AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-0881

Autonomous flight for Multi-copters flying in UTM -TCL4+ sharing common airspace

Anjan Chakrabarty *& Corey Ippolito †
NASA Ames Research Center, Moffett Field, CA 94035

Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881

NASA's UAS Traffic management (UTM) research initiative has demonstrated the flight of multiple UAVs flying in urban environment. Most of the deconfliction has been demonstrated with separate operational volumes through different USSs. For high density operations in urban environments autonomous vehicles should be able to fly in the same shared airspace take on-board decisions to handle different scenarios and land safely beyond visual line of sight. In this paper we present a complete autonomous architecture for flying an UAV in dense urban environments. The planning architecture consists of a global planner and local planner. The global planner plans path to the goal and alternate landing sites along the way. The local planner is involved in avoiding obstacles along the approved path. An UTM client talks with the connected USS to get flight approvals from UTM and make contingency plans.The entire system is tested in the Reflection Simulation Architecture in the NASA Ames Campus simulation demonstration.

Keywords: UTM, USS, Autonomy, Path planning, BVLS, V2V communication

I. Introduction

Autonomy technologies will be key in operating unmanned aerial vehicles(UAVs) beyond visual line of operations.(BVLOS) citemorris2019addressing. Flying autonomous UAVs in dense urban regions require development of both vehicular technologies as well as save utilization of the airspace.

Future autonomous operations using UAVs in dense urban regions will involve flying multiple vehicles simultaneously sharing common airspace. Sharing airspace includes sharing airspace with other UAVs as well as manned aircrafts. Beyond visual line of sight flights require autonomous vehicles capable of taking decisions and acting safely under all conditions.

NASA has been developing UAS Traffic Management System (UTM) from 2013[1] to handle large scale UAS operations in the national airspace. UTM is a collection of infrastructure, policies and procedures to support low altitude UAS operations.[2]UTM has provided the much needed infrastructural requirements of operating simultaneous operation of large number of UAVs.

UAS operations in dense urban areas still remains a challenge in the UAS community. It is assumed that TCL-4 will have BVLOS flight with autonomous vehicle-to-vehicle or internet connectivity.[2] We are developing on-board capabilities for complete autonomous UAS BVLOS operations. Next we present our reference autonomy architecture. It was developed for the last 50 feet of operations is dense urban regions.

NASA UTM TCL4 flight tests have successfully demonstrated UAS operations in urban environments like downtown Reno, Nevada, and Corpus Cristi at Texas. TCL4 has laid the foundations of UAS operations in city landscape. Building on that success our environment provides complete solution to autonomous environments using on-board autonomy.

TCL4+ consists of flights beyond flight tests conducted underTCL4. These include flights operating completely autonomously and beyond visual line of sight; flights sharing common volumes; high density flights etc.

We propose a complete autonomous flight in the NASA Ames Campus in the TCL4+ framework. This paper describes the different components of designing a complete autonomous system that can fly in the urban setting using UTM framework. The main contributions of the paper are building an automated voxel map of the environment from open source maps. Planning global plans and submitting them to USS. And an on-board local planner that can follow the plans to fly in presence of other vehicles.

---

[*] Research Engineer, SGT Inc(KBR), NASA Ames Research Center, Moffett Field, CA 94035. AIAA Member. anjan.chakrabarty@nasa.gov
[†] Aerospace Scientist, NASA Ames Research Center, Moffett Field, CA 94035, AIAA Member

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881

A. UTM architecture

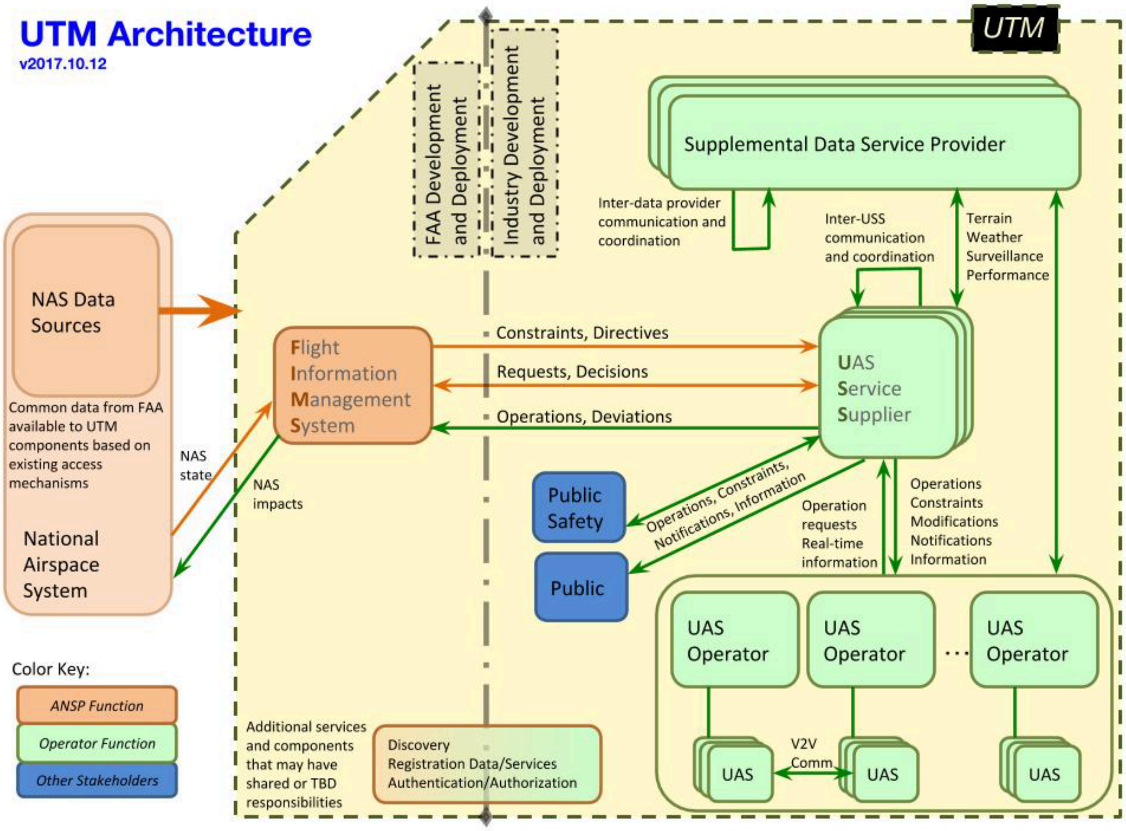


Figure 1. Proposed UTM Architecture for TCL 4.

NASA has been working with FAA for developing UTM concepts. The Federal Aviation Administration (FAA) has outlined a UTM concepts of operations.[3] The document is based on evolution of NASA's UTM concept through different use case development for the last 5 years. The FAA expects that this will be a joint venture with other industries as final rules are placed into effect.. See[3] for the full overview and use cases as envision by NASA/FAA. Here we summarize the relevant portions that are needed for this paper and the section where on-board autonomy interfaces with the larger UTM system.

Figure 1 presents a proposed UTM architecture showing the various actors and components. The UTM is a decentralized communication sharing and exchanging platform with multiple actors working independantly. Most UAS operations are envisioned as individual operators operating their own UAS. FAA does not need to intervene and regulate each UAS operation seperately. Rather FAA will manage only the airspace and the actors operate co-operatively in the airspace sharing situational awareness only. Any deviation from proposed plans are notified in the entire network.

Flight Information Management System(FIMS) is the gateway for data exchange between the different actors. FIMS is being developed at NASA, in collaboration with FAA that support information exchanges and different protocols between operators and stake holders including FAA so that the Operators can work together cooperatively.

At the center of UTM architecture is the UAS Service Supplier (USS). USS is in charge of conduction an UAS operation while maintaining all the rules and regulation. USS is responsible for getting operations approved and executed in a timely manner.USS acts as the communication bridge between different service providers and UAS operations. It is envisioned there will be multiple USS operating in the same geographical location operating co-operatively (for eg. Amazon and Walmart both can act as USS and deliver products to the same neighbor hood).

Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881

Different USSs also communicates with each other relevant data needed for co-operative UAS operations.

UAS operations are confined inside "operational volumes". These are 4-D regions including airspace volumes and time. It is the responsibility of the UAS operator to safely conduct the UAS operation inside the approved volume. Any discrepancies are reported to the USS which handles each case. All flights in UTM TCL1 to TCL4 have had a single vehicle inside an approved operational volume. Most operational volumes that have been flown are manually designed. In this paper we automate the process of generating operational volumes.

Future operations will need many UAVs flying autonomously sharing approved volumes. In the current set up the operator is responsible for completing a particular operation within the stipulated time. As the density of operations increases, one operator will be in charge of multiple UAVs. These vehicles will operate autonomously BVLOS with minimal supervision. We propose sharing the same approved airspace to increase the density of operations.

It is assumed that BVLOS UAS share responsibility with other similar vehicles and manned aircraft for collision avoidance. While collision with manned aircrafts can be mitigated by USS service providers, vehicles should have on-board collision avoidance algorithm to operate BVLOS in dense urban regions. One of the means of on-board collision avoidance is V2V communication between vehicles flying in close proximity.

Other requirements from FAA include that the operator should be able to track their vehicles in almost real time. Operators should notify FAA if the vehicles are in violation of the approved volume. V2V communication is an important aspect in case of declaration and receiving of emergency messages.

This paper describes an automated procedure to develop and register your flight plan with the NASA UTM system through a connected USS. Then follow the plan using on-board local planners to the desired destination. We use a NASA developed USS called AOLUSS to interface to the UTM architecture.

II. SAFE50 Architecture

The NASA SAFE50 study[4] aimed to develop an autonomous UAS for notional last 50 feet of operations, the most difficult phase of autonomous operations. SAFE50 architecture consists of on-board autonomy and decision making. The SAFE50 architecture has been implemented in out Reflection[5] simulation environment. The Reflection Architecture is a real time component based plug and play architecture for rapid development of embedded vehicle systems developed here at NASA Ames.

Detailed design and implementation of a fully autonomous and programmable autopilot system for small scale autonomous unmanned aerial vehicle (UAV) aircraft using the Reflection architecture is describe here.[6] Specific SAFE50 study of the Reflection is addressed in[7].[8]

Figure 2 shows the overall Reflection architecture. With the Reflection plug-and-play architecture, simulation and hardware can used interchangeably. We develop all the components here in simulation which will flight tested. The main components of the Reflection software architecture are sensing and perception, decision making and planning and control. The sensing and perception unit consists of SLAM processing, object detection and vehicle sensors. The planning and control subsystem comprises of path planning subsystem, local planner and autopilot. The Decision making module is responsible for the overall behavior of the system. The Flight Management System(FMS) handles the overall data management (figure 2,).

The AutopilotSystem class is responsible for communicating with the rest of the Reflection system and maintaining the two main objects in the system: the FMS (flight management system) and the Controller. The FMS is responsible for maintaining the list of commands which specify FMS mode instructions. The mode instructions are used by the FMS to provide targets to the controller. The controller is responsible for implementing the control loops which control the aircraft through the vehicle?s actuators.

The Decision Making(DM) module is responsible for the overall behavior of the UAS.[9] The DM communicates with the rest of the UTM system and ensures the overall feasibility if the system. From several feasible Trajectories (1,2,...n) generated using the global planner the DM decided the final waypoints the vehicle has to fly. Together with the list of way-points and the approved volume around them, the Local Planner module generates feasible trajectories inside the approved volume avoiding other vehicles sharing the same volume. DM module decided which is the best possible trajectory for the vehicle to follow.

For this paper major modifications to the SAFE50 architecture includes the development of the UTM client which connects to the USS(UAS service supplier).[10] The UTM client communicates with the Ground Control System (GCS) and obtains the global plan and contingency plans from the decision maker.

Note that the modules described in rectangles are all on-board modules which are carried on by the vehicles. (See the legend for clear distinction between on-board and ground modules.)

We also develop the Path planning System, which is the global path planning system in this paper. The next

Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881

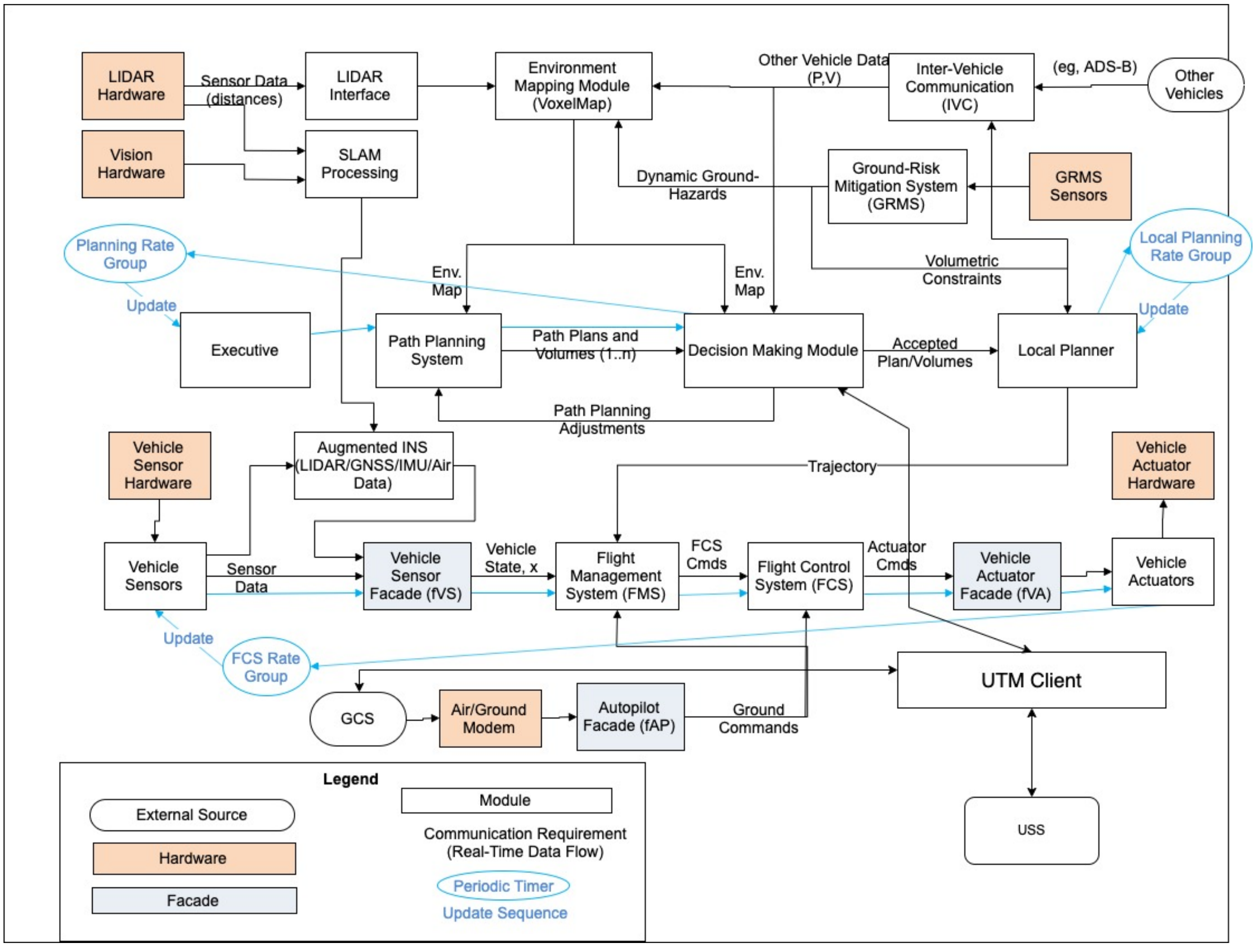


Figure 2. Reflection Software Architecture Diagram.

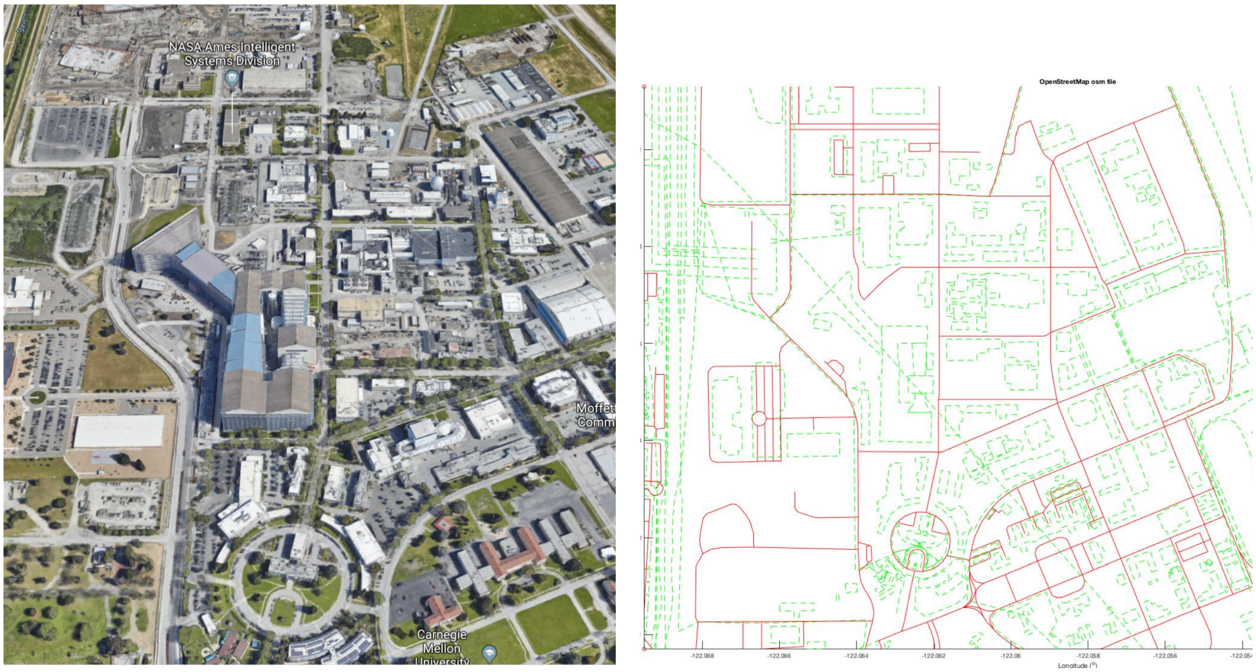
Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881

sections describe the different path planning modules employed in the architecture.

III. Global Path Planning

As shown in the SAFE50 architecture a global path planner always provides a path to the destination and is updated at regular intervals. The global planner also determines other possible paths to closest emergency spots in case the original path cannot be navigated; called the contingency plan. The global planner determines possible paths based on on-board maps the vehicle carries. The on-board maps are simple voxel representations of the environment where the vehicle flies.

A. Voxel Map



(a) Google Map Screen shot of NASA Ames Campus

(b) Building and Street information obtained from Open Street Map for the given NASA Ames map

Figure 3. Information from open source maps for generating voxel map

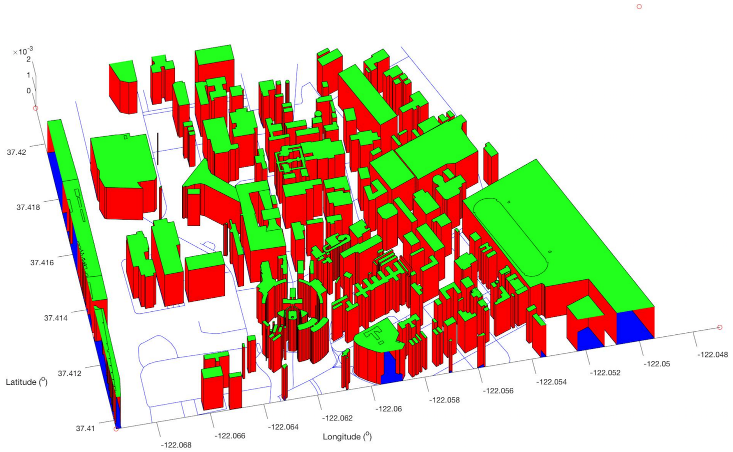


Figure 4. 3D map generated from Open Street Map.

Volumetric representation of the space is important to find feasible paths in an environment.[11] Voxel is a 3d representation of the occupancy map of the environment. Unlike, maps for autonomous cars, a 3D representation of every location is not readily available. In this paper we present a simple method of determining a a-priori voxel representation of any location from open source maps.

To develop 3d maps for any region we utilize the open source 2-D maps available for autonomous cars. Road networks around the world are described using a geo-jason file. A typical geo-jason file containing relevant information is shown below :

```
 <way id="41890574" visible="true" version="3" changeset="66892121" timestamp="2019-02-04T
  <nd ref="518303534"/>
  <nd ref="518303535"/>
....
  <nd ref="3988127778"/>
  <nd ref="518303534"/>
  <tag k="building" v="yes"/>
  <tag k="name" v="N-253"/>
 </way>
```

3-d building informations can be extracted from this data. Such Building data can be obtained for most cities in US and europe and is available for download. These can be used to create 3-d maps for on-board path planning and approval.

Figure 3 shows the Google Map screen shot of the proposed site where we want to conduct flight tests. From open street map we collect the local map data and extract information of all the buildings and roads in the vicinity of the proposed test site. Figure 3 b shows the relevant street and building information extracted from the given map.

Once we have the building information, a 3-d map with constant height can be constructed from it. Figure 4 shows the 3-D representation of the space. More detailed map of building height information can be obtained from different sources but are too large be carried on board. The size of this detailed 3D map with constant altitude is too large to be carried on-board for real time path planning.

For global path planning a simpler on-board map is required. This is done by simplifying this 3D map. Complex shapes of the buildings are encompassed with simple shapes. Complex building shapes are simplified using rectangular parallelepiped. Buildings that are too close together are coalesced into one single occupied unit. After simplifying the given 3D map, a 3D voxel map, ie 3D occupancy map is created which can be carried on-board by the vehicile.

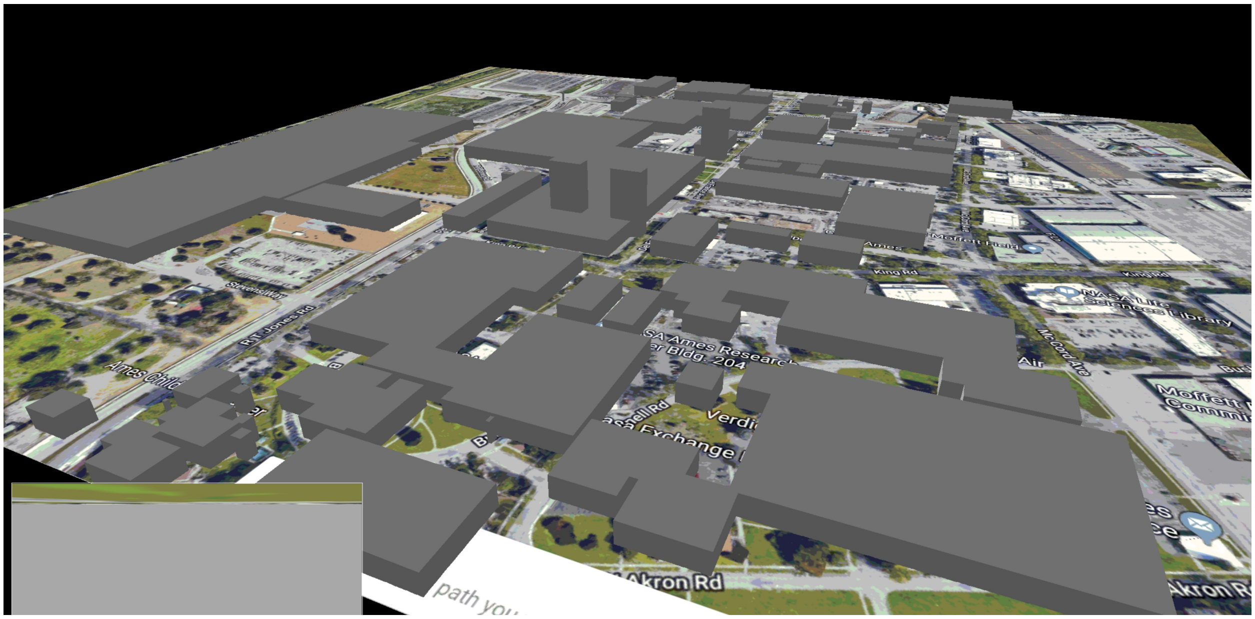


Figure 5. Voxel map generated of NASA Ames Campus.

Figure 5 shows the voxel map created of NASA Ames Campus. The figure shows the voxel map in Reflection

Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881

Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881

simulation. The total size of the voxel map is 48MB which can be carried on-board for real-time planning and re-planning.

This voxel map is used to find optimal real time path planner on-board. The next section describes the path planning architecture.

B. RRT* planner

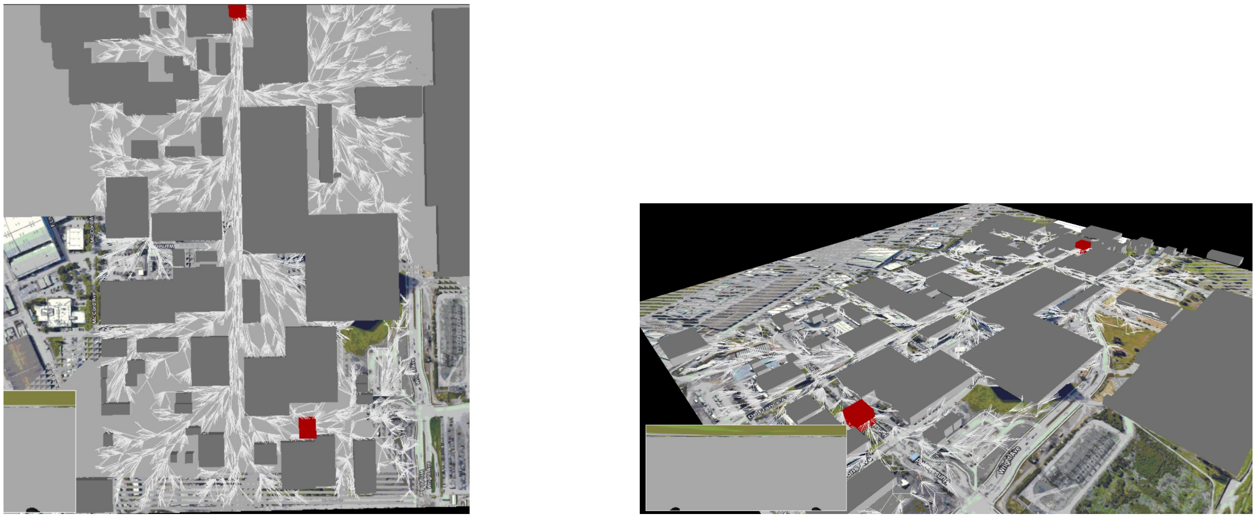
Path planning for aerial robots is an active area of research. For a comprehensive survey of motion planning algorithms for autonomous UAVs see[12] There are plenty of algorithms for path planning, most notable being A* algorithm.[13] Autonomous cars use A* algorithm for finding optimal paths to their destination. The major drawback for A* algorithm is that A* operates in a predefined grid. For our application a sampling based motion planning algorithm is more suited.

The Rapidly-exploring Random Tree (RRT)[14] algorithm, based on incremental sampling of all the motion primitives calculates feasible solutions for any robot. It is designed to effectively search nonconvex high-dimensional spaces by randomly building space filling trees. RRT algorithms have been sucessfully impleted in many application with one major drawback: the RRT algorithm is not optimal.

RRT*[15] algorithm extends the RRT algorithm by identifying optimal paths. The RRT* algorihm modifies the RRT solutions by adding cost to each node. The algorithm then re-wires the tree if it finds paths that lead to lower cost with more sampling. For detailed understanding of the RRT* algorithm see,[15].[16]

We developed a RRT* algorithm for this application to quickly find feasible paths to the destination from the current location of the vehicle. From the initial starting point a Tree is generated using RRT* algorithm. Figure 6a shows the growth of the Tree in the Reflection simulation environment. The algorithm runs in the background and continuously improves upon plans generated and eventually converges to an optimal solution.

Also note that we have an optimal solution to any location in the entire space. RTT* solution not only provides a path to the goal but also to all possible contingency locations in case of emergency.



(a) RRT* paths generated on a voxel map

(b) RRT* paths generated on a voxel map.

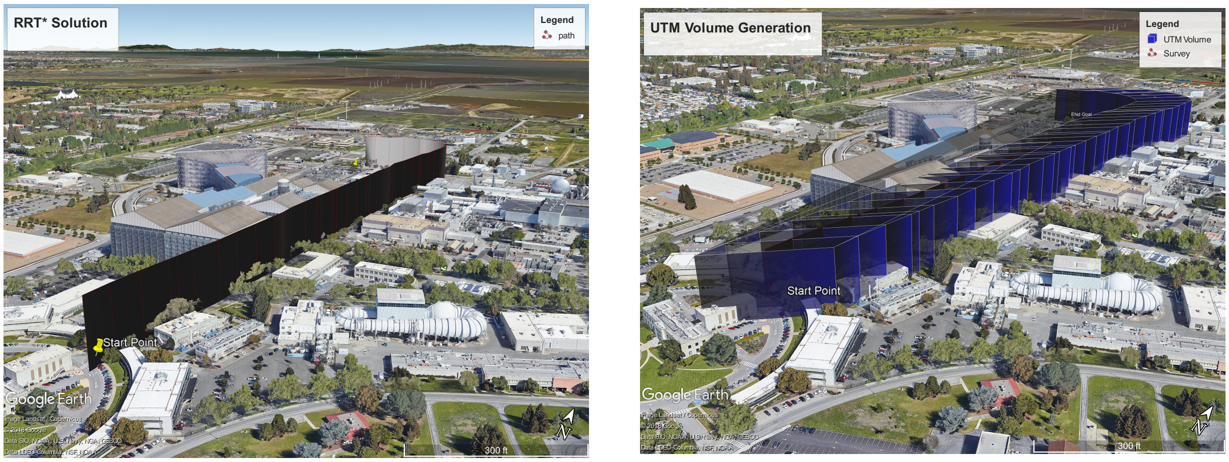
Figure 6. RRT* paths generated on a voxel map of Downtown Indianapolis

Figure 6b shows successful paths generated by the RRT* algorithm in a voxel map. The algorithm finds not only the optimal solution but also other routes to the destination. This is needed in case of the original planned path being unavailable, we have alternate paths to get to the goal.

UTM TCL4 requires vehicles to have paths defined to the goal and contingency volumes at all times during the flight. Thus the RRT* algorithm always runs in the background providing feasible paths to the destination and other contingency volumes.

Figure 7a shows the paths generated by the reflection solution using generated voxel maps on Google Earth. The paths when converted from body frame to the global co-ordinates shows successful avoidance of building in Google Earth. each segment of the path is converted into a volume which are them registered with UTM for flight authorization.

Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881



(a) RRT* paths generated on a voxel map

(b) RRT* paths generated on a voxel map.

Figure 7. RRT* paths generated on a voxel map of Downtown Indianapolis

The next section describes the USS format required to register the automated plans generated.

IV. USS

Every flight that participates in UTM flight has to go through an authorization procedure. Currently there is no means of automated authorization of flight approvals.We are working with our partners to develop this automated authorization of developed flight plans.

The process of getting approval with UTM goes through an USS. We are using our NASA USS called AOLUSS. The process involved first getting a token from the USS server. Once we get a receipt of token the operation plan is sent over. The operation plan is set as volumes using JSON format.

JavaScript Object Notation (JSON) is a text format for the serialization of structured data.? Simply put, it is a structured way of organizing key-value pairs for the exchange of data. Text encoding is specified as UTF-8. See RFC 8259 for details. The following is an example provided by json-schema.

```
{
 "gufi":"764abeb8-e563-48ce-b31c-3780b85d47ee",
  "uas_registrations": [
    {
      "registration_id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "registration_location": "utm_registration.nasa.gov"
    }
  ],
....
      "beyond_visual_line_of_sight": true,
      "operation_geography": {
        "type": "Polygon",
        "coordinates": [
          [
              -121.000, 38.000
.......
            ]
      },
      "near_structure": true,
      "volume_type": "TBOV"
    }
```

Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881

```
  ],
......
      "contingency_response": "LANDING",
      "contingency_polygon": {
        "type": "Polygon",
        "coordinates": [
              -121.000, 38.000
.......
            ]
.....
```

As shown in the geo-jason file each operations consists of ”Polygons” with global co-ordinates. Also we have to pre register contingency response and contingency plans as shown above.

Also each volume registered with UTM has a start and end time associated with it. For simplicity we booked the entire flight volume for the entire duration of the flight. For more dense operations we can ”release” each volume after the vehicle has flown through them.

The USS then sends the flight plan through the UTM server to check with other connected USSs or no fly zones before ”Accepting” the flight plan.

During flight plan send USS our position at 1 Hz to show that we are complying with the submitted flight plan. Any deviance from the submitted plan results in the vehicle being declared as vogue.

Once the flight is completed we have to report to USS that the flight has been successfully conducted and we close the operations. These are the steps our UTM client handles int he reflection simulation. We have successfully interfaces the Reflection simulation with AOLUSS to show successful implementation of autonomous flight plan generation and UTM integration.

After the flight plan is registered and we are ready to fly on particular trajectory the local planner handles the vehicle while flying inside the approved UTM volume.

V. Local Planner

The global planner provides paths to the goal using static obstacle information, i.e the a-priori voxel map. But in a changing environment a local planner is needed to avoid other vehicles flying in the same shared airspace, or avoiding other obstacles that were not know a-priori. We have discussed in details the development of local planner in the SAFE50 context but is briefly mentioned here for completeness.

A. V2V communication

One assumption is that the volumes registered with the UTM will be used by multiple vehicles equipped with V2V communication.

A detailed simulation of V2V communication was described in.[17] Is is assumed that all vehicles participating in UTM will communicate with each other over V2V or V2X communication. In[17] we described a simulation method of V2V communication and estimation of position of vehicles flying in a shared airspace. Once we have an estimation of possible occupied volume by all the vehicles a local planner avoids those regions to give a collision free path.

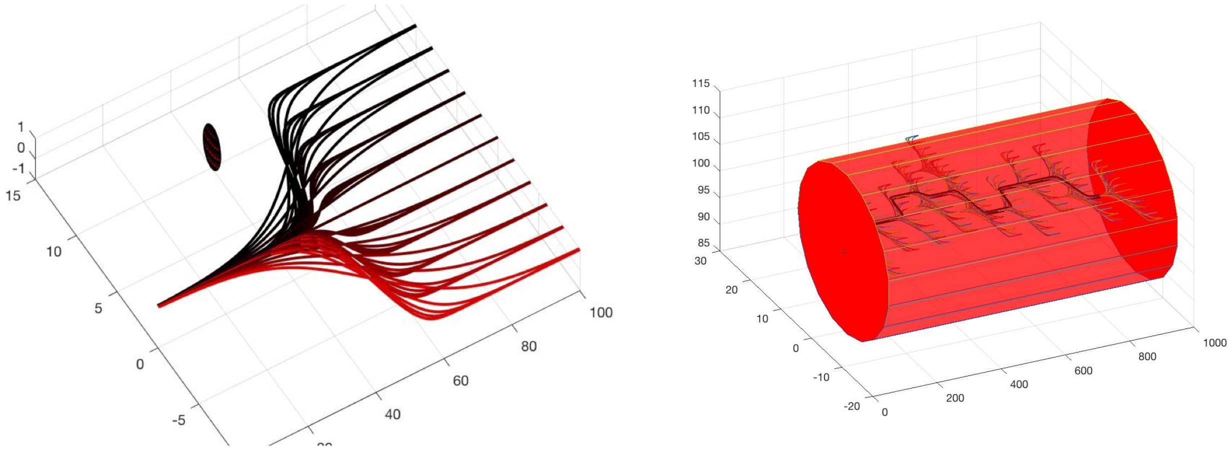
B. Tree Planner

The tree based on-board local planner is described in details here.[18] A set of motion primitives is determined for he vehicle and a local plan is developed which avoids other vehicles communicating over V2V or other obstacle detected using Lidar. The default solution of the local plan is the global plan solution. If new obstacles are observed, the local plan deviates from the global plan but returns to the original plan after mitigating the obstacle/other vehicles.

Figure 8a shows the motion primitives and figure 8b shows a local plan developed. See[18] for details of implementing multiple vehicles flying and avoiding each other using a local planner andV2V communication.

Figure 9 shows the paths generated inside UTM volumes while avoiding other vehicles using V2V communication. See[18] for details about V2V communication and on-board local planner.

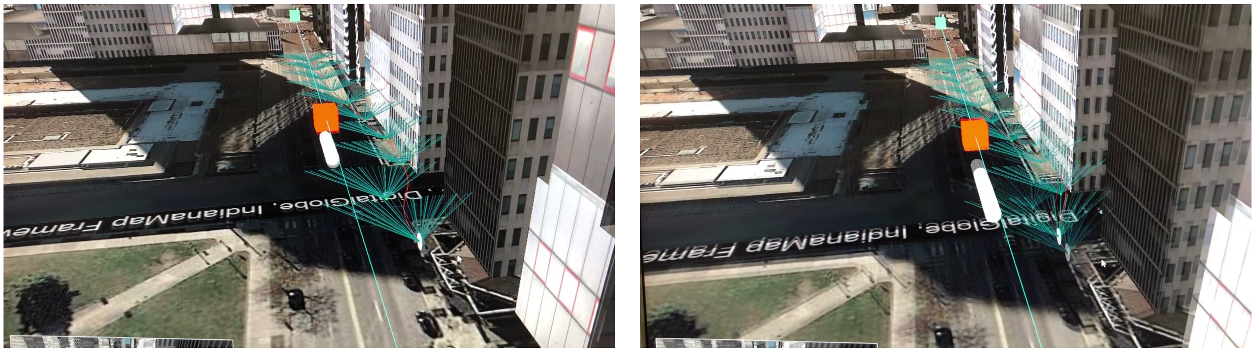
Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881



(a) Motion Primitives with cost function depicting nearness to an obstacle.

(b) Local Plan generated inside the approved volume.

Figure 8. Motion primitives and Local Plan generated at each instant of time



(a) Tree based trajectory generations

(b) All the branches of the tree are inside the UTM approved volume

Figure 9. Dynamic Obstacle Avoidance using V2V communication and Recursive Tree Planner

Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881

VI. Future Work

In this paper we have demonstrated an entire autonomous architecture for flying in dense urban environments beyond visual line of control. We demonstrated in simulation an entire framework that works with the current UTM system to fly autonomously in dense urban environments. We are working to complete the flight test at NASA Ames to test the entire system in hardware. To accomplish this we have already completed some initial hardware testing.

We have used a ground based Lidar on-board our DJI M600 vehicle to map the campus. This generated on-board voxel maps to be used by the local planner for on-board path planning.

We have also tested several Microhard modems for V2V communication. We have tested the range, delay and noise associated with the on-board modems for flight tests. We have also conducted preliminary flight tests to show on-board path planning being executed autonomously by the vehicle. We will test this simulation architecture in flight pending approvals through different agencies at NASA.

VII. Conclusion

In this paper we have described a complete autonomous architecture for flying UAVs in the national airspace under UTM. For vehicles flying autonomously in TCL4+ environments this forms a roadmap to execute complete autonomous flights. This has paper described the different subsections of the SAFE50 autonomy architecture. We have described the method to create a 3d voxel map from open source maps available. This was used to find global solution to the destination using RRT* algorithms. We have shown how the autonomous flight plans are submitted to UTM for flight approvals and the steps to achieve that. Finally the paper has also shown how this architecture has been used to fly multiple vehicles sharing the same approved airspace.

Acknowledgments

The authors would like to thank our colleagues in the NASA SAFE50 and UAS Traffic Management (UTM) projects for their collaboration and insight. We specially that George Lawton for all the help with AOLUSS simulation.

References

[1] P. H. Kopardekar, "Unmanned aerial system (uas) traffic management (utm): Enabling low-altitude airspace and uas operations," 2014.

[2] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. Robinson, "Unmanned aircraft system traffic management (utm) concept of operations," in *AIAA Aviation Forum*, 2016.

[3] "Concept of operations v1.0." [Online]. Available: https://utm.arc.nasa.gov/docs/2018-UTM-ConOps-v1.0.pdf

[4] K. S. Krishnakumar, P. H. Kopardekar, C. A. Ippolito, J. Melton, V. Stepanyan, S. Sankararaman, and B. Nikaido, "Safe autonomous flight environment (safe50) for the notional last 50 ft of operation of 55 lb class of uas," in *AIAA Information Systems-AIAA Infotech@ Aerospace*, 2017, p. 0445.

[5] C. Ippolito, G. Pisanich, and K. Al-Ali, "Component-based plug-and-play methodologies for rapid embedded technology development," in *Infotech@ Aerospace*, 2005, p. 7122.

[6] C. Ippolito, G. J. Pai, and E. W. Denney, "An autonomous autopilot control system design for small-scale uavs," 2012.

[7] C. A. Ippolito, K. Krishnakumar, V. Stepanyan, A. Chakrabarty, and J. Baculi, "Safe50 reference design study for large-scale high-density low-altitude uas operations in urban areas," 2019.

[8] C. A. Ippolito, K. S. Krishnakumar, V. Stepanyan, A. Bencomo, S. Hening, and J. Baculi, "Autonomous uas operations in high-density low-altitude urban environments," in *AIAA Scitech 2019 Forum*, 2019, p. 0687.

[9] A. Chakrabarty, R. A. Morris, X. Bouyssounouse, and R. Hunt, "An integrated system for autonomous search and track with a small unmanned aerial vehicle," in *AIAA Information Systems-AIAA Infotech@ Aerospace*, 2017, p. 0671.

[10] J. L. Rios, I. S. Smith, P. Venkatesan, D. R. Smith, V. Baskaran, S. M. Jurcak, S. K. Iyer, and P. Verma, "Utm uas service supplier development: Sprint 2 toward technical capability level 4," 2018.

[11] J. Amin, J. Boškovic, and R. Mehra, "A fast and efficient approach to path planning for unmanned vehicles," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, p. 6103.

[12] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous uav guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, p. 65, 2010.

[13] A. Stentz, "Optimal and efficient path planning for partially known environments," in *Intelligent Unmanned Ground Vehicles*. Springer, 1997, pp. 203–220.

[14] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[15] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1478–1483.

[16] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online uav replanning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 5332–5339.

[17] A. Chakrabarty, C. A. Ippolito, J. Baculi, K. S. Krishnakumar, and S. Hening, "Vehicle to vehicle (v2v) communication for collision avoidance for multi-copters flying in utm–tcl4," in *AIAA Scitech 2019 Forum*, 2019, p. 0690.

[18] A. Chakrabarty, V. Stepanyan, K. S. Krishnakumar, and C. A. Ippolito, "Real-time path planning for multi-copters flying in utm-tcl4," in *AIAA Scitech 2019 Forum*, 2019, p. 0958.

Downloaded by NASA AMES RESEARCH CENTER on January 13, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2020-0881