Tracer: Interactive AI camera man that frees your hands and record your own cinematic footage

Heran Zhang, Shurui Li, Junjie Lu, Chengdong Sun, Zengyang Pan, Wenjia Luo, Kexin Li
Department of Electrical and Electronic Engineering, Imperial College London, SW7 2BT
Email: {hz4915, sl10215, jl15315, cs5015, zp815, wjl15, kl2215}@ic.ac.uk
Supervisor: Dr. Yiannis Demiris

Abstract

Tracer is an AI camera man designed to help people taking video with interactions. It is capable of tracking individual and acquired with superb object avoidance abilities. Moreover, Trace can interact with people on different levels through body gestures and voice commands. The robot was built upon a peoplebot with two Kinect cameras, a LiDAR laser sensor and a customized camera platform.

1. Introduction

There has been a dramatic increase in the use of video contents after the introduction of social media like YouTube, Instagram, Facebook and Twitter. Views of branded video content have risen by 99% on YouTube and by 258% on Facebook from 2016 to 2017[1]. Moreover, a video Tweet is 6 times more likely to be shared than a photo Tweet. Mark Zuckerberg, Facebook's CEO predicted that he wouldn't be surprised if in the next 5 or so years, most of the content shared on a day-to-day basis on Facebook is in video form.

Besides, the growing trend in the use of video contents has been further proved after a release of a social media app TikTok. The app provides a short video platform for people to showcase their creativity. Up till June 2018, in China alone, the daily active users had gone over 150 million and over 300 million monthly active users [2]. In the first quarter of 2018, TikTok was officially downloaded by 4.58 million times in iOS, overtaking YouTube and become the world's most downloaded iPhone app [3]. These statistics define the leading position of video contents on the social network.

However, there are some occasions exist during the process of filming the videos, especially for the individuals without a professional photographer, they have to hold a camera by themselves which restricts the filming distance due to the arm length. Therefore, the visual effect would be limited to a great extent. Therefore, we introduce Tracer as a perfect hypothesized companion for video taking.

This report will first state the hypothesis we are trying to test, followed by comparison between the related work and Tracer. Next, the report will explain the details of the system design and how the experiment was set up and the methodologies to validate our hypothesis. This report will then reveal and discuss the experimental results to analyse our hypothesis. Finally, it will state some future work to further improve the robot.

2. Hypothesis

The purpose of this project is to investigate the abilities for the robot to film the third-person perspective videos for individuals or the groups. The robot can be qualified based on below hypotheses:

1. The robot should recognize individuals and tracks them to perform follow up shots, detects and avoids the obstacle in the outside environment.
2. The robot should perform different functions upon receiving hand gestures or voice commands.

The robot will be displayed and experienced by different individuals, the result videos produced by the robot will then be rated by participants to evaluate the filming skill of the robot. Methodologies to validate the robot's abilities will be further discussed later in the Experimental Validation section.

3. Background

There has already been a substantial amount of works accomplished in this field, self-flying drones in the market that tracks user to perform film recording, the most renowned ones are selfie-drones from DJI and Skydio R1. However, these drones are only capable of taking long angle shots and not suitable for close-up shots. In addition, these drones are not the best choice when audio recording is required. In most cases, background music is imported and used rather than raw audio input recorded by the drone as the sound from the propellers will be recorded in. Moreover, when the filming needs to be taken indoor, drones are not preferable, since they require more active space in general and space limitation could also limit its capabilities. On the other hand, PhotoBOT from the class 2017 is another perfect example. It is a robotic photographer on wheels which responsible for taking pictures of people. Furthermore, it can interact with the users via speech and mounted touch screen interface. It shares some similar functionalities with our proposal, such as simple movements on wheels, object avoidance, speech recognition and navigation [4]. However, PhotoBOT is only capable of taking photo shot but not video recording and we proposed to use hand gesture recognition for commanding the robot during shooting mode in order to develop novel features in human interaction. Therefore, tracer not only provides close-up shot and distinct original audio recording but also offers indoor video recording and gesture recognition functions.

4. System Design

4.1. Overall Design

The Tracer robot is a complex system integrated with both ROS and multiple AI techniques, it is designed to act as a personal cameraman and supposed to do similar job. It can be fully controlled by voice and gesture, without using any physical controller or app, just like a real human.

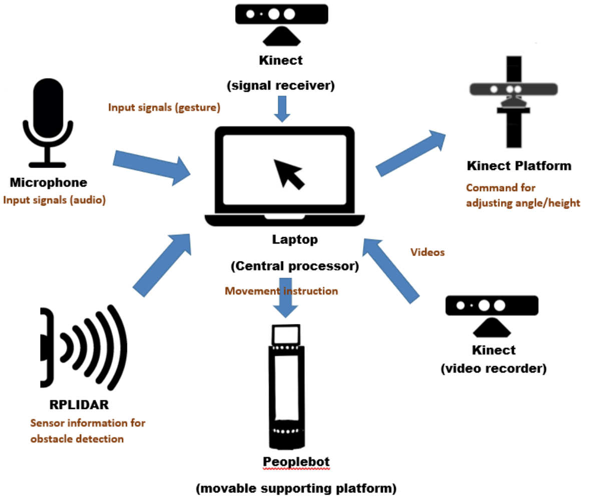

Figure 1: High-level system overview

Fig 1 shows the high-level system overview for the tracer. Two Kinect cameras are implemented with one used for gesture recognition and human tracking, the other used as a camera to record video and audio. Both of them transmit the data to the laptop. Laptop is a central processing unit which not only fed the signal from the Kinect sensor into different algorithms to generate corresponding output, i.e. movements for the Peoplebot, angle and height adjustments for the Kinect platform, but also merges the video and audio to generate the finished video.

The robot will automatically follow users when it detects an identity with calibration with the psi pose. The algorithm will let the robot keep a roughly constant distance from user to ensure the user is at the centre of the video while the actual distance is determined by the user's command or the robot's learning algorithm.

However, voice commands are not suitable during filming since the user's voice commands will be recorded as well. Therefore, gesture commands are introduced to compensate this problem as using gesture to control the robot is more natural than voice.

There are several alternative methods that can be used to track human, the methods are described and compared in the following table.

Human following method	Calibration	Description
Face recognition	No calibration needed	Easy to implement, cannot track when face not shown in the picture.
Skeleton Marker	Calibration with psi pose needed	Can be used to track even when facing the camera, but not very reliable.
OpenPose[7]	No calibration needed	High accuracy of detection, high hardware requirements.

Table 1: Human tracking method comparison matrix

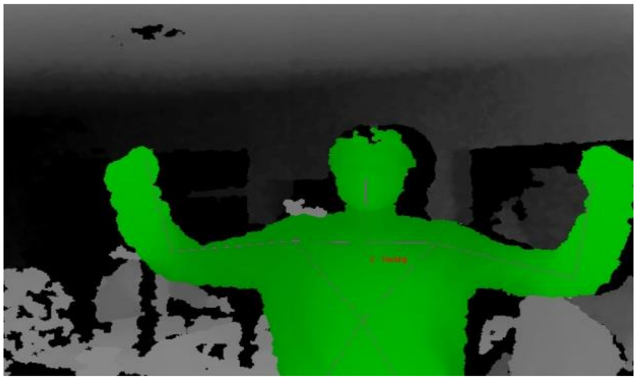

Figure 2: Calibration with the psi pose

To implement the functions of the human following and gesture detection, we use Skeleton Marker to publish a list of joint markers returned by the openni_tracker package for viewing in RViz. However, OpenPose has a better performance than Skeleton marker because OpenPose can jointly detect a human body, hand, facial, and foot key points on single images for multi-person. However, the implementation of OpenPose has some prerequisites which our laptops are not able to satisfy.

4.2. Software Architecture

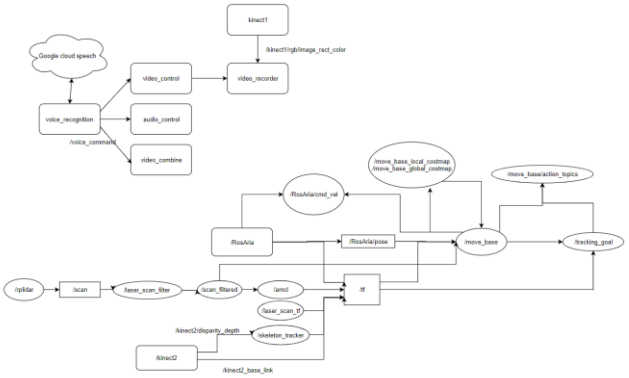

Figure 3: Final system design of Tracer

The above diagram illustrates out final system design. It is a leaf diagram with a fairly accurate representation of the communications between the nodes and topics even though some interactions were simplified for clarification.

4.3. Recording

4.3.1 Video Recording

Video recording node make use of the OpenCV library to construct the video recorded. The video mechanism captures the image taken from the image_view at the preset FPS rate and stacks the images together to form the video at the end. However, the image taken from image_view cannot be directly used by the OpenCV library; hence we need to convert the image through BGR8 encoding method into an OpenCV supported image type. The images shown in image_view is subscribed from the Kinect rectified compressed colour image topic.

4.3.2 Audio Recording

Human voice can be recorded clearly using Kinect built-in microphone. The audio_capture package is utilized to records audio from microphone and transport it to a destination for playback, after that rosbag_record is used to record the audio topic into a bag file [5]. Then the ros-audio-converter package is applied to convert a bag file into a wav file and outputs the audio messages to local speakers.

4.3.3 Merge Recording

Video recording and audio recording functions are recorded separately by Kinect, they are then merged synchronously with the same speed to generate the video.

4.4. Vision

4.4.1 Human Tracking

This module is responsible for locating a human individual and continuously sending navigation goal to the move_base topic of the robot. We have been using an existing package called skeleton_marker which provides human skeleton tracking via ROS and OpenNI. Skeleton marking is done through extensive analysis of depth data and construct coordination mapping to the joints of the human body. The skeleton_marker package constructs the skeleton frame through 12 joints (insert skeleton marker image) and publish each as a tf transform coordinate with respect to the depth_frame of the kinect camera [6][8].

In order to perform human tracking task, we construct a skeleton_tracker module which looksup the transform from the /torso frame to the /openni_depth_frame through a transform listener and this can provide us information about the human coordinates in 3D space related to the robot. Then formulate a navigation goal base on the transform provided and send to move_base to set up a navigation plan for the robot.

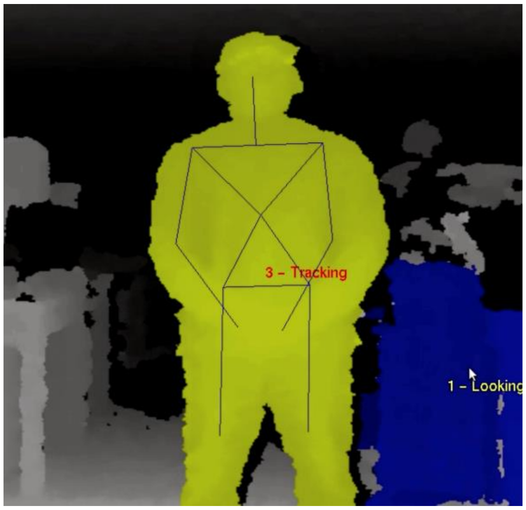

Figure 4: Skeleton tracking example with 12 joints

4.4.2 Gesture Recognition

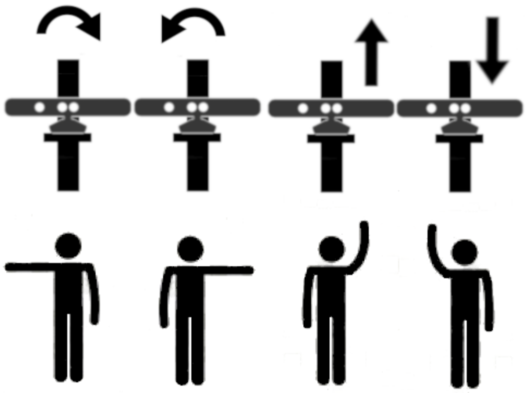

Figure 5: List of gesture commands for Kinect platform movements

Currently our system has four gestures in total, and the gesture node for our system make use of the skeleton_marker package mentioned above. In a similar manner to the skeleton_tracker node, it looks up the transform for the right elbow, right hand, left elbow and left hand. By setting multiple thresholds to the coordination of the transform getting back, it can publish different messages to the target topic according to various gestures.

4.5. Navigation

The navigation module used by out robot allows it to keep track on the person while equipped with a robust object avoidance system for both static and dynamic objects encountered. For obstacle detection, we rely on a RPLIDAR A1 sensor which is mounted at the base of our robot. The standard laser scan topic provides a 360-degree data; however, it also scans the two columns of the peoplebot along the way, which is undesirable for navigation as it always detects obstacles on the sides. Therefore, we had specified a laser scan filter configuration file to filter 180-degree data along the rolling axis and take in only the other 180-degree data [9]. Since our robot was designed for dynamic environment, thus we do not need to provide a static map for the robot, it makes use of the local cost map configuration to build local cost map on the run and avoid obstacles.

After the move_base node taking in the navigation goal from the tracking_goal node in skeleton_tracker package, the robot builds a local planar to navigate the robot in the surrounding environment. The move_base node is also running the amcl node for a good localization of the robot.

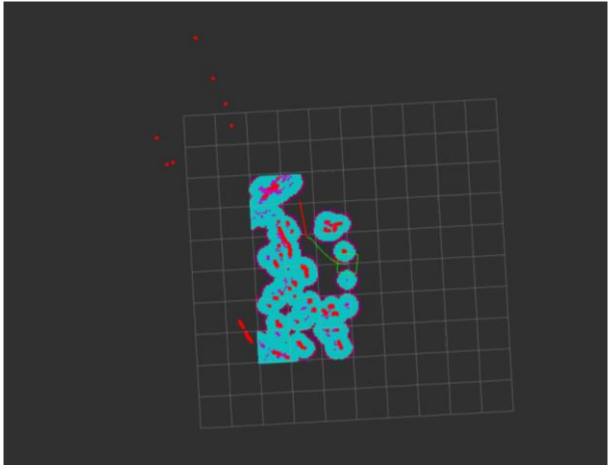

Figure 6: Local cost map and filtered scan data for navigation visualized in RViz

4.6. Voice Recognition

Speech recognition technique and hot word detection are implemented to recognize user's voice commands. The voice command system is separated into two parts, system activation and commands recognition.

4.6.1 System activation

Since Tracer is a filming robot that record videos for users, it is clearly that the users will talk while the robot is running. Thus, the voice command function cannot be always on, otherwise there will be a lot of mis-triggered commands and seriously affects the filming process. As discussed in the design report, the solution is to use a trigger word to activate the voice command function and out trigger word is 'Tracer Tracer'. The trigger word is carefully selected so that under most circumstances it will not be triggered when user is not intended to do so since it's rare that someone will repeat 'Tracer' twice during filming.

To implement the trigger word, Snowboy hot word detection package in python was used. This package collects several recorded sample of desired trigger word/voices and trains a unique convolutional neural network model for the trigger word [10]. In order to generalize our model to fit most users, several of our group members recorded their own voice sample to better train the model. After the model is trained, the hot word detection function can be used in python scripts and it's quite simple – the function holds the python scripts from continuing until it detects the trigger word from microphone.

This solution of using Snowboy package is proved to be the best solution for its simplicity, flexibility and robustness. The system rarely mis-triggers, thanks to the uniquely trained neural network model.

4.6.2 Command recognition

After trigger word is detected the command recognition function is activated, which analyses the user's speech input from microphone and converts to text. The translated text is further analysed and classified into different pre-set commands or return an error.

Python's Speech Recognition package is chosen for this task, by using Google's cloud speech engine, the package converts input voice to text with a high accuracy in quiet environment. However, this function will consistently listen to the microphone's input until there's a moment of completely quietness by default. In real world there will always be background noise and will not only affects the recognition accuracy but also makes the function 'listens' forever sometimes. Several approaches are implemented to counter the mentioned problems to ensure a good recognition accuracy in real noisy environment. Speech Recognition package's built-in function for adjust ambient noise is used in order to reduce the impact of background noise. The function will listen to background noise for some time (we use 0.5s) before recognizing the voice, then it subtracts the noise signal from voice input to remove noise. This function is pretty useful for consistent background noise. The time for the function that listen for a voice command is set to 4 seconds, to stop the function from forever listening. Moreover, we developed algorithms to analyse the translated test so that if the actual command lies in the translated text the command will be detected.

4.7. Kinect Platform

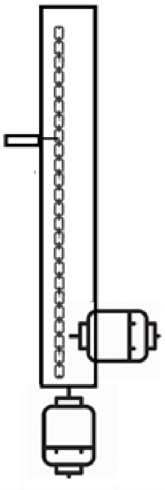

Figure 7: Kinect platform diagram

Since camera filming direction and Height need adjustment while Tracer is tracking objects, a Kinect camera Platform was implemented into tracer.

Kinect camera is held on a small plastic plate installed on a 50cm metal rod. There is a servo motor beneath the rod to rotate device from -90 to +90 degree. Camera holder is also connected to link chain controlled by a 9V dc motor to move camera up and down.

Our group used Arduino Uno programmable board to control system. It has PWN pin available which can easily adjust motor spinning speed and Arduino allows communication with ROS and its own programming environment with *rosserial* package. Circuit schematic and Breadboard implementation has shown below. Noticeably, programmable pins' maximum Voltage and current outputs are only 5V and 40mA which is not powerful enough to drive whole system. Thus, a 9V battery was added to supply additional power.

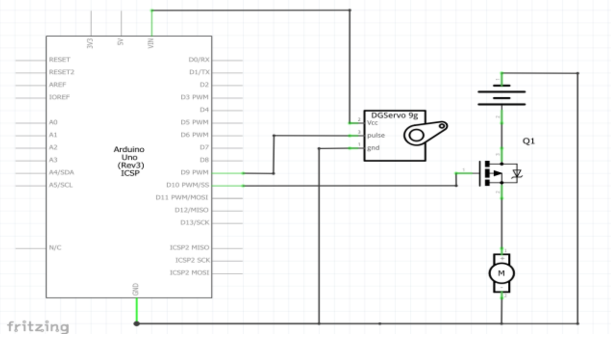

Figure 8: Platform control circuit schematic

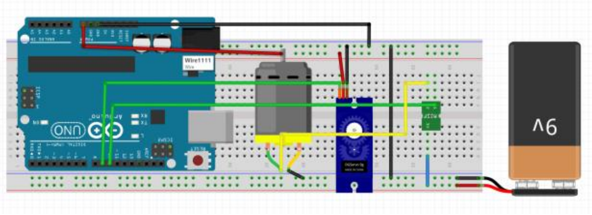

Figure 9: Platform control circuit breadboard implementation

5. Experimental setup & methodology

In this section, a variety of methodologies are outlined to test our hypotheses of our robot's abilities. Our hypotheses are explicitly divided into four different categories: navigation, hand gesture, speech recognition and video quality. Therefore, different experiments have been used to test each category.

The first experimental task is designed to validate the navigation function of the robot, since Tracer needs to be used in large and open-space indoor environments such as common rooms and corridors, we have chosen the common room on fifth floor to be the location of this experiment. The common room provides a spacious location and one of our team members stood in front of the robot at the beginning. The robot was initialized and recognized the team member through calibration gesture, our team member then began to walk away from the robot slowly, and the robot will follow him in the same direction. The robot will try to keep the distance constant between itself and the team member as he walks away. Afterwards, the same task was repeated with the team member for further tests with quicker walking pace and recognized fixed position obstacles to avoid collisions in the progress. For each trial, we measured the time needed to complete the task.

In another experiment, our team investigated the hand gestures instructions. Team members with different height demonstrated corresponding hand gestures in front of the robot to convey the adjustment instructions. In this experiment, all the hand gestures commands were performed, and the robot will try to decode those commands and adjust the filming angle accordingly. Throughout each trial, team members who interacted with the robot were required to evaluate about its' performance which can be used for improvements.

Speech acts as a secondary form of communication between the robot and the human which an intuitive way of communication for issuing commands. In order to test our real time continuous speech recognition function, we chose to conduct initial experiments in a quite common room first. Team members tested voice commands used to start/stop video filming. In addition, different team members were tested to ensure the robot can recognize human voice clearly and then transcribed into English text accurately. In order to test the feasibility of speech recognition, additional tests were conducted in public indoor environments (EEE building), where noises would be a considerably problem for the accuracy of speech recognition and quality of audio under outdoor conditions.

In addition, the quality of video recorded was evaluated through tests, the results based on personal satisfaction about the video recorder by Tracer, team members are asked to assess the video quality to reflect their determination on videos.

Finally, after all the individual experiments were carried out, an overall experiment was designed to evaluate all testing conditions stated above, that is to let users (non-team members) with various height and voices to demonstrate hand gestures and speech recognition tasks in front of the robot in a public environment each time. Users being filmed can walk randomly in a space and perform hand gestures to instruct the robot. At the same time, the robot is supposed to adjust its camera orientation while following the users and avoid possible collisions.

For the purpose of evaluate different functions of Tracer, tests are carried out for users to experience. At the end, a short survey is required to fill in for assessment as shown below.

Survey Questions	**Answer Format**
Rate the precision of human detection	Rating in 1-10
Rate the speed and angle change of following human	Rating in 1-10
Rate the effectiveness of obstacle avoidance	Rating in 1-10
Rate the agility of hand gestures	Rating in 1-10
Rate the intelligibility of speech recognition	Rating in 1-10
Rate the responsivity of voice commands	Rating in 1-10
Rate the quality of the video	Rating in 1-10
Rate the quality of the audio	Rating in 1-10
Give your overall usability ratings	Rating in 1-10

Table 2: Survey sample

6. Results

The robot was tested by 10 users at Imperial College to validate our hypotheses and let participants to fill the survey form.

The table illustrates the average ratings of the users to the subcomponent and the whole system after experimenting the robot and receiving their own finished video.

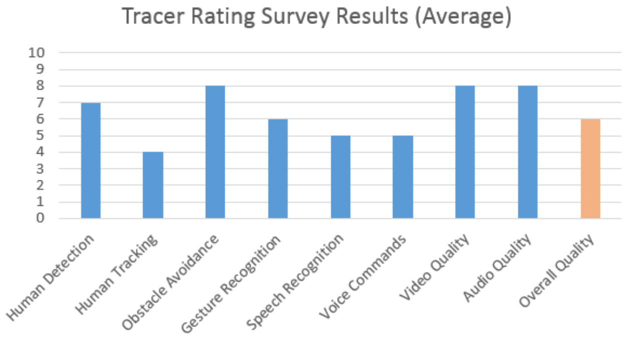

Figure 10: Survey results bar chart

It can be seen that our robot was ordinarily rated by our users. The functionalities for video recording, obstacle avoidance and human detection are fairly stable, while others like human tracking, hand gestures and speech recognition still have a lot of room to improve. When preforming human tracking task, the robot failed to move in a steady pace and stops constantly. For gesture recognition, the robot is able to recognize the gestures correctly only if the individual does not move, otherwise the gesture recognition is likely to fail as the transform information or the joints are too unstable. Voice recognition tasks is irresistible to noise when performed in public space and commands are either misconducted or completely missed out in most cases.

On the other hand, obstacle avoidance is spot on, the robot can recognize both dynamic and static obstacles ahead and reroute the local planar to the desire location automatically, however, due to the laser scan filter, and obstacles behind the robot cannot be spotted at once. As a filming robot, the video output had a satisfying result with no time difference between the audio and video. Although, there is a second of blacked out frame in the video, the overall result is acceptable. Finally, despite the fact that the time for the robot to recognize a human can vary from fraction of a second to dozens of a second, it never fails to mark the skeleton correctly, thus the task was welly executed.

As well as the answer to the survey, individual test subjects to make comments on the system and some key points were noted. The robot can avoid a majority of obstacle while it was turned on, but some minor situations happen in which the robot cannot respond fast enough to make a turning before colliding with the obstacle. For the agility of the hand gesture, some users are not satisfied with the accuracy of detecting the hand gesture and hope the robot can respond faster to the gesture commands.

7. Discussion

Throughout a series of experiments and tests with the users, a number of issues had been risen that our team did not consider before. Some customers find it is confusing to interact with Tracer at the first time due to lack of status indication. It is interesting to see that including some easy instructions for customers or adding a screen that display current status and show easy customer instruction would have significantly mitigate this issue. The Kinect camera also caused some confusion to Tracer customers, several people tends to stand surrounds Tracer which makes the robot difficult to recognize who is the right person to track with. This issue can be avoided by designing a user interface for customers to select the right following object at the beginning.

8. Future Work

On the aspect of the technical part of the tracer, the future work will focus on improving the imperfect functions reflected in the previous results and questionnaire, such as the function of obstacle avoidance and the response time of the gesture commands. In theory, the obstacle avoidance function can be improved by either better algorithms or more detective devices which can help the tracer detect more details of the surrounding and then figure out where is the small obstacle. We can also combine these two solutions to make a more excellent and tolerant result. However, in some occasions, some undetectable small stones and unavoidable obstacle like traffic hump might not affect tracer’s movement, but it might affect the filming performance. Hence, design and assemble a 3-axis stabiliser could be implemented in the future scheme. For the response time and detection accuracy of the gesture commands, the machine learning technique could be considered to implement in the tracer. Tracer can study human behaviours to predict the next moving of the commander and thereby decrease the response time. Implementing OpenPose will improve human tracking accuracy and reliability.

In the future, the assessment will be divided into a three-stage step-by-step process: individual performance comparison, simple circumstance test, complex real-world test. Individual performance comparison means re-implementing the individual tests which are mentioned before then compare the results before and after the adjustment. The test will then be undertaken under a simple circumstance such as the common room with less noise and a simple background to evaluate the overall performance. After that, the tracer will be brought to the outdoor environment with more noise and interruption to investigate the abilities of the robot and further improvement will be implemented if needed.

9. Conclusion

In conclusion, although Tracer is able to taking quality video for individuals, it does have a number of glitches that needs to be improve or fixed as shown in the above report. However, as a proof of concept experiment, it does have a lot of room for improvements and can be expected to complete the task at a higher standard.

10. References

[1] Wyzowl. (2018). Why Video is Exploding on Social Media in 2018. [online] Available at: https://www.wyzowl.com/video-social-media-2018/ [Accessed 29 Oct. 2018].

[2] Sohu.com. (2018). Tiktok user. [online] Available at: http://www.sohu.com/a/235554198_601684 [Accessed 29 Oct. 2018].

[3] Shen, K. (2018). Tik Tok is surging social network ecology — the brand-new interaction among the 00s generations. [online] Medium. Available at: https://medium.com/@kaichenshen/tik-tok-is-surgingsocial-network-ecology-the-brand-new-interactionamong-the-00s-generations-e7bfa572e829 [Accessed 29 Oct. 2018].

[4] Balassa, L., Karandejs, N., Karolcik, S., Matas, J., Olexa, P. and Pulmann, T. (n.d.). Human Centered Robotics Final Report: PhotoBOT. pp.1-4. [Accessed 30 Oct. 2018].

[5] Wiki.ros.org. (2018). audio_capture - ROS Wiki. [online] Available at: http://wiki.ros.org/audio_capture [Accessed 16 Dec. 2018].

[6] GitHub. (2018). pirobot/skeleton_markers. [online] Available at: https://github.com/pirobot/skeleton_markers/tree/46e8cee640834fd95bdf6785a953066a604e04b4 [Accessed 15 Dec. 2018].

[7] GitHub. (2018). CMU-Perceptual-Computing-Lab/openpose. [online] Available at: https://github.com/CMU-Perceptual-Computing-Lab/openpose [Accessed 15 Dec. 2018].

[8] zouxy09 (2018). Kinect开发学习笔记. [podcast] Kinect开发学习笔记之（七）骨骼数据的提取. Available at: https://blog.csdn.net/zouxy09/article/details/8161617 [Accessed 16 Dec. 2018].

[9] Wiki.ros.org. (2018). laser_filters - ROS Wiki. [online] Available at: http://wiki.ros.org/laser_filters [Accessed 16 Dec. 2018].

[10] Snowboy.kitt.ai. (2018). Snowboy Hotword Detection. [online] Available at: https://snowboy.kitt.ai/ [Accessed 15 Dec. 2018].