience, the screen paradigm



(a) Original Image



(b) Depth map image.

Fig. 3: Original and depth map images obtained using the LIBELAS library.

was change, instead of displaying an image at full resolution and move it around the QT scene, a ROI of the image was selected and placed in a fixed position inside the QT scene. In this version rather than moving the image, the ROI over the image is the element that moves. This allow the user to control better the ROI and the zoom inside the image. Leaving them without any overlap between images.

It is worth to mention that the development team of the hardware prototype has finished the third prototype, in Figure 1, it now has an improved holding system that ensures that the camera will not move during our tests. Previously, despite our efforts to keep the camera movements at minimum, every time we started recording after calibration, the movement of the HMD or even the tension on the wire, ended up moving, and therefore decalibrating the cameras.

Related with the task 2b, the viewer has been ported to a 64 bits architecture, consequently, the number of frames that can be stored in memory has grown significantly. this will allow us to record longer dataset videos.

Related with the task 3, a color map has been added to improve the visibility of the depth. This was added because for humans is easier to see a gradient of colors rather than levels of gray.

Secondly after reading some papers about stereo [9],[16] we arrived to the conclusion that a Siamese architecture was the best option to create our neural network. A Siamese neural network is a net composed with two completely separated input layers, in our case, one for each camera. After some layers of feature extraction, we arrive to a layer where the two separated nets converge into one layer, after this layer we have some more layers where the inputs are compared and some feature extraction and feature extraction.

Taking in account that the first user testing phase has been done with a limited amount of users, we have extracted some interesting conclusions, Table 2. It seems that there are two types of users, the ones that feel comfortable since the beginning of the test and do not feel the need to change the presets, subjects S1 and S3; and the users that need to customize the distance between images to feel comfortable, subjects S2, S4, S5 and S6. We notice that the users that did not change much the distance between image did not feel dizzy when using a setting in different distance than the original. On the contrary, the users that did change the distance between images mostly felt dizzy when using a setting set for a different testing distance than they were seeing. A new round of user testing session with more users will be done to verify if these results are correct.

At the same time, we notice that in most cases, both types of users feel when using the close setting in large distances objects are seen further away, and vice versa in case of far settings in closer distances.

After these first user testing sessions we notice some issues in the protocol and the viewer that need to be addressed. First we need to ensure that the users like S1 and S3 are correctly seeing the stereoscopic 3D. Second, some users found difficult to set the vergence in the close distance for lack of a relevant 3D object or form, for this reason we propose to introduce an object into the test to ease the setting up. Third, during the testing of the settings in other distances, we think that it is necessary to make the test also in the distance where the setting was set to verify if the user still feels that it is the correct setup. Fourth, we notice that

some users found difficult to find the keys used to move the images, for that reason we decided to move the keys that change the position of the right image from i,j,k and l to the arrow keys. Fifth, after the testing was done we notice that the cameras were slightly out of focus and the brightness, contrast and color balance was not correct, as this can also impact in the user comfortability, for now on before every user testing session, the general camera configuration will be check.

6 PROVISIONAL CONCLUSIONS

First, after the finishing the user testing session, we notice that users felt more comfortable in general, this is mainly due to the improvements in the prototype itself and because the improvement on the viewer that now uses the full screen instead of leaving black margins. Having said that, there is still work to do to include a dynamic change in the vergence using the distances obtained by the depth map. Second, after the results of the first user testing session, we concluded that further analysis and therefore more user testing must be done to prove that the pattern that we discovered is true. If our theory confirms, we will start working on the dynamic vergence.

7 ACKNOWLEDGMENT

This work is supported in part by a CVC transfer project with ProCare Light company, and partially funded by the Spanish Ministry of Economy and Competitiveness and FEDER under grants TIN2014-56919-C3-2-R and TIN2017-89723-P.

REFERENCES

- [1] Coen Antens. Unpublished: User testing resume. 2018.
- [2] Atlassian. Trello. <https://trello.com/>, 2018. Last access March 9th 2018.
- [3] Kieran Carnegie and Taehyun Rhee. Reducing visual discomfort with hmds using dynamic depth of field. *IEEE computer graphics and applications*, 35(5):34–41, 2015.
- [4] The QT Company. Qt. <https://www.qt.io/>, 2018. Last access March 9th 2018.
- [5] HTC Corporation. Vive. <https://www.vive.com/us/>, 2018. Last access March 10th 2018.
- [6] Anderson David J. Lean software development. [https://msdn.microsoft.com/en-us/library/hh533841\(v=vs.120\).aspx](https://msdn.microsoft.com/en-us/library/hh533841(v=vs.120).aspx), November 2011, updated in November 2012. Last access March 9 2018.
- [7] Pro Visu Foundation. Panel snellen. https://www.provisu.ch/images/PDF/Snellenchart_es.pdf, 2018. Last access May 24th 2018.
- [8] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*, 2010.
- [9] Tewodros Habtegebrial, Kiran Varanasi, Christian Bailer, and Didier Stricker. Fast view synthesis with deep stereo vision. *arXiv preprint arXiv:1804.09690*, 2018.
- [10] David M Hoffman, Ahna R Girshick, Kurt Akeley, and Martin S Banks. Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of vision*, 8(3):33–33, 2008.
- [11] Facebook Inc. Oculus rift. <https://www.oculus.com/>, 2018. Last access March 10th 2018.
- [12] GitHub Inc. Github. <https://github.com/>, 2018. Last access March 9th 2018.
- [13] GitHub Inc. Github desktop. <https://desktop.github.com/>, 2018. Last access March 9th 2018.
- [14] Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.
- [15] Google LLC. Tensorflow. www.tensorflow.org, 2018. Last access May 24th 2018.
- [16] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016.
- [17] Jason Orlosky, Peter Kim, Kiyoshi Kiyokawa, Tomohiro Mashita, Photchara Ratsamee, Yuki Uranishi, and Haruo Takemura. Vismerge: Light adaptive vision augmentation via spectral and temporal fusion of non-visible light. In *Mixed and Augmented Reality (ISMAR), 2017 IEEE International Symposium on*, pages 22–31. IEEE, 2017.
- [18] Andrzej Pronobis and Barbara Caputo. The kth-indecs database. *Kungliga Tekniska Hogskolan, CVAP, Tech. Rep. CVAP297*, 2005.
- [19] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, pages 31–42. Springer, 2014.
- [20] Ivan E Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.
- [21] Opencv team. Opencv library. <https://opencv.org/>, 2018. Last access March 10th 2018.
- [22] Kasim Terzic and Miles Hansard. Causes of discomfort in stereoscopic content: a review. *arXiv preprint arXiv:1703.04574*, 2017.

Subject	Feelings	Close		Medium		Far	
		ΔX	ΔY	ΔX	ΔY	ΔX	ΔX
S1	Comfortable	0	12	6	0	-18	0
S2	Discomfort	6	0	-258	0	-114	30
S3	Comfortable	6	0	6	0	6	0
S4	Discomfort	6	-6	-66	12	-72	-6
S5	Discomfort	-24	0	-132	0	-84	0
S6	Discomfort	-48	0	-78	10	-90	-12

Table 2: Results of the vergence in the user testing sessions. The first column represents the general feelings of the users when changing the settings. Note that ΔX means separation between the two images in the X axis and ΔY means separation in the Y axis.

[23] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

A OBJECTIVE AND TASKS LIST

1. Modular HMD Viewer
 - (a) Preset Shortcuts
 - (b) Better program configuration
 - (c) Better viewer scene
2. Dataset generation
 - (a) Calibration dataset
 - (b) Recording module
 - (c) Video dataset
3. Disparity map
 - (a) Calibration
 - (b) Depth map
 - (c) Calibration dataset
 - (d) Depth map evaluation
4. Accommodation-Vergence
 - (a) Preset creation
 - (b) Disparity map
 - (c) Neural network
 - (d) User testing
5. Depth of field blur
 - (a) Mock-up blur
 - (b) Disparity map Version
 - (c) Neural network Version
 - (d) User testing
6. Third eye
 - (a) Integration of the viewer
 - (b) Research mixture of images
7. Screen quality

B GANTT PLANNING

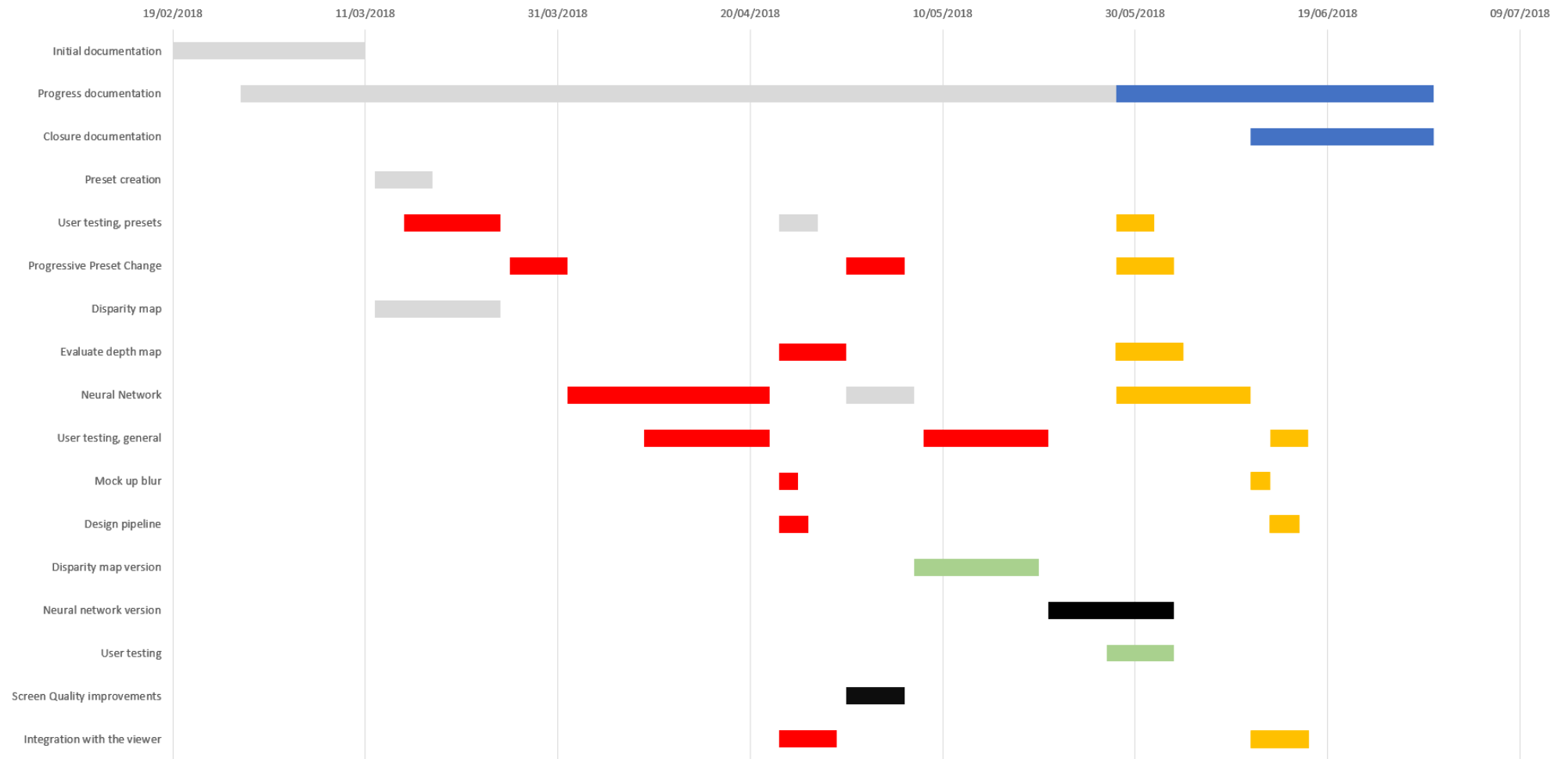


Fig. 4: Gantt diagram of the project. Note that blue tasks are remaining tasks, grey task are done tasks, red tasks are tasks not done in the scheduled time, yellow tasks are rescheduled tasks, black tasks are task that will not be completed and green tasks are tasks that probably will not be completed due to time constraints