Fog-Based Visual Gesture Control and External Stabilization for Micro-UAVs

Anandarup Mukherjee[1]*, Sudip Misra[1], Nilanjan Daw[2], and Debapriya Paul[2]
[1]Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India
[2] Department of Computer Science and Engineering, Institute of Engineering and Management, Kolkata, India

***Abstract*—We propose a method for a single ground camera-based visual gesture control of a quadrotor mUAV platform, making its flight more responsive and adaptive to its human controller as compared to a human controller using keypads or joysticks for controls. The proposed camera-based gesture control scheme provides an average accuracy of 100% gestures detected, as compared to accuracies obtained using expensive Kinect-based hardware, or processing intensive CNN-based pose estimation techniques with 97.5% and 83.3% average accuracies, respectively. A fog-based stabilization mechanism is additionally employed, which allows for flight-time stabilization of the mUAV, even in the presence of unbalanced payloads or unbalancing of the mUAV due to minor structural damages. This allows the use of the same mUAV without the need for frequent weight readjustments or mUAV calibration. This approach has been tested in real-time, both indoors as well as outdoors.**

***Index Terms*—mUAV controls, Gesture-based control, UAV stabilization, Fog-based stabilization.**

I. Introduction

An Unmanned Aerial Vehicle (UAV) is a pilot-less flying vehicle or platform, which depends on remotely stationed pilots or autonomous algorithms for navigation and guidance. In this paper, we restrict our work to quadrotor micro UAVs (mUAVs), which have a twin-pair of fixed pitch, counter-clockwise rotors, located at four distal ends of the aircraft. The use of autonomous/semi-autonomous control makes it easy to handle the multi-rotors by a wide variety of users ranging from novice to experts. The use of autonomous platforms has applications in search and rescue operations, communication, surveillance, as well as military operations. However, in certain challenging scenarios – survivor localization in damaged buildings, checking the extent of damage, the presence of hazardous fumes in buildings or caverns – the most optimal solution is manual control. Manual control can easily account for the unknown nature of the flying environment in the path of a mUAV, in which autonomous algorithms fail, as these algorithms are mostly tailored for the general environment. A more robust algorithm for mUAVs requires the inclusion of more sensors to the mUAV system. This not only increases the mUAV's payload, thereby affecting its maneuverability, but also reduces its flight time due to the additional power requirements. The mUAVs deployed in such scenarios are generally minuscule, supporting only bare-basic sensors to allow for their restricted payload and easy navigation within confined spaces. In this paper, we present a line-of-sight based control method for mUAVs using visually recorded human-reflexive gestures. It is shown that gesture-based reflexive controls [1] provide more agility and better control as compared to joystick-based controllers [2]. The additional inclusion of fog-based gesture detection and control generation for the mUAV further enhances the UAV's capabilities. The command and control method of these mUAVs play

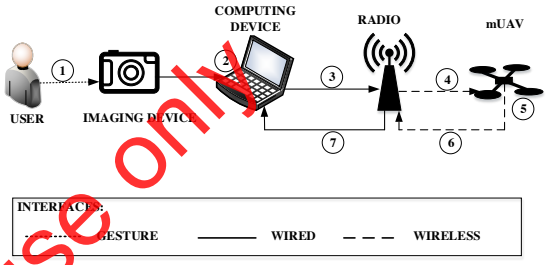


Fig. 1: A graphical work-flow for the proposed method

an important role in various mission-specific tasks. Operation of these quadrotors in GPS denied situations – *indoors*, and *cloudy outdoor conditions* – visually guided navigation is the only precise approach. However, in scenarios where a fast and efficient deployment of these mUAVs is required, dependence on external camera-guided navigation is not feasible. A visual line-of-sight (LOS) based manual control approach provides robust and precise control in previously uncharted and hazardous terrains without proving heavy on the mUAV processing and energy. Therefore, in our work, we keep the gesture-detection interface as simple as possible by choosing a video-based reflexive gesture detection method, which has a simple interface and avoids errors due to sudden or jerky movements, as is true with motion sensor based gesture detection units.

A. System Description

In our work, a quadrotor mUAV is tethered to a fog node (here, a computer) over a $2.4GHz$ wireless radio link, allowing the computer to send commands to the mUAV and the mUAV sensor values to be received by the fog node. The computer is set-up with a camera which allows a human-controller to manipulate the mUAV's flight based on reflexive hand-gestures to the camera. These simple visual gestures are translated to mUAV control sequences after being processed by a simple image-based gesture detection algorithm on the fog node. The functional steps used in our work are enumerated concerning Fig. 1 as follows:

1) A human user wearing a colored glove (Blue), gestures in front of the imaging device. For this

work, the gestures are restricted to a few directional movements of the gloved hand only – *Up*, *Down*, *Left*, and *Right*.
2) The imaging device is connected to a fog processing unit, which detects human-hand reflexive gestures using color-tracking.
3) Control signals for the mUAV are generated based on the detected gestures and are forwarded to a transmitter or radio module.
4) The radio module scans the presence of the mUAV, connects to it, and then forwards the control signals to it.
5) The mUAV interprets the control signals and actuates its motors accordingly.
6) The mUAV's onboard sensors periodically poll the system flight parameters (ψ, θ, ϕ) and status. These are then forwarded to the radio.
7) The radio returns the forwarded status signals to the processing unit, which interprets it and checks against the next set of control signals. Steps 3 to 7 form a closed loop control system, which stabilizes the mUAV and balances it. This protects the mUAV against external disturbances, as well as sudden and false over-actuation.

B. Contributions

The following are the contributions of our work:

1) A method for visual reflexive gesture-based control for enhancement of visual-range operational mUAV control has been developed, which works both indoors (under supervised lighting conditions) as well as outdoors (under unsupervised lighting conditions).
2) A network-based stabilization method has been applied, which detects and adjusts excessive imbalance due to flight maneuvers, or external factors like wind, or inter- nal factors like sensor noise, or structural damage.

II. Related Works

Ongoing works in the field of manual or semi-autonomous UAV control include gesture-based controls. Several gesture-based UAV control approaches use the popularly worked upon Kinect-based 3D-gesture recognition system for interaction with UAVs [3]. Lementec *et al.* [4] propose a gesture-based control system for UAVs using multiple orientation sensors. They implement it in a laboratory environment. In the previously mentioned works dependent on gesture recognition for remote control of a system, the task of utmost criticality is the correct detection of gestures and its translation to control signals. Valiallah *et al.* [5] implement a computer-vision based gesture recognition system to control UAVs in a multi-robot system. However, these systems either require an elaborate and costly setup or are confined to operations within a very close range.

The generation of control signals for remote operation is followed by stabilization of the UAV, which may be either sensory stabilization or stabilization of the system as a whole. UAV stabilization concerning its position and attitude have been extensively studied [6]. Previous works on UAV stabilization have mostly focused on hardware-based stabilization. Hoffmann *et al.* [7] implemented a flight control system for a quadrotor helicopter test-bed. They used a feed-forward compensation to serve aerodynamic flight motions for the effective rejection of large systematic disturbances. Their feed-forward compensation through the control loops of a *STARMAC* quadrotor protects the platform from any uncertainties in the model. In the control system of a small helicopter, Roy *et al.* [8] used an inner (position control)– external (altitude control) looped structure that stabilizes the hover flight of a UAV in the presence of wind-gusts. The outer loop is designed for robust back-stepping to control translational trajectory, while the inner loop employs a PID-controller for the stabilization of the UAV altitude. Similar works by Vinaykumar [9] report better results with Kalman filter on UAV control systems as compared to Alpha-Beta filters.

Synthesis: Vis-a-vis the works mentioned above and many more, our system provides a computationally effective, low-cost, easily deployable system for gesture-based control of mUAVs which achieves excellent results — both indoors as well as outdoors. Additionally, our approach works equally well on stabilized mUAVs as well as destabilized ones. Our approach does away with the need for mUAV recalibration or load readjustment each, and every time the UAV suffers minor damages.

III. UAV Dynamics

A. Flight Dynamics

Quadrotor mUAVs are generally four rotor aerial vehicles, each of which provides angular velocities ω_i to the mUAV for its motion. The four rotors are grouped diagonally into pairs of two. One pair rotates in the clockwise direction and the other pair in the anti-clockwise direction. This configuration cancels out the overall rotating effect of the rotors over the mUAV, thereby stabilizing it. The flight of mUAVs are generally governed by four parameters – *roll*, *pitch*, *yaw* and *thrust*. For a linear velocity v, the drag force F_D on the mUAV is calculated as $\frac{1}{2}\rho C_D A v^2$, where ρ is the environmental mean fluid density, A is the sweep-area of the mUAV's rotor blades, and C_D is a dimensionless constant. For an angular velocity ω, rotor-blade radius R and proportionality constant b, the torque due to the frictional drag τ_D is calculated as $b\omega^2$. Assuming a point source for force, the torque τ_z required along the z-axis is calculated as $b\omega^2 + I_M\hat{\omega}$, where I_M is the moment of inertia, $\hat{\omega}$ is the angular acceleration of the rotor, and b is the drag coefficient. Simplifying and assuming a steady flight, the total torque along yaw-axis (τ_ψ) is given by $b(\omega_i^2 - \omega_j^2 + \omega_k^2 - \omega_l^2)$.

Similarly, assuming any two motors i and j lying along the roll axis, the roll torque (τ_ϕ) is given as $Lk(\omega_i^2 - \omega_j^2)$, and the corresponding pitch torque τ_θ is calculated as $Lk(\omega_k^2 - \omega_l^2)$. In these equations L is the distance between the center of gravity of the quadrotor mUAV and any of the four rotor centers. The *anti-clockwise* and *clockwise* motions require equal thrust on the four motors, numbered from 1 to 4. The motion

along the z-axis is achieved by the change in propeller pair velocities. The resultant angular velocity for the corresponding motor, $k = 1, 2, 3, 4$, is denoted by ω_k. In case of actual motion along the xy-plane, during *left* or *right* motion, the two of the opposite motors should have the same $\omega_k|k \in [1,3],[2,4]$. However, the adjacent motors should have varying ω_k such that the following condition is satisfied:

$$\omega_k|k \in [1,3] \neq \omega_k|k \in [2,4] \tag{1}$$

B. Sensor Stabilization

During an mUAV's flight the on-sensors such as the gyroscopes providing the mUAV's orientation have a tendency to drift away from their initial calibration and need a constant stabilizing algorithm to counter this drifting effect. The on-board mUAV sensors are stabilized using a derived form of Kalman filter. This is used to estimate the parameters of interest, in our case, the on-board derived sensor values – ψ, θ, ϕ. The Kalman filters need only the previous and current state values to predict the correction parameters for future states. Considering the yaw axis, the state estimate for yaw (ψ) is denoted by $\hat{\psi}(k|k)$. Thus, $\hat{\psi}(k+1|k)$ is the estimation parameter for the immediate future step $\psi(k+1)$ such that

$$\hat{\psi}(k+1|k) = F(k)\hat{\psi}(k|k) + G(k)z(k) \tag{2}$$

where $\hat{\psi}(k|k)$ is the historical prediction value. $F(k)$ and $G(k)$ are experimentally determined constants. This estimation correction parameter is then used to predict the change in measurements needed, and is represented as $\hat{z}(k+1|k) = H(k)\hat{\psi}(k+1|k)$. The state estimate is again updated during a second iteration, and is denoted by $\hat{\psi}(k+1|k+1)$, which equals $\hat{\psi}(k+1|k)+w(k+1)v(k+1)$, where $v(k+1)$ is the measurement residue. For an expectation matrix $H(k)$, the Kalman filter gain is given by $w(k+1)$ and is calculated as

$$v(k+1) = z(k+1) - \hat{z}(k+1|k) \tag{3}$$

$$W(k+1) = P(k+1)P(k+1|k)H(k+1)^{'}S(k+1)^{-1} \tag{4}$$

The error covariance of the state estimates for the elements are denoted by $P(k|k)$ for $z(k), z(k-1)$, and $P(k+1|k)$ for $z(k+1), z(k)$. Additionally, the error estimation parameter is denoted by $S(k+1)$. The prediction estimation $P(k+1|k)$ and error estimation $S(k+1)$ are calculated as $F(k)P(k|k)F(k)^{'} + Q(k)$, and $H(k+1)P(k+1|k)H(k+1)^{'} + R(k+1)$, respectively. Thus, the estimated parameter value $\hat{\psi}(k+1|k+1)$ is a measure of the amount of compensation that is mixed with the yaw readings to make them reliable. Similarly, the estimated parameter values for pitch (θ) and roll (ϕ) are given by $\hat{\theta}(k+1|k+1)$ and $\hat{\phi}(k+1|k+1)$, respectively.

C. UAV Stabilization

The mUAV is stabilized by using the Proportional-Integral-Derivative (PID) controller scheme. The three correcting terms, viz., the proportional term, the integral term, and the derivative term are summed up to formulate the output as

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{5}$$

Here, K_p, K_i & K_d are constants of proportionality, and $e(\cdot)$ is a time-varying parameter, such that $e(\cdot) \longmapsto f(t)$. The Laplace transform ($L(s)$) of the above function is $K_p + \frac{K_i}{\eta} + K_d\eta$, such that η is the complex number frequency. The proportionality constants – K_p, K_i & K_d – are tuned manually until the optimum values are reached, which stabilizes the mUAV flight.

IV. Visual Gesture Detection

Gesture detection by visual means initially involves the acquisition of a three-channel Red-Green-Blue (RGB) color-image ($I \in (m \times n \times 3)$). A color-mask based background subtraction operation is performed, and the modified image is converted to a single channel gray-scale image ($I \in (m \times n \times 1)$). Subsequently, white noise is removed using blurring and morphological transformation on the gray-scale image. A threshold operation is then performed on the gray-scale image to obtain a binary image such that $\mathcal{I}(m \times n)|m, n \in (1, 0)$. Subsequently, a blue-color tracker is implemented for inferring the direction of motion of the blue-gloved hand. Since the RGB color space is highly susceptible to lighting changes, we port the image into the Hue-Saturation-Value (HSV) color space, which is much more stable than the RGB color space. A color-mask is prepared in the blue spectrum, which masks everything other than blue objects. For each of the pixels in the image $I(t)$, its value is obtained and is then subtracted with the pixels corresponding to the ones in the mask range value at the same position, and is denoted by $I(f) = I(t) - I(m)$, where $I(f)$ is the final image obtained after applying the color mask.

A. Threshold and Binarization

The 3-channel HSV image is converted to gray-scale intensity image ($I(f)_{m\times n\times p}$), which discards the hue and saturation values. The gray-scale image is then binarized using an unsupervised method, which considers the 0^{th} and the 1^{st} order cumulative moments of the gray-scaled image histogram. This converts the image pixels from 8-bits to 1-bit binary values, optimizing both spatial and temporal requirements for future calculations. The threshold range for image binarization is calculated by minimizing the weighted within-class deviation ($\sigma_w^2(t)$), where t is the threshold value calculated experimentally. The weighted within-class deviation is calculated as

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{6}$$

For the probability of background class occurrence, denoted by P_i, $q_1(t)$ and $q_2(t)$ are given by $\sum_{i=1}^{t} P(i)$, and $\sum_{i=t+1}^{I} P(i)$, respectively. We consider the variables $\mu_1(t), \sigma_1^2(t)$ and $\mu_2(t), \sigma_2^2(t)$ such that

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)}, \quad \sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \tag{7}$$

$$\mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}, \quad \sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \tag{8}$$

The threshold, thus obtained, is applied on the image $I(f)_{m\times n\times p}$, which converts it into a single-channel image $I(f)_{m\times n}$. This operation is mathematically represented as $I(f)_{m*n}$, which is similar to $I(f)_{m\times n\times p} \times \sigma_w^2(t)$. In the rest of the paper, $I(f)_{m\times n}$ is denoted as $\mathcal{I}$ for ease of representation.

B. Image Noise Filtering

A blurring operation followed by a morphological transformation of the blurred image is performed to rule out the effect of noise in detecting gestures. The gray-scale image $\mathcal{I}$ is passed through the first round of filtering, which is the blurring phase. The image pixels are convolved with a kernel ($f(x, y)$), which is described by a Gaussian Function in two-dimensions. This results in smoothing of edges and removal of pixel-level aberrations. For cells (n) in the flattened image matrix, the convolution function is mathematically represented as

$$f(x, y) * \mathcal{I}[n] = \sum_{m=-\infty}^{\infty} f[m].\mathcal{I}[n - m] \tag{9}$$

The Gaussian kernel $f(x, y)$ is generated as $\mathcal{I}.exp(-(\frac{(x-x_0{}^2)}{2\sigma_x^2} + \frac{(y-y_0{}^2)}{2\sigma_y^2}))$, where the amplitude is denoted by the pixel intensity, the center as (x_0, y_0), and the standard deviation is denoted as (σ_x, σ_y) in the x and y axes, respectively.

The morphological transformation uses mathematical set theory for the analysis of images. The blurred image $\mathcal{I}(f)$ is convolved with a kernel B, such that the minimal pixel value overlapped by B is assigned as the new value $\mathcal{I} \ominus B$ to that anchor point such as $\{z \in E | B_z \subseteq \mathcal{I}\}$, where B_z is the vector z's translation of kernel B and equals $\{b+z | b \in B\}, \forall z \in E$. This entire transformational operation is often termed as *opening of the image*. Opening results in the removal of multi-pixel aberrations or specks caused due to Gaussian noise. The image obtained at the end of this pipeline is mostly free of commonly afflicting noise in digital images.

C. Gesture Detection

Eventually, gesture detection and its subsequent recognition, involves the identification of the target gesture object. The gesture is computed using an oriented gradient signal, denoted by $G(x, y, \theta_a)$, for a binary image $\mathcal{I}$. The algorithm executes by anchoring a circular disc at the x, y coordinates, which is divided into two semi-circular discs inclined at an angle θ_a to each other. The gradient G is defined by χ^2 and is given as

$$\chi^2(g, h) = \frac{1}{2} \sum_{i}^{\infty} \frac{(g(i) - h(i))^2}{g(i) + h(i)} \tag{10}$$

The cell coordinates are denoted by $g(i)$ and $h(i)$, where i is the distance of the center of the circular disc to a point of interest in the image matrix. The result obtained is a possibility matrix for the position of the target object. A 2^{nd} order Savitzky-Golay filter is applied to enhance and smoothen the detection zenith's orthogonal to the angle θ_a. This operation increases the signal-to-noise ratio. This filtering is adapted here to remove noise in the probability distribution without distorting the matrix elements beyond permissible limits. This operation is mathematically abstracted as

$$Y_j = \sum_{i=-\frac{(m-1)}{2}}^{+\frac{(m-1)}{2}} \mathcal{I}_i y_{j+i}, \frac{(m+1)}{2} \leq j \leq n - \frac{(m-1)}{2} \tag{11}$$

$$= \frac{1}{35}(-3y_{j-2} + 12y_{j-1} + 17y_j + 12y_{j+1} - 3y_{j+2}) \tag{12}$$

where $\mathcal{I}(x, y)$ is a set of n tuples in the gradient matrix, y is a dependent variable governed by x, and is treated as a set of convolution coefficients. This matrix is then compared with the previously available gradient matrix to determine the course of movement, and eventually detect the directional gesture being executed by the user.

V. Methodology

The camera being used for detecting visual gestures records images at a rate of 30 frames per second (fps). This rate is too fast for generating command signals, which for our case, should be at a maximum rate of $10Hz$, which corresponds to the mUAV's sensor polling rate. An approximation function takes the mode of 10 consecutive frames and outputs the detected gesture as a signal to the mUAV at a rate of $3Hz$, which for our implementation is sufficient to control the mUAV without over-whelming its sensors. The proposed method is applied to the open-source micro-UAV platform – Crazyflie. A regular off-the-shelf webcam is used for gesture detection. The experimental flight tests are done, both indoors, under controlled lighting conditions and environment, as well as outdoors, under natural day-time lighting and environmental effects – winds and external EM noise. To gauge the effectiveness of our approach, we used an unbalanced mUAV, with an additional load on the front right quadrant of the mUAV.

A. mUAV Fog-based Stabilization

Real-time UAV flights pose some serious challenges as, if the UAV is unbalanced, the gyroscopes tend to drift away, resulting in the UAV turning turtle. To avoid such occurrences, a secondary fog-based stabilization mechanism is implemented, besides the regular hardware stabilizers. The telemetry data generated on-board the mUAV is fed-back into the gesture recognition system on the fog node. These telemetry values are used for generating new PID values, which are sent to the mUAV. It is a multi-step process, which begins with the generation of an estimation of the PID values required to fly the mUAV according to the user's visual gestures. The current values are used for generating an error estimate δ. For an estimated value $\mathcal{E}$, and feedback value $\mathcal{F}$, this error estimate is the correction needed in the flight stabilizers, and is given by $\delta_{\psi,\theta,\phi}$, which equals $(\mathcal{E}_{\psi,\theta,\phi} - \mathcal{F}_{\psi,\theta,\phi})(\mathcal{E}_{\psi,\theta,\phi})^{-1}$. The δ obtained is used as the correction parameter to generate the new pitch, roll and yaw values $\theta_{correct} = \theta_{feedback} + \delta_\theta$, $\phi_{correct} = \phi_{feedback} + \delta_\phi$, and $\psi_{correct} = \psi_{feedback} + \delta_\psi$. Finally, these compensated telemetry values are fed back into the mUAV for a stable flight.

B. mUAV- Gesture Control Loop

The mUAV stabilization system is forked into two main branches – 1) stabilisation of the mUAV as whole, and 2) stabilization of the on-board sensors. The sequence of events during over-the-network stabilization of the mUAV are represented in Fig. 2, and enumerated sequentially in the following stages:

Stage-0: The mUAV initializes with its original PID constant values (K_p, K_d, K_i). The initialization starts the

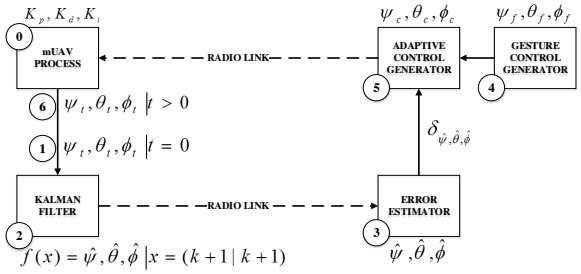


Fig. 2: The control system loop for our proposed method

on-board data-logging process, which starts polling all the sensors and actuators connected to the system.

Stage-1: The process generates the ψ_t, θ_t, ϕ_t values, which is the initial value at time $t = 0$, and generally, quite unstable.

Stage-2: A Kalman filter based hardware stabilization is applied which generates stabilized sensor outputs. The stabilized outputs derived from the parameters corresponding to the orientation sensor – Gyroscope – at instant k is given as $\hat{\psi}, \hat{\theta}, \hat{\phi}$, for the instant $k+1$. These state estimates are transmitted to the remote control-station over a radio-link.

Stage-3: The received state estimates from the mUAV are used to estimate the error between the state estimated value and the feedback value from the mUAV. The generated error $\delta_{\hat{\psi},\hat{\theta},\hat{\phi}}$ is forwarded to the next processing block.

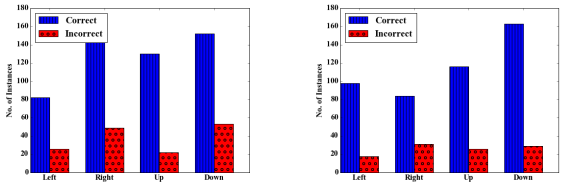
Stage-4: A gesture-control generator generates control signals for the mUAV in the form of ψ, θ, ϕ values. These generated values from the gesture control generator are referred to as ψ_f, θ_f, ϕ_f, respectively.

Stage-5: The ψ_f, θ_f, ϕ_f values, along with the error estimate $\delta_{\hat{\psi},\hat{\theta},\hat{\phi}}$, are used for computing the corrected command signals, which are sent to the mUAV over the radio link in the form of ψ_c, θ_c, ϕ_c.

Stage-6: For $t > 0$, the updated ψ_t, θ_t, ϕ_t control the mUAV and the whole cycle repeats from Stages 1 to 6.

VI. Experimental Results

We divide this section into two parts for adjudging the – 1) effectiveness of gesture detection, and 2) effectiveness of the fog-based stabilization on the mUAV. It is to be noted that only a part of the temporal telemetry values of the mUAV's flight parameters ψ, θ, ϕ are represented for ease of interpretation.



(a) Indoors (b) Outdoors

Fig. 3: Performance of various hand gesture detection without gesture approximation.

A. Effectiveness of Gesture Detection

Fig. 3 shows the performance of visual detection of gestures under various environments. Specifically, Figs. 3(a) and 3(b) depict the performance of gesture detection algorithms in indoor and outdoor environments, respectively. Fig. 3(a) shows the comparison of correct and incorrect detection instances of various gestures in an indoor environment, with stable lighting conditions. The incorrect detections appear for a fraction of a second and are attributed to little involuntary variations of the human hand concerning a comparatively faster image capturing device. Similar to the indoor environment, the gesture-detection method, when tested outdoors, provides the statistics shown in Fig. 3(b). Unlike the indoor environment with controlled lighting conditions, the outdoor gesture-detection proves to be more challenging. However, the fixed color-tracking of the gloved hand proves beneficial for outdoor gesture detection. Same as before, the approximation algorithm reduces the chance of the erroneously detected gestures to be translated to mUAV commands.

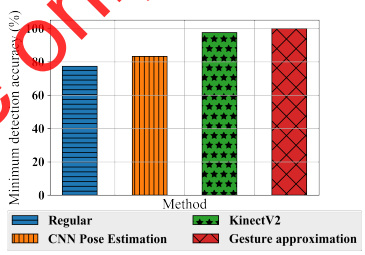
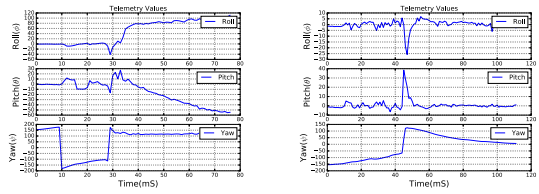


Fig. 4: Comparison of minimum gesture detection accuracies of the proposed scheme with other approaches.

Fig. 4 compares the effectiveness of the proposed gesture approximation method with the usual approach. Two additional methods are also compared with our approach — an expensive Kinect-based gesture detection scheme, termed Many Parameters Restriction Algorithm (MPRA), and a computationally expensive CNN-based pose estimation method. It is seen that the CNN-based pose estimation [10], and the MPRA-based UAV control scheme [11] give minimum gesture detection accuracies of 83.3% and 97.5%, respectively. Vis-a-vis, the proposed gesture detection approach without gesture approximation has a minimum correct detection accuracy of 77.4%, and with the gesture approximation scheme, which takes the mode of the gestures detected, achieves an accuracy of 100% due to exclusion of outliers generated as a result of chance misdetections.

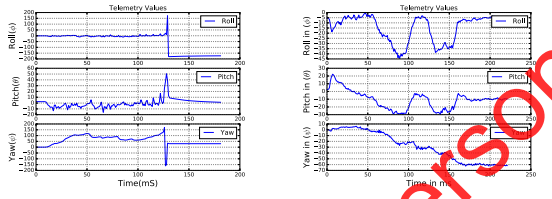


(a) Uncompensated Indoors (b) Compensated Indoors

Fig. 5: Effect of network feedback on gesture-based control of mUAV under controlled indoor lighting conditions.

B. *Effectiveness of Stabilizing algorithm*

Fig. 5 shows the mUAV flight parameters during gesture-based control in an indoor environment. The flight tests are carried out for two gestures – *left* and *right*. Figs. 5(a) and 5(b) represent the mUAV flight parameters during left gesture as command for unstabilized and network stabilized methods, respectively. In Fig. 5(a), it is seen that, upon detection of the control signal to move left, which primarily changes the yaw (ψ) of the mUAV, the sensor values change and stay in negative yaw for some time ($20ms$ approx.) before returning to the positive yaw axis. The pitch (θ) and roll (ϕ) values change because of the unbalancing load attached to the mUAV. However, in Fig. 5(b) it is seen that ψ changes gradually, allowing for much smoother control of the mUAV, as compared to the un-stabilized approach. The other associated parameters, θ and ϕ, show transgression from their stable value, but recovers quickly due to our implemented network-based stabilization. It is observed in our test flight that the mUAV sensors tend to overload and freeze beyond a certain value ($\phi, \theta, \psi = \pm 180°$). Providing control signals to the mUAV and forcing it to move in the same direction repeatedly, in an un-stabilized mUAV, causes it to lose control and veer-out uncontrollably. These sensor overloading and freezing problems are also avoided using our network-based stabilization approach.



(a) Uncompensated Outdoors (b) Compensated Outdoors

Fig. 6: Effect of network feedback on gesture-based control of mUAV during uncontrolled lighting conditions in an outdoor flying environment.

Fig. 6 shows the mUAV flight parameters during gesture-based control in natural lighting during outdoor flight. Figs. 6(a) and 6(b) represent the mUAV flight parameters during left gesture as command for the unstabilized and fog stabilized methods, respectively. In Fig. 6(a), upon receiving the control signal for turning left, for a larger duration of time, the sensory overload at $t = 125ms$. This overloading affects the ϕ values, causing it to behave abruptly, which eventually results in a crash. Crossing the ±180° causes the mUAV to become uncontrollable, and eventually crash. It is noteworthy to mention that the mUAV sensors tend to overload, which is attributed to the effect of wind on it. This causes the mUAV to be already unbalanced, even before getting control signals from the user. However, using the fog-based stabilization, as shown in Fig. 6(b), even the effects of wind do not destabilize the mUAV as is evident from the ϕ and θ values of the mUAV's flight. The effects of wind can be seen from the hump and valleys formed between $t = 90 - 150ms$ in the ϕ and θ values of Fig. 6(b).

VII. Conclusion

The work presented in this paper uses a low-cost approach, requiring minimal set-up time to establish a gesture-based control of a mUAV, which can be used in specific challenging and restrictive scenarios. Besides the low-cost gesture detection approach, which depends on reflexive gesture-based controls to attain impeccable command over mUAVs, our approach of employing a fog-based stabilization mechanism, in addition to the mUAV's onboard stabilization system, allows the mUAV to operate even when it is structurally unbalanced or unequally loaded. This network-based stabilization mechanism prevents the mUAV from flipping-over and crashing by restricting the user-generated commands to stay within limits of the normal sensor operating range to avoid overloading of the mUAV's sensors.

In the future, we plan on increasing the functionalities of the mUAVs by increasing the number of commanding gestures and employing this approach for the control of multiple mUAVs or swarms. Additionally, the deployment of more advanced hardware-based on-board mUAV stabilizing mechanisms is being worked upon.

References

[1] R. R. Yan, K. P. K. P. Tee, Y. Y. Chua, H. H. Li, and H. H. Tang, "Gesture Recognition Based on Localist Attractor Networks with Application to Robot Control [Application Notes]," *IEEE Computational Intelligence Magazine*, vol. 7, no. 1, pp. 64–74, Feb 2012.

[2] R. K. Megalingam, V. Rangan, S. Krishnan, and A. B. E. Alinkeezhil, "IR Sensor-Based Gesture Control Wheelchair for Stroke and SCI Patients," *IEEE Sensors Journal*, vol. 16, no. 17, pp. 6755–6765, Sept 2016.

[3] K. Pfeil, S. L. Koh, and J. LaViola, "Exploring 3D gesture metaphors for interaction with unmanned aerial vehicles," in *Proceedings of the 2013 international conference on Intelligent user interfaces*. ACM, 2013, pp. 257–266.

[4] J.-C. Lementec and P. Bajcsy, "Recognition of arm gestures using multiple orientation sensors: gesture classification," in *The 7th International IEEE Conference on Intelligent Transportation Systems, 2004. Proceedings.* IEEE, 2004, pp. 965–970.

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

[6] A. Mohamed, R. Clothier, S. Watkins, R. Sabatini, and M. Abdulrahim, "Fixed-wing MAV attitude stability in atmospheric turbulence, part 1: Suitability of conventional sensors," *Progress in Aerospace Sciences*, vol. 70, pp. 69–82, 2014.

[7] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control Engineering Practice*, vol. 19, no. 9, pp. 1023–1036, 2011.

[8] T. K. Roy, M. Garratt, H. Pota, and H. Teimoori, "Hover flight control of a small helicopter using robust backstepping and PID," in *10_{th} World Congress on Intelligent Control and Automation (WCICA), 2012.* IEEE, 2012, pp. 1688–1693.

[9] M. Vinaykumar and R. K. Jatoth, "Performance evaluation of Alpha-Beta and Kalman filter for object tracking," in *International Conference on Advanced Communication Control and Computing Technologies (ICACCCT).* IEEE, 2014, pp. 1369–1373.

[10] J. Bolin, C. Crawford, W. Macke, J. Hoffman, S. Beckmann, and S. Sen, "Gesture-Based Control of Autonomous UAVs," in *Proceedings of the 16^{th} Conference on Autonomous Agents and MultiAgent Systems.* International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 1484–1486.

[11] M. Dong, L. Cao, D.-M. Zhang, and R. Guo, "UAV flight controlling based on Kinect for Windows v2," in *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI),*. IEEE, 2016, pp. 735–739.