Design of multiple UAVs Using Computer Vision, Human Body Gusture and QR Code Recognition with Indoor Self-navigations

Hao Minsheng, Zhang Ruidong, Zhong Qingyang, and Lu Geng*

Department of Automation, Tsinghua University

* lug@tsinghua.edu.cn

1. Introduction

In the latest 8th mission of IARC, some extra capability is required for the UAVs and competetor, such like autonomous swarm control, remote oral control, QR code merging and recognition, obstacle detection and avoidance, human body gusture recognition, target box recognition and location, as well as acception of rescue laser beams. Team from Tsinghua will participate this mission and have an introduction of some key technology using in the competition.

1.1 Obstacle avoidance based on RGBD Image recognition

In recent years, image recognition has been used more and more widely with the development of deep learning. Fingerprint identification, face recognition and medical image segmentation are common applications. In the International Aerial Robotics Competition, We can use RGBD cameras to perform obstacle detection, gesture recognition and box recognition tasks.

There are two cameras on the UAV, one is ordinary, and the other is deep camera.When flying in the air, the UAV needs a more complicated method to detect and avoid obstacles, compared with cars on the land. As for obstacle detection, we designed an algorithm to avoid any obstacle within 3m so that the UAV is able to fly smoothly in a complex environment. Without this technology, the UAV is impossible to accomplish missions automatically in practice.

For identifying tasks, the image taken by ordinary camera only has RGB information, and the box is separated from the background according to the color information. After the center of mass of the box is obtained, the flight direction and speed of the UAV are adjusted to fly towards the box. When the UAV flying near the box, the flight attitude is adjusted so that the depth camera is aligned to the two-dimensional code exactly, and the information is identified and subsequent flight is carried out.

In this year's mission, the UAV is required to be able to respond to gesture commands. Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. In view of the actual operating environment of the UAV, we focus on the skeleton recognition and hand gesture recognition for single people.

Compared with the traditional manual control system, UAV has many advantages, but it also includes many risks. For some certain missions, UAV have to operate in a complicated environment, which has both fixed and moving obstacles. Therefore, it is of great value to improve the security and stability of UAV flight. In order to achieve real autonomous control, obstacle avoidance approach is the foundation of getting rid of human manual operation.

Before the vision-based method achieved significant development, traditional methods are widely used in this area. Gageiket al.[1]designed an innovative and simple solution optimized for and evaluated with quadrotors which uses low-cost ultrasonic and infrared range sensors. The problem is that though the exploited sensors are much cheaper, it is noisier than expensive ones like laser scanners. As a result, it need to be taken into consideration of the implementation and parametrization of the system model. As a result, it increases the extra computational burden, which is memory and time-consuming for localization and mapping.

Because of the significant development of the stereo camera systems and GPU-based image processing algorithms, it is quite efficient to use the vision-based measurements since it is compact, lightweight and accurate and it has many practical applications in different fields [2].

A paper by Kumar and Ghose demonstrates that a navigation and guidance law achieves both waypoint tracking and collision avoidance[3]. However, this method is restricted only in a 2-D plane. Kwag and Kang[4] also proposed a radar sensor system. They exploit an image processor, which could detect multiple obstacles in images that is obtained from a 2-D vision sensor. Stefan Hrabar proposed a navigation and guidance technique that achieves multi-obstacles collision avoidance in unknown environments. They use a 3D map to represent the environment and update the state using stereo vision data for path planning and dynamic path updating. This method could be applied to the field of helicopter-based structure inspections and ensure the security of UAV flying safely close to the inspections. Chen et al.[5] first present an algorithm which combined the tangent vector field guidance (TVFG) and the Lyapunov vector field guidance (LVFG). They find that TVFG outperforms the LVFG as long as a tangent line is available between the UAV's turning circle and an objective circle, which is a desired orbit pattern over a target. They then proposed a shortest path planning algorithm adopting the hybrid version of TVFG and LVFG. The algorithm is quite efficient when given a target position and current UAV dynamic state.

1.2 Speech recognition

According to the requirements of the competition, the player needs to use non-electronic means to control the mobile assistant robots in order to complete mission 8. Therefore, a new method of control needs to be designedto control the quadrotor UAVseffectively.

After years of development, speech recognition technology has become increasingly mature and has been widely used in the fields of information processing and automation due to its convenience and well interaction. Using voice to send commands to UAVs directly is suitable for controlling UAVsunder this circumstance and thus has become our top choice.

This competition requires the on-site player and UAVs to complete complex tasks together: the player running on the field needs to interact with the UAVs in a timely and effective way. After several tries, we finally decide to use the off-line command lexicon and Voice Activity Detection (VAD) technology to help UAVs recognize the pre-set command words. Our speech recognition module is capable of real-time control, high accuracy and good robustness in complex field environment. In this part, we will talk about the methods we use as well as the results we have achieved in the simulation environment.

The pronunciation of words is consisted of phonemes, which are used as units in speech recognition.Further division of phonemes leads to a more detailed phonetic unit, state.Because sound is a kind of wave, the collected signals need to be processed first. After de-noising and filtering, the voice is subdivided into frames and then frames are identified as states which will be further combined into phonemes. At last, the phonemes will be combined into words to realize speech recognition.

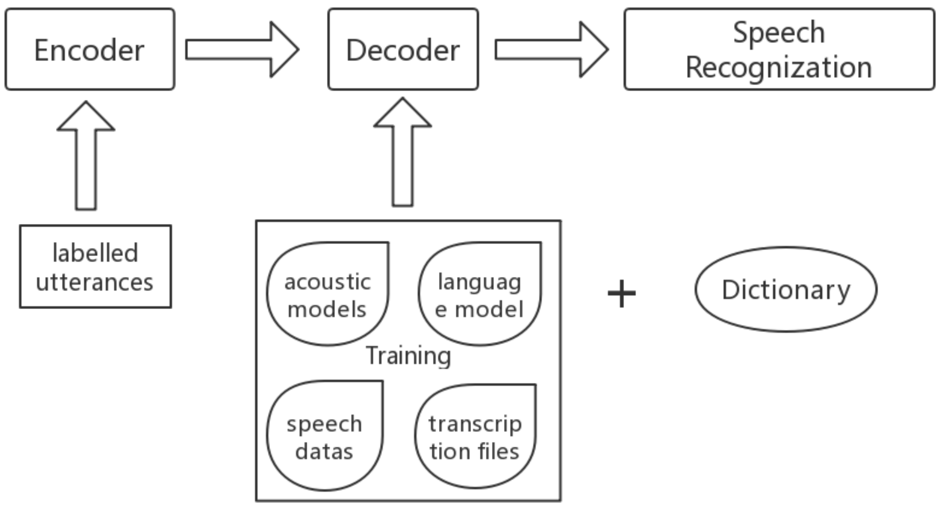


Figure 1: the structure of acoustic model

The existing acoustic models can generally be divided into two kinds: traditional mixed acousticmodels and end-to-end acoustic models using neural networks. In this project, we first try to use convolutional neural network (CNN) and Connectionist Temporal Classification (CTC) to construct a Deep-Learning-Based Chinese Speech Recognition System.

By using a large number of Chinese speech data sets for pre-training, voice is transcribed into Chinese phonetic alphabetfirst and then the phonetic alphabet sequence is converted into Chinese text through language model.

Before training the model, the wav speech signal is converted into two-dimensional spectrum image signal through frame-dividing and windowing operations, which is needed by the neural network and used as the input parameter of the acoustic model.The main body of the acoustic model is built based on Keras and Tensorflow referring to the structure of VGG net.Combining the model with CTC method at the output of the acoustic model, the sound waveform signal can be transcribed into Chinese Pinyin sequence. Finally, the sequence can be converted into Chinese text through statistical language model.

Experiments show that the prediction results of the model are of high accuracy. Since the pinyin-to-Chinese conversion uses a dynamic programming algorithm, basically the same as the searching algorithm for shortest path, it is easy to decode speech to text and complete the task of speech recognition.

1.3 QR code recognition

QR Code is designed to encode information that can be easily recognized and decoded by a computer. Due to its convenience and simplicity, it has now been widely used in fields like logistics, goods management and mobile payment. Apart from its easy-to-recognize feature for a computer, it is also unreadable to human eyes, thus can be used in scenarios where human interventions are not wanted.

In this mission, QR Code is used to encode the lock code. A complete QR Code is split into 4 quadrants and displayed on 4 iPad screens separately, each containing slightly more than one-fourth of the original complete QR Code. Only with all four quadrants combined can we decode the original message correctly.

To address this problem, we attach a camera to each of our robots. The robots will take photos of the four quadrants simultaneously under our operator’s voice control. The photos will be transferred back to the server with high-speed Wi-Fi. We then adopt various digital image processing techniques to identify and segment areas containing QR Code in each photo, and combine these four quadrants together to perform the recognition.

QR Code stands for the Quick Response Code. A QR Code stores information in the arrangement of black and while blocks. A typical QR Code, as shown in Fig 2, contains three relatively large black-and-white square rings used as anchor points to determine the position of the code area. It may also contain alignment blocks to provide information of alignment in large QR Codes.

Fig 2. QR Code for “7890”

The recognition of QR Code can be split into several steps including image denoising and enhancement, direction correction, code area extraction, key point extraction and decoding. QR Code standard offers different levels of error correction, at the cost of reducing capacity of storage.

a. Otsu binarization

Otsu binarization is a global thresholding algorithm originally proposed by Otsu in 1979 to perform automated clustering-based image thresholding. It was based on the idea of maximizing inter-category variance. Otsu thresholding often does well in scenarios where there are two distinct classes in the image.

In this mission, the QR Code pieces are presented on the iPad screens, which makes it a suitable fit for the Otsu thresholding algorithm.

b. OpenCV

OpenCV is an open source cross-platform image processing library. It provides various interfaces for basic image processing, computer vision, pattern recognition, etc. Latest techniques involving machine learning, virtual/augmented reality, robotics, etc. are also integrated into the project, making it even more powerful for developers. Starting from version 3.4.4, it supports QR Code recognition with its object detection library. Most of our algorithms are based on OpenCV.

2. Approach

We disassemble the task into three subtasks: obstacle detection, object recognition,gesture recognition, speech recognition and QR code recognition. The following describes the approach of tasks as follows,

2.1 Obstacle detection

Obstacle detection and avoidance is an important task in Unmanned Aerial Vehicles(UAV) flight control and navigation. Our aim is to achieve the detection and avoidance targets through the RealSense camera in order to avoid the UAV crash into the obstacle. The technologies we have applied including frame transmission, image alignment, get the core parameters, exponential smoothing and Kalman filtering.

About the frame transmission, this part of the work is reflected on the camera getting the size of picture and the rate.For getting the core parameters, we refer the official documents about the RealSense and we get the relation between the real depth and the depth matrix. As for image alignment, because we need the transmission between the world coordinate system and the camera coordinate system. This transmission will make the picture process easier.For about exponential smoothing, we recall the RTT settings and dynamic update in

Computer Network, so we use the same method to smooth the distance in order to avoid the sharpchange making the UAV instable.

2.1.1 Space conversion

The relative distance between the obstacle and the camera is needed in the calculation of force. Considering that the application range of the program is the UAV system with acceptable speed, the picture coordinate is transformed into the camera coordinate system.

Firstly, we understand that the process of camera imaging is to transform camera coordinates into image coordinates by projection, and then to form discrete imagery by discretization.Graphic coordinates can be obtained by inverse transformation of pixels from discrete to continuous, and camera coordinates can be obtained by inverse transformation of the relative distance between the graphic coordinates and each pixel point and camera.The formula is as follows.

$$\begin{cases} x = \frac{f x_c}{z_c} \\ y = \frac{f y_c}{z_c} \\ z = z_c \end{cases} \tag{1}$$

Among them,x_c、y_c、z_c are three-dimensional coordinates in discrete case，f is focal length，x、y、z are three-dimensional coordinates in continuous case.

Summarize the conversion formula：

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{2}$$

2.1.2 The image segmentation

Not all points in the image are obstacles in the flight path. After obtaining the specific three-dimensional coordinates,screening of obstacle points is necessary.

We turned the depth image to a 3-channel image by combing 3 same images together. Then we turned it into a grey image, adding a threshold to make it a binary image. The threshold would be the clipping distance, which is 3 meters in our program. In consideration of the instability of the image data, it would be hard to recognize if we only select the required part by the threshold of 3 meters. Also, there would be noise which interfere the valid region selection. Sowe used morphological operations to enhance the accuracy and reduce the noise. Dilation and erosion were used to do the close operation so that the small holes would be filled, adjacent objects would be connected and the boundaries would be smoothed.

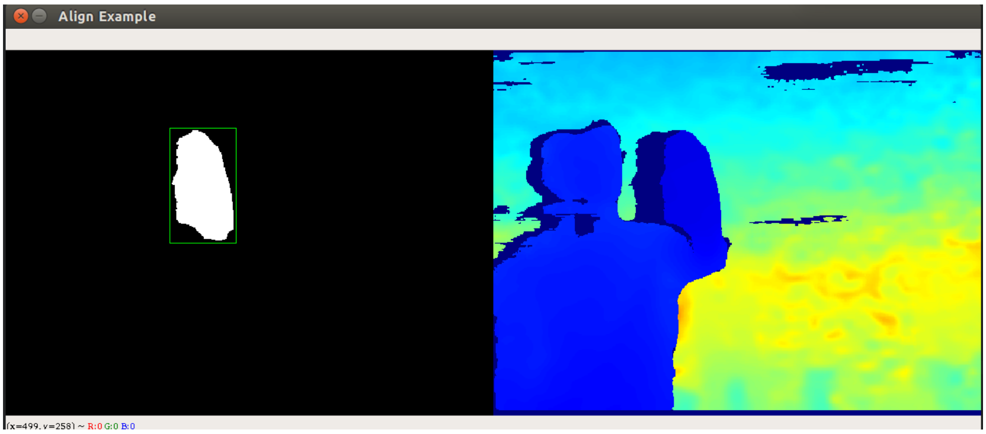


Fig 3. Result of image recognition

After the morphological operations, the selected image would be clear enough for further calculation. We calculated the angle between the camera and the points in the selected area. The angle would be useful for the repulsive force calculation.

2.1.3 The Force Model

The $F = \frac{k}{d^2}$ model is used to simulate the force situation, enlarging the influence weight of the close object on the obstacle avoidance process, and relatively diluting the situation of the long-distance object.The force is decomposed into the camera coordinate system, that is, the XYZ coordinate system. The formula can be as follows:

$$F_x = \frac{kx * z}{(x^2 + y^2 + z^2)^{1.5}}$$

$$F_y = \frac{ky * z}{(x^2 + y^2 + z^2)^{1.5}} \tag{3}$$

$$F_z = \frac{kz * z}{(x^2 + y^2 + z^2)^{1.5}}$$

Considering that the object is not a particle, the force equivalent to a point on the camera can be calculated by averaging the resultant force, so that the influence of obstacle shape on obstacle avoidance performance can be ignored.

Considering the relative position of the equivalent point in the obstacle avoidance object, it can be obtained that when the object is at a long distance; the equivalent point is in the center of the object. When the object is only at a distance, the equivalent point is near the camera, and the closer the object is to the camera, the more obvious the offset is.

The disadvantage is that considering the force of distant objects, when the environment of UAV flight is full of objects, all the objects in the range of camera perspective will produce force, which also includes the objects in the orbit of no longer UAV. However, because of the force jump in time domain, the smoothness of the force will be impacted when the object first appears on the orbit, so the filter should be considered to reduce the damage to the mechanical structure.

In order to solve the problem, we select the rectangular window to select the time-domain space, that is, the information of the XY axis in the camera coordinate system is selected symmetrically about the origin.In addition,the force model is moved downward to a certain extent (as shown below).

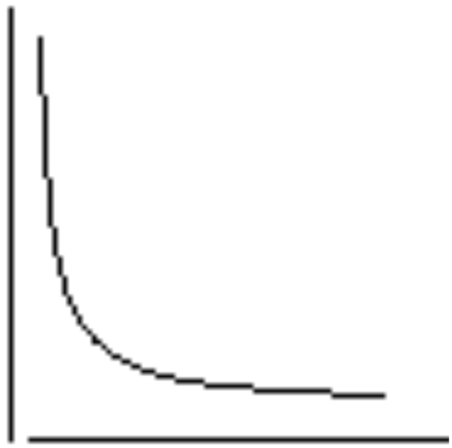


Fig 4.　Original Force model

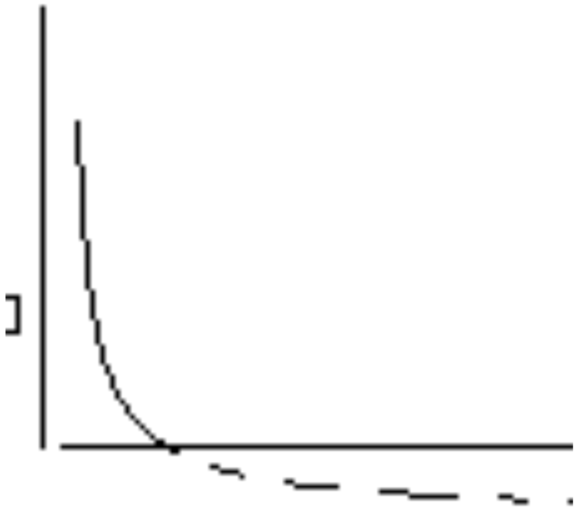


Fig 5. OptimizedForce model

Finally, we send the calculated repulsion to the UAV control system to complete the autonomous obstacle avoidance task.

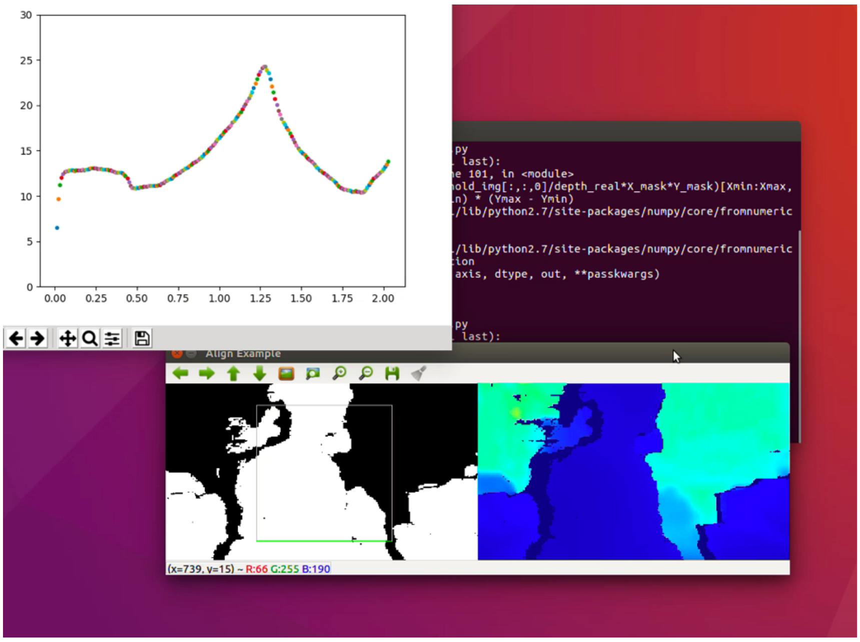


Fig 6. Final results

With space conversion, force model, spatial selection and image segmentation, the UAV gains much better performance while flying through different kinds of obstacles. This approach is easy for the UAV to compute and get proper avoidance strategy quickly, also the possibility to misjudge is reduced to a low level.

2.2 Object recognition

Before the competition, the size of boxes to be identified is unknown, so some input parameters need live debugging. What's more, floor, wall and ceiling patterns and lighting parameters are unknown, so depth information and color information will be used to varying degrees. In the debugging before the formal competition, we use the following box to do experiments.

Fig 7. Cartonusedin project development

Takingthe limitation of facilities carried by the UAV into consideration, the recognition of the coffer is supposed to be degenerated as a special session of obstacle avoidance, also processing by visual information. A customary way in automatic flying control or intelligent driving to deal with visual elements is deep learning. Meanwhile, traditional morphologic processing of digital pictures is also a simple but useful instant method. After discussion about the court and requirements of the competition, it is proposed that the environment is quite simple and clear with few disturbance and high identifiability. To follow the principle of simplicity, we launch an attempt of the second method.

2.2.1 Algorithm

An inspiring mature algorithm is Grabcut, which reaches a fair result under most of the circumstances while requires human being interaction. By drawing on the idea of this useful algorithm, we realized that the coffer always causes a sudden change in both distance and color on the background. Therefore, we use Sobel operator on RGBD pictures taken by RealSense camera to get the edge information:

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \qquad Gy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A \tag{4}$$

$$\mathbf{G} = \sqrt{\mathbf{G_x}^2 + \mathbf{G_y}^2}$$

By this way an initial prediction of barycenter is formed. Due to the simplicity of the background, to calculate the average color of relative far points or corner points is practical.

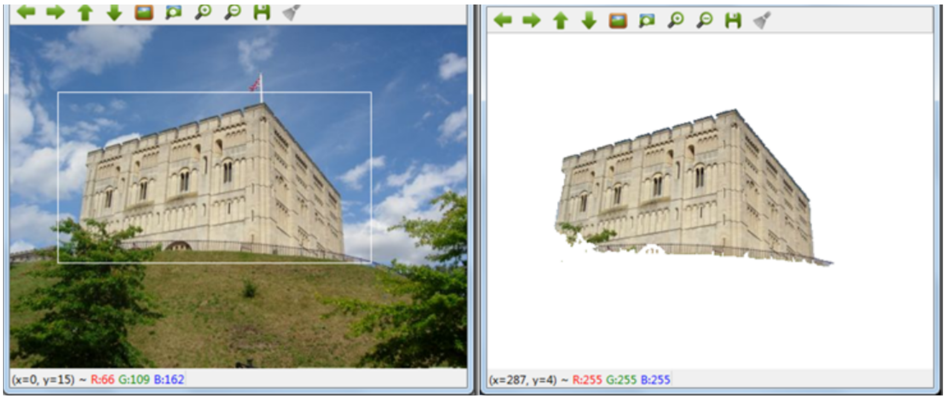


Fig 8. Example of Grabcut Algorithm

To estimate the foreground carving, the surrounding points of initial barycenter are screened by Euclidean distance in RGB space. Since there may be some small objects disturbing the color curving, the depth information and area of foreground are combined to get size reference: $S_r = d(p_{center}) * N(foreground)$. Set the recognizing threshold and avoid the disturbance.

When the UAV comes nearby the coffer, the barycenter of the coffer needs to be precisely reported in need of QR code recording and scanning:

$$x_{center} = \frac{\sum x_{count(i)}}{n(x_{count})}, y_{center} = \frac{\sum y_{count(i)}}{n(y_{count})} \qquad (5)$$

2.2.2 Experiment

To reduce the difficulty and enhance the robustness of the algorithm, we make the camera vertically downwards and thus get the coffer picture as a square. After receiving the coordinate result, the flight control module adjusts the position of the UAV to align up the center point and take the photo of QR code. The quality and efficiency of proceeding is not completely stable, so the coordinate of barycenter need to be low-pass filtered before sending out. Under the current solution, the frame rate is around 16, and we set 8 frames as a unit for judging and getting mean value.

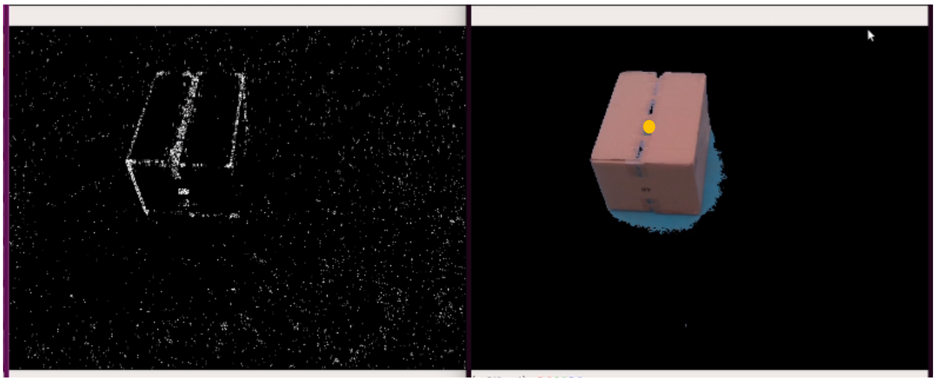


Fig 9. Coffer recognized

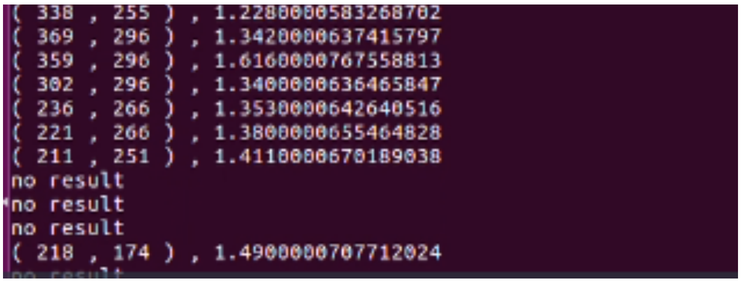


Fig 10. Console output of coordinate and depth

2.3 Gesture recognition

2.3.1Skeleton extraction

For gesture recognition, we use the combination of RGB image and depth image to segment the human body from the scene. By reasonably designing the middle layer representation of the body components, difficult pose estimation problems can be mapped to simple pixel-by-pixel classification problems.

However, training such classifiers requires a large amount of labeled data and a large number of hyper parametric experiments. So we used the middle layer firmware provided by the third party to get the skeleton joint of the human body. Using the information of the nineteen joints it provides, it is easy to get the distance between the joints:

$$d_{ij} = \sqrt{(\mathrm{x_i} - \mathrm{x_j})^2 + (\mathrm{y_i} - \mathrm{y_j})^2 + (\mathrm{z_i} - \mathrm{z_j})^2} \qquad (6)$$

After getting the distance between the joints, we can further calculate the angle between the joints. For any three joints, we can use the cosine theorem：

$$\theta_{i-jk} = \arccos \frac{d_{ij}^2 + d_{ik}^2 + d_{jk}^2}{2 d_{ij} d_{ik}} \qquad (7)$$

where θ_{i-jk} represents the angle between joint j and joint k in triangle ijk. Through the angle and distance, the various postures of the human body can be defined and identified by mathematical methods.

2.3.2 Algorithm

Based on the API provided by the third party, we can get the relative coordinate of skeleton joints in the image which is captured by a RealSense Camera. Although the API is designed to detect nineteen joints, several detected joints are not very stable and few joints even miss in some situations. Thus, we only select 5 joints, head, left wrist, right wrist, left side of the hipbone, right side of the hipbone, which can be detected stably as inputs for the algorithm.

Our algorithm is designed to recognize five gestures, left hand up, right hand up, hands up, stand and hands crossed on the chest. Here we denote the coordinate of each joint as (x, y, z) and the origin is the upper left corner.

In order to recognize 'hand up' gesture, we need to compare the y-value of three joints. For example, if y-value of the left-wrist is smaller than the head and y-value of the right-wrist is bigger than the head, the algorithm will judge the gesture is 'left hand up'. In addition, if we use more joints' information, we can even calculate the direction of arms and provide more gesture commands for flight control.

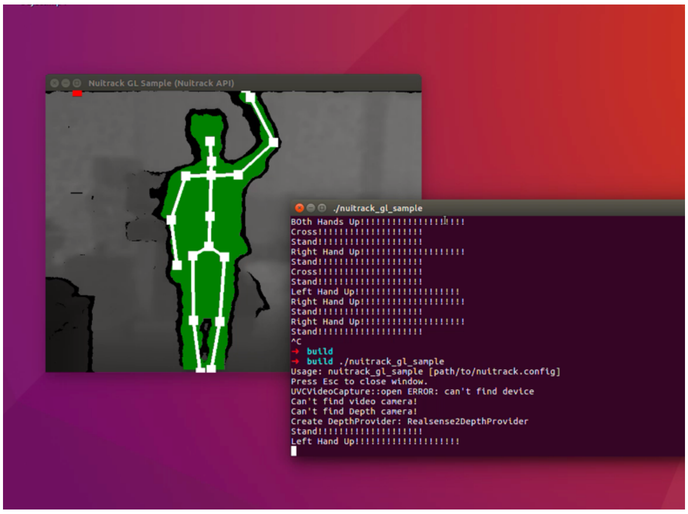


Fig 11. Reconition result: Left hand up

The algorithm for determining 'stand' gesture is similar to the abovementioned algorithm. We only need to compare the y-value of left-wrist, right-wrist, left-hipbone and right-hipbone. If y-value of the left-wrist is bigger than the left-hipbone and y-value of the right-wrist is bigger than the right-hipbone, the algorithm will judge the gesture is ‘stand’.

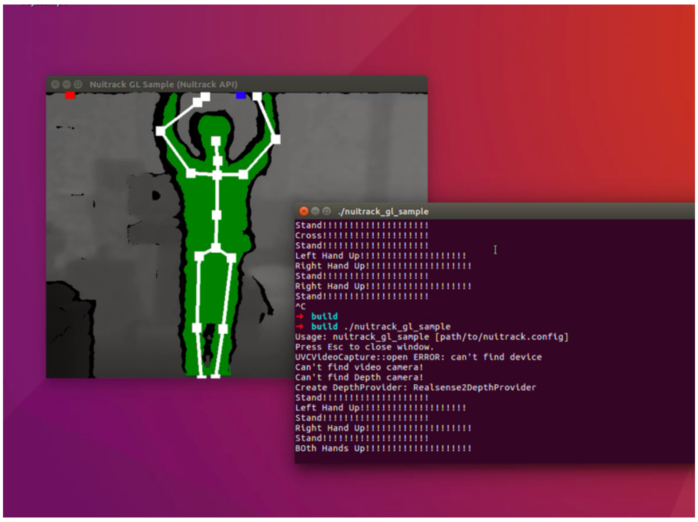


Fig 12. Recognition result: Both hands up

The algorithm for determining ‘hands crossed on the chest’ gesture is more difficult than abovementioned algorithms. Apart from comparing the y-value and x-value of left-wrist, right-wrist and core, we also need to use the cosine theorem to calculate the angle between left-wrist and right-wrist in triangle left-wrist- right-wrist- core. If the angle is within the set range as well as x, y-value of 3 joints, the algorithm will judge the gesture is ‘hands crossed on the chest’.

2.4 Speech recognition

Although using neural network to construct acoustic model can achieve good prediction results, recognizing every word emitted by the player may lead to long delays.

In fact, UAV does not need to “understand”every word of the player rather than be responsive to specific commands. The commands given to UAV are usually isolated words or a few restricted words, so the recognition of these words can be transformed into a simpler classification problem, using off-line speech recognition. We finally decide to use the off-line command lexicon instead of general speech recognition.

The real-time offline command word recognition system consists of two components. The first module makes use of voice activity detection techniques, aiming at dividing real-time input audio into valid human voice pieces. The second module takes the responsibility of recognizing human voice pieces and gives out recognition results with confidence probability.

We utilize voice activity detection(VAD) technique in the first module. VAD can detect the endpoints of input audio, based on which we can split the audio into several valid human voice pieces. VAD regards the detection of audio endpoints as a binary classification problem, which will classify the input audio into valid human voice class or environment noise class. The VAD detection principle is to divide the input spectrum into six sub-bands according to the spectrum range of human voice like 80Hz~250Hz, 250Hz~500Hzetc. After calculating the sub-bands' energy respectively, a logarithmic likelihood ratio function can be obtained by using the probability density function of the Gaussian model. The module will finally classify the audio taking globally and locally logarithmic likelihood ratio into consideration. We are able to get valid human voice audio pieces according to procedures above and then the audio pieces are sent to the second module for recognition.

In the second part, we apply the offline command word recognition module from Iflytek, which is a Chinese software enterprise specializing in intelligent voice and voice technology research. The application can take a short audio file as input and generate the recognition result with confidence probability offline, which is why we first split the real-time audio into pieces.

The module incorporates a special deep neural network called Deep Fully Convolutional Neural Network into traditional word recognition methods. According to self-defined command word file, the traditional recognition module will help map the neural network's output into command word space and the final result with confidence probability can be derived from that. The well-pre-trained recognition model is capable of most of command recognition tasks with low error rates. Even if sometimes the first module cannot successfully offer valid human voice piece, this module can give out recognition result with extremely low confidence probability so that the mistake from the first module can be corrected.

As for the programming language, the firstpart is accomplished with python while the second part is accomplished with Linux C. In order to connect these two modules, socket technique based on TCP/IP protocol is applied to accomplish the communication between python program and C program. What's more, some techniques are utilized to optimize the programs. We make use of multi-thread to prevent the problem of delay from word recognition. Besides, we set a confidence probability threshold which can be modified to guarantee that the accuracy should stay stable regardless of changes from environment.

According to the requirements of the competition, wedetermine the specific process of voice control and pre-define the instruction set.Speech recognition module can be used for the following scenarios:

1. The player can control the flight behavior of UAVs;
2. The playercan call the UAVs to block the attack of the sentry robots;
3. Theplayercan call the UAVs to shoot healing light;
4. Theplayercan call UAVs to identify the box, take photos and get password.

The specific process of using voice to control UAVs is as follows:

1. The player needs to wear portable wireless microphones. Due to the performance limitations of wireless microphones, the distance between the player and the ground station must be less than 40 meters in an indoor environment with obstacles.
2. The instructions of the player are sent back to the ground station by wireless microphone for identification processing. The ground station sends the instructions to the corresponding UAV through MAVROS protocol according to the recognition results, thus realizing the simultaneous control of single or multiple UAVs.
3. Only specific predefined command words can be recognized by speech recognition module. Speech recognition module can resist noise interference to a certain extent. All kinds of other noise in the environment or wrong commands issued by the player will not trigger command sending. Soonly the correct commands can change the UAV's motion behavior.

After adjusting the model to the optimal threshold, we integrate the speech recognition module with ROS, so that the speech recognition module can be carried by UAVs and achieve real control.

The effect of speech recognition module in simulation environment is as follows:

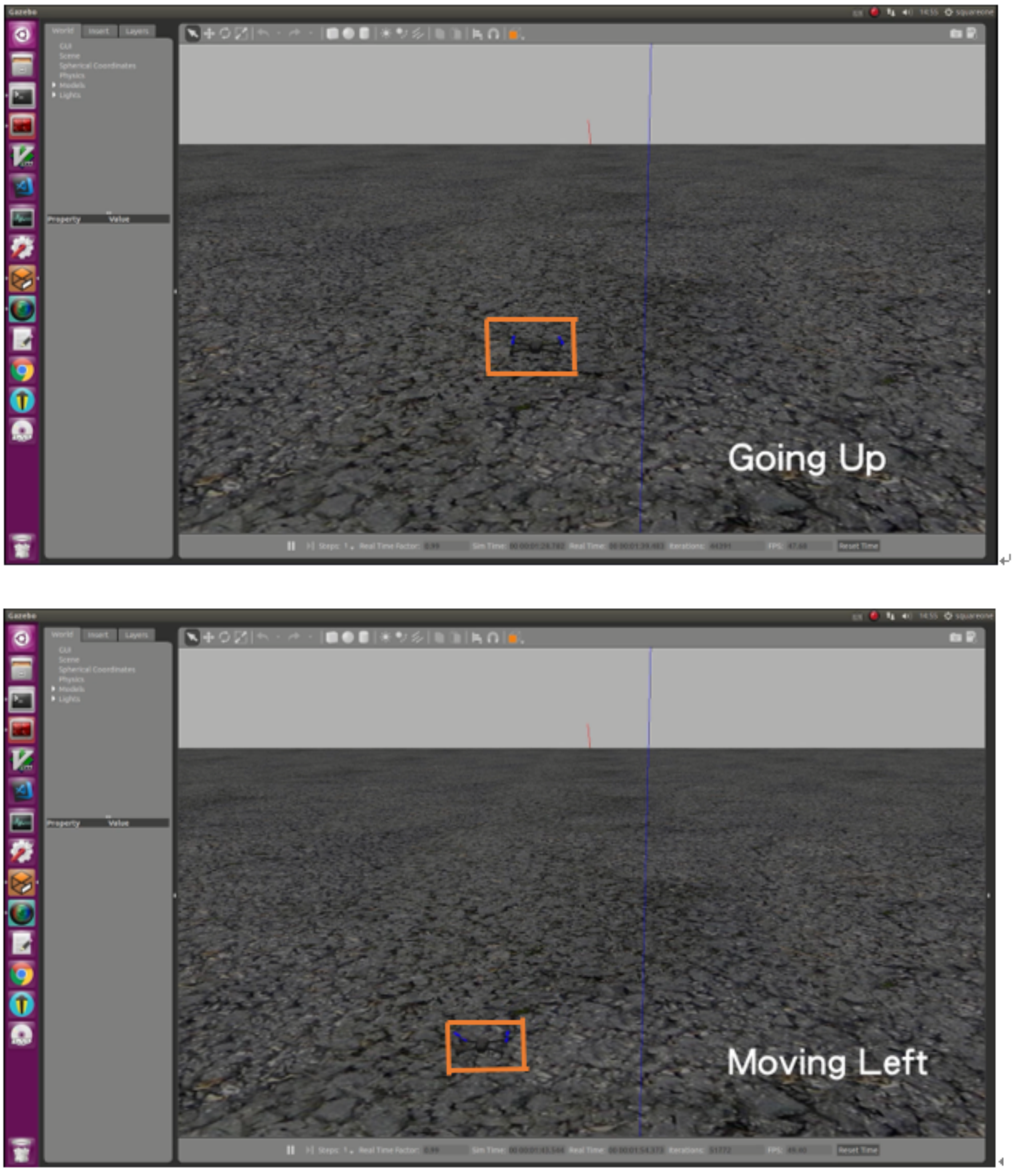
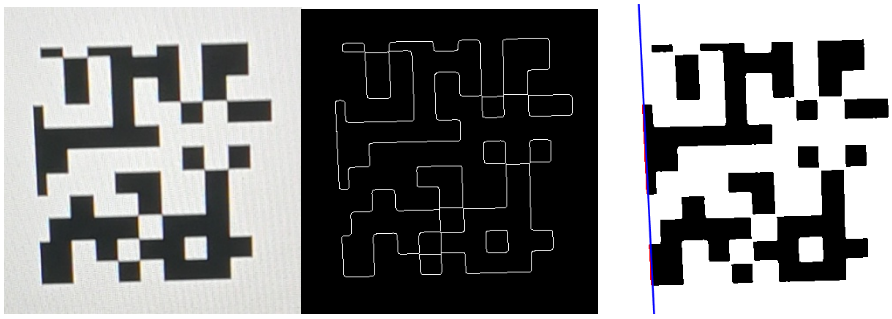


Figure 13. The effect in simulation environment

2.5 QR code recognition

a.Gradient-based edge detection

In this mission, a QR Code is split into four quadrants and displayed on four iPads separately. The screen of iPad will provide a relatively stable lighting condition, but also brings about problems such as reflections and moiré patterns. To better distinguish areas containing QR Code parts from those do not, we adopt gradient-based edge detection algorithms to extract the edges of the black and white blocks in a QR Code. To remove minor noises and moiré patterns, a pre-process containing image resizing and denoising is performed prior to edge detection.



(a) (b) (c)

Fig 14. The result of QR pre-process
a) original image, b) edges, c) regressed leftmost boundary

The widely-used Canny algorithm is good at detecting significant as well as minor edges in an image, thus tending to produce more noise. Therefore, we adopt Prewitt operator to perform edge detection. The magnitude of the gradient is used to determine the edges while the direction of gradient is used to correct the overall rotation of the image.

b. Boundary detection

Normally, a QR Code is oriented using the three or four large black-and-white squares on the upper-left, upper-right and lower-left of the code. These four squares are easy to detect, and can be used to determine the boundaries of the code. However, in this mission, the complete QR Code is split into four parts, leaving the three anchor points separated. In this way, the anchor points cannot be used to accurately determine the boundaries of the code area. Therefore, we adopt a different method based on morphological algorithms to determine the exact boundaries of the code area.

The goal of this step is to obtain the outermost boundaries of the area containing QR Code. Meanwhile, the boundaries have to be compact and exact, otherwise the quality of recognition cannot be guaranteed.

Prior to the boundary detection step, the image is rotated at a specific angle determined by the directions of gradients previously calculated. If the original image is not rotated in the first place, directions of gradients will mostly be focused around 0, 90, 180 and 270 degrees. On the other hand, if the image has been rotated before the step, whether it has been deliberately arranged so by the organizers of the competition, or it is due to the view angle of the robot while taking the photo, the direction map will produce a different distribution of angles. Thus, by looking into the histogram of the direction map, we are able to rectify the rotation of the image before continuing to the boundary detection part.

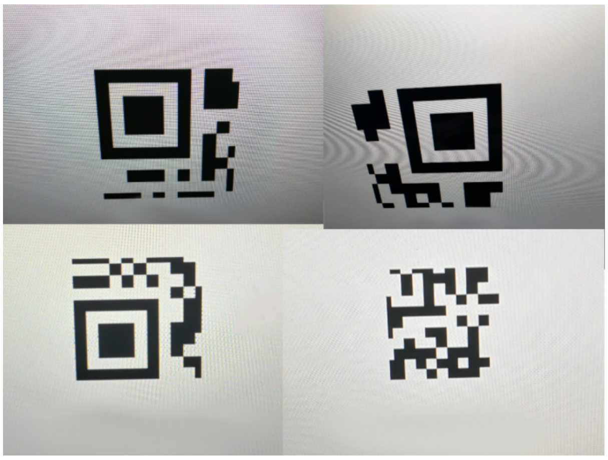
The boundary detection step uses the edges extracted in the previous steps. By scanning horizontally and vertically through the edge map, four outermost edges are extracted, corresponding to the left, right, top and bottom boundary of the code area. The extracted boundary lines are organized in the form of point lists, each containing the points consisting of the line. A linear regression algorithm is then adopted to calculate the equation represent the four boundary lines respectively. After the equations of boundary lines have been determined, the intersection points can then be calculated. The four points are the vertexes of the skewed QR Code area. A projective transformation is then performed to transform the selected area to a uniform square.

In addition to the algorithms described above, we also use Hough Transform for straight line detection. The Hough Transform algorithm produces decent outcomes most of the time, but may fail to detect some lines if they are too short. Therefore, we combine the two algorithms together to balance efficiency and accuracy. The major steps of our algorithm are illustrated in Fig (2) and Fig (3).

c. QR Code recognition

After all four quadrants have been extracted from the original image and transformed to uniform square, we can then combine them to produce the complete QR Code. The original images instead of the detected edges are uniformed. Then each uniformed quadrant is binarized using the Otsu algorithm to eliminate the influence of lumination. For the upper-left, upper-right and lower-left quadrants, the original anchor point square can be used to determine whether the quadrant should be rotated or not. For the lower-right quadrant, it will be rotated 90 degrees a time until recognition is successful.

Since each quadrant contains slightly more than one-fourth of the complete QR Code, all quadrants will have to be cropped in order to make the whole picture. A correlation operation is performed near the intersection part of the two upper quadrants after they have been correctly oriented. The overlying size is then determined and applied to crop all four quadrants. We then use OpenCV to extract the message in the QR Code and send it to our operator via Wi-Fi.



(a)

(b)

Fig 15. QR code reconstruction result

a) images from camera, 25% each, b) reconstructed

2.6 Rescue laser receiver

When player is hit by the sentry robot laser beams, the player will be hurt. After ten such hits, the player will be dead. So the helper robots should heal the player with a surgical laser. The laser module is designed to emit a healing laser the covers a big range. Also, this module should have a good interface to be easily controlled by on-board computer.

The healing laser is coded to be 650nm red light with the frequency of 13kHz . It is suggested to use a 555 chip to generate that frequency, according to the design manual. But we choose an Arduino Nano controller to do that, since it’s cheap and easy to be controlled. The circuit is shown in Fig 16 and 17.

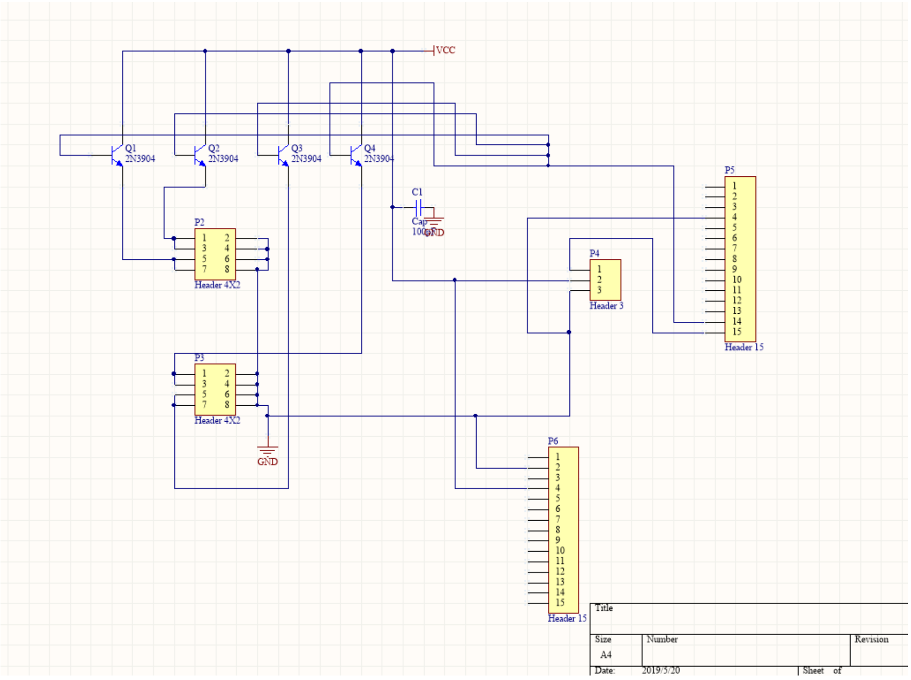


Fig 16. The shematic diagram of the transmitter circuit

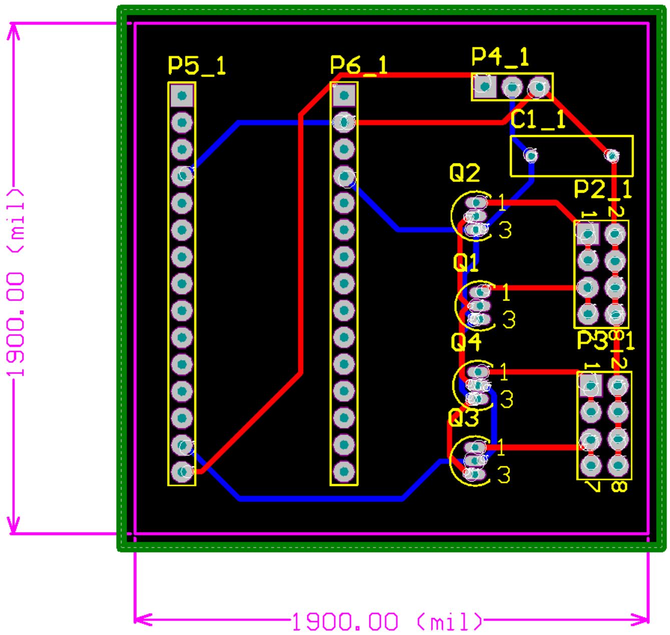


Fig 17. The PCB layout of the transmitter circuit

By setting up a timer of 0.038ms , the Arduino generates a pulse of approximately 13kHz. This pulse controls switch triodes powering the lasers, and therefore lasers are lighted at the same frequency. This healing laser module can be easily programmed to be a kill laser module, which can be used on a sentry robot for practices.

To make sure the laser hits the player, 8 lasers are used on this board.

This module is placed in a 3D-printed box, attached on the bottom of the helper robot. The interface is three pins. A GND pin and a +5V pin powers the module, and an enable pin to control the laser module.

We build a sensor helmet for testing and practicing.

Twelve ISO103 sensors are linked to an Arduino Nano controller. The controller will check all sensors one by one. If a pulse of certain frequency is detected, the controller will keep checking the same sensor, until it stops receiving that pulse, or the pulse continues for 1 second.

If this pulse continues for 1 second, the counter will increase or decrease, depending on the type of laser, by 1. Also the beeper will be enabled to tell the player that a hit or healing laser has occurred. Then the controller will stay inactive for the next 5 seconds.

The circuit is shown in fig 3.

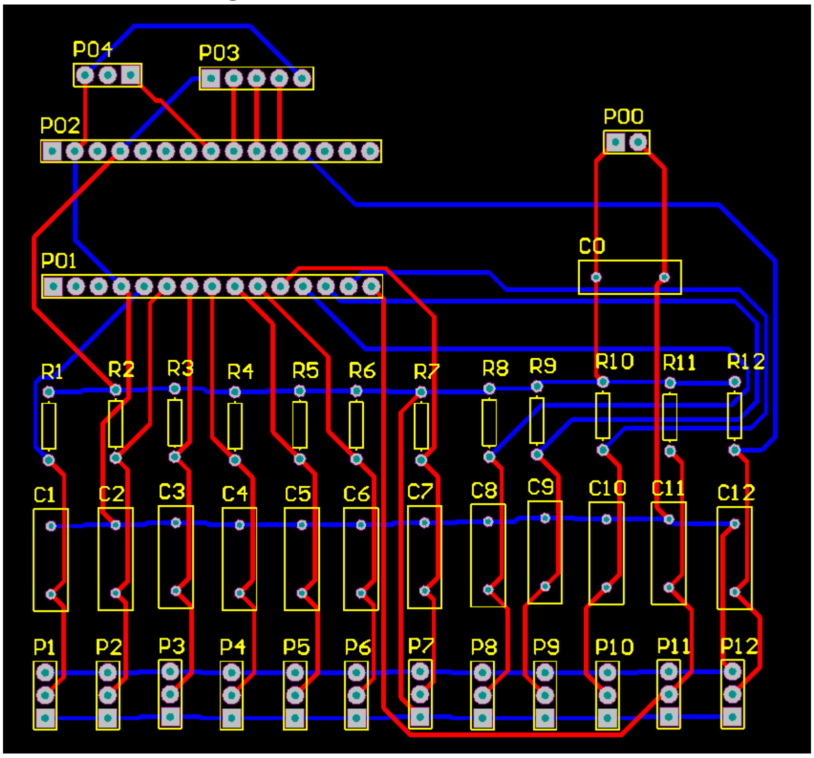


Fig 18. The PCB layout of the receiver

4. Conclusion

UAV combined with RGBD camera can complete a variety of complex tasks, especially the ability to interact with human-computer interaction.

The other part of the UAV system, such like speech recognition, QR code recognition and rescue laser receiver also works fine.

References

[1] N.Gageik, P. Benz, and S. Montenegro. Obstacle detection and collision avoidance for a uav with complementary low-cost sensors. IEEE Access, 3:599–609, 2015.

[2] Yoko Watanabe, Anthony Calise, and Eric Johnson. Vision-Based Obstacle Avoidance for UAVs.

[3] B. Ajith Kumar and D. Ghose. Radar-assisted collision avoidance/guidance strategy for planar flight, Jan 2001.

[4] Y.K.Kwag and J.W.Kang. Obstacle Awareness and Collision Avoidance Radar Sensor System for Low-Altitude Flying Smart UAV" Digital Avionics Systems Conference. October 2004.

[5] H. Chen, K. Chang, and C. S. Agate. Uav path planning with tangent-plus-lyapunov vector field guidance and obstacle avoidance. IEEE Transactions on Aerospace and Electronic Systems, 49(2):840–856, APRIL 2013.

[6] Otsu N. A threshold selection method from gray-level histograms[J]. IEEE transactions on systems, man, and cybernetics, 1979, 9(1): 62-66.

[7] Gu Y, Zhang W. QR code recognition based on image processing[C]//International Conference on Information Science and Technology. IEEE, 2011: 733-736.

[8] Gaur P, Tiwari S. Recognition of 2D barcode images using edge detection and morphological operation[J]. International Journal of Computer Science and Mobile Computing, 2014, 3(4): 1277-1282.

[9] Singh C, Bhatia N, Kaur A. Hough transform based fast skew detection and accurate skew correction methods[J]. Pattern Recognition, 2008, 41(12): 3528-3546.