# ROS-based UAV Control Using Hand Gesture Recognition

Yangguang Yu[1],Xiangke Wang[2],Zhiwei Zhong[3] Yongwei Zhang[4]

1. the College of Mechatronics & Automation, National University of Defense Technology, Changsha, 410072
E-mail: yuyangguang11@nudt.edu.cn

2. College of Mechatronics & Automation, National University of Defense Technology, Changsha, 410072
E-mail: xkwang@nudt.edu.cn

3. the College of Mechatronics & Automation, National University of Defense Technology, Changsha, 410072
E-mail: zhongzhiwei1987@163.com

4. the College of Mechatronics & Automation, National University of Defense Technology, Changsha, 410072
E-mail: zhangyongwei93@163.com

**Abstract:** As the first step towards the natural interaction between human and multi-UAV system, we present a novel human-UAV interaction method based on hand gesture recognition. We design different hand gestures corresponding to different UAV commands and propose a stable algorithm to distinguish these gestures. Then, considering the fact that the operation of multi-UAV system may need multiple cooperative operators, an algorithm based on position information and color information is designed to identify multiple operators. Thus multi-user recongnition by hand gestures is achieved. Lastly, we combine the results of hand gesture recognition of multiple operators with UAV control under Robot Operation System (ROS) framework and use AR_Drone platform to test the whole system's performance.

**Key Words:** ROS, UAV Control, Hand Gesture Recognition

## 1 INTRODUCTION

Recent years, unmanned robot system, such as UAV, got a booming development. But it has become a challenge that how to control them with a good real-time performance. In this thesis, we intend to find a both quick and reliable method to achieve that target. The commutation among human gives us enlightenment. Human can accurately and naturally express his purpose just using simple hand gestures. Actually, sending command using hand gesture is a good option both for human operator and UAV. For human operators, so many instruction sets are not needed and operators can command the UAV in a more natural way, which is very helpful in emergency situations. For UAV, it only needs a camera and a processor to understand the command, which is economical and easily accessible.

In the past, the problems of gesture and action recognition have received considerable attention [1]. Many methods are used to recognize hand gestures, such as histogram of local orientations [2] and a precursor of HOG [3]. Ying Yin proposed a novel method hand recognition algorithm based on flattened hierarchical hidden Markov Model [4]. In her thesis, she focused on how to smoothly handle two different kinds of gestures and determining how and when the system should respond to gestures. Aimed to avoid hundreds of training for hand gesture recognition, a transfer learning approach was proposed by Bellocchio E, Valigi P, et al [5]. This approach exploits both data downloaded from the web and gestures collected from other users and permits to learn a set of person-specific classifiers. When people make different gestures, the muscles on hand will have distinct changes. Based on this theory, Rui Fukui achieved hand

(a) Asus Xtion Pro Live camera (b) AR_Drone

Figure 1: experiment hardware

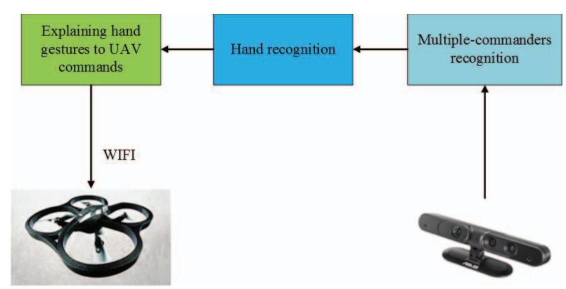
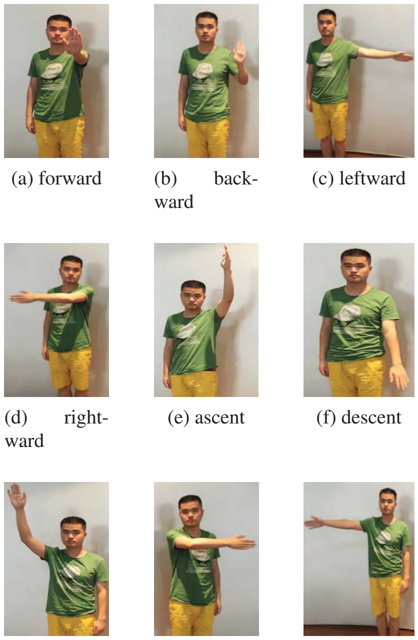



Figure 2: main framework of the overall system

gesture recognition using wearable wrist contour measuring device [6].

People have a long desire to achieve natural interaction with robots and many efforts have been paid towards this target. Gaze is used to regulate conversations between a humanoid robot and two human participants [7]. Valiallah (Mani) Monajjemi gave an example of creating, modifying and commanding teams of UAVs by an uninstructed human [8]. Using machine vision techniques, Jawad Nagi gave an approach which enables human operators to command and control Parrot drones by giving them directions to move,

978-1-5090-4657-7/17/$31.00 



(a) forward (b) backward (c) leftward

(d) rightward (e) ascent (f) descent

(g) hovering (h) rotating clockwise (i) rotating anticlockwise

Figure 3: the designed hand gestures

using simple hand gestures [9]. Onboard depth camera was also utilized to recognize simple hand gestures and many tasks can be achieved such as UAV person-follow [10].

Almost all the literature we read focuses on the interaction between UAV and single operators. But for UAV swarm, which usually has dozens of UAVs, it may needs multiple operators to work cooperatively. To solve this issue, operators' identities should be identified firstly. Therefore, we design a simple but effective algorithm to distinguish different operators based on position information and color information. As for hand recognition, we present a novel algorithm to recognize nine different hand gestures corresponding to nine different UAV commands. At last, we achieve the complete control loop from hand gesture to UAV control under ROS framework and final experiment proves the whole system performs well.

## 2 APPROACH

Our approach consists of three main components, namely hand gesture recognition, multi-operator recognition and UAV control. The main framework of the whole system is illustrated in Fig. 2. We employ an Asus Xtion Pro Live (see Fig. 1.a) to acquire the image. Firstly, we observe the fact that the operators usually won't move frequently. Therefore, we distinguish operators' identities based on their position. In order to avoid confusing, we choose to use a dictionary learning and sparse representation method for identity recognition when operators are close to each other. We mainly utilize the position variation of hand during the process of making gestures to achieve gesture recognition. Finally, we use the AR_Drone (see Fig. 1.b) as our platform and achieve the communication between the hand recognition and UAV control under ROS framework.

**Algorithm 1** HAND GESTURE RECOGNITION

**Input:** Current camera image $I$
**Output:** the hand gesture flag set $\mathscr{F}_g$

```
Get position of both hands
Compute the position error between current frame and last frame
if The error corresponding to a certain gesture exceeds a pre-designed threshold then
    Corresponding hand gesture count number increases by 1
else
    Set corresponding hand gesture count number zero
if The gesture count number reaches a certain level then
    Set the flag of corresponding hand gesture TRUE
else
    Set the flag of corresponding hand gesture FALSE
update all the hand gesture flags
return result
```

### 2.1 Hand gesture recognition

The basic states of UAV can be divided into nine types, i.e. moving forward, moving backward, moving leftward, moving rightward, ascent, descent, rotating clockwise, rotating anticlockwise and hovering. Therefore we design nine different hand gestures to represent nine different commands, depicted as Fig. 3.

To acquire the hand gesture images of operators, we choose the Xtion produced by ASUS, a kind of depth camera similar to Kinect. Using Xtion, we can acquire the color information as well as depth information of every pixels, which give us much convenience to distinguish different kinds of gestures. In fact, distinguishing different gestures only using vision images is a complex work. It usually needs thousands of lines code as it is often related with a series of image processing. But thanks to the Nite library developed by PrimeSense, we can save much time and energy for image processing and get the hand gesture in a more accessible way.

By using the functions in the Nite library, we can easily get the absolute coordinates of both hands in world frame. But as the position of the operator in the image is uncertain, we cannot judge the hand gesture directly by the position of hands. Therefore, we design a robust algorithm to distinguish designed hand gestures. The steps of algorithm are reported in Algorithm 1.

### 2.2 Multi-operator recognition

As our final target is to achieve the Human Robot Interaction (HRI) with multi-UAV system, some practical problems must be considered. One of the problems we faced is that the control for a large number of UAVs may need two or even more operators. Therefore, as the first step, we want to achieve distinguishing different operators and recognizing their hand gesture simultaneously.

Firstly, we consider the facts that operators won't move frequently or fiercely and usually hold a distance from each other when giving commands. Therefore, operators can be

**Algorithm 2** MULTIPLE-COMMANDER RECOGNITION

**Input:** Current camera image $I$ and image dictionary set $D$
**Output:** Identity_flag

**for** i=1, 2,..., $N_t$ **do**
  Detect the areas which belongs to operator i
  Cut the detected area into image patch set $X$
  $Identity\_flag[i] = 0$
  **for** j=1,2,..., $N_t$ **do**
    $TotalError_j = 0$
    **for** k=1,2,..., $N_s$ **do**
      $\alpha_k = arg\min_{\alpha_k} \|\alpha_k\|_1 \; s.t. \|x_k - D_j\alpha_k\|_2 \leq \varepsilon$
      $error_{jk} = \|x_k - D_j\alpha_k\|_0$
      $TotalError_j = TotalError_j + error_{jk}$
$Identity\_flag[i] = arg\min_j TotalError_j$
**return** result

idetified based on their position history. If position error between current frame and last frame is less than a certain threshold value, we can make a judge that the person detected in current frame should be the same person detected in last frame. But when operators stand close enough (for example, when operators are exchanging their position), this algorithm may won't work. Then, we consider the fact people usually wear different clothes and this condition is also very easy to satisfy. Thus, the vision feature of clothes is utilized to distinguish different operators. When the operators stand close enough, the method we use is dictionary learning and sparse representation. Firstly, as long as the process of recognition start, we record the first frame of image. Let $N_t$ denotes the number of operators and $N_s$ denotes the number of small patches that a image is cut into. Then we detect the areas that belongs to $N_t$ operators and cut corresponding area into $N_s$ small patches respectively. Then, we use these patches to form image directory set $D = \{D_1, D_2, D_3, \ldots, D_{N_t}\}$.

Finally, we use these image directories to make a distinction for the following image frame. The algorithm are reported in Algorithm 2.

But as the dictionary learning method require much computation and computation time, we will still distinguish multiple operators based on position information when operators hold a certain distance. At last, the complete algorithm of multiple-operator recognition is depicted as Fig. 4.

### 2.3 UAV control

After hand gesture is successfully recognized, we need to send the recognition result to UAV and transfer it to control command. The Robot Operation System (ROS) provides a good interface for message transformation.

In ROS, programs which have different functions can run simultaneously in different nodes. The message between different nodes can be transferred via a certain topic recognized by a unique name. In our work, two nodes runs simultaneously, namely the hand gesture recognition part and the UAV control part. The recognized type of hand gesture is sent from recognition part to control part via a

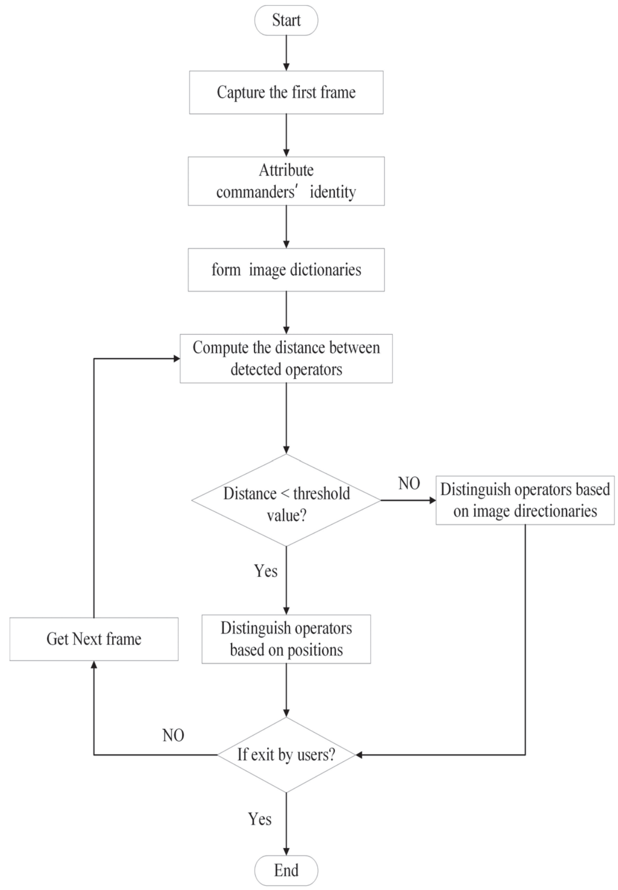



Figure 4: Complete algorithm of multiple-operator recognition

topic named /hand_gesture. Then the UAV compute the output of rotors according to different hand gestures received. The control structure graph is given in Figure 5. At the final step, we choose AR_Drone UAV, which is a small quadcopter UAV equipped with camera, ultrasonic sensors and IMU sensors, as our experiment platform. Using AR_Drone, we test the whole system's performance.

## 3 EXPERIMENTAL RESULTS

In this section, we will demonstrate the experiment results of our approaches. The goal of our approach is to achieve following functions: **(i)** the system can recognize the designed hand gestures with a low mistake rate. **(ii)** the system can be operated by multiple operators and distinguish their identity correctly. **(iii)** the overall system can allow person to control UAV using simple hand gestures.

### 3.1 Hand gesture recogniton

To evaluate the performance of our algorithm for hand recognition, we invited three volunteers who didn't take any training at all in advance to test the algorithm. After a simple explanation, volunteers made designed hand gestures arbitrarily and we made records about what hand gestures they made and the success rate. The experiment results are given in Table I. From the results, it can be seen that the average recognition rate is over 90%.

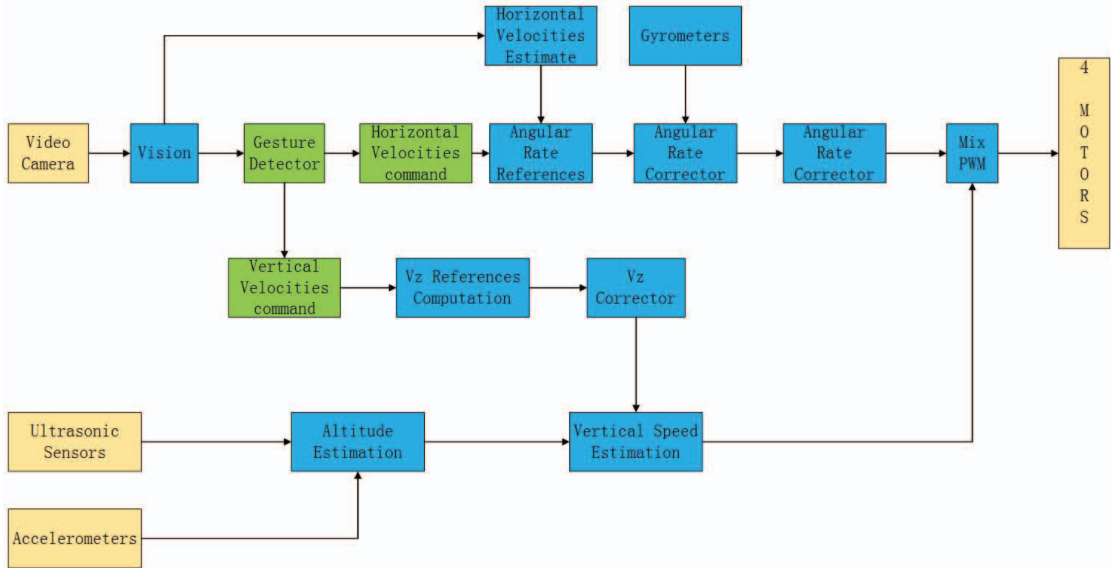



Figure 5: The control structure graph of the whole system

Table 1: EVALUATION HAND GESTURE RECOGNITION PERFORMANCE

| Commands | Person1 | Person2 | Person3 |
|---|---|---|---|
| **Forward** | 85% | 90% | 100% |
| **Backward** | 100% | 93% | 95% |
| **Leftward** | 100% | 100% | 100% |
| **Rightward** | 100% | 80% | 92% |
| **Ascent** | 100% | 100% | 100% |
| **Descent** | 90% | 97% | 100% |
| **Hovering** | 87% | 98% | 100% |
| **Rotating clockwise** | 95% | 100% | 100% |
| **Rotating anticlockwise** | 94% | 100% | 96% |

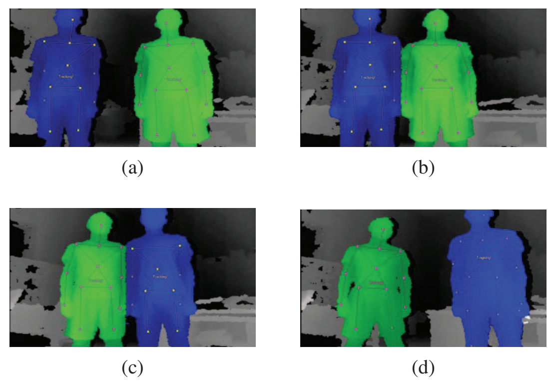



Figure 6: Multiple-operator recognition while operators are exchanging position

### 3.2 Multi-operator recognition

Considering there won't too many operators in practical situation, we tested our recognition algorithm in two-operator scenario. We designed a scenario where two operators are exchanging their positon and the recognition performance of system is recorded during this process. The results are depicted in Fig. 6. In Fig. 6.a, two operators stand holding a distance in the start. In Fig. 6.b, two operators are approaching each other. In figure Fig. 6.c, two operators have exchanged position successfully. In Fig. 6.d, two operators are departing from each other. During the whole process, the operators' identities were distinguished without mistakes. The result indicates that the recognition algorithm works well.

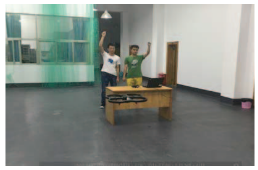
### 3.3 Full system

To evaluate the overall performance of the whole system, we firstly set some barriers in room as depicted in Fig. 7.a and hoped AR_Drone can fly through them under the hand gesture of single operator. Then, in order to verify the performance of multi-operator control, we designed a scenario where UAV was controlled by two operators. We supposed one operator was the main operator and the other was security guard, who can land the UAV in emergency situations. The operators gave gesture commands in front of the Xtion and controlled the AR_Drone as shown in Fig. 7.b. During the experiment, the AR_Drone flied smoothly and went through the barrier without much difficulty. We also tested the security guard land the UAV in emergency and the system worked well. The video attachment to this experiment for full system is available at [11]. The result clearly proves that hand gesture control UAV is an effective and promising HRI method.

## 4 CONCLUSION AND FUTURE WORK

In this paper, we present a novel method for controlling UAV, namely using the hand gesture. Firstly we solved the hand recognition problem based on hands position information and got good experiment results. In order to pave the way for controlling UAV swarm in next step, we

(a) single operator test



(b) multiple operators test

Figure 7: Experimental scenario

designed an algorithm to distinguish multiple operators. Lastly, we achieve the whole system under the ROS system and got expected performance. The experiment result indicates that using hand gesture to control UAV is an accessible and effective method. We believe this method bears a large potential for practical applications in the future.

In the future, we may continue our work in the following domains: i)our experiments were mainly conducted in indoor environments. In fact, we test the system in outdoor environments and the recognition performance is poor. We will improve the system adaptability for the outdoor environment in next step. ii) At present, we can just send some simple operators to UAV using hand gestures, which is somehow awkward. In the future, we want to achieve a smarter and more convenient controlling method for multiple UAVs using hand gestures.

## REFERENCES


[1] J. M. Chaquet, E. J. Carmona, and A. Fernandez-Caballero, A survey of video datasets for human action and activity recognition, Comput. Vis. Image Underst., vol. 117, no. 6, 2013, pp. 633659.

[2] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition, in FG, volume 12, 1995, pp. 296-301.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection, in CVPR, volume 1, June 2005, pp.886-893,.

[4] Yin Y. Real-time continuous gesture recognition for natural multimodal interaction. Massachusetts Institute of Technology, 2014.

[5] Costante G, Bellocchio E, Valigi P. Personalizing vision-based gestural interfaces for HRI with UAVs: a transfer learning approach, presented at Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. IEEE, 2014, pp.3319-3326.

[6] Fukui R, Hayakawa N, Watanabe M, et al. Hand gesture interface for content browse using wearable wrist contour measuring device, presented at Intelligent Robots and Systems (IROS) 2015 IEEE/RSJ International Conference on. IEEE, 2015, pp.1222-1229.

[7] B.Mutlu, T.Shiwa, T.Kanda, H.Ishiguro, and N.Hagita, Footing in human-robot conversations: how robots might shape participant roles using gaze cues, in Proc. of Int. Conf. on Human Robot Interaction (HRI), 2009, pp.6168.

[8] Monajjemi V M, Wawerla J, Vaughan R, Hri in the sky: Creating and commanding teams of uavs with a vision-mediated gestural interface, presented at Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013, pp. 617-623.

[9] Nagi J, Giusti A, Di Caro G A. Human control of uavs using face pose estimates and hand gestures, presented at Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction. ACM, 2014, pp.252-253.

[10] Naseer T, Sturm J, Cremers D. Followme: Person following and gesture recognition with a quadrocopter, presented at Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013, pp.624-630.

[11] `http://v.youku.com/v_show/id_XMjYwNTM1MzQ0NA==.html?from=s1.8-1-1.2&spm=a2h0k.8191407.0.0`.