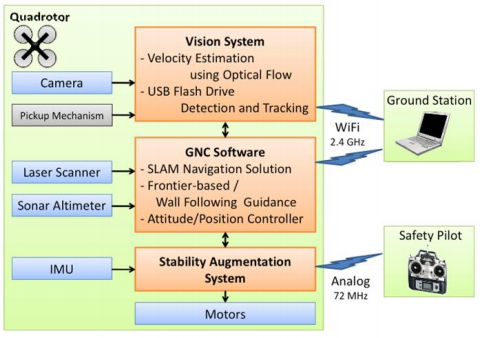
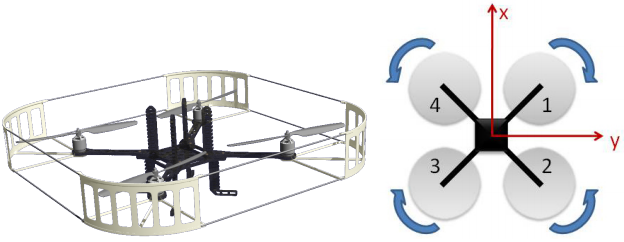
**Georgia Tech Team Entry for the 2012 AUVSI International Aerial Robotics Competition**

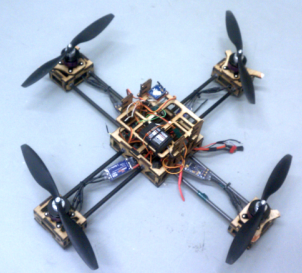
* “We use an elaborate navigation algorithm that fuses information from a laser range sensor, inertial measurement unit, and sonar altitude sensor to form accurate estimate of the vehicle attitude, velocity, and position relative to indoor structures. We leverage the fact that all indoor structures have walls to design a guidance algorithm that detects and follows walls to ensure that the navigation solution maintains its fidelity and maximum indoor area is explored in a reasonable amount of time. We use a control architecture that augments a proven baseline proportional-derivative controller with an optional adaptive element that aids in mitigating modeling error and other system uncertainties.”



* Although many Quadrotors in aerial robotics community with diamond configuration, we selected the square configuration to allow for more flexibility in sensor mounting locations.



* The vehicle weighs about 0.85kg with the following avionics and equipment: four motors, propellers, and speed controllers, Xbee transceiver, Copter Control SAS, wiring harness, receiver, and power regulator. A 0.38kg 5000mah 3C battery allows for approximately 10 minutes of flight while the motors provide over 4kg of thrust for a payload capacity of around 3kg. An I2C to PWM converter allows use of the GTQ1/2 SAS board. The vehicle weighs about 1.25kg with the above battery, which leaves about 0.25kg for a LIDAR system (0.16g), camera (70g), and a Gumstix board for GNC software.



**Guidance System**

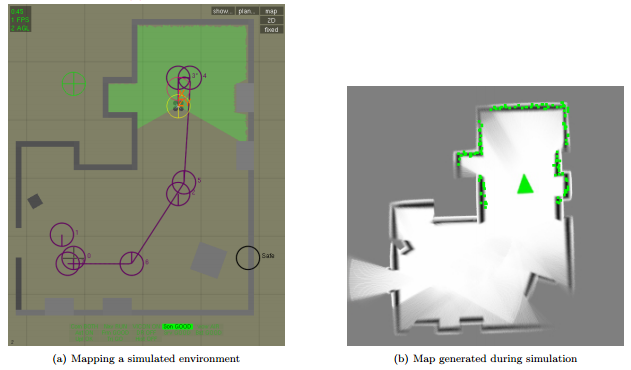
The developed guidance system uses only locally available information gathered through the onboard sensors, which include a laser range scanner. The system maintains previous tree-based guidance systems as backup for the newly developed graph based frontier exploration system. Scan frontiers points are used to add nodes to the undirected exploration graph at each scan point. A node is added as the midpoint of each independent frontier of at least a particular arc length. Each new node is checked against existing nodes. Should a node exist within a specified Cartesian distance from the new node, a node will not be added, but a graph connection will be made between the existing node and the node at the current scan point. As the vehicle captures a node, a new scan is performed and the next node is chosen based on a weighting system. Distance to the node, the arc length of the frontier used to create it, and a directional bias are used to select the next target highlighted in red. The intended graph path is highlighted as well. The directional bias is used should multiple vehicles be operating at the same time as an exploration efficiency enhancement. Should the vehicle not reach the target node in a specified amount of time, or should the GNC software detect that the vehicle is stuck in an unfavorable position, the vehicle will be commanded to backtrack to the last reached node. Stuck detection works by applying three criteria, all of which increase the confidence in the stuck detector output. Firstly, the mean velocity of the vehicle over a rolling time window is calculated and checked against a threshold set in the software. If the velocity is below this value, the probability of a stuck detection is increased. Secondly, in that same time window, should velocity of the vehicle flip on itself more than a number of allowable times, the probability is yet again increased. Lastly, if the vehicle does not leave a circle of a specified radius within the time window but has traveled a specified distance, the stuck probability is again increased. Timeout is forced should probability reach a specified value.



**Navigation Algorithm**

The algorithm used for the preliminary research, called CoreSLAM, was chosen primarily because it is simple, easy to implement, and it uses integer math where possible to improve computational speed. There are two main parts to any SLAM routine. The GTAR team has developed an Iterative Closest Point (ICP) scan-matching algorithm in-house. The algorithm was developed to prioritize localization with respect to the immediate environment, and place lesser emphasis on building and maintaining highly accurate global maps. Hence the “map" in the ICP algorithm consists entirely of a single previous scan. However, with the ICP algorithm individual scans can be saved and post-processed into a global map if desired.

The inherent nonlinearities in the vehicle dynamics and the measurement sensors are handled through the use of an Extended Kalman Filter (EKF). An existing EKF-based navigation filter architecture developed at the Georgia Tech UAV Research Facility is utilized as part of the indoor navigation system. The navigation algorithm developed by the GTAR team augments the existing EKF based architecture to function without GPS signals by using the range information obtained from the laser range scanner and the sonar altimeter.



**Memory Stick Detection and Retrieval**

Memory stick detection and retrieval is achieved with a combination of open-source software and an original tracking algorithm optimized for use on computers with low computational power. The object detection algorithm is part of the open-source software package OpenCV. The detector first is trained using a series of images of the target object, called the positive image set. The positive image set can be created from multiple images of the object, or from a single image that is artificially distorted to simulate viewing the object from many angles. Approximately 1000 images were used. Additionally, a set of 1000 images without the object, called the negative set, is passed to the function. The output of the training algorithm is a cascade of Haar classifiers that can be used to efficiently identify the object in a sample image. Generating the classifier typically took 8 hours of processing time on a standard desktop computer with a Core 2 Duo processor. Processing an image with the classifier could be performed in 0.2 seconds on the GTQ onboard computer. Once the target memory stick is located, a specialized guidance system generates a trajectory for the vehicle for landing on and retrieving the memory stick. The guidance system estimates the three-dimensional location of the memory stick using a downward-facing camera mounted on the vehicle. A cascade filter identifies the pixel location and pixel area of the memory stick in each image from the camera, and an EKF estimates the three-dimensional location based on the pixel measurements. The vehicle is then commanded to descend onto the location of the memory stick and retrieve it. One preliminary retrieval mechanism design is a pad of adhesive tape on the bottom of the vehicle.