Autonomous Navigation in GPS Denied Indoor Environment Using RGBD Sensor, Kinect

Vihari Piratla*, Suraj B Malode*, Sachin Kumar Saini*, Anurag Jakhotia§

Anil K Sao¶, Bharat Singh Rajpurohit¶, Genemala Haobijam¶

§Student, School of Engineering.
*Student, School of computer science and engineering.
¶Professor, School of computer science and engineering

IIT Mandi, India

ABSTRACT

This paper describes the autonomous quadrotor designed at IIT Mandi for participation in International Aerial Robotics Competition (IARC) 6^{th} mission, capable of autonomous navigation in a GPS denied and cluttered indoor environment. A distributed nodal architecture is implemented using RGBD sensor, sonar sensors and a camera with Simultaneous Localization and Mapping (SLAM) algorithm to map the environment. A system of hierarchical visual odometry algorithms are fused with IMU (Inertial Measurement Unit) using EKF (Extended Kalman Filter) is implemented to ensure globally consistent localization, navigation and exploration of the environment. The secondary sensor suite along with obstacle avoidance algorithm ensures real-time path planning and pattern recognition.

I. INRODUCTION

Recent advances in cost effective inertial sensors and accurate navigation systems, such as the GPS, have been key determinants of the feasibility of UAV systems. Most current Navigation systems use conventional navigation sensors such as standard IMUs for orientation, GPS for localization. These systems rely heavily on GPS which may not be available in cluttered environments like urban and indoor environments. Furthermore, the substantial weight, power and computational constraints preclude the use of conventional sensors in such areas. On the other hand, visual sensors are passive, lightweight and can provide rich information about the aircraft self-motion and surroundings structure.

Our motivation for this work is to develop a fully embedded, lightweight, and low-cost solution for autonomous localization in arbitrary and unknown environment using 3D visual sensors like kinect.

A. Problem Statement

The Problem statement of 6^{th} mission of IARC requires an autonomous aerial vehicle to drive

through a cluttered environment, decipher some Arabic symbols, search and recognize a pre-defined memory stick to replace it with a decoy. All this has to be done in a time-span of 10 minutes and covertly.

B. Conceptual Solution

Team SPARTA at IIT Mandi, has designed a quadrotor Autonomous Unmanned Aerial Vehicle (IMAUAV) capable of navigating indoors without any external navigation aids like GPS. At the base level, the on-board IMU, the flight controller and the main processor create a feedback loop to stabilize the quadrotor at an update frequency of ~500Hz. Further, the realtime visual odometry algorithm estimates the vehicle's position relative to the local environment, while an EKF combines these estimates with the IMU outputs to provide accurate state estimates of the position and velocity at 15Hz. Figure 1 shows the overall system architecture. These pose estimates enable the flight controller to navigate the quadrotor through the cluttered environment stably.

To mitigate cumulative errors from the odometry algorithm, we use SLAM using both EKF and ICP algorithm to create a global map, ensuring consistent pose estimates. As the robot navigates it simultaneously builds the map and localizes itself in the map, this way SLAM provides globally consistent position estimates of the vehicle from the map created.

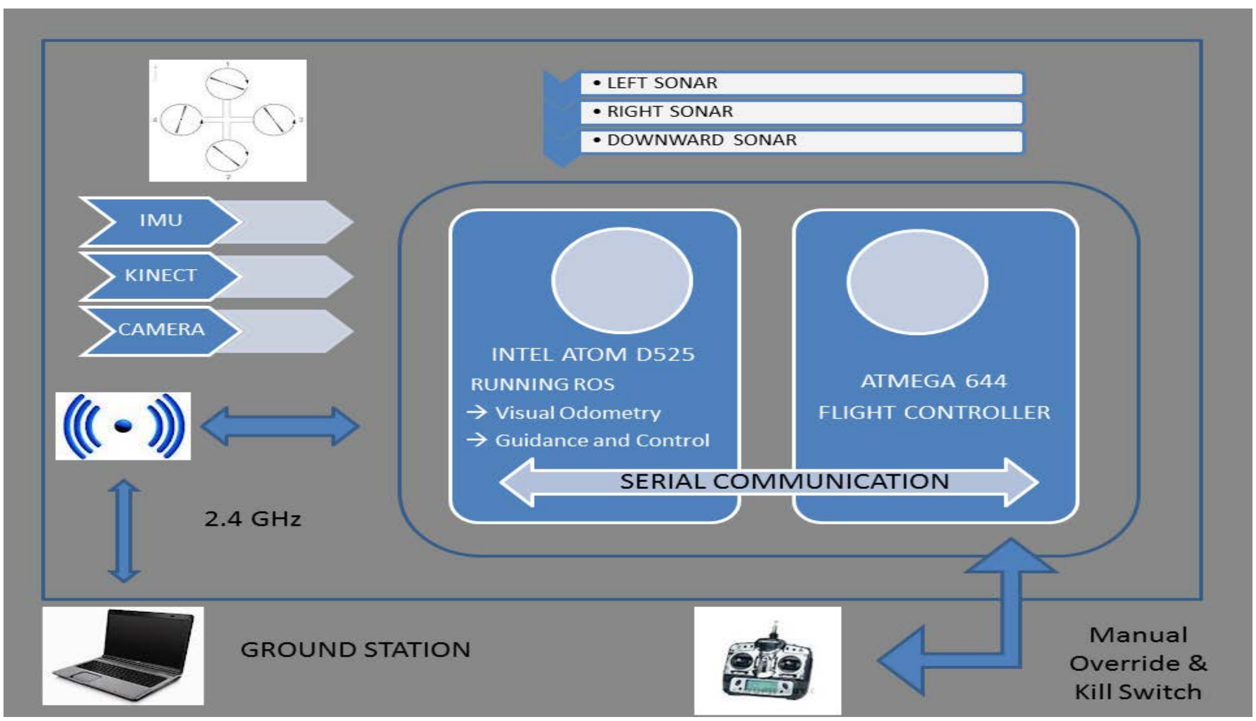


Figure 1: Overall System Architecture.

C. Yearly Milestones

In our first attempt at IARC, we intend to develop an Unmanned Aerial Vehicle (UAV) capable of autonomous navigation in unknown cluttered indoor environment and complete IARC mission 6. Based on the work over the past 1 year, we intend to implement covert operation of the vehicle along with real time target tracking and retrieval mechanism in the current year.

II. Aerial Vehicle

Team SPARTA is using a quadrotor with two pairs of rotors spinning clockwise and counter-clockwise respectively. The very design of quadrotor makes hovering over an area very stable due to cancelling of reaction torques. This type of hardware design makes quadrotors an ideal choice for indoor environments, where stable flight and hovering is primarily important. The vehicle is capable of carrying requisite sensor payload (*refer section 4*).

An interlocking assembly was designed to flexibly mount the kinect sensor and the camera. Sensor positions are aligned so that Field of View (FOV) is not obstructed by the quadrotor components. The quadrotor is operated in diamond configuration to ensure a clear FOV. The weight of the vehicle along with the components and the payload is ~ 1500g. We selected L4-ME Quadro basic set made by Mikrokopter (*www.mikrokpter.de*) as the base platform. The vehicle structure and motors have been modified to suite the mission requirements.

A. Propulsion and lift System

The Quadrotor is controlled by four inputs (roll, pitch, yaw and thrust) by varying the lift forces and the balance of reaction torques through changing the rotating speed of the rotors. Roll and pitch can be changed by increasing the speed of one rotor and decreasing the speed of the other rotor in a pair of rotors. The quadrotor requires a low level attitude control due to its unstable dynamics, which is provided by the on-board flight controller.

The vehicle generates lift by four brushless electric motors. We use 10 inch propellers for extra stability and meeting the payload requirements.

B. Guidance Navigation and Control System

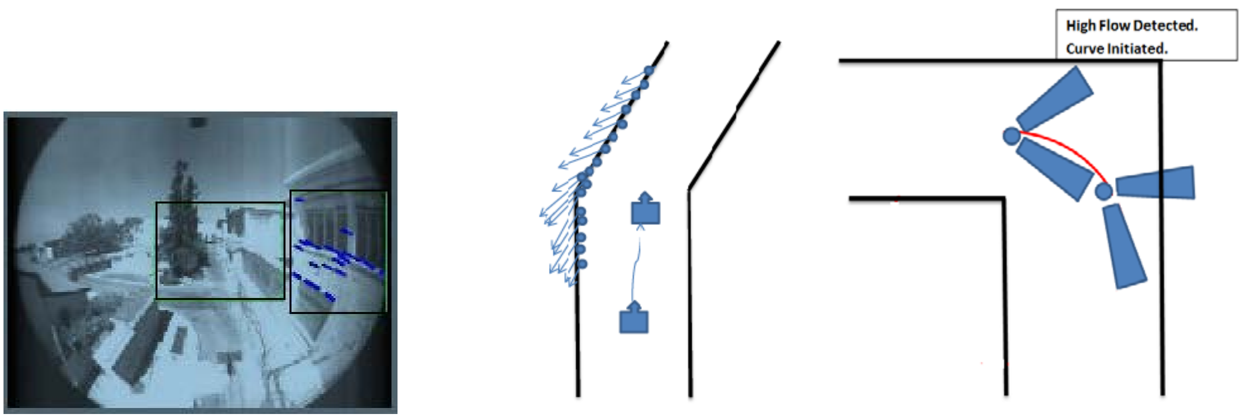
1. Guidance

The guidance system of IMAUAV uses information gathered by the sonars and fish eye camera. The Guidance algorithm is controlled by two factors, one is sonar pair measurements and the other is optical flow measurement by front facing fish eye camera. Both systems have one common goal, to centralize the vehicle in the environment and to determine the topology of the unknown environment.

With sonar an equal distance is maintained from the walls, change in sonar readings provides an insight into change in environment structure. If a pathway is detected on either side, the guidance system commands vehicle to explore the area, if it is unexplored, otherwise appropriate steps are taken in accordance with primary goals published by the mission control.

Fish-eye cameras are known for their high FOV 190˚, hence side walls are visible in the left and right peripherals of the image. Optical flow [1] is recorded in both the left and right peripherals of the image in the camera. The vehicle is commanded to move away from the side of higher flow. It ensures that the vehicle centralizes itself and turns are navigated gradually. This aids in successful navigation at inside and outside corners. Figure 2(a) below shows an image in which optical flow is detected in right periphery of the image. Figure (b) and Figure (c) are depiction of different topologies of environment.

Scenario with fish-eye camera under various Environments:



(a) *(b)* *(c)*

Figure 2 (a) Flow is detected in the right periphery of the image. (b) High flow is detected (c) High flow detected in the left portion of the image, hence commanded to turn right

Fish-eye cameras cannot be of any help when there is an opening on both the sides, as the flow may be equal on both the side's, it is sonar that is of help in these instants.

The vehicle moves with the aim of exploring the unexplored areas i.e. forward in to the arena. Whenever the camera detects an opening, the corresponding area is searched for the Arabic symbols, if it matches with the target only then it is explored. If the recognized symbol is not the desired one, then it continues its exploration and if no symbol is detected then it would interpret it as sideway and navigates through it.

Entry point Window detection:

IMAUAV uses shape detection technique to recognize the window. Once the vehicle gets to a desired height, a disparity image is captured by the RGBD sensor and is probed for square shapes. If detected, the spatial moments of the shape will be the goal position for the vehicle.

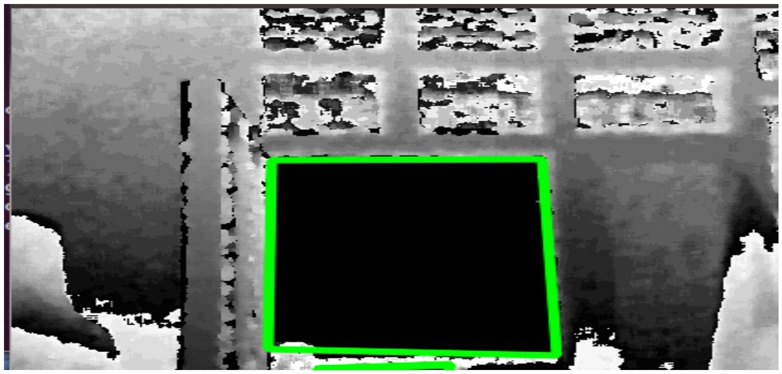


Figure 3: This image shows the window detected in green captured from the disparity image.

2. Stability Augmentation System (SAS)

The Quadrotor platform has a very unstable dynamics, without controlled input it will drift continuously and may eventually crash. The relationship between control inputs and thrust is nonlinear. The proportional and integral terms in the PID controller compensate for rate at which the error is changing and the rate at which the system state is evolving. These PID control equations are implemented on the firmware of FC by the manufacturer, Mikrokopter.

3. Control System

Quad-rotor has 6 Degrees of Freedom (DOF) which gets very complex when the vehicle is controlled in all the DOFs. So, to reduce complexity of the dynamics is reduced to 3DOF by keeping altitude, roll and pitch constant for most of the flight. This constrains the vehicle to move in the 2D plane coplanar with it.

The PD control equations implemented [2] takes in as input the desired velocity and desired position and outputs the roll and pitch commands to be maintained to achieve the goal position. The navigation stack provides x, y positions and a yaw angle every 100ms and these PD controllers generate outputs at the same rate These PD control commands are provided to the flight controller by the onboard processor. The flight control then issues commands to the motor through I2C bus which is converted to PWM before passing to the ESC (Electonic Speed Controllers).

$$w_\theta = K_p(w_{x*} - w_x) + K_d*(w_{x*} - w_x)'$$

$$w_\varphi = K_p(w_{y*} - w_y) + K_d*(w_{y*} - w_y)'$$

$$^{B}Rw(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}$$

$$\begin{bmatrix} B_{\theta*} \\ B_{\varphi*} \end{bmatrix} = R(\psi)\begin{bmatrix} w_{\theta*} \\ w_{\varphi*} \end{bmatrix}$$

$w_{\theta*}$ notes the pitch input in world frame and w_{x*} is the reference input which the quadrotor should follow. w_x is the current position of the quadrotor in the world frame and this data is provided from the SLAM.

$B_{\theta*} = \cos\psi . w_{\theta*} - \sin\psi . w_{\varphi*}$ $\qquad\qquad$ $B_{\varphi*} = \sin\psi . w_{\theta*} + \cos\psi . w_{\varphi*}$

The world and body frames are related by yaw angle transformation. Finally, the commands in the body frame are given by $B_{\theta*}$ and $B_{\varphi*}$.

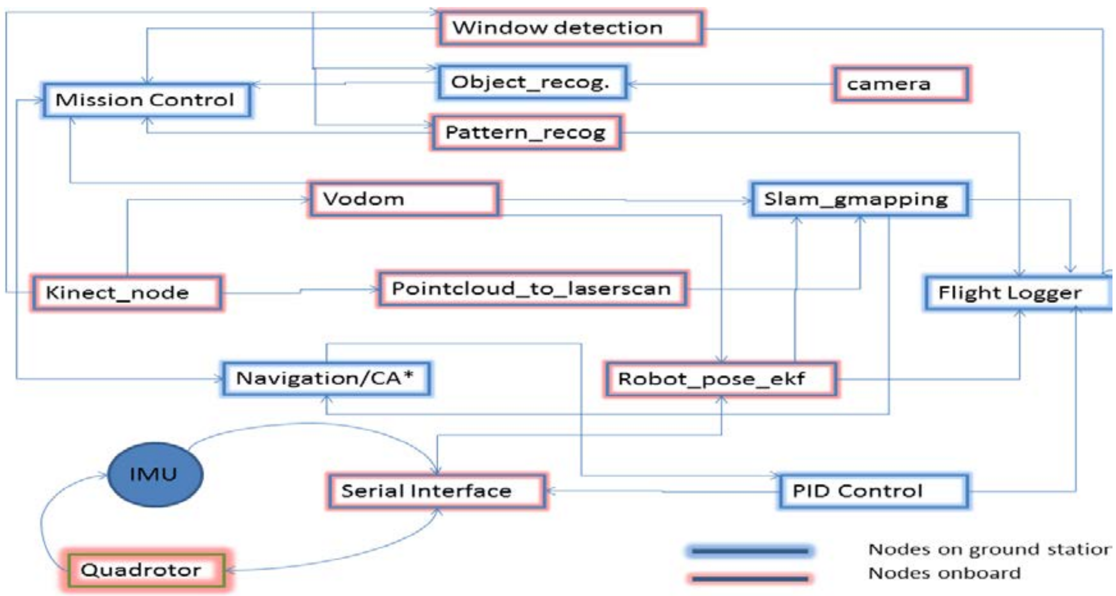


Figure 4: Control System Architecture

(ROS nodes communicating in IMAUAV System)

4. Navigation System

IMAUAV depends on Visual odometry, IMU readings and ICP scan matching algorithms to determine the accurate pose of the vehicle. SLAM algorithm is used to produce globally consistent pose estimate and to produce map, which is used for autonomous navigation once a goal pose is determined.

For SLAM we used Gmapping slam package available in ROS and since we are not using any laser scanner, we converted the point clouds of kinect to laser scans before passing it to the SLAM algorithm. This reduces the bandwidth requirement to communicate point clouds among ROS nodes. The slam package is fed with the odometry values from the EKF fused IMU and Visual Odometry.

Visual Odometry(VO)[5] works by extracting the features in the current image, compares the image features in the previous frame which when cascaded with the 3D values given by kinect; results in two 3D point clouds belonging to the previous and current instants. These Point clouds are RANSAC [4] filtered obtain inliers (corresponding to the defined RANSAC threshold). Then a least square estimation is done on the inliers detected for the pose change between the previous and the current frames.

RANSAC selects 3 random points from the filtered point cloud, say a1, a2 and a3 in the previous frame and corresponding points in the current frame say b1, b2, b3. Rotation and Translation transformations are determined from these three points as shown in the equations below.

$$a_x = (a_2 - a_1)/|a_2 - a_1| \qquad b_x = (b_2 - b_1)/|b_2 - b_1|$$
$$a_y = (a_3 - a_1) - a_x(a_x.(a_3 - a_1)) \qquad b_y = (b_3 - b_1) - b_x(b_x.(b_3 - b_1))$$
$$a_z = a_x \, x \, a_y \qquad b_z = b_x \, x \, b_y$$

$$R_A = [a_x \;\; a_y \;\; a_z] \qquad R_B = [b_x \;\; b_y \;\; b_z]$$
$$R = R_{B.}R_A^T \qquad b_1 - RA_1 \rightarrow translation$$

Once the Rotation and translation matrices are obtained from the random sample set, these values are used to estimate new point cloud by transforming previous point cloud. The estimated point cloud is then compared with the current point cloud obtained by the sensor and if the error falls below RANSAC threshold then the sample set is declared as inliers. Once the inlier set is obtained, Singular Value Decomposition (SVD) is used to estimate the Rotation and translation transformations in world frame.
The Imu readings from IMU are fused with the visual odometry values with an EKF filter with the help of robot_pose_EKF node of ROS. These readings are then fed to the SLAM package. Below is an image that shows the map that is produced as IMAUAV traversed a closed room with windows.

Team SPARTA modified Navigation stack from ROS (fig. 4) which carries out the autonomous navigation, once it is fed with the goal position, sensor streams and odometry. This stack makes sure that the vehicle does not collide with obstacles in the FOV, by commanding the vehicle with safe velocities. These are in turn communicated to PD control loop which makes sure that the desired velocity is maintained which in turn communicates with Serial interfacing Node.

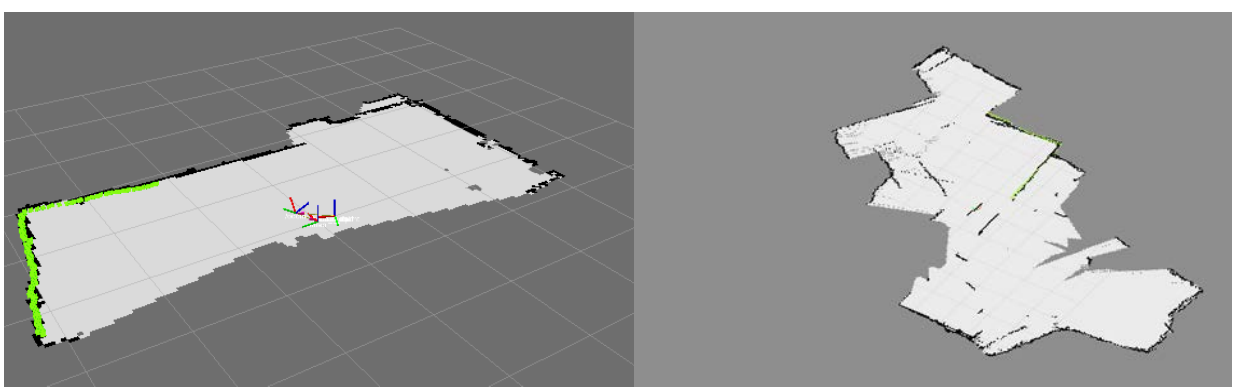


Figure 5: Map generated by SLAM node

C. Flight Termination System

For safety, the vehicle can be switched from autonomous to manual control through a safety pilot's transmitter. The ground station constantly monitors the RC channel for signal strength, and allows for manual override. The transmitter also provides a "kill-switch", to be used in case the mission fails. In the event of low battery or RC/data-link connection loss, the vehicle will attempt a controlled descent.

IV PAYLOAD

The quadrotor is capable of carrying 800 gms of payload.

A. Sensor suite

1. GNC Sensors:

Mikrokopter IMU: The IMU employed consists of tri axis gyroscopes and tri axis accelerometer.

Kinect (RGBD sensor): IMAUAV uses an RGBD camera from Microsoft which is equipped with a color camera and a depth camera which results in 3D coordinate at each and every pixel at a rate of 640*480@30 fps. It has a maximum range of 3.5-4 m and minimum range of 0.6 m. It has a Horizontal FOV of 57 ˚ and vertical FOV of 43 ˚.

Ultrasonic Range Finder EZ0: This has a max range of 6.45 m's and a minimum of 0.15 m, weighs around 2 gms and is used for altitude estimation of quadrotor.

2. Mission Sensors:

Orthogonal Sonars: Two sonars are placed perpendicular to the direction of motion of the vehicle which is basically Ultrasonic Range Finder EZ0 and follows the description as given above.

Fish-eye camera: This camera (FOV 190˚) is placed facing slightly downward for memory stick recognition. Though kinect has a camera, this camera is needed because of the low FOV of kinect (Vertical FOV is 43˚) which doesn't suffice for both symbols and memory-stick recognition.

2.1 Target identification:

The target to be retrieved is very small and has to be first located and then replaced by the decoy. This requires acquiring accurate position of memory stick. We have divided target identification into two basic steps i.e.

(a) Recognition of target location by deciphering the Arabic symbols;

IMAUAV system solves (a) by turning wherever sonar detects a sideway to probe for an Arabic symbol and if desired symbol is found then navigates that sideway. For deciphering the Arabic symbols, we use Scale Invariant Feature Transform (SIFT)[3] based feature matching, results of which can be seen in Figure 9: taken at a most probable perspective in the arena.

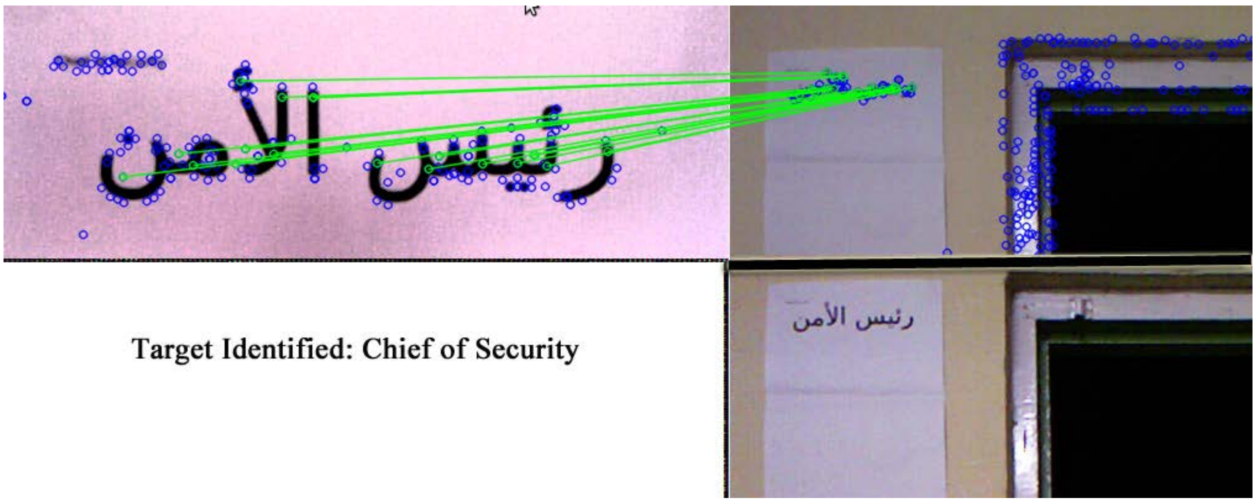


Figure 6: This image shows the rightly deciphered Chief of Security office , the template is shown to the left and the matching features are shown to the right with the original image below to it.

(b) Recognizing of memory stick in the room susceptible;

Memory stick is recognized by comparing edge extracted template and camera images. The matching is done using SIFT.

Recognition/Retrieval of memory stick:

Once the memory stick is recognized by the downward facing camera, the IMAUAV is commanded to move right or left along the orthogonal direction to the camera and with height of the vehicle constant.

The pixel position of the memory stick is measured again after this command and the difference in X coordinate gives the disparity estimate i.e. the depth of the memory stick can be determined from stereo triangulation, once the position of the object is accurately calculated then the vehicle may be destined to that position for the retrieval of the memory stick based on the trajectory generated by the navigation module. For retrieval, a sticky pad is used with a large surface to compensate for target localization error.

B. Communications

The data communication is done through 2.4GHz wifi link to ground station using an wireless NIC (Network Interface Card) onboard operating on 802.11n mode. The ground station node communicates among them through a separate adhoc wifi link. For safety pilot/ kill switch a separate 2.4 GHz RC link is used with an onboard receiver interfaced with flight controller.

C. Power Management System

The system uses a four cell Lithium polymer Ion battery. It provides power to both motors and to the onboard processor. The battery has power rating of 2200mAh with a full charge voltage of 16.2V and nominal voltage of 14.8V. The flight controller has an on board battery eliminator circuit (BEC). The BEC takes care of controlled descent in the event of low battery.

5. OPERATIONS

A. Flight Preparations

Some conditions are needed to be ensured before each flight test of the vehicle to avoid any mishap. These are described below:

1. Flight Checklist

- ✓ Make sure all the batteries used are fully charged and tested for polarity.
- ✓ Make sure the wireless communication between workstations in the ground cluster is active and working as expected.
- ✓ Ensure that all the propellers, camera mounts, nuts and other sensors are firmly fixed to avoid any vibration and hardware damage during flight.
- ✓ Switch on the vehicle to ensure that all the sensors and the wireless link between vehicle and the ground station are in working condition.
- ✓ Check if all the ROS nodes are up and running.

B. Man/Machine Interface:

A ground station based on the ROS (Robot operating System) environment will continuously monitor the vehicle and display health and status information during the flight. The vehicle will continuously send information about its pose, velocity, health status, environment (real time images), etc. to the ground station (over a 2.4GHz wireless LAN data link) preconfigured with ROS making it easier to interpret the data received and present it in an informative way. A safety pilot link is available through a separate 2.4 GHz radio uplink. All primary goals are assigned by the mission control on ground station to the onboard processor.

6. RISK REDUCTION

A. Vehicle Status

The vehicle flight controller board is equipped with status LEDs and alarms which start working in a pre-established way if any component malfunctions occur. Moreover, the ground station continuously monitors the vehicle status using the data received checking for any potential malfunctions, signal strengths and battery status.

1. Shock/Vibration Isolation

Attempts have been made to restrict the vibration as much as possible to avoid any hardware damage and stray readings from different sensors. All the mechanical joints are firmly fixed to the mainframe. Heavy equipment like kinect, battery, etc. are all placed near the center of gravity of the vehicle for further stabilization and reduced vibration. The landing gear is made up of fiber restraining any shock to the vehicle and other equipment while landing. Further, the frame is designed so that any shock is dampened ensuring maximum stability and shock absorption.

2. Electromagnetic Interference (EMI)/Radio Frequency Interference (RFI) Solutions

To curb down EMI effects on the kinect sensor, the microprocessor board and the kinect sensors have been mounted at a good amount of distance from each other. Kinect sensor doesn't entertain considerable amount of EMI. Brushless motors used have reduced EMI signature compared with conventional motors. Since, both data communication link and safety pilot link both operate on 2.4 GHz frequency, appropriate and separated channels are being used for communication, so as to ensure minimum overall RFI.

B. Safety

Effective measures have been taken to ensure safety during the autonomous flight. In case, of a collision approach, safety propeller protectors are provided to prevent damage to the vehicle as well as by standers. Further, controls can be manually overridden to avoid any vehicle damage or injury to any person. Further, the vehicle motion can be instantaneously killed using a "kill" switch to avoid any potential injury or crash.

C. Modeling and Simulation

The Quadrotor is modeled using the "Mikrokopter Tool" provided by the Mikrokopter. The software contains effective tools to control and test flight controllers, motors and other equipment. It was also used to determine the PID gains and calibrating onboard sensors.

The entire software base of IMAUAV system is modeled based on ROS architecture, which is a system with processes running as nodes and communicating over TCP by publishing messages. Every node in IMAUAV system outputs its messages or publishes on a common output stream. Nodes subscribe to other nodes in order to get input. Messages are generic data structures used in ROS to exchange data.

A ROS package was created with a master node acting as mission control publishing primary goal poses to associated nodes. All the algorithms were simulated in the ROS environment on datasets recorded using ROS. ROS with Rviz provides a safe and powerful testing environment for debugging before the software is tested on vehicle in in real scenarios. Maps were created by simulating pre- recorded data using kinect in ROS for higher level planning.

D. Testing

The IMAUAV system has been subjected to flight testing in different conditions. The kinect based navigation and object recognition algorithms have been heavily tested for any failures. The distributed nodal communication among ROS nodes have been tested by deploying nodes on different workstations working under a single master. The validation of algorithms has been done extensively using ROS by logging and recording data and simulating it.

VII. CONCLUSION

In this paper we have presented, the development of a quadrotor Unmanned Aerial Vehicle intended for navigating cluttered GPS denied indoor environment. We describe our solution to this problem using RGBD sensor by Microsoft; Kinect. We have developed a hierarchal suite of algorithms which augments an effective mechanism for autonomous navigation, planning, localizing and mapping of unknown environment.

A distributed nodal architecture has been implemented, aided by an extroceptive sensor suite that helps in achieving stable flight behavior and give attitude control commands to the vehicle. A novel solution; usage of fish eye camera for obstacle avoidance is employed in the navigation of the vehicle. The team SPARTA intends to compete in the 2012 IARC competition with this vehicle. In future, we hope to successfully extend this system to an integrated system for 3D navigation with advanced reconnaissance features so that vehicle can be deployed in any environment.

VIII REFERENCES

[1] Stefan Hrabar and Gaurav S. Sukhatme, Peter Corke, Kane Usher and Jonathan Roberts, *Combined Optic-Flow and Stereo-Based Navigation of Urban Canyons for a UAV*

[2] Inkyu Sa, *Mikrokopter Documentation*, Cyphy Laboratory, 2011

[3] David G Lowe , *Distinctive Image Features from Scale-Invariant Keypoints, 2004*

[4] Martin A Fischler and Robert C Bolles, *Random Sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. ACM,24(6):381-395, June 1981*

[5] Andrew Howard, *Real time visual odometry for Autonomous Ground vehicles*. IEEE/RSJ, Sept 2008