The multimodal speech and visual gesture (mSVG) control model for a practical patrol, search, and rescue aerobot

Ayodeji O. Abioye[1], Stephen D. Prior[1], Glyn T. Thomas[1], Peter Saddington[2], and Sarvapali D. Ramchurn[1]

[1] Faculty of Engineering & the Environment, University of Southampton, UK.
[2] Tekever Ltd, Southampton, UK.

Abstract. This paper describes a model of the multimodal speech and visual gesture (mSVG) control for aerobots operating at higher nCA autonomy levels, within the context of a patrol, search, and rescue application. The developed mSVG control architecture, its mathematical navigation model, and some high level command operation models were discussed. This was successfully tested using both MATLAB simulation and python based ROS Gazebo UAV simulations. Some limitations were identified, which formed the basis for the further works presented.

Keywords: mSVG (Multimodal Speech and Visual Gesture), aerobot (Aerial Robot), HCI (Human Computer Interaction), visual gesture, speech, MCPU (Multimodal Control Processing Unit), nCA (Navigational Control Autonomy), SBC (Single Board Computer)

1 Introduction

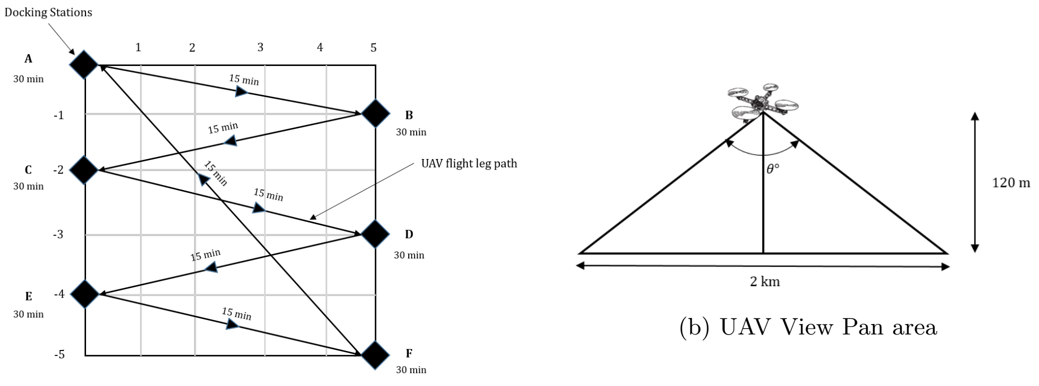
The rise in the popularity of small unmanned aerial vehicles (UAVs) in agriculture, aerial survey and inspection, transportation and logistics, surveillance and monitoring, search and rescue operation support, photography and videography, entertainment, sports, health care, law enforcement, environmental conservation, etc. has led to a significant increase in the number of operators, users, developers, researchers, beneficiaries, and other stakeholders with varying interest levels. This paper is interested in how the increasing leagues of human operators interact with small multi-rotor UAVs. According to [1], “It is clear that people use speech, gesture, gaze and non-verbal cues to communicate in the clearest possible fashion.” [2, 3] identified the need for smart and intuitive control interaction methods for aerobots (aerial robots) on higher nCA autonomy levels. This paper focuses on a UAV patrol, search, and rescue application scenario in the Alps. In this paper, an alternative method of interacting with multirotor aerial robots, otherwise known as aerobots [2], is presented.

Consider a case in which a small multirotor UAV is being developed to 1) patrol a dangerous region of the Alps, 2) provide signposting information to climbers, 3) alert search and rescue teams in case of any incidence, and 4) support search and rescue efforts or team operations. If such a patrol UAV would be required to interact with climbers when needed, how would the UAV register the climbers requests? As climbers would not normally have the UAVs RC controller, an intangible HHI-like multimodal speech and visual gesture interaction method would seem more suitable for such scenarios.

1.1 A case of the Matterhorn

An application of the mSVG interaction method on a patrol, search, and rescue robot could be the Alps in Southcentral Europe, where sporting activities such as climbing, mountaineering, hiking, cycling, paragliding, mountain biking, rafting, skiing, snowboarding, curling, and snow shoeing, are quite popular. People visiting these places for the first time could easily get lost if not very careful, climbers could fall when anchored on deceptively rigid surfaces, and people new to certain sports could get hurt, particularly when caught in stormy weathers. Therefore, patrol UAVs could be dispatched to assist local search and rescue operation efforts immediately after a storm. For example in the Matterhorn, about 1,700 rescue missions are conducted annually [4]. These high number of rescues is because the Matterhorn is home to numerous glaciers, which are laced with countless deep crevasses, many of which are hidden by snow that can give way without warning, swallowing up climbers and skiers in the process [4]. With the limited number of resources and man power, how can this operation be run more efficiently? Aerobots could be particularly useful in performing patrol, searching areas, and transporting small supplies. Patrol UAVs could fly along predefine routes, warning climbers of hidden crevasses, because signs on routes are easily covered by snow. The patrol UAVs could also be used to keep track of climbers progress during it regular flyovers. This could help find climbers in distress even before a call for help is made, which reduces the time between fall and call, potentially shortening rescue time, which in turn increases a rescued climbers survival chances - especially in situations where every second counts. In addition to this, a fallen climber may be unconscious, and therefore may be unable to call for help themselves, the UAV could make the call after failing to establish communication with the climber. Also, climber tracking information collected during patrol could be used to narrow down the search area in the event of an emergency or if a climber goes missing.

The UAV could also be used to provide verbal signposting, alerting climbers of their proximity to deep crevasses hidden by snow. Signs are probably unusable here because they could be easily covered by snow. According to [4] "people never know how close they are to the limits, every mistake they make could be their last". A typical signposting interaction could be - a UAV informs a lone climber, *"hello, deep crevasse 400 m ahead"*, and the lone climber could respond with *"ok"* as an affirmative or *"repeat"* to have the UAV repeat itself. Prior to the climb, all climbers and mountaineers would have already been briefed on appropriate UAV responses, and how to ask for help. They may also choose to opt out of being helped by the UAV during it routine fly over, if they think that this may be distracting, in which case the UAV avoids interacting with the climbers. The climber stick a *"don't disturb"* QR code patch on their backpack, which the UAV scans on approach and flies away instead, except it is an emergency. Patrol UAVs could also warn climbers of rapidly changing weather conditions and advise them on the nearest refuge points. In the event that the patrol UAV comes across a fallen lone climber, it could potentially alert climbing parties nearby to act as first responders, if they are quite close to the incident site, while also alerting the central control room of the emergency.



(b) UAV View Pan area

(a) UAV Patrol Grid

Fig. 1: 5 × 5 km^2 patrol area grid in six flight legs and with six docking stations.

An array of UAVs with front and downward facing cameras, thermal cameras, on-board navigational aid (GPS), on-board sense and avoid (proximity sensors), microphone, speaker, etc. could be used to execute this patrol operation as described in Figure 1a. Immediately after a storms, such small patrol UAVs augmented with on-board computation abilities via single board computers with microphone arrays, cameras, a speaker, and other sensors, could be dispatched to patrol and search specific hotspot areas in grids of 5×5 km^2. But how should such an mSVG control interface be effectively developed? This paper describes our approach in developing an mSVG control method for a small multirotor UAV. We developed a mathematical model, converted this into a program algorithm, tested this in MATLAB, and then performed a graphical simulation using a ROS Gazebo firefly UAV simulator. The idea is to run all the computation required for the mSVG on a single board computer, and couple this to the flight controller of a typical UAV, with the SBC communicating with the UAV flight controller via waypoint navigation (nCA Tier 1-III model).

2 Literature Review

Due to the complexity of the current HCI control interfaces, sometimes two operators may be required to control a single UAV [5]. For example, in a search and rescue mission, it may be difficult for the pilot to, simultaneously, control the UAV while effectively searching for missing persons. In order for robots to reduce human workload, risk, cost, and human fatigue-driven errors, it is crucial to make the human-robot interaction effective, efficient, and natural, through multiple modalities of contact, dialogue, and gestures [6]. The need for intuitive control interaction interfaces for aerobots beyond tier-one components of the nCA autonomy model opens up an opportunity to explore smart novel interaction techniques [3].

2.1 Multimodal Interface overview

It is often assumed that "Multimodal interfaces can support flexible, efficient, and expressive means of human-computer interaction, that are more akin to the multimodal experiences human experience in their physical world" [7–9]. Hence HCI researchers are constantly trying to find ways to endow computers, machines, and robotic systems with intuitive and natural multimodal interaction abilities similar to the human-human experience. This is possible because of the advancement in non-desktop embedded computing, more powerful mobile devices, and more affordable sensors [9]. According to [10], humans tend to exhibit more implicit behaviours, utilizing a combination of short verbal and nonverbal gestures in communicating their intentions, when performing tasks under stressed conditions, with resource constraints, and under time pressure, as is often the case in the space, military, aviation, and medical domains. [11] discovered that HERMES, a humanoid robot assistant, appeared more user-friendly, intelligent, and cooperative, when endowed with the ability to interact via a multimodal combination of speech, vision, and haptics. According to [12], soldiers often use a combination of verbal and visual lexicons, to communicate manoeuvres with each other, hence incorporating robots into these existing human ISR teams often presents a human-robot interaction challenge. [13] observed that speech control is particularly effective for task requiring short control communications, but may perform poorly for longer communication tasks, or in longer continuous operations. Therefore, multimodal interfaces are explored further.

2.2 Multimodal interfaces in aerial robotics

A multimodal speech and gesture communication with multiple UAVs in a search and rescue mission, was investigated by [14] using the Julius framework [15] and Myo device for speech and gesture respectively. The result of their simulation experiment showed that a human operator could interact effectively and reliably with a UAV via multiple modalities of speech and gesture, in autonomous, mixed-initiative, or teleoperation mode. [16] investigated the use of natural user interfaces (NUIs) in the control of small UAVs using the Aerostack software framework. Their project was aimed at studying, implementing, and validating NUIs efficiency in human UAV interaction. In their experiment, they captured whole body gestures and had visual markers (for localization and commands) via a Parrot AR Drone 2.0 camera, captured hand gestures via the Leap motion device, and speech command was captured via the ROS implementation of the CMU PocketSphinx library. These researchers demonstrated that natural user interfaces are effective enough for higher level UAV communication. [12] and [17] investigated the performance of a speech and gesture multimodal interface for a soldier-robot team communication during an ISR mission, even considering complex semantic navigation commands such as *"perch over there* (speech + pointing gesture)*, on the tank to the right of the stone monument* (speech)*"* [18, 17]. In a related research by [19], the researchers suggested that multimodal speech and gesture communication was a means to achieving an enhanced naturalistic communication, reducing workload, and improving the human-robot communication experience, especially when factoring in that only a minimal training is required to execute this communication method by the operator. [20] also investigated the effectiveness of speech and gesture communication in soldier-robot interaction. [21] observed participants' behaviour around UAVs and studied how the participants' interacted with the UAV, particularly how the users combined speech and two hand gestures in communicating control intentions to the UAV. [22] and [23] conducted elicitation study to determine intuitive gestures for controlling UAVs. [24] investigated human and UAV swarm interaction using spatial gestures.

3 Multimodal Speech and Visual Gesture (mSVG)

3.1 The mSVG Model

The mSVG technique is basically the multimodal combination of speech and visual gesture, a method that leverages familiar human-to-human type interaction in human aerobotic interaction. This combination could be simultaneous, sequential, or complementary. The underlying architecture of how this technique is designed to work is as described in Figure 2. Let the speech and visual gesture input be s and v respectively, and let f and g be the respective processing functions, which generates the control symbols $f(s)$ and $g(v)$ as shown in Fig 2.

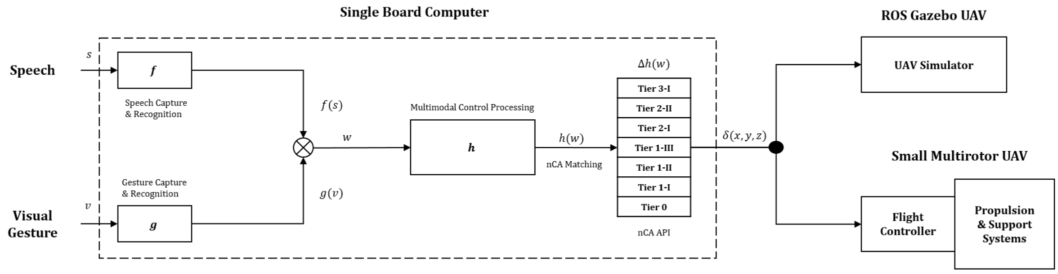


Fig. 2: mSVG design architecture control capture, processing, and execution.

Then the control command w processed at the multimodal control processing unit (MCPU) is a function of control symbols $f(s)$ and $g(v)$ as shown below:

$$w = f(s) \otimes g(v) \tag{1}$$

$h(w)$ is the resultant control output generated after the multimodal combination of both the speech and the visual gesture input at the MCPU. This is then passed on to an nCA autonomy model API to match $h(w)$ with the coordinate increment/decrement parameters $\delta(x, y, z)$ via a delta Δ parameter, depending on the nCA autonomy level of the UAV [3]. This is described by the following equation as:

$$\delta(x, y, z) = \varDelta h(w) \tag{2}$$

The delta parameter is a function generated by the nCA API to modify the MCPU output, $h(w)$, to enable compatibility with different nCA navigational control autonomy levels. For the Tier 1-III nCA model component, $\varDelta = 1$. Therefore, $\delta(x, y, z) = h(w)$. The coordinate increment/decrement parameters $\delta(x, y, z)$ specifies the change in the aerobots 3-dimensional position with respect to its current position.

3.2 Mathematical model

The processing operation in the multimodal control processing unit (MCPU) stage can be mathematically described through the use of relational set theories. The universal command set used consists of navigational and scenario commands, which are as presented in Table 1 and Table 2. The symbol "u" is a numeric modifier parameter specifying amount of navigational increment/decrement as used in Table 1. Control keyword and modifiers are also highlighted in block letters. The current position of the aerobot in the world environment is represented by the x, y,and z coordinate components. Where dx, dy, and dz are unit conversion parameter from simulation to world environment, for example $dx = dy = dz = 1$ in the simulation test.

Table 1: Navigaitional commands and control expresisions with example usage

S/N	Navigational Command	Control Expression	Command Example	
1	Forward	$x + udx$	Go FORWARD HALF metre	$x + 0.5dx$
2	Backward	$x - udx$	Go BACKWARD ONE metre	$x + dx$
3	Right	$y + udy$	Step RIGHT	$y + 0.5dy$
4	Left	$y - udy$	Step LEFT TWO metres	$y - 2dy$
5	Up	$z + udz$	Climb UP ONE metre	$z + dz$
6	Down	$z - udz$	Go DOWN HALF metre	$z + 0.5dz$
7	Hover	$z = u$	HOVER THREE metres	$z = 3$
8	Land	$z = 0$	Land	$z = 0$

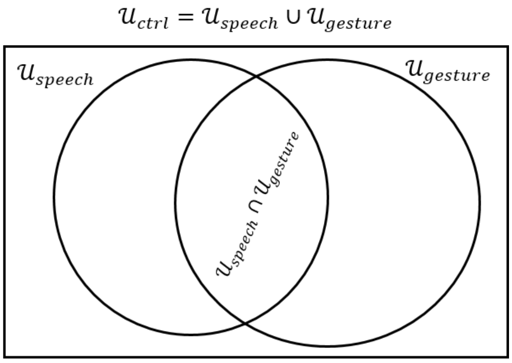

$u_{ctrl} = u_{speech} \cup u_{gesture}$
u_{speech}
$u_{gesture}$
$u_{speech} \cap u_{gesture}$

Fig. 3: Speech and Gesture control command set model.

Let us consider that the universal set of control commands u_{ctrl}, listed in Table 1 and Table 2, could be issued as either speech or gestures commands. Then the universal command set can be described as $u_{ctrl} = u_{speech} \cup u_{gesture}$. Where u_{speech} is speech only commands, and $u_{gesture}$ is gesture only commands. Figure 3 presents a set model describing this relationship. Commands that can be issued via either speech or gestures are represented as commands found at the intersection of both method $u_{common} = u_{speech} \cap u_{gesture}$.

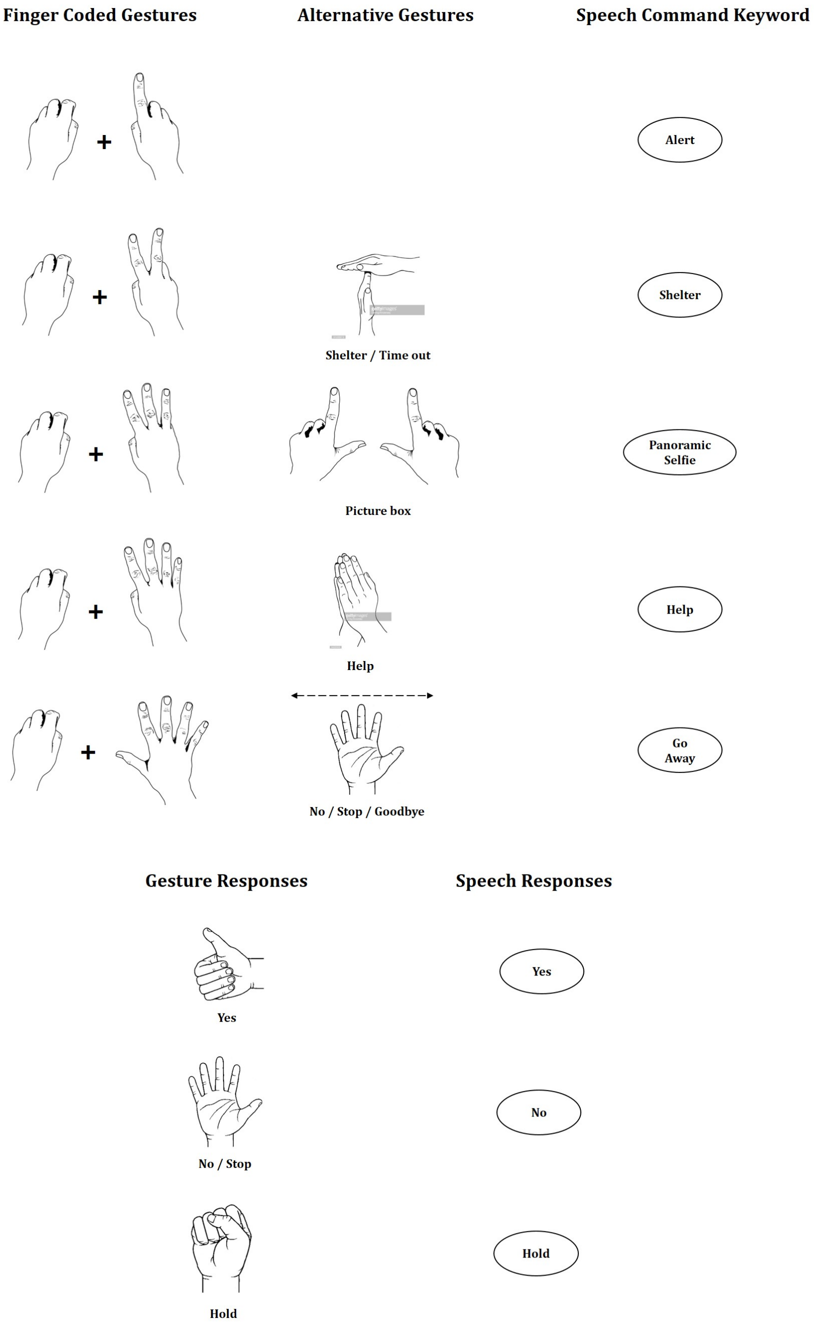


Fig. 4: Finger coded climber gestures and corresponding speech commands.

For the first phase of this work, all commands were implemented as speech commands, and only a few higher level scenario commands were implemented as gestures as described in Figure 4. Lets consider an hypothetical set of speech commands,

$$u_{speech} = [\{ok\}, \{weather\}, \{signpost\}, \{there\}, \{that\}, \{selfie\}, etc.] \tag{3}$$

Which can be denoted as

$$u_{speech} = [s_1, s_2, s_3, ..., s_n] \tag{4}$$

Where n is the total number of climber-user speech control vocabulary available. Similarly, $u_{gesture}$ could be an hypothetical set of climbers' visual gesture commands,

$$\begin{aligned} u_{gesture} = [&\{\text{ok - thumbs up}\}, \{\text{leave - wave away}\}, \{\text{what - open palm}\}, \\ &\{\text{that - point object}\}, \{\text{there - point place}\}, \\ &\{\text{selfie - picture board symbol}\}, etc.] \end{aligned} \tag{5}$$

Which can also be denoted as

$$u_{speech} = [g_1, g_2, g_3, ..., g_n] \tag{6}$$

Where m is the total number of climber-user gesture control vocabulary available. Using these notations, a typical series of control commands could be a sequence of

$$(t_1, G_1), (t_2, G_2), (t_3, S_3), (t_4, G_4 + S_4), (t_5, S_5 + G_5), (t_6, S_6), etc. \tag{7}$$

Where t_i is the sequential time component, S_i is the speech command component, and G_i is the gesture command component. These commands could be sequential, for example - (t_1, S_1) followed by (t_2, G_2) and so on; or simultaneous, as in - $(t_4, G_4 + S_4), (t_5, S_5 + G_5), etc.$ While sequential commands consist of only one gesture or speech components in each time component t_i, simultaneous commands consist of both speech and gesture component at the same time component t_i. In spatio-temporal terms, a speech and gesture command is considered simultaneous if the time between capture is no more than $0.5s$, otherwise it is considered a sequential command and one would be executed after the other. In order words

$$\text{Command Selection} = \begin{cases} \text{Simultaneous}(G + S) & \text{if } t \leq 0.5s \\ \text{Sequential}(G, S) & \text{if } t > 0.5s \end{cases} \tag{8}$$

Simultaneous command could be emphatic or complementary. A simultaneous command is emphatic if it repeats the same command using the alternative modality, whereas it is complementary when it provides additional information not given in the alternative modality. For example, *"Hold* (Speech) *+ Fist* (Gesture for hold)*"* issued within $0.5s$ apart only emphasis the command for the UAV to "hold" its position. Whereas a command *"Go Forward* (Speech) *+ Two-fingers* (Gesture for numeric modifier two)*"* issued within $0.5s$ of each other, results in the aerobot advancing two metres in the Forward direction. In this case, the gesture command complements the speech command.

3.3 High level command operation

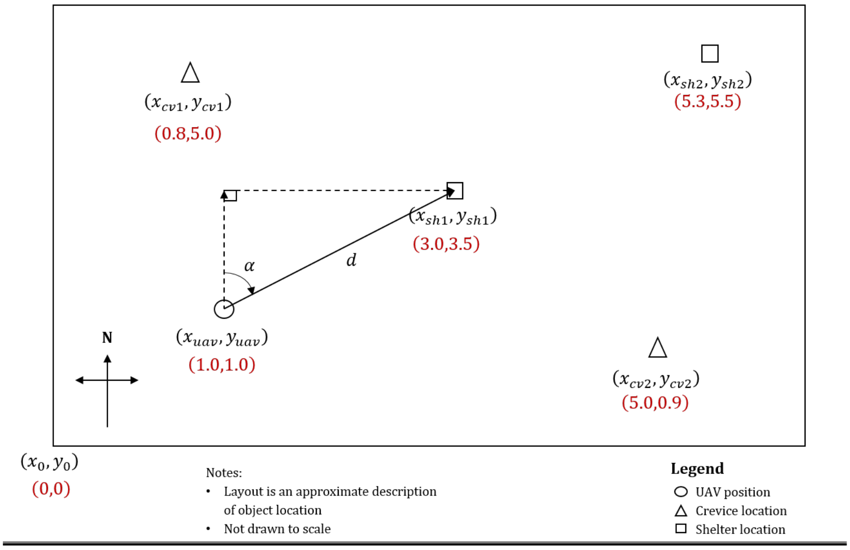
The last section discussed navigational control command operations. This section discusses examples of scenario command operations that could be executed by a search and rescue patrol aerobot in the wild. Navigation commands emulates low level nCA interaction while scenario commands emulates higher nCA level models. Table 2 presents some scenario commands.

Table 2: Scenario commands and synonyms

S/N	Scenario Command	Command Synonyms
1	Alert	-
2	Shelter	-
3	Panoramic Selfie	Panoramic, Selfie, Panorama
4	Help	Operator
5	Go Away	Away, Patrol

Shelter command computation

The UAV knowledge base includes the UAVs world map shown in Figure 5, as a computable lookup table, accessible during multimodal control processing. Figure 5 shows the crevice and shelter map/table implemented in the MATLAB and Python based Gazebo Simulation test.



Notes:
- Layout is an approximate description of object location
- Not drawn to scale

Fig. 5: UAV world map showing UAV position, crevices, and shelter locations.

From Figure 5, the relative North degree direction of the crevices, shelters, and other objects of interests can be computed as follows:

$$\tan\alpha = \frac{y_{sh1} - y_{uav}}{x_{sh1} - x_{uav}} \tag{9}$$

Therefore, the bearing angle

$$\alpha = \arctan(\frac{y_{sh1} - y_{uav}}{x_{sh1} - x_{uav}}) \tag{10}$$

The distance between the UAV and the shelter shown in Figure 5, can be computed as

$$d^2 = (x_{sh1} - x_{uav})^2 + (y_{sh1} - y_{uav})^2 \tag{11}$$

Therefore, the nearest shelter to climber can be said to be *"d km in the $\alpha°$ North direction"*, where x and y components are measured in km. In general, the equations for computing the distance and direction of any object located at point 'x, y' on the map, from the user/climber/operator/UAV 'x_{uav}, y_{uav}' are

$$\alpha = \arctan(\frac{y - y_{uav}}{x - x_{uav}}) \tag{12}$$

$$d = \sqrt{(x - x_{uav})^2 + (y - y_{uav})^2} \tag{13}$$

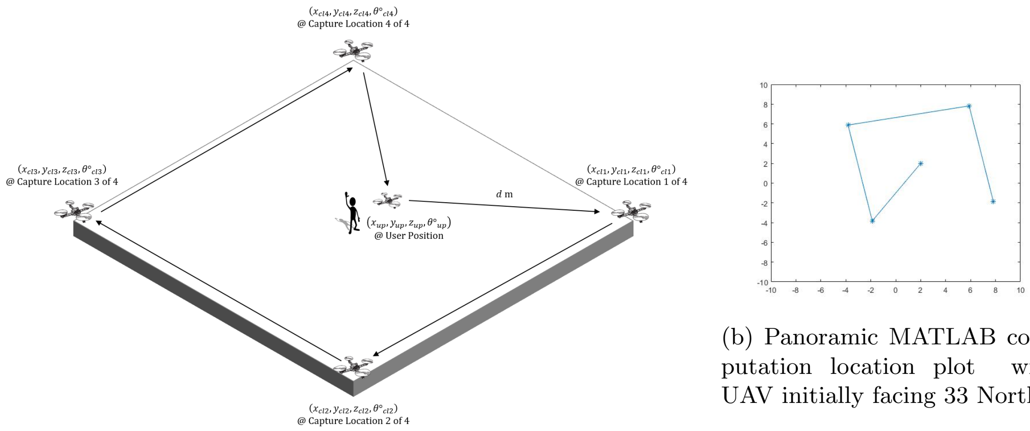
Interpreted as the map object of interest is *"d km in the $\alpha°$ North direction"* from the user. In addition to these, the UAV could beam a *"red arrow"* light on the ground with the arrow pointing in the direction of travel. Also, the UAV could travel the initial 50 metres with the climber, before flying away.

Panoramic selfie command computation

From Figure 6a, capture location 1 components can be computed as

$$x_{cl1} = x_{up} - d \sin \theta_{up} \tag{14}$$

$$y_{cl1} = y_{up} - d \cos \theta_{up} \tag{15}$$


$(x_{cl4}, y_{cl4}, z_{cl4}, \theta°_{cl4})$
@ Capture Location 4 of 4
$(x_{cl3}, y_{cl3}, z_{cl3}, \theta°_{cl3})$
@ Capture Location 3 of 4
$(x_{cl1}, y_{cl1}, z_{cl1}, \theta°_{cl1})$
@ Capture Location 1 of 4
d m
$(x_{up}, y_{up}, z_{up}, \theta°_{up})$
@ User Position
$(x_{cl2}, y_{cl2}, z_{cl2}, \theta°_{cl2})$
@ Capture Location 2 of 4
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

(a) Panoramic selfie capture location description.

(b) Panoramic MATLAB computation location plot with UAV initially facing 33 North.

Fig. 6: Panoramic selfie capture.

For capture location 2 - 4 :

$$x_b = x_a + \sqrt{d^2 + d^2 \sin(\theta_a + 90)} \tag{16}$$

$$y_b = y_a + \sqrt{d^2 + d^2 \cos(\theta_a + 90)} \tag{17}$$

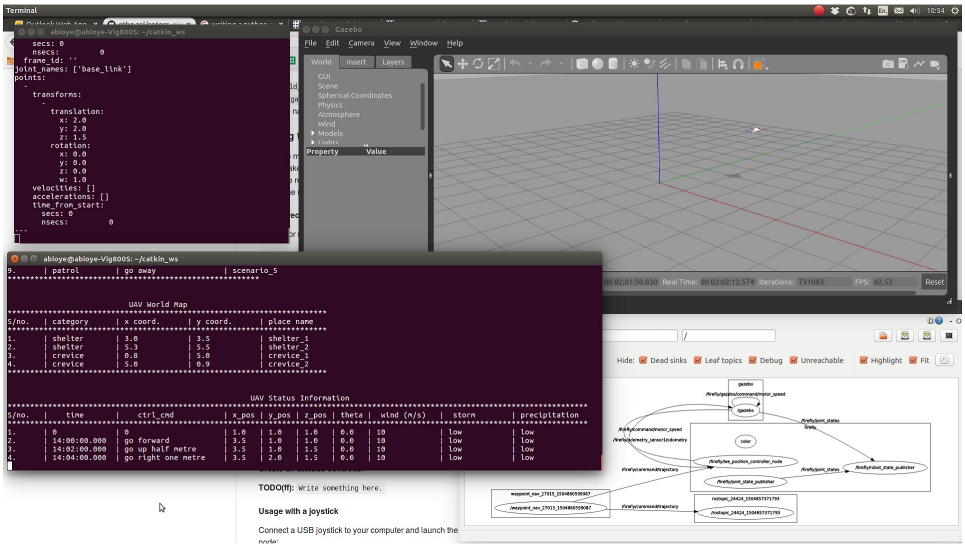
Where subscript 'a' denoted parameters refers to parameters from the previous location, and 'b' subscripted parameters refers to the parameters for the current location being computed.

Patrol command computation

As was shown in Figure 1a, the patrol operation can be broken down into the following components: 1) UAV flies at 0.5 m altitude, 2) Briefly stopping at 4 intermediate stop points between two docking stations (e.g. A and B) to pan and scan area under. The stop points between point A and point B is computed using the following expression in both the MATLAB and Python implementation

$$(x_n, y_n) = (x_p + 0.2(x_b - x_a), y_p + 0.2(y_b - y_a) \tag{18}$$

Five points including final point B. Where p is the previous state and n is the next state. And the 'a' and 'b' subscripted coordinate 'x' and 'y' corresponds to the UAV's location A and B coordinate component. Figure 7 shows this python based ROS Gazebo implementation of the UAV patrol operation.



```
abioye@abioye-Vig800S: ~/catkin_ws
    secs: 0
    nsecs:          0
  frame_id: ''
joint_names: ['base_link']
points:
  -
    transforms:
      -
        translation:
          x: 2.0
          y: 2.0
          z: 1.5
        rotation:
          x: 0.0
          y: 0.0
          z: 0.0
          w: 1.0
    velocities: []
    accelerations: []
    time_from_start:
      secs: 0
      nsecs:          0
---
```

```
abioye@abioye-Vig800S: ~/catkin_ws
9.      | patrol        | go away                | scenario_5
*************************************************************


                        UAV World Map
***********************************************************************
S/no.   | category       | x coord.       | y coord.       | place name
***********************************************************************
1.      | shelter        | 3.0            | 3.5            | shelter_1
2.      | shelter        | 5.3            | 5.5            | shelter_2
3.      | crevice        | 0.8            | 5.0            | crevice_1
4.      | crevice        | 5.0            | 0.9            | crevice_2
***********************************************************************


                                                   UAV Status Information
*******************************************************************************************************************************
S/no.   |     time       |     ctrl_cmd           | x_pos | y_pos | z_pos | theta |  wind (m/s)    |  storm          | precipitation
*******************************************************************************************************************************
1.      | 0              | 0                      | 1.0   | 1.0   | 1.0   | 0.0   | 10             | low             | low
2.      | 14:00:00.000   | go forward             | 3.5   | 1.0   | 1.0   | 0.0   | 10             | low             | low
3.      | 14:02:00.000   | go up half metre       | 3.5   | 1.0   | 1.5   | 0.0   | 10             | low             | low
4.      | 14:04:00.000   | go right one metre     | 3.5   | 2.0   | 1.5   | 0.0   | 10             | low             | low
```

Terminal
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Real Time: 00 02:02:12.574 Iterations: 731083 FPS: 62.52 Reset
Hide: Dead sinks Leaf topics Debug Unreachable Highlight Fit
gazebo
/gazebo
/firefly/gazebo/command/motor_speed
/firefly/command/motor_speed
/firefly/odometry_sensor1/odometry
/firefly/joint_states
/firefly
color
/firefly/lee_position_controller_node
/firefly/joint_state_publisher
/firefly/robot_state_publisher
/firefly/command/trajectory
waypoint_nav_27015_1504860599087
/waypoint_nav_27015_1504860599087
rostopic_24424_1504857371793
/rostopic_24424_1504857371793
TODO(ff): Write something here.
Usage with a joystick
Connect a USB joystick to your computer and launch the node:

Fig. 7: Python implementation with rotors gazebo UAV simulator.

4 Result and Discussion

4.1 Test and Implementation

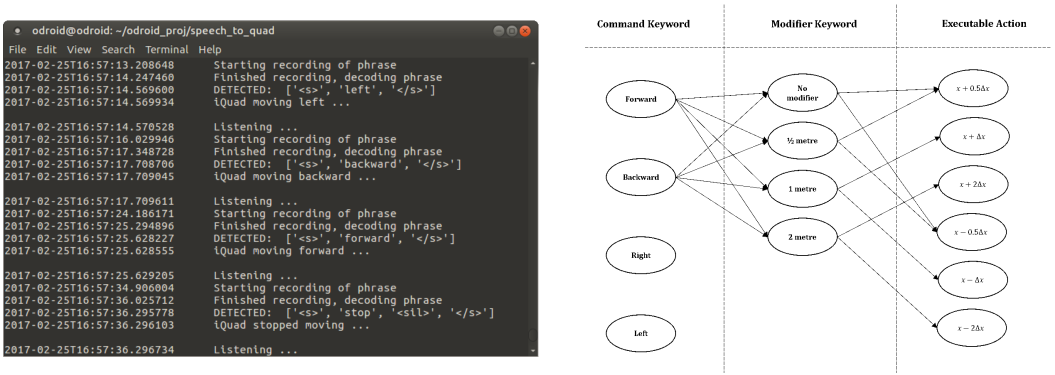
Speech is captured via a microphone, processed and recognised using the CMU Sphinx ASR with custom-defined phonetic dictionary containing only the set of command vocabulary, in order to increase recognition speed and accuracy. Figure 8a shows a screenshot of the speech recognition engine identify captured speech, and describing aerobot navigation response such as "iQuad moving forward..." Visual gesture is captured via a camera connected to the aerobot single board computer (SBC) computer. In the preliminary work, a simple finger-coded visual gesture control commands set was developed to be recognised through a combination of two OpenCV algorithms Haar cascade for hand tracking and convex hull for finger counting.

4.2 MATLAB and ROS Gazebo Simulation

Based on the mathematical set model, the mSVG control navigation was simulated in MATLAB, which was then implemented in python for easy integration of algorithm on a single board computer (in this case, Odroid XU4 SBC), and simulated on a rotors gazebo firefly UAV simulator in an open world environment. In each case, a series of command such as 'go forward', 'go up half metre', 'go right one metre', 'hover at three metre', 'and', 'hover', 'go forward backward two half metre', 'patrol', etc. were successfully tested. Figure 8b shows a portion of the decision tree as implemented for forward and backward navigation command operations.

5 Conclusion - Limitations and Further Works

In this paper, the authors described the growing application of small multi-rotor UAVs otherwise called aerobots, with a particular focus on high level nCA operation such as a patrol, search, and rescue application scenario in the Matterhorn area of the Alps. An overview of multimodal interfaces was presented with further emphasis on related multimodal interface research in aerial robotics. The multimodal speech and visual gesture (mSVG) architecture model developed by the researchers was then presented along with some mathematical model description and a few high level command operation examples. A summary of the implementation and test was given as a result consequence of the developed models.



```
odroid@odroid: ~/odroid_proj/speech_to_quad
File  Edit  View  Search  Terminal  Help
2017-02-25T16:57:13.208648      Starting recording of phrase
2017-02-25T16:57:14.247460      Finished recording, decoding phrase
2017-02-25T16:57:14.569600      DETECTED:  ['<s>', 'left', '</s>']
2017-02-25T16:57:14.569934      iQuad moving left ...

2017-02-25T16:57:14.570528      Listening ...
2017-02-25T16:57:16.029946      Starting recording of phrase
2017-02-25T16:57:17.348728      Finished recording, decoding phrase
2017-02-25T16:57:17.708706      DETECTED:  ['<s>', 'backward', '</s>']
2017-02-25T16:57:17.709045      iQuad moving backward ...

2017-02-25T16:57:17.709611      Listening ...
2017-02-25T16:57:24.186171      Starting recording of phrase
2017-02-25T16:57:25.294896      Finished recording, decoding phrase
2017-02-25T16:57:25.628227      DETECTED:  ['<s>', 'forward', '</s>']
2017-02-25T16:57:25.628555      iQuad moving forward ...

2017-02-25T16:57:25.629205      Listening ...
2017-02-25T16:57:34.906004      Starting recording of phrase
2017-02-25T16:57:36.025712      Finished recording, decoding phrase
2017-02-25T16:57:36.295778      DETECTED:  ['<s>', 'stop', '<sil>', '</s>']
2017-02-25T16:57:36.296103      iQuad stopped moving ...

2017-02-25T16:57:36.296734      Listening ...
```

(a) CMU ASR speech recognition on Odroid XU4 single board computer (SBC) terminal screen.

(b) Relative 1-D (x-axis, forward-backward) Navigation Symbol Decision Tree.

Fig. 8: Speech ASR and control navigation implementation.

The main limitations of the proposed system is 1) its susceptibility to speech corruption during capture, due to the noise generated by the multirotor propulsion systems and other loud ambient noise such as in stormy weathers, 2) the effect of poor visibility level on visual gesture capture, as could be the case at night, or in cloudy or misty weather. The next phase of this research is already underway to determine the range of effectiveness of the mSVG method under varying noise and visibility levels. This could inform the possibility of working around this or developing techniques that may extend this range, thereby extending the usefulness of the propose mSVG technique over a much wider application area. Also, a comparison of the mSVG and RC joystick in terms of training time, same nCA Tier task completion rate, and cognitive workload requirement, is currently being conducted. Further works may also consider the application of some artificial intelligence algorithm such as deep neural network in recognising hand gestures in complicated outdoor environments.

Acknowledgement

This research was financially supported by the Petroleum Technology Development Fund (PTDF) of the Federal Government of Nigeria. Accessible via the following PTDF Reference Number: 16PHD052 and PTDF File Number: 862.

References

1. Green, S., Chen, X., Billinnghurst, M., Chase, J.G.: Human Robot Collaboration: an Augmented Reality Approach a Literature Review and Analysis. Mechatronics **5**(1) (2007) 1–10
2. Abioye, A.O., Prior, S.D., Thomas, G.T., Saddington, P.: The Multimodal Edge of Human Aerobotic Interaction. In Blashki, K., Xiao, Y., eds.: International Conferences Interfaces and Human Computer Interaction, Madeira, Portugal, IADIS Press (2016) 243–248
3. Abioye, A.O., Prior, S.D., Thomas, G.T., Saddington, P., Ramchurn, S.D.: Multimodal Human Aerobotic Interaction. In Isaías, P., ed.: Smart Technology Applications in Business Environments. IGI Global (2017) 39–62
4. Root, S., Air Zermatt: The Matterhorn 101 - This is all you need to know about the Matterhorn. https://www.redbull.com/int-en/the-horn-air-zermatt-matterhorn-rescue-team (2016) Available: 2016-10-17; Accessed: 2017-06-07.
5. Aeryon Labs Inc.: Whitepaper - intuitive control of a micro uav. https://aeryon.com/whitepaper/ituitivecontrol (2011) First Available: 2011-02-07; Accessed: 2016-01-22.

6. Fong, T., Nourbakhsh, I.: Interaction challenges in human-robot space exploration. In: Proceedings of the Fourth International Conference and Exposition on Robotics for Challenging Situations and Environments. Number January 2004 (2000) 340–346
7. Oviatt, S.: Multimodal interfaces. In Jacko, J.A., Sears, A., eds.: The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications. First edn. Lawrence Erlbaum Associates, Incorporated, London (2003) 286–304
8. Preece, J., Sharp, H., Rogers, Y.: Interaction Design: Beyond Human-Computer Interaction. Fourth edn. John Wiley & Sons Ltd, Glasgow (2015)
9. Turk, M.: Multimodal interaction: A review. Pattern Recognition Letters **36**(1) (2014) 189–195
10. Shah, J., Breazeal, C.: An Empirical Analysis of Team Coordination Behaviors and Action Planning With Application to Human-Robot Teaming. Human Factors: The Journal of the Human Factors and Ergonomics Society **52**(2) (2010) 234–245
11. Bischoff, R., Graefe, V.: Dependable multimodal communication and interaction with robotic assistants. In: Proceedings - IEEE International Workshop on Robot and Human Interactive Communication. (2002) 300–305
12. Harris, J., Barber, D.: Speech and Gesture Interfaces for Squad Level Human Robot Teaming. In Karlsen, R.E., Gage, D.W., Shoemaker, C.M., Gerhart, G.R., eds.: Unmanned Systems Technology Xvi. Volume 9084. SPIE (2014)
13. Redden, E.S., Carstens, C.B., Pettitt, R.A.: Intuitive Speech-based Robotic Control. U.S. Army Research Laboratory (Technical Report ARL-TR-5175) (2010)
14. Cacace, J., Finzi, A., Lippiello, V.: Multimodal Interaction with Multiple Co-located Drones in Search and Rescue Missions. CoRR **abs/1605.0** (2016) 1–6
15. Lee, a., Kawahara, T., Shikano, K.: Julius an Open Source Real-Time Large Vocabulary Recognition Engine. Eurospeech (2001) 1691–1694
16. Fernandez, R.A.S., Sanchez-lopez, J.L., Sampedro, C., Bavle, H., Molina, M., Campoy, P.: Natural User Interfaces for Human-Drone Multi-Modal Interaction. In: 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA USA, IEEE (2016) 1013–1022
17. Barber, D.J., Howard, T.M., Walter, M.R.: A multimodal interface for real-time soldier-robot teaming. **9837** (2016) 98370M
18. Borkowski, A., Siemiatkowska, B.: Towards semantic navigation in mobile robotics. Graph Transformations and Model-Driven Engineering (2010) 719–748
19. Hill, S.G., Barber, D., Evans, A.W.: Achieving the Vision of Effective Soldier-Robot Teaming : Recent Work in Multimodal Communication. Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts (2015) 177–178
20. Kattoju, R.K., Barber, D.J., Abich, J., Harris, J.: Technological evaluation of gesture and speech interfaces for enabling dismounted soldier-robot dialogue. **9837** (2016) 98370N
21. Ng, W.S., Sharlin, E.: Collocated interaction with flying robots. Proceedings - IEEE International Workshop on Robot and Human Interactive Communication (2011) 143–149
22. Cauchard, J.R., Jane, L.E., Zhai, K.Y., Landay, J.A.: Drone & me: an exploration into natural human-drone interaction. Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (2015) 361–365
23. Obaid, M., Kistler, F., Kasparaviciute, G., Yantaç, A.E., Fjeld, M.: HowWould You Gesture Navigate a Drone? A User-Centered Approach to Control a Drone. In: Proceedings of the 20th International Academic Mindtrek Conference, Tampere, Finland, ACM New York, NY, USA (2016) 113–121
24. Nagi, J., Giusti, A., Gambardella, L.M., Di Caro, G.A.: Human-swarm interaction using spatial gestures. IEEE International Conference on Intelligent Robots and Systems (Iros) (2014) 3834–3841