Investigation of Annotation-assisted User Performance in Virtual Reality-based Remote Robot Control

Thanh Long Vu, Dac Dang Khoa Nguyen, Sheila Sutjipto, Dinh Tung Le, Gavin Paul
University of Technology Sydney: Robotics Institute (UTS:RI)
Thanh.L.Vu@student.uts.edu.au, {Khoa.Nguyen, Sheila.Sutjipto, DinhTung.Le, Gavin.Paul}@uts.edu.au

Abstract

This paper investigates the use of point cloud processing algorithms to provide annotations for robotic manipulation tasks completed remotely via Virtual Reality (VR). A VR-based system has been developed that receives and visualises processed data from real-time RGB-D camera feeds. A point cloud processing algorithm is introduced to annotate targets, and simulated experiments were conducted to validate the efficacy of the proposed algorithm. A real-world robot model has also been developed to provide realistic reactions and control feedback. The targets and the robot model are reconstructed in a VR environment and presented to users with different modalities. The modalities and available information are varied between experimental settings, and the associated task performance is recorded and analysed. The results accumulated from 288 experiments completed by 12 participants indicated that point cloud data is sufficient for task completion. Additional information, neither image stream nor preliminary processes presented as annotations, was found to have a significant impact on the completion time. However, the combination of image stream and colored point cloud data visualisation modalities was found to greatly enhance a user's performance accuracy, with the number of target centres missed being reduced by 25%.

1 INTRODUCTION

Extended Reality (XR) bridges the gap between digital and physical worlds, encompassing technologies such as Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR). When combining VR with sensors, such as LiDARs (Light Detection and Ranging) and depth cameras, it is possible to capture the state of real-world objects in real-time within a VR environment. Conversely, rather than using sensors to render the world in a virtual environment, users that are physically present, with the ability to clearly see the robot and environment, can leverage AR to view computer-generated perceptual information that enhances real-world objects. Thus, by providing a user with additional information in the form of overlaying annotations that are ordinarily unavailable in the real world, this work hypothesises that annotations improve the efficacy of remote manipulation tasks.

This paper presents the findings ascertained from the observation of users performing real-time remote control manipulation tasks in a VR environment. For the VR environment, streamed sensor data such as point clouds and images are used in conjunction with annotations in several settings in the user study. A point cloud processing algorithm is also presented, which detects multiple targets above an estimated ground level and generates three-dimensional (3D) bounding boxes as annotations. The annotations aim to help distinguish objects from the background, allowing them to be more noticeable in the scene. Thus, a study was designed for participants to remotely perform a robotic task in VR that required manipulating an end-effector to targets in the environment. The VR environment is used to render the 3D model of the robotic platform and its surrounding environment. By monitoring the manipulator's joint states, the model can be updated in real-time to reflect the actions of the real robot providing the participant with visual feedback. Relevant measurements of participant performance under various settings were recorded, including their completion time and precision score.

2 RELATED WORK

With further development in sensing technologies, VR facilitating human-robot interaction has become a prevalent research area. One particular goal is to understand how to render real-life scenes appropriately, and potentially control objects in the scene using a robotic manipu-

lator. VR is conducive to creating an intuitive and effective control interface for remote control robotics systems while also having the benefit of enhancing the user's perception through 3D data visualisation. However, achieving this requires investigating a suitable method of interaction with the VR environment and understanding how to render the sensor data within VR. Consequently, research continues to focus on quantifying human performance during manipulation tasks to obtain insights into the effect of different visualisation settings and control modalities [Whitney *et al.*, 2020][Le *et al.*, 2020].

Another facet of the VR experience that continues to be an active research area is on enhancing the interactions between the users and the VR scene. [Vélaz *et al.*, 2014] investigated various modes of interaction in VR, including mouse, haptic device, motion capture devices, etc. A custom control interface, which consists of a 6DOF robot, an underwater LiDAR and cameras, has been created to remove underwater munitions [Gharaybeh *et al.*, 2019]. Another approach has leveraged motion-capture technologies to ascertain the pose of the human wearing a glove, which provides information regarding joint positions of the fingers and enables intuitive fine control of a virtual hand[Kumar and Todorov, 2015].

The aforementioned improvements in visual and haptic feedback, and methods for intuitive interactions have enabled the exploration of VR for training platforms. The effectiveness of such training methods has been explored for the control of manipulators [Pérez *et al.*, 2019] and human-robot collaborative assembly processes [Rückert *et al.*, 2018]. Consequently, as training platforms become more prevalent, systematic approaches for adopting VR become necessary to ensure the efficacy of such systems. This requires the evaluation of input and output devices and their suitability for the desired task. In particular, these evaluations have been considered in the context of manufacturing when interacting with collaborative robots [Paul *et al.*, 2016] [Malik *et al.*, 2020].

AR leverages the user's presence in their environment and thus is focused on providing users with graphical overlays on existing objects in sight [Kipper and Rampolla, 2012]. A user study has compared the performance of AR and VR for manipulating an object in 9 degrees of freedom (translation, rotation, and scaling) [Krichenbauer *et al.*, 2017]. The study's results indicate that AR consistently outperforms VR for the object selection and transformation task based on completion time.

The overlaid information provided in AR can constitute documentation or guidance for the ongoing task [Gong *et al.*, 2019] [Yew *et al.*, 2017], or processed sensor data. This has been extended to include results from an object detection algorithm and utilising inverse kinematics and motion planning algorithms to generate a robotic manipulator's movement automatically[Gradmann *et al.*, 2018]. More sophisticated methods of interaction, such as using a gesture-based interface, enable operator movements to be converted into control commands in real-time [Lin *et al.*, 2016]. However, these types of systems lack the haptic feedback that is provided by external controllers, which has been shown to aid the user in following a pre-planned path [Ni *et al.*, 2017][Sutjipto *et al.*, 2020].

3 METHODOLOGY

3.1 System Overview

The system (Figure 1) consists of a custom-built robotic manipulator with remote joystick control and a calibrated camera system. The camera system includes two static Intel Realsense D435 RGB-D cameras with different viewpoints and partially overlapping fields of view. The cameras were mounted such that they achieved a high visual coverage and limited the data loss from occlusions caused by the manipulator during motion. Both color and depth images are captured and utilised to generate point clouds in real-time within the VR environment. Object detection and bounding box creation are performed using the combined data from the cameras. The bounding boxes are then transmitted to the VR environment for visualisation.

3.2 Custom Robotic Manipulator

The manipulator is a 5-DOF robotic arm, with a prismatic rail as the first joint and four revolute joints. The last three revolute joints are actuated by linear motors; the linear-to-angular control and feedback conversions are done automatically by the onboard controller. The end-effector is chisel-shaped with a small area of contact, and can be moved to various locations in a one-meter-squared work volume in front of the robot.

A kinematic and dynamic model was constructed from the CAD model and fed into a kinematic solver that utilises the MoveIt Motion Planning Framework [Coleman *et al.*, 2014]. The robot is equipped with joystick control allowing the user to send control commands via ROS, in either the joint space or the end-effector Cartesian space. All target locations are within the robot's reachable volume, and no joint states exhibited by the robot are near singularities or joint limits. The robot's joint state is displayed in real-time as the model reflects the robot's real state in the virtual environment. To ensure that the point cloud of the manipulator is accurately superimposed on its respective model in VR, the cameras are extrinsically calibrated with respect to the manipulator.

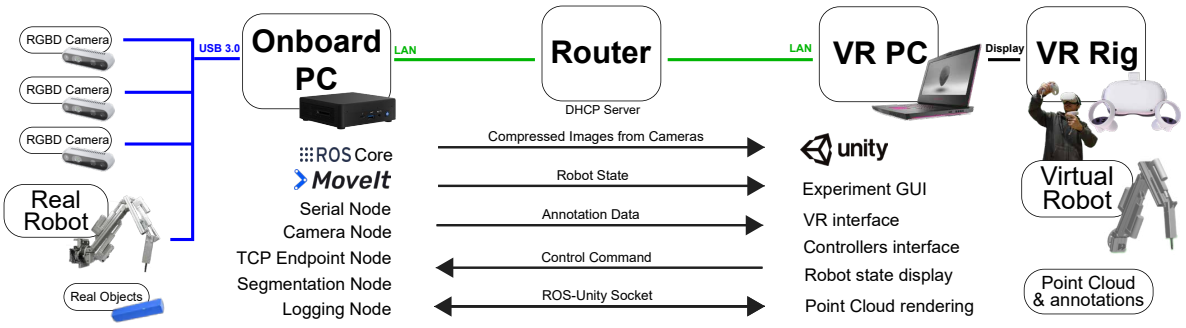


Figure 1: System Overview

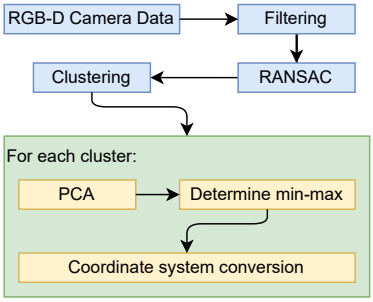


Figure 2: Point Cloud Processing Algorithm Overview

3.3 Point Cloud Render and Data Transmission

The pipeline used to transmit data from the sensors and render it within the VR environment is based on an existing framework presented in [Vu *et al.*, 2021]. The previous work describes a system capable of transferring point cloud data from a ROS machine to another machine that is running the VR environment. In [Vu *et al.*, 2021], the point cloud frame rate and resolution were variables that were systematically altered to obtain their effect on user performance as a trade-off exists between the frame rate and resolution.

3.4 Point Cloud Processing Algorithm

The point cloud processing pipeline, shown in Figure 2, combines multiple algorithms to produce bounding boxes that highlight objects above the scene's ground surface. The pipeline builds upon prior work [Vu *et al.*, 2019], in combination with additional algorithms from the open-source Point Cloud Library (PCL) [Rusu and Cousins, 2011]. The algorithm utilises the point cloud that is generated by multiple RGB-D cameras and can be received directly without additional processing steps. Concurrently, the point cloud rendered in the VR environment is generated through the fusion of color and depth images. Assumptions that were made about the scene in the field of view from the cameras are listed below:

- The multiple cameras must be extrinsically calibrated, and an overlap between the cameras' field of view must exist to implement the point cloud processing pipeline
- The surface on which the objects to be detected are placed must comprise a large portion of the combined field of view of the cameras.

The raw point cloud, P_r received directly from the RGB-D camera contains millions of points with noise that proportionally increases with the distance between objects and the camera lens. Due to the overwhelming amount of information and noise in P_r, preliminary processes need to be applied. Consequently, P_r is downsampled with the voxel grid filter from PCL and additionally, a distance threshold is applied to filter points that are outside of the required bounds. The distance filter aims to remove noisy data recorded from distanced objects [Vu *et al.*, 2019].

After filtering, the point cloud, P_f remains from P_r. Due to the assumptions listed above, the surface where objects are located will be the surface determined by RANSAC [Fischler and Bolles, 1981]. The list of points that belong to this surface will be denoted as S_R.

Before the clustering step, all the points that reside on and below S_R relative to the camera position, c must be removed. The method implemented to determine the point's relative position to a surface is shown in Algorithm 1. Let P_{temp} be the point cloud, P_f after removing P_R. The projection point of c onto S_R is denoted as p_j. The vector from the projected point to c is $\vec{v_{pc}}$. For every point, p_i in P_{temp}, vector, $\vec{v_{pp}}$ is the mapping from p_i to p_j. The relative position of point, p_i to a surface, S_R and c is determined based on the dot product dp of $\vec{v_{pc}}$ and $\vec{v_{pp}}$. If $dp < 0$, p_i and c are on the same side relative to S_R, or are on opposite sides if $dp > 0$. All points p_i that satisfy $dp < 0$ are stored in point cloud, P_{ss}

The Euclidean distance algorithm [Trevor *et al.*, 2013] is applied to P_{ss} to obtain a set that contains clusters of points. Each cluster, C_i represents an individual object on the surface, S_R. To generate a bounding box that

Algorithm 1 Two Points Relative Position to Surface

$P_{temp} = P_f \setminus P_R$
for $p_i \in P_{temp}$ **do**
  $dp = \vec{v_{pc}} \bullet \vec{v_{pp}}$
  if $dp < 0$ **then**
    p_i and c are on the same side relative to S_R
    $p_i \rightarrow P_{ss}$
  end if
  if $dp > 0$ **then**
    p_i and c are not on the same side relative to S_R
  end if
end for

encompasses each of the objects, a three-step process is implemented. The three steps involve the application of Principle Component Analysis (PCA), extracting the minimum and maximum values of the axes generated by PCA and converting the corners of the bounding box from the PCA coordinate frame to a camera's coordinate frame.

PCA is a multivariate statistical technique that extracts important information from an existing set of data and represents the set accordingly. This results in a set of new orthogonal variables labeled principal components [Abdi and Williams, 2010]. For each cluster, C_i, a three-dimensional coordinate system, O_i, can be derived from the eigenvectors obtained from the PCA algorithm. Each Cartesian point, $p_j^W \in C_i$ is converted from the camera coordinates, O_c, to point, p_j^O of coordinate O_i. The minimum and maximum values on each of the axes, $[x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}]$ among all points, p_j^O are utilised to define the set of corners of the bounding box, B_i^O, wrapping the cluster, C_i. The corner points are then converted from PCA coordinates, O_i, to the camera frame, O_c, for visualisation. The set of corner points in O_c coordinate system is denoted as B_i^W.

4 EXPERIMENT

The point cloud processing algorithm is evaluated through simulated experiments. For these experiments, a system of two RGB-D cameras was utilised to capture the scene and generate the point cloud input for the algorithm. Additionally, a human participation study utilising the proposed point cloud algorithm was designed to investigate the practicality of incorporating point cloud processing as an overlay and additional information in VR. The user study conducted involved a remote manipulation task that required participants to move the custom-built robotic manipulator's end-effector to specified targets. The criteria for assessing participant performance is based on the completion time and precision.

4.1 Point Cloud Processing Algorithm

A simulated Gazebo environment was created containing multiple simple-shaped objects and a system of two calibrated cameras. The experiment includes seven unique trials, and for each trial, the cameras are placed facing the objects from multiple viewpoints. The environment used for the first three trials contains one object, as shown in the first row of Figure 3. The object's shape and location vary between trials. For the latter trials, multiple objects were placed within the view of the camera and remained stationary between trials, as seen in the second row of Figure 3.

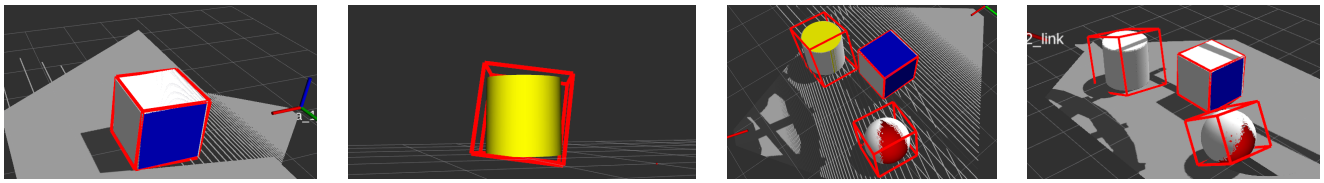
Figure 3 shows the bounding box generated by the algorithm in Rviz. The algorithm is evaluated by comparing the centre of the generated bounding box and the centre of the object in a simulated environment. The objects' centres are ground truth values obtained directly from the simulation. The error is calculated as the Euclidean distance between the bounding box centre and the objects' centre. The bounding boxes' centre and error values are shown in Table 1.

Table 1: Bounding boxes centre and error values

Exp	Bounding Box Centre (m)		Error (m)
1	Box	0.000,-0.402,2.836	0.002
2	Sphere	0.263,-0.395,4.494	0.0086
3	Cylinder	-0.124,-0.403,4.198	0.0059
4	Box	0.003,-0.401,3.836	0.0032
	Sphere	-1.499,-0.394,5.120	0.041
	Cylinder	1.400,-0.401,4.797	0.0221
5	Box	0.001,-0.237,3.956	0.0033
	Sphere	-1.761,-0.685,5.159	0.07
	Cylinder	1.770,-0.540,4.864	0.0319
6	Box	0.000,-0.825,3.689	0
	Sphere	-1.756,-1.902,4.393	0.0778
	Cylinder	1.761,-1.611,4.236	0.0437
7	Box	0.000,-0.827,3.684	0.0054
	Sphere	-1.748,-1.903,4.358	0.1094
	Cylinder	1.753,-1.606,4.230	0.0526

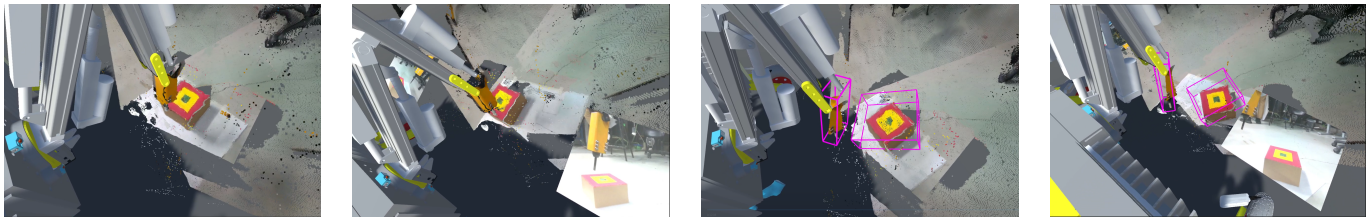
4.2 VR-based User Study

In order to evaluate if the annotations computed from point clouds, or additional information such as image streams would have a significant impact on user performance, a VR-based user study was conducted. Eight healthy participants with no neuromuscular or debilitating visual impairments volunteered to partake in the study. Prior to completing the study, participants provided informed consent approved by an ethics committee (UTS, Australia, approval number ETH21-5929). The participants were asked to perform the experiments pertaining to 4 sets of six repetitions.



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

Figure 3: Bounding boxes from the simulated experiments: (a), (b) individual objects experiment; (c), (d) multiple objects experiment with different cameras viewpoints



(a) Setting 1 (b) Setting 2 (c) Setting 3 (d) Setting 4

Figure 4: Participant's view in VR with differing settings: (a) Setting 1 - Point cloud only; (b) Setting 2 - Point cloud and two image streams; (c) Setting 3 - Point cloud and annotations; (d) Setting 4 - Point cloud, two image streams, and annotations

Table 2: Experimental setting

Setting	Available Information		
	Colored PC	Images	Bounding Box
1	A	N/A	N/A
2	A	A	N/A
3	A	N/A	A
4	A	A	A

4.3 Experiment Design and Procedure

Participants were given a set of written instructions and shown how to fit and adjust the VR headset. Participants were asked to remain seated with the VR headset on during the experiment for their health and safety.

The experiment required participants to perform a series of remote manipulation tasks. The task designed for the study involves using a joystick controller to manoeuvre a robotic manipulator whilst wearing a Virtual Reality headset. Within the VR environment, camera data ascertained from the 2 RGB-D cameras is rendered as a point cloud. Depending on the task configuration, the bounding boxes and color image streams may be presented.

Each participant is required to perform the manipulation task 24 times in four sets of six. The scene layout was varied after each repetition. Six different scene layouts were designed and were presented to the participants in a randomised sequence. The four different settings shown in Figure 4, which visualise the environment with a combination of data modalities, were used as shown in Table 2. For each repetition of the experiment, the four settings sequence are randomised, and one target is in the sensors' fields of view. The participant's aim is to move the robot's end-effector to the centre of the target, indicated by the green area, as quickly as possible (Figure 5). The additional colors are shown as supplemental information available to the participant.

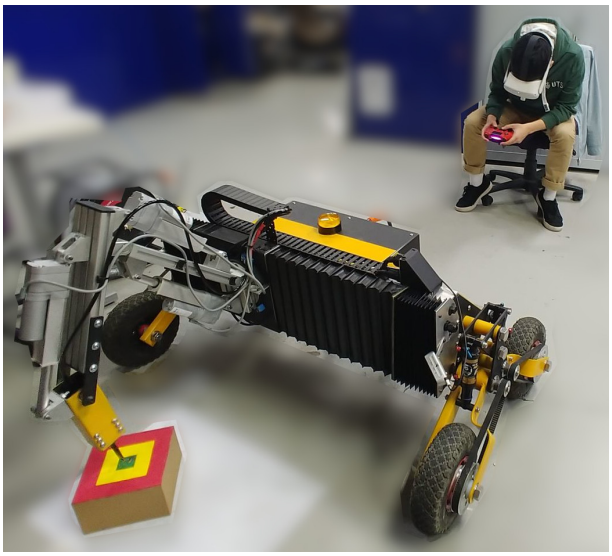


Figure 5: A participant manipulating the end-effector to the green area of the target while in VR

Before putting on the VR headset, participants were given three minutes to perform the manipulation task

while looking at the real-world experiment setup. This is to aid participants in familiarising themselves with the task, thus reducing the effect of the learning component. The layout used for this practice is a unique scene layout and is not one of the six scene layouts used for the study.

The participants are placed in a VR waiting room after the VR headset is put on. They are asked to not move the manipulator while waiting. When the scene layout is prepared, the administrator begins the VR simulation, and the scene is revealed to the participants in VR. During the experiment, the participants must verbally indicate when the end-effector of the manipulator has touched the target based on their own judgment. After the participants have indicated that the task is complete or two minutes has elapsed, the VR simulation is terminated, and the participants are returned to the VR waiting room. The administrator then configures the scene for the next task and returns the manipulator to its predefined home position. After a set of tasks, the participants remove the VR headset and rest for 3 minutes before continuing the next set of repetitions. This minimises the effect of fatigue on the participants' performance.

4.4 Experimental Results

Experimental results collected from 12 participants performing a total of 288 repetitions of the task are shown in Figure 6.

Objective Task Completion Time

The task completion time is determined as the time from when the participant starts operating the manipulator till when they verbally indicate that the task is completed. A one-way ANOVA test was conducted with the null hypothesis being that there is no significant difference between the mean completion time of the various settings implemented during the study. An F-statistic value of 1.71 and a p-value of 0.1644 were obtained from the analysis, showing no statistical significance.

Objective Task Precision

To quantify the accuracy of the participants during the experiment, the participants receive a point score based upon whether the end-effector touches the target. The participants are given 5 points if the end-effector is touching the green area on the target surface when they indicate they have completed the task. Similarly, if the end-effector stops when touching the yellow area, the participants receive 3 points, and 1 point is allocated for touching the red area. If the end-effector is not touching the target surface, the participant is not given any points.

Similar to section 4.4, a one-way ANOVA test was also conducted with the null hypothesis being that there is no significant difference between the mean accuracy of the above-mentioned settings. An F-statistic value of 2.37 and a p-value of 0.0705 were reported, supporting the previous statement.

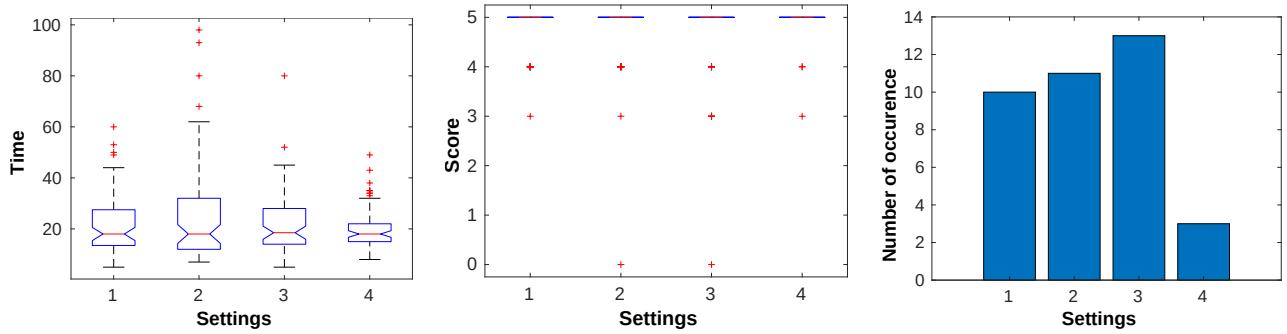
5 DISCUSSIONS

The experimental results indicate that the availability of the information presented through various mediums, namely 2D images, and bounding box annotations, did not significantly impact the objective task completion time, as seen in Figure 6a. The median completion time of the participants collectively for all settings is similar, despite the varied fluctuation range. Figure 6b shows that among the four settings investigated, participants are able to manipulate the end-effector to the target's centre in most trials. However, Figure 6c demonstrates a difference in the resulting score when images are available. In settings where images are available (Setting 2 and 4), participants failed to hit the target's centre 14 times in total. Within the 14 misses, there were 12 occurrences (85.7%) where the end-effector touched the edge of the green target area, resulting in a score of 4. On the other hand, for other settings, participants missed the centre 23 times. The participants only scored 4 a total of 15 times (65%). Amongst all the investigated settings, participant performance improved with Setting 4, as indicated by the low number of times the participants missed the green area.

The experimental results suggest that the non-annotated visualisation that the presented colored point cloud alone, was sufficient for this remote-controlled manipulation task. However, this may be due to the controlled and ideal indoor setup where depth sensors were not exposed to excessive infrared noise, such as from the sun. Furthermore, bounding box annotations did not impact the overall performance as the target could be easily identified in the scene. In other scenarios, where the target cannot be distinguished through color or if the color information is unavailable, the target becomes less recognisable, and annotations may better assist a user.

6 CONCLUSIONS

This paper investigated the effect of annotations on participant performance when executing a remote manipulation task in VR. The annotations are generated and presented as overlays on the real-time 3D point cloud data displayed in VR. The point cloud processing algorithm integral to the creation of the annotations is also presented. A simulated experiment was conducted to determine the accuracy of the point cloud processing algorithm in a controlled environment. In addition, the experimental results from the user study indicate that a combination of point clouds and images greatly improves the task precision. On the other hand, there are no sig-



(a) (b) (c)

Figure 6: Experimental results for each setting: (a) Completion time; (b) Precision quantified score; (c) Histogram of repetitions where participants missed the green target area

nificant differences between the four settings examined in terms of completion time.

Future work requires undertaking further experiments regarding the implementation of annotations in VR, especially for scenarios where the targets are difficult to see without additional assistance. Furthermore, this research necessitates experiments to be conducted in more challenging environments. Outdoor environments, particularly those with bland colors, are expected to result in more RGB-D camera data error, which is predicted to significantly reduce a user's performance and increase the need for annotations.

7 Acknowledgement

This research is supported by UTS:RI, The Commonwealth of Australia's Department of Industry, Science, Energy and Resources (Innovative Manufacturing CRC Ltd) and Perenti, via its subsidiary Ausdrill. T.L. Vu, D.D.K. Nguyen, D.T. Le and S. Sutjipto are supported by Australian Government Research Training Program Scholarships.

References

[Abdi and Williams, 2010] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[Coleman *et al.*, 2014] David Coleman, Ioan Alexandru Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *ArXiv*, abs/1404.3785, 2014.

[Fischler and Bolles, 1981] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[Gharaybeh *et al.*, 2019] Zaid Gharaybeh, Howard Chizeck, and Andrew Stewart. Telerobotic control in virtual reality. In *OCEANS 2019 MTS/IEEE SEATTLE*, pages 1–8, 2019.

[Gong *et al.*, 2019] L. L. Gong, S. K. Ong, and A. Y. C. Nee. Projection-based augmented reality interface for robot grasping tasks. In *Proceedings of the 2019 4th International Conference on Robotics, Control and Automation*, ICRCA 2019, page 100–104, New York, NY, USA, 2019. Association for Computing Machinery.

[Gradmann *et al.*, 2018] Michael Gradmann, Eric M Orendt, Edgar Schmidt, Stephan Schweizer, and Dominik Henrich. Augmented reality robot operation interface with google tango. In *ISR 2018; 50th International Symposium on Robotics*, pages 1–8. VDE, 2018.

[Kipper and Rampolla, 2012] Gregory Kipper and Joseph Rampolla. *Augmented reality: An emerging technologies guide to AR*. Elsevier, 2012.

[Krichenbauer *et al.*, 2017] Max Krichenbauer, Goshiro Yamamoto, Takafumi Taketom, Christian Sandor, and Hirokazu Kato. Augmented reality versus virtual reality for 3d object manipulation. *IEEE transactions on visualization and computer graphics*, 24(2):1038–1048, 2017.

[Kumar and Todorov, 2015] Vikash Kumar and Emanuel Todorov. Mujoco haptix: A virtual reality system for hand manipulation. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 657–663. IEEE, 2015.

[Le *et al.*, 2020] Dinh Tung Le, Sheila Sutjipto, Yujun Lai, and Gavin Paul. Intuitive virtual reality based control of a real-world mobile manipulator. In *16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 767–772, 2020.

[Lin *et al.*, 2016] Yuan Lin, Shuang Song, and Max Q-H Meng. The implementation of augmented reality in a robotic teleoperation system. In *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 134–139. IEEE, 2016.

[Malik *et al.*, 2020] Ali Ahmad Malik, Tariq Masood, and Arne Bilberg. Virtual reality in manufacturing: immersive and collaborative artificial-reality in design of human-robot workspace. *International Journal of Computer Integrated Manufacturing*, 33(1):22–37, 2020.

[Ni *et al.*, 2017] D Ni, AWW Yew, SK Ong, and AYC Nee. Haptic and visual augmented reality interface for programming welding robots. *Advances in Manufacturing*, 5(3):191–198, 2017.

[Paul *et al.*, 2016] Gavin Paul, LiYang Liu, and Dikai Liu. A novel approach to steel rivet detection in poorly illuminated steel structural environments. In *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–7, 2016.

[Pérez *et al.*, 2019] Luis Pérez, Eduardo Diez, Rubén Usamentiaga, and Daniel F García. Industrial robot control and operator training using virtual reality interfaces. *Computers in Industry*, 109:114–120, 2019.

[Rückert *et al.*, 2018] Patrick Rückert, Laura Wohlfromm, and Kirsten Tracht. Implementation of virtual reality systems for simulation of human-robot collaboration. *Procedia Manufacturing*, 19:164–170, 2018.

[Rusu and Cousins, 2011] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pages 1–4. IEEE, 2011.

[Sutjipto *et al.*, 2020] Sheila Sutjipto, Yujun Lai, Marc G Carmichael, and Gavin Paul. Fitts' law in the presence of interface inertia. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 4749–4752. IEEE, 2020.

[Trevor *et al.*, 2013] Alexander JB Trevor, Suat Gedikli, Radu B Rusu, and Henrik I Christensen. Efficient organized point cloud segmentation with connected components. *Semantic Perception Mapping and Exploration (SPME)*, 2013.

[Vélaz *et al.*, 2014] Yaiza Vélaz, Jorge Rodríguez Arce, Teresa Gutiérrez, Alberto Lozano-Rodero, and Angel Suescun. The influence of interaction technology on the learning of assembly tasks using virtual reality. *Journal of Computing and Information Science in Engineering*, 14(4), 2014.

[Vu *et al.*, 2019] Thanh Long Vu, Liyang Liu, Gavin Paul, and Teresa Vidal Calleja. Rectangular-shaped object recognition and pose estimation. In *Australian Conference on Robotics and Automation (ACRA)*, 2019.

[Vu *et al.*, 2021] Thanh Long Vu, Dinh Tung Le, Dac Dang Khoa Nguyen, Sheila Sutjipto, and Gavin Paul. Investigating the effect of sensor data visualization variances in virtual reality. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, pages 1–5, 2021.

[Whitney *et al.*, 2020] David Whitney, Eric Rosen, Elizabeth Phillips, George Konidaris, and Stefanie Tellex. Comparing robot grasping teleoperation across desktop and virtual reality with ros reality. In *Robotics Research*, pages 335–350. Springer, 2020.

[Yew *et al.*, 2017] AWW Yew, SK Ong, and AYC Nee. Immersive augmented reality environment for the teleoperation of maintenance robots. *Procedia Cirp*, 61:305–310, 2017.