WaveToFly: Control a UAV using Body Gestures

Shixin Li Henrik I. Christensen*

***Abstract*— Incompetent user interface limits the effective control of unmanned aerial vehicles (UAVs) and consequently prevents the wide adoption of UAVs. This paper proposes WaveToFly, a framework that leverages static and dynamic gestures to control a single UAV. Combining the state-of-the-art body skeleton detection tools and gesture detection algorithms, WaveToFly allows an operator to control a UAV in a natural, intuitive, and responsive manner even under an unstructured environment. WaveToFly can handle multiple gestures and is capable of scaling up with more commands with little effort.**

I. Introduction

UAV, being highly mobile and responsive in 3D spaces, are playing important roles in delivery, urban search and rescue, aerial photography, etc; however, being able to control a UAV in a natural, intuitive and responsive manner remains challenging.

In addition to traditional teleoperation by remote controllers, common control modalities include speech, gesture, facial expression, etc [1]. In this paper, we focus on using gestures to convey the commands to a UAV. Some prior works explore interactions through gestures: Waldherr et al. propose an interface that uses gestures to guide a service robot to specific locations [2]; Triesch et al. use hand tracking to let the user indicate how to grasp the object and where to put it [3]; Christensen et al. keep track of human hand positions for interaction between a human user and a service robot [4]; there are also projects that use gestures to control a swarm of UAVs [5], [6].

Considering the issues including change of lightning, blurring, etc, challenges still remain for using only visual inputs from the operator to enable seamless and stable control of a UAV in an unstructured environment.

Our goal is to design a gesture control framework that integrates human pose estimation, dynamic gesture recognition, static body gesture recognition, command mapping and sending. WaveToFly supports effective interaction even in an unstructured environment like outdoor fields; growing gesture command in the framework requires little efforts. Section II presents the pipeline. Section III introduces the human pose estimation algorithm and a Hidden Markov Model-based classifier for dynamic gestures. Section IV discusses the future work.

II. Framework

WaveToFly contains several modules: skeleton extraction, gesture classification, command mapping and sending. This section introduces the overall pipeline and the designed gesture modality.

*Department of Computer Science and Engineering, University of California, San Diego. La Jolla, CA 92093, USA. {lshixin, hichristensen}@ucsd.edu

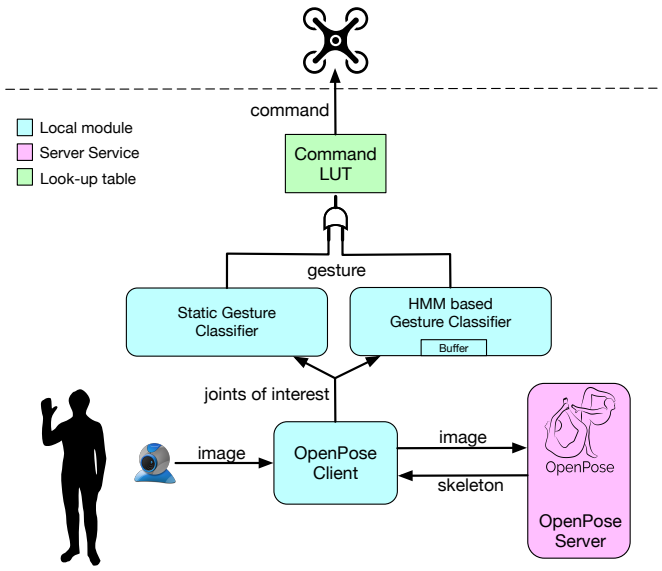

Fig. 1. The figure shows the pipeline of WaveToFly. We use an RGB camera to capture the image of the operator. The OpenPose [7] client receives the captured image and sends it to a server running OpenPose pose estimation service. The server returns the estimated skeleton back to the client. A template-matching static gesture classifier and an HMM-based dynamic gesture classifier further use the joints of interest to recognize. Static classifier has a priority over the dynamic classifier. If the classifiers recognize a legal gesture, the command look-up table maps the gesture to command and the program sends the command to the UAV.

Fig. 1 presents the overall pipeline of WaveToFly. The operator stands in front of a camera and conducts the gesture to indicate his or her intention. A local client program captures the images from a webcam and sends them to a remote OpenPose server via remote procedure call (RPC), and it receives back the coordinates of joints including neck, middle hip, elbows, and wrists. Static gesture classifier takes the upper-body skeleton and matches it with the predefined templates to recognize. Dynamic gesture classifier uses Hidden Markov Model [8] to recognize a dynamic hand gesture from a series of continuous frames. The program maps a gesture to a command using a look-up table and sends the command to the UAV by RPC.

We design several gestures as shown in Fig. 2. The corresponding commands are take-off, hovering, turning left, turning right, going up, going down and landing.

III. Technical Methods

Deep learning-based pose estimation algorithm provides a stable skeleton extraction approach. It equips the following gesture recognition module with robustness to deployment

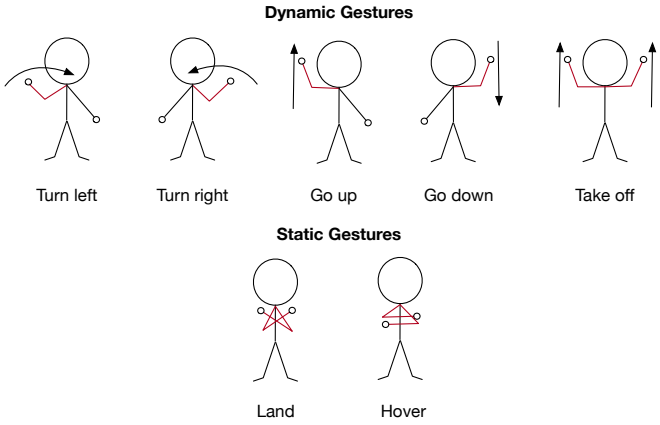

Fig. 2. The designed gesture and corresponding command.

Fig. 3. Pose estimation result of an operator standing in a messy office. The red dots on the images represent the joints of interest.

in unstructured environments. And the HMM-based gesture classifier is data-efficient.

A. Human Pose Estimation

OpenPose [7] realizes a real-time approach to detect the 2D pose of multiple people in an image. We adopt OpenPose to extract upper-body joint coordinates of the user in the image frame. For static gesture recognition, we use the joints of left wrist/elbow and right wrist/elbow. For dynamic gesture recognition, we use the left and right wrist coordinates. We also use joints of the neck and middle hip to normalize and re-center the coordinates. Fig. 3 shows a pose estimation result when the user is performing hand waving. OpenPose can produce a fairly good result on blurring images with a noisy background.

B. Dynamic Gesture Recognition

HMM is a popular tool to handle time-series data. It assumes the system being modeled is a Markov process with unobserved states. Each hidden state connects to the next state with a transition probability $T \in \mathbb{R}^{N\times N}$ with N being the number of hidden states, and generates an observation with either discrete or continuous emission probability. Since we are using wrist coordinates in image frame as observation variable, the emission probability is $B = \{b_j(x)\}$, where $b_j(x) = \phi(z_t;\mu_j,\Sigma_j)$ is Gaussian distribution with μ_j as mean and Σ_j as covariance. The Gaussian distribution is two dimensional because we are using both x and y coordinates of the wrist of the left or right hand.

Training procedure of HMM is to find a set of parameters to maximize the likelihood of the given K unlabeled observation series $D = \{z^{(k)}_{0:T}\}^K_{k=1}$. For left and right hand, we train HMM models respectively. In total, the number of HMM models equal to the number of gestures performed by left and right hand. During the test phase, each HMM model evaluates the probability of the observation series $p(O|\theta)$ as the score, where $O = \{z_{0:T}\} = \{x_t, y_t\}$ is a test series, and θ is the parameters of the model. The gesture corresponding to the model outputting the largest score is the classified result for the test data. To include identification of the case for "no gesture", we introduce a threshold so that the score below the threshold means no gesture recognized.

So far, we train four HMM models, which are corresponding to the gestures of left hand waving to right, right hand waving to left, left hand waving up and right hand waving up. We collect data from 10 people. The training set has around 40 instances for each gesture from 4 to 6 people, and the test set has around 15 instances for each gesture from 3 people. The test accuracy is 95.7%. It shows that our method is data-efficient, so introducing more gestures into the framework requires little effort.

IV. Future Work

Future research should consider using 3D human pose and include interpreting pointing gestures. We hope this framework can apply to controlling a swarm of UAVs. A promising application is to use gestures to deploy a subgroup of UAVs and control them to finish tasks like mapping and exploration.

References

[1] M. A. Goodrich, A. C. Schultz *et al.*, "Human–robot interaction: a survey," *Foundations and Trends in Human–Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2008.

[2] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, vol. 9, no. 2, pp. 151–173, 2000.

[3] J. Triesch and C. Von Der Malsburg, "A gesture interface for human-robot-interaction," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. IEEE, 1998, pp. 546–551.

[4] H. I. Christensen, D. Kragic, and F. Sandberg, "Computational vision for interaction with people and robots," in *Proc. Int. Conf. Mechatronics and Machine Vision in Practice*, 2001.

[5] J. Alonso-Mora, S. H. Lohaus, P. Leemann, R. Siegwart, and P. Beardsley, "Gesture based human-multi-robot swarm interaction and its application to an interactive display," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5948–5953.

[6] A. Suresh and S. Martínez, "Gesture based human-swarm interactions for formation control using interpreters," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 83–88, 2019.

[7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[8] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.