A HEAD AND EYE TRACKING SYSTEM FOR HANDS-FREE CONTROL OF A ROBOTIC EXOSCOPE

BAŞ VE GÖZ TAKİP SİSTEMİNE DAYALI, ELLER SERBEST ÖZELLİKLİ ROBOTİK EKSOSKOP

AŞKIN AŞKIN

PROF. DR. HASMET GÜRCAY

Supervisor

Submitted to Institute of Informatics of Hacettepe University as a Partial Fulfilment to the Requirements for the Award of the Degree of Master of Science in the Department of Computer Graphics

2019

ABSTRACT

A HEAD AND EYE TRACKING SYSTEM FOR HANDS-FREE CONTROL OF A ROBOTIC EXOSCOPE

Aşkın AŞKIN

MSc of Institute of Informatics, Department of Computer Animation and Game Technologies

Supervisor: Prof. Dr. Haşmet GÜRÇAY

Co-Supervisor: Dr. Shahram EIVAZI

May 2019, 44 pages

An exoscope is a high definition video teleoperated system for performing microsurgery and it has been proposed as an alternative method for the operating microscope. It provides a visualizing magnified image of application areas on a display helping the surgeons to complete the operation. The advantages of the exoscope are providing a wide field of view and deep focus of application area. Moreover, the system is minimizing the need for repositioning and refocusing. On the other hand, the exoscope has also limitations such as performing under 2D motion images. Additionally, conducting fast and safe interactions with a surgical microscope or exoscope are one of the basic principles applied in microsurgery and the exoscope requires improved hand and eye coordination for high precision work. Studies revealed inefficiencies of the current manual hand-grip, foot-pedal, and mouth-switch interaction techniques with these magnifiers. Adjusting the field of view is a frequent source of unwanted delay and interruption when using a microscope or exoscope in microsurgery. Consequently, in this thesis, we present an experimental investigation of hands-free control for an exoscope system. We report the development of a head and gaze-contingent robotic exoscope for camera viewpoint automation. We recorded empirical measures, namely task completion time and the number of errors, and examined the user workload using NASA task load index. The results show the task completion time is shorter for hands-free exoscope compared to hand-based interaction technique. We further show the challenges and difficulties related to mental, physical, and temporal demand as well as performance, effort, and frustration level of participants for the design of the hands-free surgical exoscope.

Keywords: Hands-free, Robotic, Exoscope

ÖZET

BAŞ VE GÖZ TAKİP SİSTEMİNE DAYALI, ELLER SERBEST ÖZELLİKLİ ROBOTİK EKSOSKOP

Aşkın AŞKIN

Yüksek Lisans, Bilgisayar Animasyonu ve Oyun Teknolojileri Bölümü

Tez Danışmanı: Prof. Dr. Haşmet Gürçay

Eş Danışman: Dr. Shahram Eivazi

May 2019, 44 sayfa

Eksoskop, mikro cerrahide kullanılan uzaktan kontrollü yüksek çözünürlüklü görüntü izleme sistemidir ve ameliyat mikroskobuna alternatif bir yöntem olarak önerilmiştir. Cerrahların operasyonu tamamlamalarına yardımcı olmak için, ekranda uygulama alanlarının büyütülmüş bir görüntüsünü yansıtır. Eksoskobun yararları, uygulama alanının geniş bir görüntüsünü sağlamak ve derinlemesine odaklanmasına yardımcı olmaktır. Ayrıca, sistem yeniden konumlandırma ve yeniden odaklama gereksinimini en aza indirir. Öte yandan, eksoskobun 2-boyutlu hareketli görüntülerin gösterilmesinde sınırlamaları da vardır. Ek olarak, cerrahi bir mikroskop veya eksoskop ile hızlı ve güvenli etkileşimler yapmak, mikro cerrahide uygulanan temel ilkelerden biridir ve eksoskop, yüksek hassasiyetli işler için, gelişmiş el ve göz koordinasyonu gerektirir. Yapılan çalışmalar, mevcut elle kavrama, ayak pedallı ve ağızla yönlendirmeli etkileşim tekniklerinin bu büyüteçlerle çalışmasının verimsizliğini ortaya koymuştur. Mikro cerrahide, mikroskop veya eksoskop kullanıldığında görüş alanının ayarlanması, sık sık istenmeyen gecikme ve kesintilerin oluşmasına neden olur. Sonuç olarak, bu tezde, bir eksoskop sisteminin, eller serbest özellikli olarak kontrol edilmesinin deneysel bir araştırmasını sunuyoruz. Kamera görüş açısının otomasyonu için, kafa hareketleri ve göz takip sistemine dayaklı bir robotik eksoskop geliştirdik. Görevin tamamlanma süresini ve hata sayısını hesapladık ve NASA görev yükü endeksini kullanarak kullanıcı iş yükünü inceledik. Sonuçlar, görevin tamamlanma zamanının, el bazlı etkileşim tekniğine kıyasla, eller serbest özellikli eksoskop için daha kısa olduğunu göstermektedir. Eller serbest özellikli cerrahi eksoskobun tasarımında, katılımcıların performans, çaba ve başarısızlık düzeylerinin yanı sıra zihinsel, fiziksel ve zamansal gereksinimle ilgili deneyimledikleri zorlukları da gösteriyoruz.

Anahtar Kelimeler: Eksoskop, Robotik, Mikro cerrahi

ACKNOWLEDGEMENT

I would first like to thank my thesis advisor Prof. Haşmet Gürçay of the Institute of Informatics at University of Hacettepe. I also want to thank my co-advisor Dr. Shahram Eivazi from University of Tübingen. The door to Dr. Eivazi and whole Perception Engineering Group office was always open whenever I ran into a trouble spot or had a question about my research or writing. They consistently allowed this paper to be my own work but steered me in the right the direction whenever he thought I needed it.

I would also like to thank the robot expert Adrian Zwiener for his help and Prof. Zell for opening me the door of the robotics department at University of Tübingen. Without their passionate participation and input, the project could not have been successfully conducted.

I would also like to acknowledge Jun. Prof. Enkelejda Kasneci of the Perception Engineering Group at University of Tübingen for providing me a student job in the group and let me a part of this wonderful team.

I thank to my friend Dr. Saniye Sinem Yılmaz for reviewing my thesis and her effort. She was always there whenever I need help. I cannot deny her support and encouragement in this period.

Finally, I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Aşkın Aşkın,

January 2019, Berlin

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations

VR	Virtual Reality
ROS	Robot Operating System
HMD	Head-mounted Device
SDK	Software Development Kit
OS	Operating System

1 INTRODUCTION

Teleoperation can be defined as doing work at some distance and the content of the work could include almost every situations or environments completing by telerobotic devices. In this content, the distance can emphasize two different meaning. It may define a physical distance or a change in scale. This definition depends on the situation and environment of the work content, that teleoperation systems assist the human operator to fulfil complex, uncomfortable, limited tasks such as microsurgery and dangerous, costly, less structured environments such as nuclear plants, battlefields, space, and underwater operations.

Teleoperation consists of two main parts. First part is the control module called a cockpit where the human operator (master) controls the telerobotic device. The second part is the telerobotic device (slave) at the remote location.

The aim of teleoperation is providing some advantages to human operators in the situations of which the manual control cannot be used due to the dangerous, hazardous and limited environmental conditions. The most typical examples of the teleoperation are the handling of nuclear materials, microsurgeries, space, and underwater exploration tasks. The modern teleoperation has a rich history and it has begun at the end of the 40s in the Argonne National Laboratory. The first master/slave telemanipulator was demonstrated for handling the chemical and nuclear materials [1].

After the first implementation, the teleoperation developments have been accelerated. In the beginning, the first concern was the dexterity. The teleoperation systems only had to support simple task such as handling some liquids and pour them from one bottle to another. After fulfilling dexterity, other needs came up such as increasing the distance between the human operator and the operation environment because of safety precautions. Therefore, the more complex mechanical transmission has been required. Although the operator could still have a direct view of the slave and the operation field, implementation of mechanical transmission has provided the ability of moved the slave through wires by the human operators. Due to the mechanical separation between master and slave, two severe problems have raised: the lack of the vision and the time-delay.

Therefore, after dexterity and mobility, researchers have started to focus on improving the other sensing capabilities which can be named as telepresence. In 1987, telepresence was described as the ideal of sensing information at the remote site and transferring this information to the human operator. The aim of telepresence is letting the human operator perform his/her tasks as he/she was within the remote area. Implementation of video technology and feedback mechanism to teleoperated systems have made the telepresence possible. Implementation of the advanced control loops at teleoperators has become possible by the development of current computer technology. Finally, virtual reality has raised as a major tool for teleoperated systems. Virtual Reality (VR) technology is advanced to a level where it’s possible to transfer visual and audio stimulus at near-real-time speeds over long distances [2].

The main purpose of the teleoperation is based on the idea that providing an improved control module for human operators, so they can feel that they are actually working in the teleoperation area. Due to the fact that the current researchers have focused on improving better and more effective teleoperation methods and systems by implementing advanced feedback and vision systems for the most difficult tasks.

This thesis project “Virtual reality teleoperated control robot using eye and head tracking” connects the HTC Vive virtual reality set to a Kinova Jaco2 robotic arm. One of the primary uses of teleoperated robots is to allow operators to interact with the environment which is physically distant from the location of the operator. Typically, the human operator has a sense of being on location, with cameras that detect sensations of vision. A live video can then be presented at the operator location by screen or Virtual reality glasses.

We provide a live 3D magnified view to the surgeon’s natural field of view through augmented reality glasses. The augmented reality glasses assist the user to rotate the webcams mounted robotic arm, along with the eye tracker allows adjusting the contrast of the display.

In this thesis, we aimed to prove the hands-free based system is more preferable to compare to handheld-controller based system in surgeries. We set up 3 tasks to experiment with the systems. As an experimental setup, we utilized one board with three nails and two nuts. The users asked to move from nail to another by a tweezer as a typical experimental setup for the surgeries.

After the experiment, we compared the completion time of hands-free and handheld controller systems. Furthermore, two different questionnaires were asked to the users to obtain negative and positive sides of both systems. One of the questionnaires is NASA Task Load Index which is utilized to grade the grade experiment. We asked the users about mentally and physically effort to scale their performance in the system. Finally, we discuss the comparison of both systems to show that our proof of concept is more convenient for the surgeries.

1.1 History

In the previous section, teleoperation tasks are defined as doing work in a distance. Therefore, the poking of fire could be considered as one of the first example for teleoperation in history. Actually, the poking the fire can be categorized as a telemanipulation which can be considered as the primitive stage of teleoperation systems. A human hand is a perfect tool for handling daily tasks unless the environment is hostile, dangerous for a human. When the environment has become dangerous and hostile like hunting and fighting or they might have wanted to ease their fundamental tasks like cooking, then the more adequate tools were needed.

In the 1940s, after the building the first nuclear reactor because of the military applications, the necessity of the teleoperation system has become more obvious, because the radioactive nuclear materials must have been handled safely that were placed in a hot cell. The hot cell was a dangerous environment for a human operator, so he/she must manipulate this situation at the outside of the cell by remote control in order to protect. Therefore, R. Goertz and his group have developed the first modern master/slave teleoperator system [1]. Their first system was based on mechanical systems that both energy and decision making were completely done by the human operator. The first update on the system was done by implementing energy into the system with feedback servo control. The idea of implementing the new updates into the system was mitigating the operators' heavy workload both in physically and mentally.

After the first demonstration of the teleoperation systems and the exploration of various advantages, the range of application areas has extended. Deep-sea exploration researches are one of the first areas that the teleoperation systems have been utilized. Today, most of the deep-sea organizations have been using teleoperated submarines calling as a Remote

Operated Vehicles (ROV) because of the hostile environment of deep-sea for humankind. These ROVs contains different teleoperation systems in order to perform various tasks under the sea [3].

In both nuclear and deep-sea exploration kinds of research, the environment is dangerous and hostile for a human operator. In space applications other than dangerous situations, the astronauts need extra equipment's which are much more expensive than a teleoperation system.

1.2 Applications

After the first implementation of teleoperation systems in the 1940s for the military applications, the various teleoperation systems have been developed for a wide field of application areas ranging from microsurgery to space beside those security applications, underwater vehicles and even in forestry and mining applications. Teleoperation systems combine a highly diversified set of supporting technologies to tackle problems like large time-delay and broaden the application area of teleoperation much wider than before. This section presents brief information about developments in the most common application areas for teleoperation systems.

1.2.1 Space

The teleoperation systems have been used very frequently in space applications because the environment of space is not suitable for the physical presence of a human and operating some tasks in space requires many resources for humans. In fact, in some cases like the exploration of Sun, the mission becomes totally impossible. We can categorize the space applications into three different subgroups. First one is the space exploration robots. The space exploration robots are crucial to explore the Moon, Mars or the other planets of Solar systems and gather information as a first step. The second one is the satellites that are using for communication, GPS positioning and also more accurate weather forecast. The third category is the outer space robot arms like some robot arms on the International Space Station. These robots are being used for the various tasks in the outer space by the human operator, who is an astronaut in this case [4].

1.2.2 Undersea

As mentioned in the previous section, underwater applications are one of the first application areas that the teleoperation systems have been implemented. Teleoperation systems are commonly used in offshore oil explorations, oil platforms, maintenance on drill-heads and also geological explorations. Remote Operated Vehicle (ROV)'s are controlled by the ship at the sea surface by the human operator with video monitors and joysticks. ROVs can also be categorized based on their size, weight, ability or power.

1.2.3 Microsurgery

Microsurgery operations are also one of the application areas of teleoperation systems. It has been started at the middle of the 1980s and has been extending to almost all of the surgical specialties. The aim of the surgery with teleoperation systems is that improving the motor capabilities of surgeons to perform surgical tasks which are impossible with hands. Also, these systems can be used for recreating the area of the surgery virtually for surgeons like being at the patient's bedside which makes possible to perform surgical tasks at miles away from patients.

The first generation of teleoperation systems for surgical applications were based on performing image-guided precision tasks in the neurosurgical field. After the first implementation of these systems into the medical field, robotic systems have developed some disadvantages situations that prevent them to become more invasive.

Surgeons use a magnified view of the operation area with a microscope. During surgery, adjusting the microscope position is a major problem that causes interruptions. These interruptions take 10% of surgery time [5]. Looking away from the microscope causes potentially a loss of awareness, error, delays and undesirable situations.

Handgrips, mouth switch, and the pedal foot switch are current ways to adjust the position of a surgical microscope. Adjusting the microscope position with all these alternative solutions has limited control and it requires removing the hands from operation filed causing an undesired situation [6].

If we make it more detailed, with mouth switch, microscope moves by surgeon's lips or teeth. Although it has been designed to help surgeons change the field of view without removing their hands from the operation area, it requires high physical and mental demands.

Additional to these extra demands, it provides a limited degree of freedoms during the surgery microscope movements, zooming and focusing have become nearly impossible. Furthermore, the microscope zooms by a feet pedal limits the surgeon's movements and microscope reposition is not possible. According to the experiences of surgeons, the feet pedal is uncomfortable to use during the operation and it requires many years of training to become an expert.

Due to the insufficient use of a feet pedal for surgery, another solution has been developed which a head-mounted switch is moving by the surgeon's head. The microscope control is attached to the surgeons' head. The positioning of the microscope area is arranged with the surgeon's mouth and chin. The surgeon placed his/her chin into the device. When the surgeon opens his/her mouth, an electrical switch is pressed by the chin and the microscope is released to move on all planes of movement. However, it has some disadvantages which are restricted to the free movement of the head. Besides, it is extremely unergonomic for the surgeon and also requires many years of pieces of training [7].

A microscope can also be adjusted by voice commands. In this method, speech recognition causes an error and it is not suitable for continuous commands such as zoom, focus, or movements. In eye control switch, the microscope moves by the surgeon's eyes. It is hard to distinguish between unintentional looking and real command on the eye-controlled method and tilting is not possible [8].

The exoscope is a telescope-based optical solution which is attached to a high-definition camera to provide a magnified view on a screen in front of the surgeon. Compares to a surgical microscope, the exoscope has a much smaller size of optical components. Thus, medical assistance can freely reposition the whole device close to the operating field from any desired direction.

During operation, one of the main limitations of video exoscopy is the manual adjustment of repositioning [9]. Current systems require the surgeons to remove their hands from the operation area to adjust the instrument. A counterbalance system as the surgical microscope for exoscope system is another solution to solve this problem. This solution can be shown in Zeiss KINEVO 900 which is an example of exoscope for micro-surgeries [10].

Moreover, this example doesn’t solve the interruption problem related to repositioning the counterbalance exoscope. Manual interactions with a counterbalanced microscope resulted in disruptive events and hazardous situations.

In response to this limitation, researchers have developed new experimental systems, such as in [9], where interaction with exoscope has been achieved through robotic techniques based on a joystick control unit. MODUS V is another example of robotic exoscope which is hands-free, and 3D visualized to the screen. However, the lack of automatic repositioning and stereoscopic view are the vital drawbacks of the current robotic systems.

Our interest is to explore the potential for hands-free interaction techniques within microsurgery settings. There is a large and growing body of medical studies examining automation in robotic surgeries [11]. Our purpose here is to highlight the key works that particularly focus on the hands-free repositioning of surgical instruments.

In this project, we present our solution as a combination of head tracking surgical exoscope and augmented reality technologies. We propose the world’s augmented reality hands-free surgical exoscope. With this exoscope, surgeons will be able to see magnified surgery area and control the view by using head movement and change the contrast by eye movements.

1.3 Outline

This thesis is structured as follows: First Introduction is introduced in the related work and background Chapter 1. In Chapter 2, System Component and Background part of the project will be discussed. Then Implementation part follows as Chapter 3. The experiment task will be detailed in Chapter 4. The results are given in Chapter 5. Finally, a discussion and short outlook conclude this thesis.

2 SYSTEM COMPONENTS AND BACKGROUND

The system is composed of a robotic arm, a stereo webcam set, and a VR headset. The robotic arm used in the experiment is called "Kinova Jaco2". A stereo webcam is used to deliver the images, "Logitech QuickCam Pro 9000". Finally, HTC Vive is used as a VR headset.

The setup allows a person to teleoperate a robotic arm attached a stereo webcam by using virtual reality headset and its movements sensors. The movements of the user, read in near-real time, sent over TCP and fed into a robotic arm and the cameras attached to it. The cameras attached to the arm streams continuous video feedback to create a stereoscopic view. During the operation, the user sees a magnified 3D view of the operation area. Furthermore, the user is able to change the position of the view depending on the headset's movement just by simple head movements (Figure 2.1).

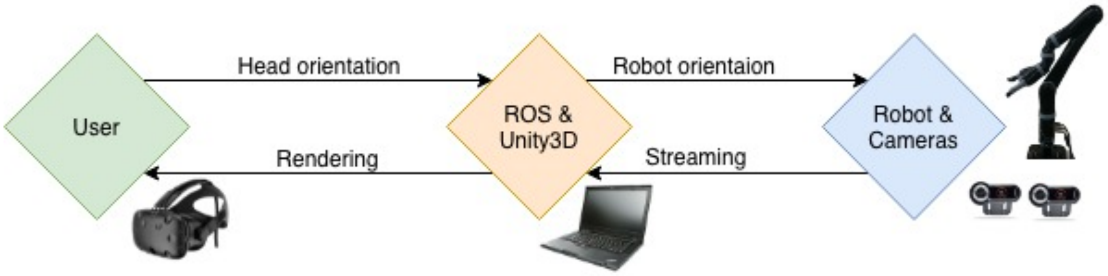


Figure 2.1 First draft of the whole system.

2.1 HTC Vive

HTC Vive is a virtual reality headset developed by HTC and Valve Corporation, released in 2016. The headset is designed to utilize "room scale" technology to turn a room into 3D space via sensors which allows the user's movements to be tracked and processed in real time in a virtual room, then transferred to the virtual world allowing the user to navigate naturally, with the ability to walk and use motion tracked handheld controllers to manipulate objects, interact with precision, communicate and experience immersive environments [12].

There are two screens in HTC Vive, both are located in front of each eye. Resolution of these screens is 1200 by 1800 pixels. Because human eyes have over 60 frames per second reading bandwidth, HTC Vive has 90 frames per second rendering capacity, which is enough to create the “suspension of disbelief” effect.

HTC Vive has multiple sensors to give feedback about user location and position. A gyro sensor, accelerometer and laser-position sensor are read and combined creating a smooth and precise rotation to the accuracy of 1/10th of a degree, which increases the sense of reality. SteamVR base stations can be also used with HTC Vive, which enables location tracking [13].

As shown in Figure 2.2, each eye displays only half of display from different view and has its own view, therefore left eye is not able see right part of the image and right eye is not able to see left part of the image itself.

If we consider the cameras as eyes in Figure 2.3, we see that brain fuses these two images as one image.

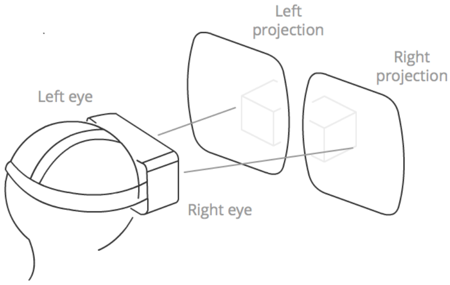


Figure 2.2. Virtual Reality Headset Left and Right Eye Projection

2.1.1 Stereo Vision

Stereo vision in human brain works by resolving the differences between two slightly different feeds from two eyes. To provide a real-life environment to an operator, we need to render two different images to different eyes, thus need to use multiple video cameras. These cameras also need to provide at least high definition image because HTC Vive requires that for optimal performance.

Stereo vision also got effected by what is the distance between the two images fed to two eyes. So Interpupillary distance (IPD) also plays a big role.

Figure 2.3 shows interpupillary distance and depth of an image from two individual cameras. Interpupillary distance (IPD) can be defined as the distance between the centres of pupils is between 50mm and 70 mm [Variation and extrema of human interpupillary distance]. We set the distance between the left and right cameras typically 65mm, depending on an adult's eyes.

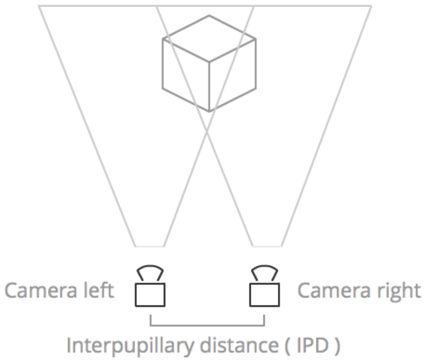


Figure 2.3. Interpupillary distance

2.2 Kinova Jaco2 Robotic Arm

Jaco2 arm is a safe and effective robot arm, which moves precisely and silently with unlimited rotation on each axis. It has six degrees of freedom and reaches all the positions we need. It also has gripper and three fingers. We don't use the fingers in our project because we want the gripper to move the robot arm only x, y and z-axis. It doesn't have a camera system itself. Thus, we created a stereo camera system attached to the gripper.

Jaco2 has an advanced application programming interface (API) which can be improved by programmers depending on their experiences. The API connects to Jaco2 via USB 2.0 connection and it allows the programmers to configure the parameters and integrate the Jaco2 to their own systems [14]. The API runs on Robot operating system (ROS), which is known as an operating platform for robotic applications.

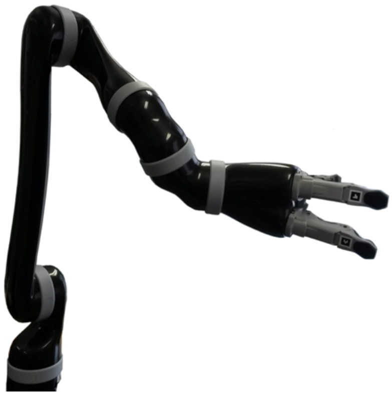


Figure 2.4. Kinova Jaco2 Robotic Arm

2.3 Robot Operating System

ROS is an open source operating system for robots, which provides hardware abstraction, low-level device control, interprocess message-passing and package management. It also has tools and libraries for software development over multiple computers [15].

The ROS framework provides support for a topic-based publish–subscribe model and a service-based request and reply model. ROS is organized by a root element, `roscore`. Every program is connected to the core as a node. ROS leverages the topic-based publish-subscribe model to provide named data pathways among distributed nodes. The service-based request and reply model is implemented as services; nodes within the ROS environment may provide functions, or services, that can be called through ROS protocols by any other node running in the ROS environment. A new node interacts with the ROS framework by subscribing to and processing existing topics and/or publishing new data over topics [16].

2.4 Unity3D

Unity is a cross-platform game engine developed by Unity Technologies [17]. While the mainstream knowledge about Unity3D is that it's a cross-platform game engine, in reality, it's also an incredibly powerful tool for non-game multimedia applications and simulations. In this project Unity3D is a perfect fit because of its vast library support (like OpenVR which is also used in this project), it has extensive video and streaming support, and the expressiveness of its architecture.

2.4.1 Visuals

`WebCamTexture` class is used in Unity3D to provide live video stream from two webcams on Jaco2 to the operator’s eyes. One `WebCamTexture` per each eye is created. Eyes must only see one of the images, so the other one is culled via `CullingMask`. This provides stereoscopic real-life like vision to the user on the head-mounted display. Then, the orientation of head as yaw, roll and pitch information gathered from HTC Vive. This information will be sent to an HMD topic on ROS server.

The code below subscribes to the HMD topic of the node.

```
IEnumerator Connect()
    {
        Debug.Log("Trying to connect.");
        w = new WebSocket(new System.Uri(server_uri));
        yield return StartCoroutine(w.Connect());
        Debug.Log("Connection established.");


        while (true)
        {
            if (bufferList.Count > 0)
            {
                string message = bufferList[0];
                bufferList.RemoveAt(0);
                w.SendString(message);

            }

            string reply = w.RecvString();
            if (reply != null)
            {
                if(showReceivedData)
                   Debug.Log("Received: " + reply);
            }
            if (w.error != null)
            {
                Debug.LogError("Error: " + w.error);
                isConnected = false;
                break;
            }
            yield return 0;
        }
        Debug.Log("Connection closed.");
        w.Close();
    }
```

2.4.2 Tracking

The HTC Vice comes with a built-in inertial measurement unit (IMU) consisting of a 3-axis gyroscope and a 3-axis accelerometer. With this IMU the user's head rotation can be estimated with an accuracy of 1/10th of a degree. In addition, to estimate the gaze position, a Pupil-Lab eye tracker hardware was mounted into the HTC Vive glasses.

With OpenVR Software Development Kit on Unity, HTC Vive VR can be accessed and tracked. SteamVR SDK provides us the rotation values of HTC Vive Headset. Depends on the user's head movement, the angle of movement from x, y and z-axis can be obtained. In Figure 2.5, orientation around y-axis as yaw, orientation around x-axis as pitch and orientation around z-axis as roll is shown.

The piece of code below shows how we can reach the x, y and z angles depending on if the device controller or headset.

```
if (Controller.triggerPressed)

    {
        orientaionWithController = true;
        hmdActive = false;
        Debug.Log("Hmd inactive!");
        pitch = Controller.transform.rotation.eulerAngles.x;
        yaw = Controller.transform.rotation.eulerAngles.y;
        roll = Controller.transform.rotation.eulerAngles.z;
    }
    else
        orientaionWithController = false;


    if(hmdActive)
    {

        pitch = headTransform.rotation.eulerAngles.x;
        yaw = headTransform.rotation.eulerAngles.y;
        roll = headTransform.rotation.eulerAngles.z;
    }
```

Roll, pitch, and yaw angles are defined as follows:

- Roll φ: angle around the Z axis while tilting the head in the frontal/coronal plane.
- Pitch θ: angle around Y axis while tilting the head in the median plane.
- Yaw ψ: angle around X axis while turning the head in the transverse/axial plane.

These three angles are received with SteamVR and then transferred to the ROS machine via the WebSocket. On the ROS machine this information is read from the WebSocket and published on a ROS topic. Similarly, for eye tracking we used Pupil-Labs eye tracking SDK for Unity to receive the gaze position from VR Eye Tracker. We received vectoral data both eyes via `PupilGazeTracker.Instance.GetEyeGaze2D (PupilGazeTracker.GazeSource.BothEyes)` method.

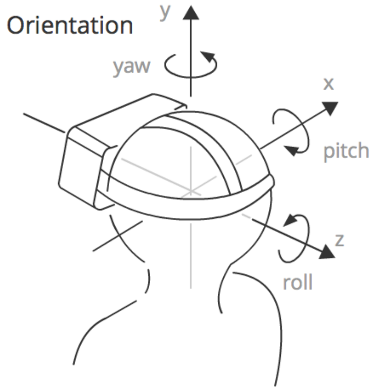


Figure 2.5. Orientation of headset

3 IMPLEMENTATION

This section summarizes a combination of ROS and Unity from different perspectives. During the setup phase, we have decided to go with a Windows machine to run Steam VR. In our trials, it wasn't possible to run Steam VR on Linux. But ROS required Linux environment, so we used a WebSocket connection between them and simply used our Windows machine as a rendering and input relay.

The Windows-based machine has Unity and SteamVR applications and HTC Vive is connected them. The Kinova Jaco2 Arm is connected to the Linux based machine via ROS application. The connection between Linux and Windows machine is via ROS bridge suite. Unity3D connects to the ROS bridge as a client via WebSocket library. The Unity3D ROS node we created requests data from ROS. It subscribes to the node which we mentioned in Section 2.3 (Figure 3.2).

3.1 Building the Environment

3.1.1 Connecting Computers

The system required one windows-installed and one Linux-installed computer. Between the computers, a network connection was set up. WebSocket library on Unity3D was created on Windows machine for the connection. On Linux machine, the connection was created via ROS Bridge. WebSocket on Unity sends data to the server on ROS machine via rosbridge.

3.1.1.1 Connection via Rosbridge

The rosbridge protocol which is a specification for sending JSON based commands to ROS was utilized to create connection between two computers via WebSocket [18]. The connection protocol for this project is shown below.

```
{ ‶op″:‶call service″, ‶id″:‶call service:/unity:1″, ‶service″:‶/unity″,
‶args″:{ ... } }
```

To install the rosbridge `‶sudo apt-get install ros-<rosdistro>-rosbridge- suite″` command was used. It was launched via `‶roslaunch rosbridge server rosbridge websocket.launch″` command. Launching the ROS bridge creates a WebSocket on port 9090 by default.

It is also possible to configure the port by setting the `~/port` param in ROS. An example launch file that will run rosbridge on port 8080 would look like:

```
<launch>

  <include file="$(find rosbridge_server)/launch/rosbridge_websocket.laun
ch" >

     <arg name="port" value="8080"/>

  </include>

</launch>
```

3.1.1.2 ROS Server

ROS Server requires to declare our node using `init node()` and then declare the service:

```
s = rospy.Service('unity', Rotations, handle rotations)
```

This declares a new service named unity with the Rotations service type. All requests are passed to handle rotations function. handle rotations is called with instances of `RotationsRequest` and returns instances of `RotationsResponse`.

3.1.1.3 Connection via WebSocket on Unity

Another WebSocket was used on Unity and created a connection between ROS and Unity utilizing rosbridge protocol which we talked above.

We used WebSocket for connection to achieve maximum portability and compatibility between two computers which were working on two different operating systems such as Windows and Linux.

3.1.2 Display

As we mentioned earlier, we aimed to provide a stereoscopic 3D view to the user. This required two webcams to create a stereo video stream which will be transmitted to the user. `WebCamTexture` on Unity3D reaches these USB-connected webcams. Two planes in front of stereoscopic camera setup were created in Unity3D, one plane per eye. Also, each camera has ability to see only one layer. This provides the stereoscopic vision required for depth perception.

Culling mask method of Unity platform was applied to create 3D view. First, we created a texture as `WebCamTexture`. Then, `WebCamTexture.devices[..].name` provided us the chosen camera. After it we rendered the texture as `this.GetComponent<Renderer>()`. Finally played the texture as `texture.Play()`.

```
        texture = new WebCamTexture();
        switch(this.eye)
        {
            case Eye.left:
                i = 0;
                this.texture.deviceName = WebCamTexture.devices[2].name;
                this.GetComponent<Renderer>().material.mainTexture = texture;
                texture.Play();
                Debug.Log("Camera Left: " + texture.deviceName);
                break;
            case Eye.right:
                i = 1;
                this.texture.deviceName = WebCamTexture.devices[1].name;
                this.GetComponent<Renderer>().material.mainTexture = texture;
                texture.Play();
                Debug.Log("Camera Right: " + texture.deviceName);
                break;
            default:
                throw new System.Exception("unexpected parameter: eye=" +
this.eye);
        }
```

For stereoscopic display, all we needed was two webcams. We preferred Logitech webcams as a brand because they are very compatible with Windows OS. In our earlier experiment, we also used PS3 cameras. However, we needed to import libraries of PS3 cameras because they are not compatible with Windows OS. This has been caused by latency and error at our experiment. Because of that reason, we switched them to the more compatible camera such as Logitech QuickCam Pro 9000.

3.1.3 Robotic Movement

As mentioned, task within ROS are usually implemented as a network of various nodes. One of these nodes called *jaco2_arm_driver* publishes the Jaco2's sensor data e.g. joint position and joint velocities. Besides, this node provides several control interfaces to command movements, e.g. there is a subscriber for joint velocities. This subscriber listens to a topic on which a vector of 6 velocities was published. This node than quarantines that the joints of the robot move by exact this velocity.

3.1.3.1 PID Controller

PID Controllers are simply feedback loops for systems which try to keep a system at the desired state. A proportional–integral–derivative controller (PID controller) does this by reading values from a system, then comparing the values read to a predefined target that it needs to match and then adjust the system input accordingly to get closer to those predefined targets. There are many different applications of PID controllers used from heating systems to irrigation, valves in space vehicles to some advanced aquarium heaters [19]. For example, in [20], a PID controller was used for an automated truck-trailer vehicle to pick up payloads.

In [21], a PID controller for a quadrotor drone was explained.

In [22], a torque controller with joint-space PID feedback was discussed for quadruped locomotion.

The linear-feedback PID controller continuously used the error e(t) between the desired set point x_d and the measured process variable $x(t)$ to correct the current state until the desired set point was reached. The control update $u(t)$ was given by

$$u(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{de(t)}{dt}, \qquad (1)$$

where K_p, K_i and K_d are proportional, integral, and derivative control gains determining the performance of the control law and giving the controller its name [23].

We tried the following method to find value of all gains. First of all, we set all gains to 0 for tuning. Second, we increased the proportional gain (K_p) until the response to a disturbance gets stable oscillations. Third, we increased the derivative gain (K_d) until the oscillations disappear. Afterwards, we repeated second and third steps until increasing the derivative gain does not stop the oscillations. Behind, we set proportional and derivative gains to the last stable values. At the end, we increased the integral gain until we get the desired setpoint with the number of oscillations.

Our goal is the synchronize the operator’s head movement with the robot arm. Consequently, a new node is written subscribing to the HTC Vive roll-pitch-yaw angles and optionally to the gaze position. Since the orientation of the operator's head and the robot can be different, we do not intend to synchronize the absolute orientation but the relative change in orientation. Therefore, the difference Δ to the last received angle was calculated:

$$\Delta\phi_k = \phi_k - \phi_{k-1} \qquad (2)$$

$$\Delta\theta_k = \theta_k - \theta_{k-1} \qquad (3)$$

$$\Delta\psi_k = \psi_k - \psi_{k-1} \qquad (4)$$

These angle differences are then used for a PID controller to control the joint velocities $\dot{q}_i$ of the Jaco2.

Let us assume without the loss of generality that joint 1 is the yaw-joint, joint 2 the pitch-joint, and joint 6 the roll-joint. Our time discrete control law then becomes

$$\dot{q}_1 = K_{p,1}e_{\Delta\phi_k} + K_{i,1}T\sum_{i=0}^{k} e_{\Delta\phi_k} + K_{d,1}\frac{e_{\Delta\phi_k} - e_{\Delta\phi_{k-1}}}{T} \qquad (5)$$

$$\dot{q}_2 = K_{p,2}e_{\Delta\theta_k} + K_{i,2}T\sum_{i=0}^{k} e_{\Delta\theta_k} + K_{d,2}\frac{e_{\Delta\theta_k} - e_{\Delta\theta_{k-1}}}{T} \qquad (6)$$

$$\dot{q}_6 = K_{p,6}e_{\Delta\psi_k} + K_{i,6}T\sum_{i=0}^{k} e_{\Delta\psi_k} + K_{d,6}\frac{e_{\Delta\phi_k} - e_{\Delta\psi_{k-1}}}{T}, \qquad (7)$$

where T is the update time, $e_{\Delta x_k} = \Delta x_k - \Delta x_{k-1}, x \in \phi, \theta, \psi$ and $K_{\lambda,j}$ with $\lambda \in p, i,$ d are the different gain values.

Depending of the true dynamics of the manipulator these gains can be different for each joint.

Moreover, if we want different reactions to the three rotations of the head, we can adjust these parameters accordingly, e.g. it can be desired that a strong yaw/axial rotation of the head affects the change of view just slightly.

Similarly, for the gaze position one can introduce two gaze angles if the camera matrix for the eye tracking is known (center c_x and c_y and focal length f_x) and f_y. Then we can calculate our gaze angles as:

$$\varphi_x = atan2(x - c_x, f_x) \qquad (8)$$

$$\varphi_y = atan2(y - c_y, f_y) \tag{9}$$

As an alternative, we can assume that these angles are small and Eq. (8) and (9) are nearly linear.

Thus, we can use the gaze position directly in a PID control law as the above for 2 new joints.

3.1.3.2 Future Work

As an alternative approach, we can use inverse kinematics and a differential motion to update the robot arms configuration, since the update of the HTC Vive's orientation is a rotation in the Cartesian space and the update rates are quite high. For this approach, the inverse kinematics solver Trac-Ik discussed in [24] was used.

First, the robot orientation is updated by difference between last and current HTC Vive orientation. Second, for this updated orientation the joint angles are calculated using inverse kinematics.

Unfortunately, the solver Trac-Ik [24] shifts the configuration to often from elbow-up to elbow-down. Therefore, the camera view changes to often and this method cannot be used.

As an improvement, in the Trac-Ik solver a cost term could be introduced so that configurations nearby have lower cost and therefore optimization will return those configurations with lower cost. Another possible improvement could be to use the Jacobian to calculate the joint velocities from the HTC Vive rotational velocities. Both approaches, however, are beyond the scope of this paper.

3.1.4 Start-Stop the Robot

As a safety feature, the Jaco2 API provides the admittance controller discussed in [25]. Thus, the robot will react to contact forces. This allows minimizing forces if the Jaco2 collides with itself or the environment. Besides, a second user is able to move the Jaco2 by simply pushing the joints into a desired configuration. This can happen simultaneously to the synchronization of head and robot.

To start synchronization of the user's motion and the robot's motion, a joystick is used a simple device to start and stop the robot. This is required, since the users view is blocked by the HTC Vive.

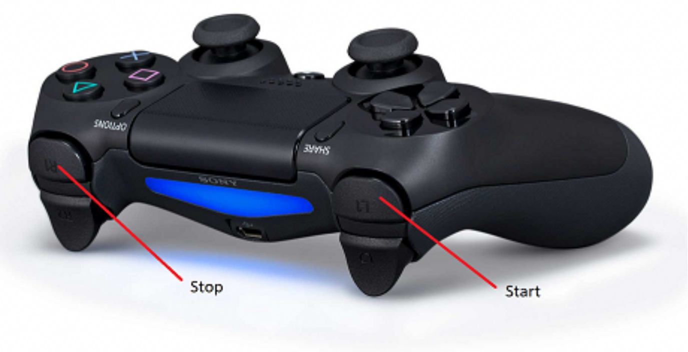


Figure 3.1. User interface for start and stop the robot.

Because of blocking the user’s vision by HTC Vive, the robot was needed to control by another user. A joystick is used as a simple device to start and stop the robot. The robot was stopped by me when the robot lost the control because of the user’s extreme head movement. After the initial position set up again, the robot was started. The control node was created for ROS system. When pressed the stop button, the joystick stop message is published to the node. The command stops the robot. Then it repeats with the start command when start button pressed. R1 button is used to active and R2 button is used to inactivate the robot. Figure 3.1 shows the position of the buttons.

3.1.5 Overview

The final section summarizes how the whole system works. The overall system in Figure 3.2 details the components. It is shown that the application integrates Unity3D engine (handling and rendering HTC Vive display), the SteamVR (receiving orientation data of HTC Vive glasses) and the WebSocket (transferring data to the ROS machine) and the ROS (routing the headset signal to the robot) to a 3D visualization display.

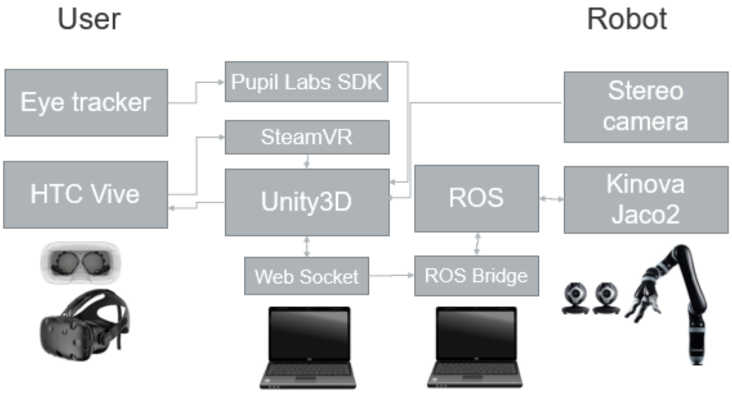


Figure 3.2. System overview displaying the application components (Unity3D, SteamVR, Pupil Labs SDK and ROS).

At the beginning of the project, we installed all the systems to one Windows-based computer. However, we realized that the robot operating system is not designed for Windows OS. After that, we decided to run all the system on a Linux-based machine. Nevertheless, SteamVR SDK was not properly working on a Linux-based machine at the earlier phase of this project.

We used two laptops who have two different operating systems and created connection via WebSocket to provide a smooth connection between the computers.

4 EXPERIMENT

In order to answer the question “Does our proof of concept work well?”, we conducted an experiment with the combination of the HTC Vive virtual reality set and a robot teleoperation platform.

For the experimental task, a board having three nails was used (See Figure 4.1). Two nuts were placed over one of the nails. The users were responsible to do three same tasks with different systems. The task was to remove the nuts from a nail to another nail. In the first experiment, the users had to move nails without using any headset or controller, so the user simply moves nuts from nail to nail in a normal manner. The second experiment was to remove nails using HTC Vive headset in handsfree mode. The third was by using the HTC Vive headset and the handheld controller. The second and third task was not always with the same order for the user in order to remove learning bias that might occur from starting always with the same tools.

We measure the completion time of the experiments to show the time differences between the handheld controller used and hands-free experiments. As the first experiment, we let the user complete the task without using any tools. Hereby, we aimed the user to understand the task better. At the second task, we measure the time of hands-free system compare to the handheld controller system to state which system is the fastest.

After completion of the tasks, we asked the users to score NASA Task Load Index (TLX) which is a multi-dimensional subjective workload rating technique [Gaw00] was used to grade the experiment. With NASA Task Load Index, we had one of the subjective feedbacks from the users. Another subjective feedback was provided by a questionnaire which has 15 questions related to the users’ performance on the tasks.

Before starting the experiment, the users were asked to get familiar with the whole experiment but with only one nut. With this practice, they learned how to do their task before they started to do a real experiment.

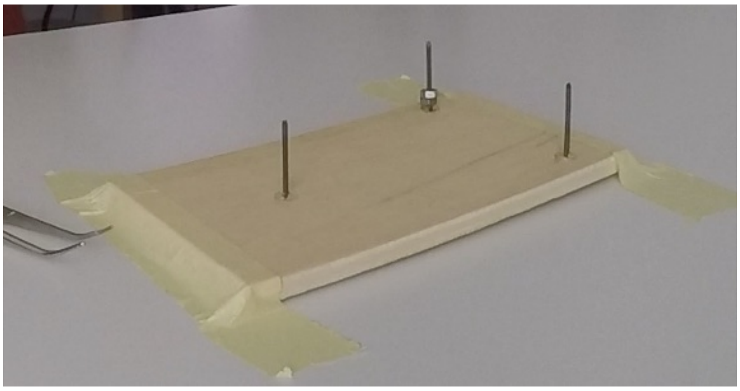


Figure 4.1. 3 nails and 2 screw placed at first position.

4.1 Experimental setup

Figure 4.2 shows the experimental setup. Camera-mounted robotic arm looks at the board from the top. The user sits in front of the table.

Three tasks were given to the users.

- Task 1: Removing the nuts from first nail to second one, from second one to third one and then from third to first one back. In this part it was not allowed to wear VR glasses or using the controller.
- Task 2: Repeating first task by wearing VR glasses (See Figure 4.2).
- Task 3: Repeating first task by controller and also wearing VR glasses (See Figure 4.3).

In all tasks we measured the time between beginning and end of the tasks. Every user always started the Task 1. After first task, Task 2 or Task 3 was randomly started as a second task. Then, the last task was applied as a third task.

On Task 2, the beginner position was arranged to see only two of the nails as on Figure. The user was needed to move the head to see the other nail. Depend on the head movement, camera mounted robotic arm moves that direction.

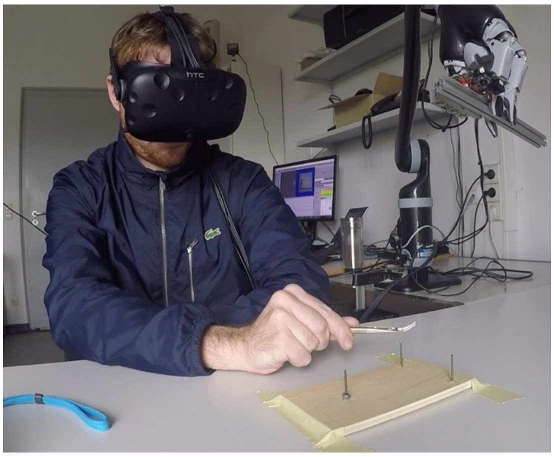


Figure 4.2. A User experiment with Headset

On Task 3, the beginner position was also set up displaying of only two of the nails. The user did the task by wearing VR glasses. VR glasses became inactive by pressing the trigger button of HTC Vive Controller. On this stage of the task, the user needed to remove the hands from the board to control the robot. The robot directed by holding the trigger button and rotating the controller. The user needed to remove the hands when the robot was needed to be directed by the user (see Figure 4.3).

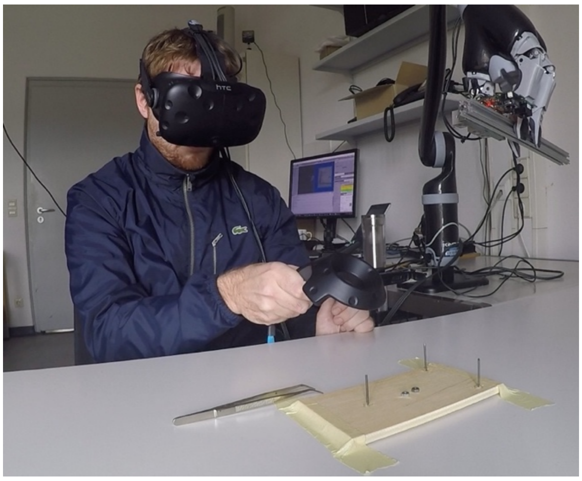
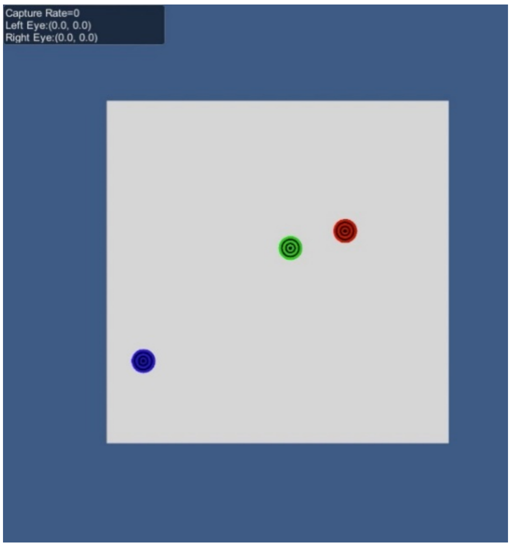


Figure 4.3. A User Experiment with Controller

4.2 Gaze Control

Pupil Labs Eye Tracker was used for Virtual Reality glasses. Via gaze detection, we were able to control the contrast of the display. When the user looks at the top right corner for 3 seconds, the contrast of the display turns maximum. Every minute is automatically changed to normal. So, the user has to increase the contrast again. At the beginning of the experiment, pupil tracker needs calibration for each user. The calibration of the pupil tracker is provided by Pupils Labs software. Pupil labs software has assets for Unity3D. We made calibration for VR eye tracker using the Unity platform. 9 calibration points appear one by one. Then, the user follows the point by his/her eyes. After calibration, left eye blue, right eye red, and both eyes are shown as green (Figure 4.4).



```
Capture Rate=0
Left Eye:(0.0, 0.0)
Right Eye:(0.0, 0.0)
```

Figure 4.4. Calibration for eye tracker

The users are not able to see the point from their screen not to let block the screen. We also keep the data coordinate points of both eyes. It can be used as future work.

In this experiment, we only change the contrast of the screen. However, for future works, it is feasible to add new futures such as settings and zooming.

4.3 Participants

12 people joined to the experiment. From which 3 female and 9 males had the experiment. All participants were from the computer science department including students, research assistant and Ph.D. candidate.

Before the experiment starts, the user was also asked information including vision status, personal experience on the robotic domain, eye tracking, VR or teleoperated systems.

6 users who had an eye condition such as myopia and hypermetropia were wearing glasses. This condition didn’t affect them to wear a headset.

5 participants had experience on the robotic domain as one user 2,5 years, one user 0,5 year, one 5 years, one 10 years and one user 2 years.

On eye tracking domain 8 participants had experience as three users 2 years, one user 5 years, one user 4 years, one user 3 years, two user 0,5 year.

Only 2 participants (one user 1 year, another user a few times in 2 years) had experience on VR domain.

In our experiment, none of the users had experience in teleoperated control systems.

4.4 User Study

Besides the performance measure, the user is given the opportunity feedback in a questionnaire. Depends on NASA task load index, the users graded the experiment from very low to very high on mental, physical, temporal demand, performance, effort, and frustration. They also graded 15 questions head and eye tracking system compare to the hand controller.

Before the experiment starts, the user is also asked information including vision status, experience on a robotic domain, eye tracking, VR or teleoperated controls systems.

In total 12 participants tested the system including students and researchers of the group. NASA Load Index questions the users’ grade on the scale from (1) very low to (20) very high were:

1. Mental Demand: How mentally demanding was the task?
2. Physical Demand: How physically demanding was the task?
3. Temporal Demand: How hurried or rushed was the pace of the task?
4. Performance: How successful were you in accomplishing what you were asked to do?

5. Effort: How hard did you have to work to accomplish your level of performance?
6. Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?

Another questionnaire was graded by the users to compare head and eye tracking system to hand console from (1) to (7).

1. It does everything I would expect.
2. It requires the fewest steps possible to accomplish what I want to do.
3. I can recover from mistakes quickly and easily.
4. It is easy to use.
5. It is simple to use.
6. It is user friendly.
7. Using it is effortless.
8. I can use it without written instructions.
9. I don't notice any inconsistencies as I use it.
10. I easily remember how to use it.
11. I learned to use it quickly and is easy to learn. 11. I easily remember how to use it.
12. I quickly became skilful with it.
13. I found the system unnecessarily complex.
14. I found the various functions in this system were well integrated.
15. I found the system very cumbersome to use.

5 RESULTS

We first present the quantitative measurement of the experiment, then we show the qualitative measurement and feedbacks from the users.

5.1 Measure

The measures taken were the duration of three tasks. Table 5.1 shows the average time of the tasks.

The completion of the task averagely takes 38 seconds without using any tools and 2:49 min with a headset. However, with the controller, it takes an average of 3:43 min. The major difficulty was to see the platform from an unnatural angle. The camera displays the task platform from the top view. However, the user has to look from the front. Another difficulty is the position of the robot. The fixed point of the robot stays behind of the user and it moves one-meter diameter from the fixed point. Accordingly, during an extreme head movement of the user, the robots come to the user's face (Figure 5.1). Another challenge is when the nut is out of the viewport, we have to bring back the nut into the middle of the platform.

The fastest time of 55 secs was achieved by an engineering student who had experience virtual reality and robotic domain. Typically, the group without virtual reality and robotics experience directed the robot very carefully and spent a long time on Task 1 and Task 2.

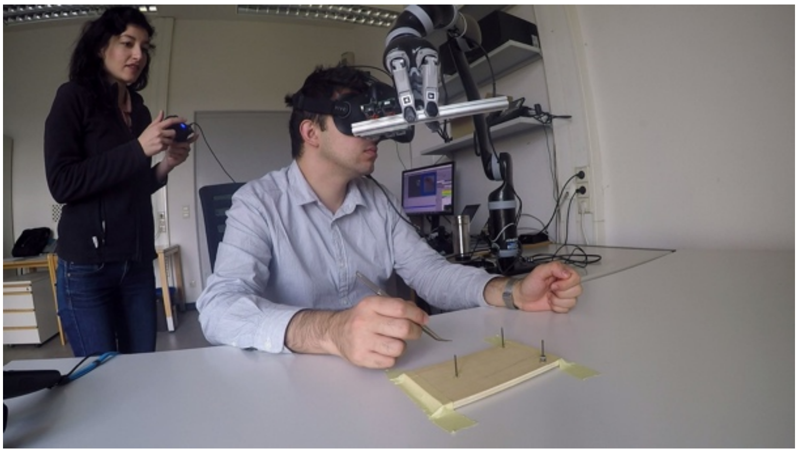


Figure 5.1. A moment of sudden head movement of the user. At this moment, the robot moves through the user's face. A joystick was used to stop the robot immediately.

We got better results from virtual reality compared to hand controller. The only exception was one participant, who completed the task 2.40 secs earlier than Task 1. She unexpectedly had difficulties her second try. It might be because of her tired eyes, which had a rare disease. Another user completed both tasks at the same time. Only one user was not able to complete the task because she had motion sickness from virtual reality glasses. Other 12 participants were able to complete all 3 tasks. All users were needed help to re-position the camera or bring the screws back to the stage during experiment.

Table 5.1. Average time of user study.

Average	Task 1	Task 2	Task 3
Total	00:38	02:39	03:43
Fast Group	00:25	01:29	02:40
Slow Group	00:51	03:50	04:45

The data from two participants who had motion sickness problems were omitted for this analysis. On average, participants finished the baseline condition (a) within 39 seconds (std. 20), the handheld controller condition (b) within 209 second (std. 75), and the hands-free condition (c) with automatic repositioning system within 128 second (std. 56). Table 5.2 shows the task completion time for each participant.

Table 5.2. Task completion time with total duration in second for each interaction condition.

Participants ID	1	2	3	4	5	6	7	8	8	10
Baseline (a)	90	46	26	22	45	29	36	39	24	30
Handheld controller (b)	146	222	228	287	186	136	351	126	145	260
Hands-free (c)	112	222	71	94	124	55	185	166	77	174

Typically, participants had less errors –when a nut falls out of the board– when preforming in hands-free (c) condition (avg. 0.4, std. 0.5) compared to handheld controller (B) condition (avg. 1.1, std. 0.9). Based on NASA task load index, the majority of the participants noted that the proposed automatic repositioning system required high mental demand (Table 5.3).

Table 5.3. Average NASA task load index for all participants in a scale between 1 to 20.

Index Demand	Mental	Physical	Temporal	Performance	Effort	Frustration
Average	11.8	8.9	4.5	15.2	10.2	7.8
Std.	5.9	6.2	3.6	3.7	5.5	4.9

On average, participants agreed that the task did not require physical effort and did create time pressures for them. Participants could easily operate our system, but not completely effortless. They mainly were frustrated from video quality of our camera and HTC Vive display.

5.2 Subjective Feedback I

NASA Task Load Index (TLX) which is a multi-dimensional subjective work- load rating technique [26] was used to grade the experiment.

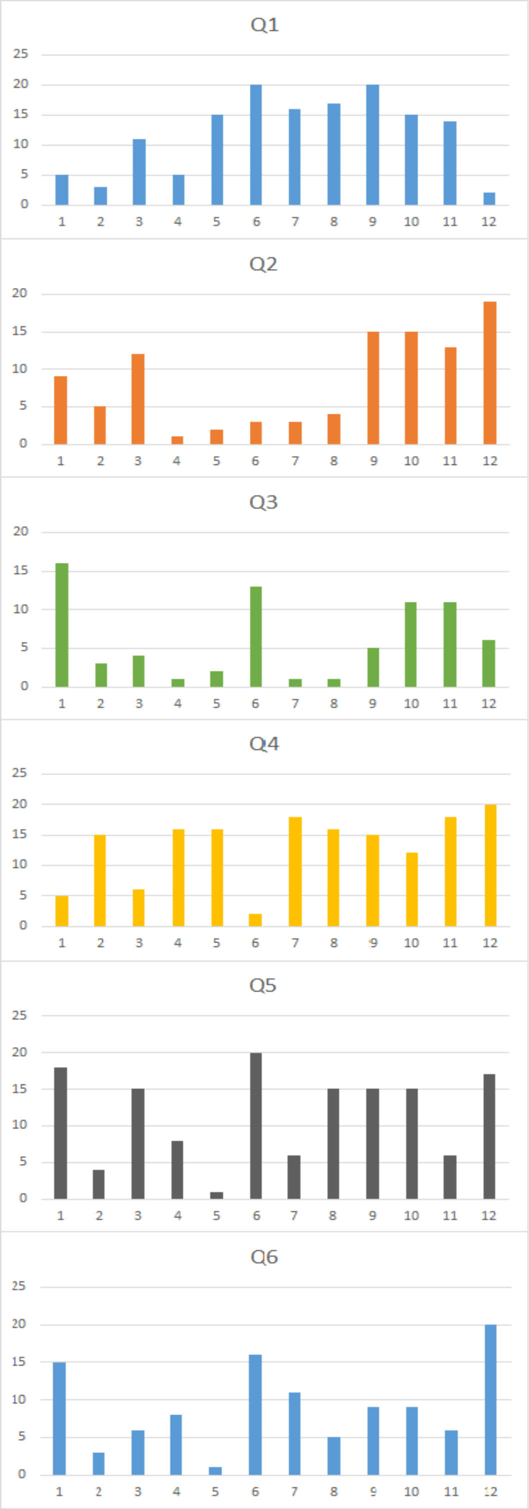
NASA Load Index questions the users' grade on the scale from (1) very low to (20) very high were:

1. Mental Demand
2. Physical Demand
3. Temporal Demand
4. Performance
5. Effort
6. Frustration

The evaluation is illustrated in Table 5.4 by the histograms of the participants' answers.

First, we asked users "How mentally demanding was the task?" (Q1) to determine how much mental and perceptual activity was required during experiment such as thinking, deciding, calculating, looking, remembering or searching. 8 participants over 12 claimed that the experiment required high mental demand.

Table 5.4. Subjective evaluation of the participants. Histograms showing the votes for the questions Q1 to Q6 of the questionnaire of NASA Task Load Index



As next question, “How physically demanding was the task?” (Q2), the majority agreed that the task didn’t required much physical effort such as pushing, pulling. Some participants agreed on directing the robot by controller didn't also required much physical effort. One participant mentioned that she required more effort because VR glasses were heavy.

About time pressure question, "How hurried or rushed was the pace of the task?" (Q3), most of the participants agreed that the task didn't make pressure on them about time.

"How successful were you in accomplishing what you were asked to do?" (Q4) was answered as high performance in accomplishing the task for majority of the participants.

"How hard did you have to work to accomplish your level of performance?" was asked to measure experienced difficulty. The answer of this question was quite mixed. Half of the participant answered that they easily got used to the system. According to other half, they had difficulties either using controller or VR glasses.

"How insecure, discouraged, irritated, stressed, and annoyed were you?", scored low by all the participants.

5.3 Subjective Feedback II

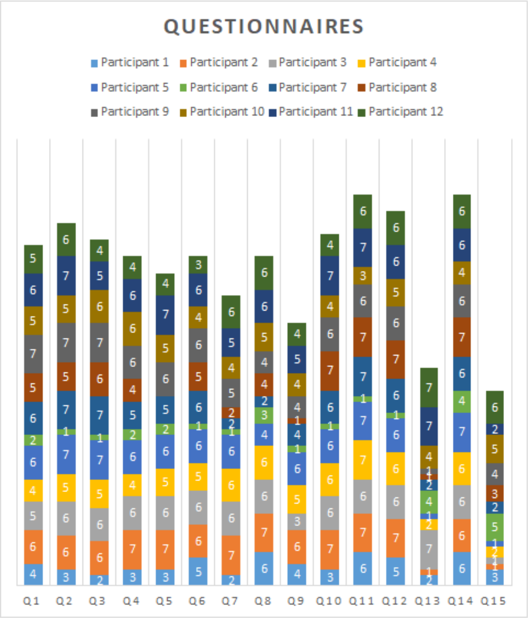
We decided to create another questionnaire to obtain more information about our test users experiences. It's graded from 1 to 7, with 15 questions about comparison between hands-free and handheld controller system. The answers of the questionnaire are shown on Table 5.5.

When we look at results of the comparison between handheld controller and hands-free systems, we see that rotating the camera-mounted robot by head-mounted display is easier than handling by the handheld controller. Answers of the questions 2,4,5,6 and 10 tells that the users easily learnt how to orientate head-mounted display. When we ask about integration of whole system in last three questions, we can claim that the system was not found complex by the participants.

The highest average scored question was the question about system integration, "I found the various functions in this system were well integrated."(Q14).

As we said in Chapter 5 titled Experiment, all participant trained the whole task to get used the experiment before they started it. Even though this experimental part helped 11 participants to understand better the real tasks, 1 participant had more difficulties on real tasks than the experimental task because the user's eyes got tired by time.

Table 5.5. The showing the votes from 1 to 7 for the questions Q1 to Q15 of the questionnaires. Each participant is shown in different colour and their votes in it.



If we compare both questionnaires, it visible that hands-free based system requires quite effort. However, in NASA Task Load, 9 participants over 12, finds that their performance is satisfying.

6 DISCUSSION AND OUTLOOK

In this chapter, the results that were depicted in chapter 5 will be discussed in detail. 12 people from the computer science department with different backgrounds combination of students, teaching assistants and Ph.D. candidates have participated in the experiment. In this experiment, 25 % of the participants were females and 75% of the participants were males. As explained in chapter 4, the experiments were categorized into three different tasks. Before the experiments, the users were asked their experiences about a robotic domain, eye tracking, VR or teleoperated systems and in addition to this information their vision statuses were investigated. 6th of the participants had problems with their vision status like myopia and hypermetropia, which did not affect the results of the experiment because of the headset. Depending on our experiments, all of the participants easily managed to control robot except one participant. This participant couldn't complete the task because of motion sickness. Rest of untrained participants completed the task successfully. Another participant had problems with her eyes, but the problem occurred due to a genetic disease that she has.

The experiment was based on three different tasks. The Nasa Task Index was used to assess the cognitive workload of these tasks with a total of 12 participants. The most difficult part of this experimental setup for the untrained participants was to view the application area via magnified view. However, this problem is a common issue in microsurgeries, and it can be overcome by practice.

Another obstacle for the experimental setup was motion sickness. Changing the camera view too often caused motion sickness and distraction for the user. This issue also can be overcome by repeating usage of the equipment by the user.

During the tests, because of how the robotic arm and the user was positioned, it was possible that the user might do certain movements which would rotate the robotic arms towards users face. This situation could be dangerous for the user. We used a joystick to stop the robot to prevent from this situation (See Figure 5.1). In real-world applications there will be enough spacing when the arm mounted to ceilings.

Our aim was to demonstrate the potential of hands-free robotic exoscope with automated camera viewpoint. In our experiment, we showed the differences between hands-free and handheld exoscope.

The experimental setup and measurements are explained in detail in the previous chapter. On average, the task completion time was 2:08 minutes for our hands-free solution. However, with a handheld controller, it took on average 3:29 minutes for participants to finish the task. We can easily see that hands-free solution nearly cut the time required for doing the task into half. The biggest reason for the time savings with the hands-free method is, with hands-free method user doesn't need to switch between tools. Moreover, participants were not trained enough about the handheld controller.

We used NASA Task Load Index for evaluating the workload and mental demands of the experimental setup. Therefore, after the application, the questionnaire was applied to participants to evaluate the experiment. The questionnaire contained six different questions on the scale from 1 (very low) to 20 (very high). The first question was about the scale of the mental demand requiring completing the task. Eight of the total participants claimed that the experiment required high mental demand with the scale of above than 10. This result was expected because we chose the participants with no experience with VR and the previous experiences with VR systems and the results of the first question were correlated against each other. The second question was about the physical demanding of the task and the majority of the participants agreed that the task did not require physical effort except the weight of the VR glasses. After investigating the mental and physical demand of the experiment, we also investigated the time pressure on the participant during the task and only 30% of the participant indicated that this task creates a time pressure on them. Question 4 was about the success rate of the accomplishing the task and majority of the participant rated this question above 10. Despite the success rate of the task, half of the participant also agreed on the difficulty of the task to complete either using the controller or VR glasses. The sixth question was about the frustration about using an experimental setup to complete the task and the average score of this question was lower than 10. This result indicated that the majority of the participants were open to using newly developed technology without hesitations and this result could be the reason for the user-friendly interface of the experimental setup. When all the results of the questionnaire were examined carefully, the developed experimental setup requires mental determination and takes more time than hands-free setup, but it can be solved by practice. Despite the inexperienced participants, the developed system provides a revolution solution in the future especially for microsurgery because it provides fewer physical demands with less time pressure and high success rate.

In the future, the system can be easily expanded with more sensors. The robot used in the system had limitations. Different robots might result in better precision with a faster completion time interval. Using integrated AR systems like Microsoft HoloLens is another option that can be used with our implementation. Integrated AR systems can provide better precision and special displays that reduce issues like dizziness. Also superimposing current environment similar to what HoloLens demonstration applications show, can create more interesting implementation options. Setting up the robot from a ceiling or a base that's far enough from the user, can create a better degree of freedom.

We emphasize that the experiment was limited to a total of 12 participants. It can be re-done on a more representative sample set (real surgeons who will use the device), and with more participants.

As we discussed on Future Work topic on Robotic Movement under Chapter 3 Implementation, there can be an alternative approach for such as using inverse kinematics and a different motion to update the robot arms configuration. Unfortunately, the solver Trac-Ik shifts the configuration often from elbow-up to elbow-down.

BIBLIOGRAPY

[1] Goertz, R.C., Grimson, J. H., Kohut, F.A., "Manipulator for slave robot". US Patent Patent 2,978,118, April 1961.

[2] Stone R.J., "Advanced human-system interfaces for telerobotics using virtual reality and telepresence technologies," in *Fifth International Conference on Advanced Robotics' Robots in Unstructured Environments*, IEEE, 1991, pp. 168-173.

[3] Madni, A., Chu, Y., Freedy, A., "Intelligent interface for remote supervision and control of underwater manipulation," in *Proceedings OCEANS'83*, IEEE, 1983, pp. 106-110.

[4] Stieber, M.F., Trudel, C.P., Hunter, D.G., "Robotic systems for the international space station," in *Proceedings of International Conference on Robotics and Automation}*, IEEE, 1997, pp. 3068--3073.

[5] Eivazi, S., Afkari, H., Bednarik, R., Leinonen, V., Tukiainen, M., Jääskeläinen, J. E., "Analysis of disruptive events and precarious situations caused by interaction with neurosurgical microscope," *Acta neurochirurgica,* vol. 157, no. 7, pp. 1147-1154, 2015.

[6] Afkari, H., Eivazi, S., Bednarik, R., Mäkelä, S., "The potentials for hands-free interaction in micro-neurosurgery," in *Proc. of 8th Nordic Conference on Human-Computer Interaction*, 2014, pp. 401-410.

[7] Pitskhelauri, D. I., Konovalov, A. N., Shekutev, G. A., Rojnin, N. B., Kachkov, I. A., Kopachev, D. N., "A novel device for hands-free positioning and adjustment of the surgical microscope," *Journal of neurosurgery,* vol. 121, no. 1, pp. 161-164, 2014.

[8] Charlier, J., Sourdille, P., Behague, M., Buquet, C., "Eye-controlled microscope for surgical applications. In Evolution of Microsurgery," pp. 154-158, 1991.

[9] Oertel, J. M., Burkhardt, B. W., "Vitom-3D for Exoscopic Neurosurgery: Initial Experience in Cranial and Spinal Procedures," *World neurosurgery,* vol. 105, pp. 153-162, 2017.

[10] Anonym, “ZEISS KINEVO 900,” [Online]. Available: https://www.zeiss.com/meditec/int/products/neurosurgery/surgical-microscopes/kinevo-900.html#robotics. [Accessed 14 January 2019].

[11] Lanfranco, A. R., Castellanos, A. E., Desai, J. P., Meyers, W. C., “Robotic surgery: a current perspective,” *nnals of surgery,* vol. 239, no. 1, p. 14, 2004.

[12] Anonym, “HTC Vive Wikipedia,” 2017. [Online]. Available: https://en.wikipedia.org/wiki/HTC_Vive. [Accessed 29 August 2017].

[13] Anonym, “HTC Vive,” 2017. [Online]. Available: https://www.vive.com. [Accessed 30 August 2017].

[14] Anonym, “Kinova Jaco Arm,” 30 August 2017. [Online]. Available: http://www.robotnik.eu/robotics-arms/kinova-jaco-arm/.

[15] Koubaa A., “ROS As a Service: Web Services for Robot Operating System,” *Journal of Software Engineering for Robotics,* vol. 1, p. 1, 2015.

[16] Anonym, “Robot Operating System,” [Online]. Available: http://www.ros.org;http://wiki.ros.org. [Accessed 30 August 2017].

[17] Anonym, “Unity,” [Online]. Available: https://en.wikipedia.org/wiki/Unity_(game_engine). [Accessed 30 August 2017].

[18] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A.Y., “ROS: an open-source Robot Operating System,” in *ICRA Workshop on Open Source Software*, 2009.

[19] Anonym, “PID,” [Online]. Available: https://en.wikipedia.org/wiki/PID_controller. [Accessed 14 November 2017].

[20] Buck, S., Hanten, R., Bohlmann, K. and Zell, A., “Multi-sensor Payload Detection and Acquisition for Truck-Trailer AGVs,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, Singapore, 2017.

[21] Salih, A.L., Moghavvemi, M., Mohamed, H.A. and Gaeid, K.S., “Modelling and PID controller design for a quadrotor unmanned air vehicle,” in *Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on*, 2010.

[22] Righetti, L., Buchli, J., Mistry, M., Kalakrishnan, M. and Schaal, S., “Optimal distribution of contact forces with inverse-dynamics control,” *The International Journal of Robotics Research,* pp. 1-19, 2013.

[23] Siciliano, B. and Khatib, O. eds., Springer handbook of robotics, Springer, 2008.

[24] Beeson, P. and Ames, B., “TRAC-IK: An open-source library for improved solving of generic inverse kinematics,” in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, 2015.

[25] Lecours, A., Mayer-St-Onge, B. and Gosselin, C., “Variable admittance control of a four-degree-of-freedom intelligent assist device,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012.

[26] Gawron V.J., Human performance measure handbook., 2000.

[27] Hudson, C., Carruth, D.W., Bethel, C.L., Lalejini, A. and Odom, B., “ANVEL-ROS: The integration of the robot operating system with a high-fidelity simulator,” in *Proceedings of Ground Vehicle Systems Engineering and Technology Symposium*, 2015, p. 378.

APPENDIX 1. Consent Form for the Experiment Participant

I, the undersigned, confirm that (please tick box as appropriate):

1.	I have read and understood the information about the project.	□
2.	I have been given the opportunity to ask questions about the project and my participation.	□
3.	I voluntarily agree to participate in the project.	□
4.	I understand I can withdraw at any time without giving reasons and that I will not be penalised for withdrawing nor will I be questioned on why I have withdrawn.	□
5.	The procedures regarding confidentiality have been clearly explained (e.g. use of names, pseudonyms, anonymisation of data, etc.) to me.	□
6.	If applicable, separate terms of consent for interviews, audio, video or other forms of data collection have been explained and provided to me.	□
7.	The use of the data in research, publications, sharing and archiving has been explained to me.	□
8.	I understand that other researchers will have access to this data only if they agree to preserve the confidentiality of the data and if they agree to the terms I have specified in this form.	□
9.	Select only **one** of the following: • I would like my name used and understand what I have said or written as part of this study will be used in reports, publications and other research outputs so that anything I have contributed to this project can be recognised. • I do not want my name used in this project.	□ □
10.	I, along with the Researcher, agree to sign and date this informed consent form.	□

Participant:

________________________ ___________________________ ________________

Name of Participant Signature Date

Researcher:

________________________ ___________________________ ________________

Name of Researcher Signature Date

APPENDIX 2. Users' details form for the experiment

Virtual reality teleoperated control robot using eye and head tracking

Participant ID

Date

First Name

Last Name

Gender

Age

Occupation

Vision status (e.g. normal, glass, contact lens, special eye illness)

Experience in robotic domain (what kind of, years)

Experience in eye tracking domain (what kind of, years)

Experience in VR domain (what kind of, years)

Experience with teleoperated control systems (what kind of, years)

By filling this questionnaire, I confirm that I voluntarily agree to participate in the experiment, I have understood the information about the experiment and that the collected data can be used anonymously. Moreover, I can withdraw from the study at any moment, I understood my identity won't be exposed in any eventual publication.

Place, date, and Signature

APPENDIX 3. Questionnaire

Questionnaires

Ranke between 1 to 7. Head and eye tracking system compare to hand console

- It does everything I would expect
- It requires the fewest steps possible to accomplish what I want to do
- I can recover from mistakes quickly and easily.
- It is easy to use
- It is simple to use
- It is user friendly
- Using it is effortless
- I can use it without written instructions
- I don't notice any inconsistencies as I use it
- I learned to use it quickly and is easy to learn
- I easily remember how to use it
- I quickly became skillful with it
- I found the system unnecessarily complex
- I found the various functions in this system were well integrated
- I found the system very cumbersome to use

APPENDIX 4. NASA Task Load Index

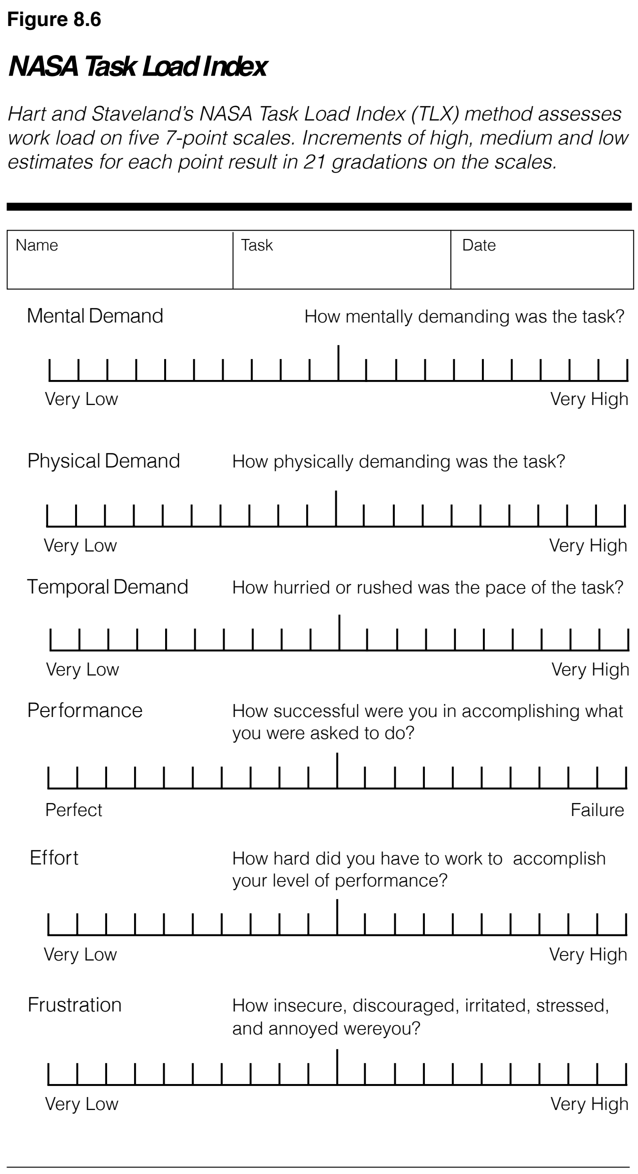


Figure 8.6

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low — Very High