JOURNAL OF AIRCRAFT
Vol. 43, No. 2, March–April 2006

Multidisciplinary Approach to Intelligent Unmanned-Airborne-Vehicles Mission Planning

Iain A. McManus* and Rodney A. Walker†
Queensland University of Technology, Brisbane, Queensland 4001, Australia

The increasing popularity of unmanned airborne vehicles (UAVs) has led to a greater desire to deploy UAVs within civilian airspace. Previous UAV platforms have possessed a limited degree of onboard intelligence. Consequently these UAVs have been constrained to operations within restricted areas or constantly supervision by a human operator. These previous approaches have not enabled the full potential of UAVs to be exploited. The onboard intelligence of UAVs needs to be increased to fully unlock their potential. This paper focuses upon increasing the UAV's onboard intelligence in one specific area, namely, mission planning. Onboard planning techniques have been developed for UAVs in the past. However, the previous approaches have not considered civilian airspace applications and as a result have not been suitable for these applications. This paper presents a novel multidisciplinary approach to onboard UAV mission planning, which combines robotics and three-dimensional graphics techniques. The paper will describe the development of the multidisciplinary approach and its application to mission planning for UAV operations in civilian airspace. Specific optimizations that have been made for the civilian airspace problem will be discussed. Results will be presented based upon testing scenarios built using real-world airspace and terrain information from southeast Queensland, Australia. These results will prove the higher performance and operational freedom achieved by this new approach compared to previous approaches.

I. Introduction

IN the last decade UAVs have increased in sophistication, advancing from deployments as target drones for military operations to performing reconnaissance and strike missions in war zones. UAVs are now routinely deployed for civilian purposes with applications such as meteorological data collection and farming (e.g., Yamaha's RMAX). However, a limit exists on how extensively current UAVs can be deployed within civilian airspace. Presently, the intelligence onboard UAV platforms is low. The consequence of this is that UAVs must be either deployed within a restricted area or must be continually monitored by trained personnel. This constrains the deployment of UAV platforms and denies access to the full capabilities of UAVs.

It is therefore desirable to increase the intelligence onboard UAVs to enable the full exploitation of their potential. The purpose of increasing the onboard intelligence is to increase the capabilities of the UAV and to reduce the workload placed upon its human operators. The research presented in this paper focuses upon the mission planning aspect of the onboard intelligence. In this context mission planning refers to the process of determining the path that the UAV needs to fly in order to meet the mission objectives.

In current UAV systems the mission planning is typically performed by a human operator. The human operator must be aviation trained, and the process of planning the mission takes some time. In this context aviation trained implies that the operator has been trained to plan missions, which will be conducted under visual-flight-rules (VFR) conditions. The planned mission is provided to the UAV, which flies the mission but which cannot autonomously make changes to the mission plan during flight. This approach is both inflexible (the UAV cannot change its own mission plan) and costly (in terms of planning time and operator training). Onboard mission planning is therefore preferred in order to achieve higher flexibility and lower costs.

The topic of onboard mission planning for UAVs has been considered to a limited extent. Previous research has focused upon military applications and has not given consideration to specific issues present within civilian airspace (e.g., airspace boundaries, prescribed cruising levels, etc). Past approaches have typically operated in two dimensions and have only permitted the UAV to fly within narrow corridors.[1,2] Limitations such as these are undesirable for civilian airspace applications as they prevent the full potential of the UAV from being exploited. The limitations of the previous approaches will be specifically addressed by the new approach presented in this paper.

The objectives of the research presented here are 1) to perform three-dimensional onboard mission planning for civilian airspace operations, where the UAV will be granted the maximum freedom of movement within the environment; and 2) to plan efficient paths in terms of distance traveled, time required, or fuel consumed.

A novel multidisciplinary approach has been adopted to achieve these objectives. This approach draws upon techniques from the three-dimensional graphics and robotics fields.

Three-dimensional graphics techniques have been employed to enable the UAV to construct and maintain its situational awareness. This situational awareness is used by the UAV to perform high-level tasks such as mission planning. The mission-planning algorithms have been adapted from techniques used within the robotics field for path planning. The combination of the three-dimensional graphics and robotics techniques enables the UAV to plan its own missions and to do so with greater freedom (of movement) than previous approaches. The advantages of this novel multidisciplinary approach will be demonstrated using scenarios based upon airspace and terrain data from southeast Queensland, Australia.

The first section of the paper discusses the autonomous path-planning problem and the development of the research framework. Following on from this the second and third sections will describe the development of the new multidisciplinary approach. These sections will examine the creation of situational awareness and the process of autonomous path planning. The simulation environment used for performing the testing will be discussed in section four. The fifth section will describe the testing scenarios used to evaluate the performance of the new approach. The results of testing performed based upon real-world scenarios will be presented in section six. An analysis of the results achieved will be presented in section seven.

Received 21 December 2004; revision received 15 March 2005; accepted for publication 16 March 2005.

*Researcher, GPO Box 2434, Cooperative Research Centre for Satellite Systems. Member AIAA.

†Project Supervisor, GPO Box 2434, Cooperative Research Centre for Satellite Systems.

The discussion of further research and the conclusion will be given in section eight.

II. Autonomous Mission-Planning Problem

This paper presents one possible solution to the autonomous mission-planning problem. However before the solution can be discussed in detail, it is first necessary to understand the problem in question. The mission-planning problem is, simply put, that the UAV is given a list of mission objectives which it must satisfy. To satisfy the objectives (i.e., solve the problem), a plan must be generated of how to reach the objectives given that there are obstacles which must be avoided.

This section will examine existing solutions to the problem and identify issues that prevent these approaches from fully exploiting a UAV's potential. Based upon the identified issues, a solution will be developed aimed at resolving these issues.

Currently most mission plans for UAVs are prepared by a human operator. This avoids the need to perform mission planning onboard, but at a cost. The human operator needs to be aviation trained (trained in VFR mission planning), and the process of planning the mission can take considerable time. Naturally the more complex the mission the longer the time required for mission planning. There are three main issues with having a human operator perform the mission planning.

Firstly, the UAV is constrained to flying along the path that the operator has planned. During a mission, the UAV cannot change the mission plan; this can only be done by the human operator. This makes it difficult for the UAV to adapt to changes within the environment during the mission, as shown in Fig. 1.

Secondly, the workload placed upon the operator is high. This is especially true if the operator is required to replan large sections of

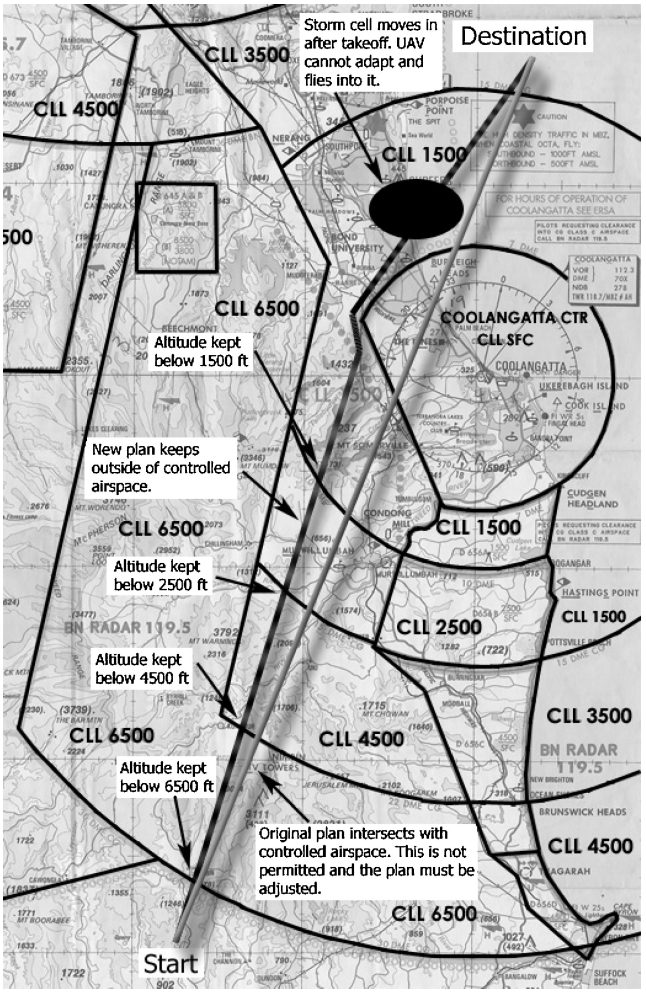

Fig. 1 Consequence of lack of adaptability in UAV mission planning.

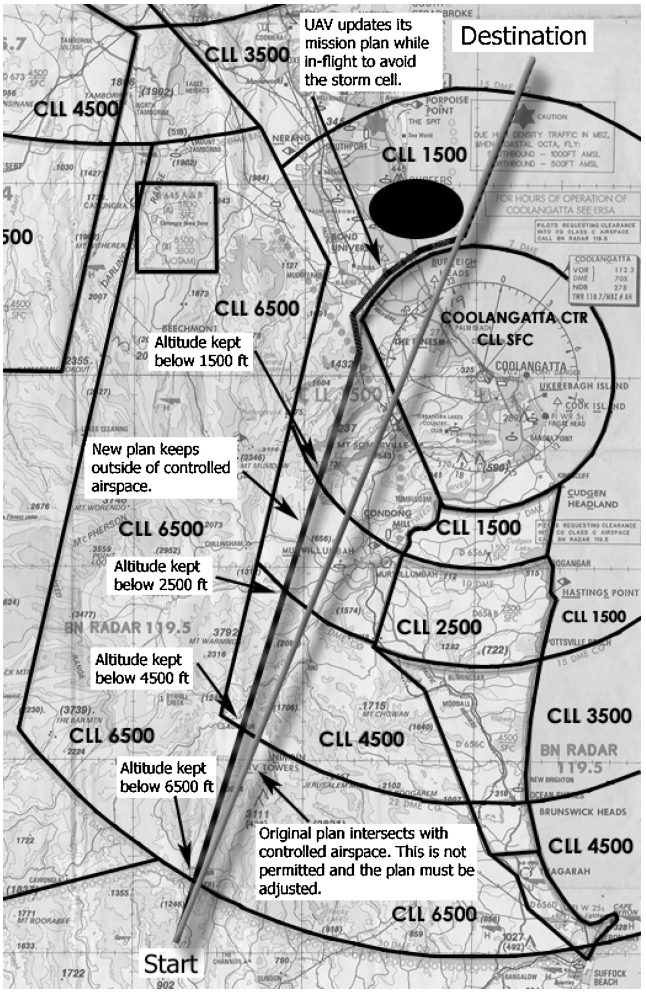

Fig. 2 Onboard planning capability enables the UAV to adapt to the environment.

the mission during a flight. Finally, the operator must be aviation trained in order to be capable of planning an appropriate path. This in turn raises the training and personnel costs for operating the UAV.

The presence of these issues therefore makes the transfer of mission planning to onboard the UAV a desirable action. As shown in Fig. 2, by providing the UAV with an onboard planning capacity it can adapt to changes in the environment.

Some UAVs that can perform their own mission planning have been developed. These systems have focused upon military operations within a war zone. They have therefore placed their emphasis upon factors such as minimizing the size of the target presented to enemy RADARs and other similar military-oriented attributes.[3] These systems, although capable, are not suitable for civilian airspace applications because of a number of issues.

Firstly, the previous approaches make no accounting for airspace boundaries or issues associated with them (e.g., being active only in certain hours). Secondly, many of the previous approaches only consider the planning problem in two dimensions (often a fixed height above ground is assumed). A two-dimensional approach is not suitable for the civilian airspace environment where stepped airspace boundaries (i.e., a three-dimensional shape) are commonplace. Finally, the previous approaches typically constrain the UAV to operations along specific paths. This limits the freedom of the UAV and potentially blocks out more efficient flight paths.

Current onboard UAV mission-planning systems are therefore not suitable for operations within civilian airspace. A mission-planning solution that addresses these issues is required. This paper describes the development of an intelligent mission planner (IMP) designed specifically to address these issues. To provide the necessary

mission-planning capability, the IMP needs to 1) operate based upon a fully three-dimensional representation of the world; 2) plan missions onboard the UAV in a manner suitable for civilian airspace applications; 3) plan missions that optimize distance traveled, time required, or fuel used; and 4) enable the UAV to fly anywhere within civilian airspace that a human pilot would be permitted to fly.

These requirements define the framework for the intelligent mission planner research presented in this paper. Consideration is not being given at this stage, of the research, to the sensing side of the problem. The assumption is made that the information required will be available. The research focuses upon using and representing the sensory information to assist with mission planning.

Consideration must also be given to the rules under which the flight will be conducted. The flight could be conducted under VFR conditions, instrument-flight-rules (IFR) conditions, or a mix of both (i.e., VFR during one part of the flight, IFR in another). VFR conditions impose the least constraints upon the operations of the UAV, which is one of the research objectives. Furthermore many civilian applications, such as crop dusting, are conducted under VFR conditions. Therefore the decision was made that the IMP would plan missions assuming that the UAV was flying under VFR. This requires the IMP to avoid entry into controlled airspace (typically VFR flights are not permitted into controlled airspace). The UAV must also fly at the appropriate hemispherical altitude based upon its heading.

To satisfy these requirements, the IMP will require a mechanism to store, represent, and interact with information that describes the operating environment. The IMP will also require mission-planning algorithms that operate based upon the current information stored about the operating environment.

The proceeding sections will discuss, in detail, the multidisciplinary techniques that have been implemented in order to create the IMP.

III. Achieving Situational Awareness

A human pilot operating an aircraft requires a high degree of situational awareness. The intelligent mission planner will also require an awareness of the environment in which it is operating. The situational awareness will need to contain sufficient information for the IMP to plan its assigned mission. The IMP will need to construct, update, and interact with its awareness of the operating environment.

A human pilot's situational awareness primarily comes from four main sources: flight instrumentation, voice-based communications, the human vision system, and maps/charts. The IMP can construct the required situational awareness in a similar manner, using flight instrumentation. Maps and charts are available and can be encoded into digital forms, which are then stored onboard the aircraft. This paper focuses on performing onboard mission planning using these digitally encoded charts. Vision presents a much more difficult problem that is outside the scope of this research. This research assumes that the required information can be sensed by some means and is available digitally (e.g., collision with other aircraft that might have relied upon the human vision system and voice communication).

Constructing a dynamic, complex three-dimensional digital world has historically been restricted to large dedicated high-performance computing systems. With the rapid advances in computing technology, constructing such digital worlds is now possible on, what would be considered by today's standards, low-end computers. These advances have made it possible to construct a digital representation of the civilian airspace environment, which incorporates terrain, airspace boundaries, weather, and other aircraft. This digital world is the key to constructing the situational awareness required by the IMP to perform onboard mission planning. As new sensor information is collected the digital world can be updated (e.g., add a new aircraft) to provide a snapshot of the current environment. The intelligent mission planner then interacts with the digital world. These interactions include determining if the straight-line path between two waypoints intersects with an entity in the world (e.g., terrain). The interactions with the digital world are performed using techniques established in the three-dimensional graphics and robotics fields that have been optimized for speed and numerical efficiency.

The first step in constructing this digital world is to identify exactly of what the UAV needs to be aware. The critical entities that must be represented are as follows: 1) terrain, 2) airspace boundaries, 3) adverse weather (e.g., storm cells), 4) other aircraft, 5) tall buildings, 6) navigation aid locations and types, 7) runways, and 8) radio frequency zones.

Other entities could be incorporated if the operational concept requires it; however, the preceding list was considered to be sufficient to meet the research objectives of this project.

The preceding entities form an important part of the civilian airspace environment, and the UAV needs to be aware of them. The majority of the entities (terrain, airspace, adverse weather, other aircraft, and buildings) are objects that a human VFR pilot would be aware of (through instrumentation, maps, etc.) and avoid. The other entities are either informational (navigation aids and radio frequency zones) or are mission start/endpoints (runways).

A number of sensing capabilities has been assumed for this research. In addition to standard attitude and position information, the UAV must be aware of the location of other aircraft and weather (e.g., storms). A number of methods exists for obtaining this information, for example, radar-based systems or the airborne dependent surveillance–broadcast can provide the location and speed of nearby aircraft. Similarly, a weather radar can also provide location information for adverse weather conditions. However, the focus of this paper is on how to represent and utilize the information that is assumed to be available. No further consideration is given as to what sensing capabilities must be implemented in order to provide this information.

In addition to the location of the entities within the world, their dimensions must also be known. This means that for each entity within the world there needs to be a description of its dimensions. This description is necessary for the entities to be "drawn" within the digital world. The creation (i.e., drawing) of, and interaction with, three-dimensional models is an area that has been studied in detail within the three-dimensional graphics field for both hardware and software applications.

The industry standard for rapidly representing three-dimensional models is to use a mesh of triangles (triangular mesh).[4–7] Triangles are geometrically and mathematically simple shapes. They are constrained within a single plane and are fast to interact with (e.g., detecting if a line intersects the triangle). Complex objects can be represented using triangular meshes, with the complexity limited only by the resources available for the storage of the mesh. Fast, efficient algorithms have been developed for interacting with triangles and by association triangular meshes.[8–10] The triangular mesh representation is used by the majority of three-dimensional graphics hardware and software systems as the standard for rapid modeling of objects.

This approach has been used to model the entities within the digital world. This is a unique application of three-dimensional computer graphics algorithms to provide situational awareness for a UAV operating in civilian airspace. Before lower-level details of the creation of the situational awareness are provided, a brief explanation of the terminology, that will be used will be provided:

1) *Entity* refers to an object from the civilian airspace environment, for example, terrain, airspace boundaries and other aircraft.

2) *Model* is the digital representation of an entity. A model stores both the properties of the entity [e.g., the type (airspace boundary, terrain, etc.)] and the shape of the entity.

3) The shape of an entity is stored (digitally) using a *primitive*. A primitive is one of a group of predefined shapes that can be used to represent an entity.

4) Finally, primitives are converted to a *triangle mesh* to facilitate fast interactions (e.g., detecting if a straight line passes through the model of an entity) with them.

These terms are shown in Fig. 3.

Every entity is converted to a triangular mesh-based model with which the intelligent mission planner can interact. These interactions include detecting if the straight-line path between two

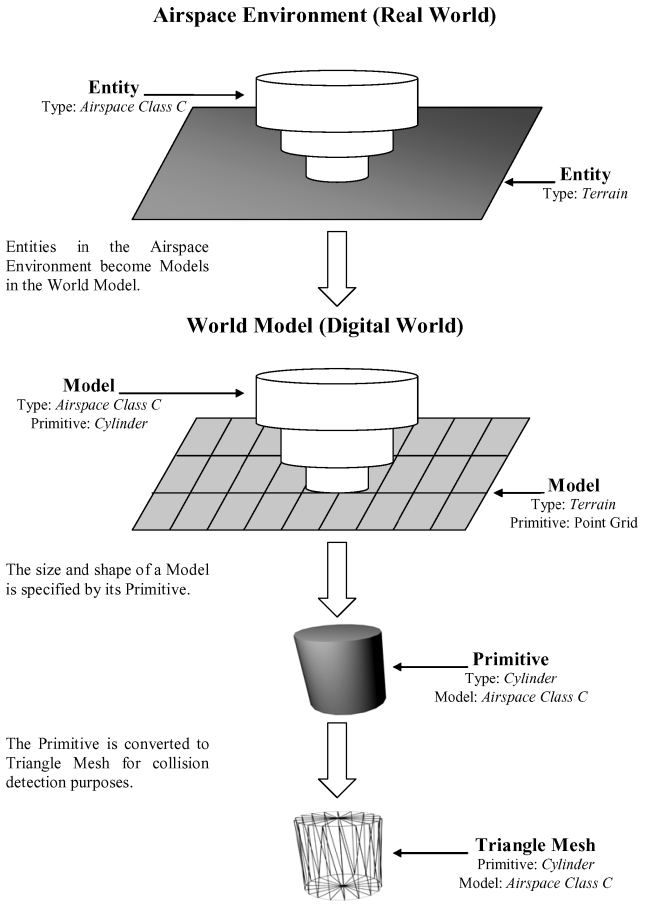

Fig. 3 World modeling terminology.

waypoints will collide with an entity and if a location lies within an entity.

The models within the digital world cannot be of arbitrary shapes. This constraint was used to limit the complexity of the modeling and interaction process. The models are constrained to being one of a given set of standard models known as primitives. Six different primitives were chosen to form the base set of shapes, which can be used to create the model of an entity. The chosen primitives are 1) box, 2) sphere, 3) cylinder, 4) extruded polygon (constant cross section in horizontal plane), 5) flat plane (runways only), and 6) point mesh (terrain only).

The flat plane and point mesh primitives have been designed specifically for modeling runways and terrain respectively. Neither runways nor terrain need to be modeled as a closed three-dimensional shape. The flat plane and point mesh primitives provide a simpler method for modeling these entities.

The extruded polygon representation is a primitive designed specifically for representing airspace, though it can be used to represent other objects. The extruded polygon has a constant cross section in the horizontal plane. All airspace boundaries can be represented using one or more extruded polygons. For this reason rather than implementing a complex three-dimensional primitive, which can represent changes in cross section in all axes, a simpler extruded polygon was implemented. This limitation greatly simplifies the modeling and interaction process by reducing the amount and complexity of the information to process.

This unique approach of using three-dimensional graphics to provide situational awareness for mission planning in civilian airspace has been successfully implemented into the intelligent mission planner. A three-dimensional digital representation of the airspace environment has been constructed using information from digital elevation models and airspace charts. The current digital world stores the terrain and airspace details for a section of the southeast coast of Australia. This section covers a region measuring 270 n miles (east-west) by 240 n miles (north-south) and contains over 70 airspace boundaries.

By continually updating the digital world based upon current sensor data, the IMP, and other UAV systems, can be provided a with high degree of situational awareness. This situational awareness provides the foundation for the mission-planning algorithms within the IMP.

IV. Autonomous Mission Planning

Once the digital representation of the world (i.e., the situational awareness) has been created, high-level activities such as mission planning can be performed. The autonomous mission-planning algorithms are the core component of the IMP. These algorithms use both the mission objectives provided by a human operator and the onboard situational awareness to plan the mission. The planning process involves determining an efficient and collision free flight path, which will achieve the assigned mission objectives. The output of the planning process is a series of waypoints (i.e., flight path), which the UAV must fly to (in sequence) in order to achieve the mission objectives.

The field of mission planning has been studied in depth by the robotics industry.[11–20] Research to date can be divided into two categories: solutions where all information is known a priori and solutions where either partial or no information is known a priori. Information in this case refers to the layout of the world and the locations and dimensions of entities within the world. Solutions that initially have a full awareness of the operating environment (i.e., all information provided a priori) can develop a complete plan from the start to the end of the mission. These plans are known as "global plans." As global plans cover the entire mission, they can be made efficient over the entire course of the mission (i.e., globally efficient).

The disadvantage of developing global plans is that the mission planner requires a large amount of a priori information in order to develop the plan. The alternative approach is to plan for the near future (e.g., 10 min). This uses only a partial awareness of the environment and requires less information. Solutions of this type can only develop plans for the immediate future (i.e., plan for 10 min into the future). These plans are known as "local plans." Local plans have the disadvantage that they might not be capable of achieving the required goal as they can become lost or stuck because of their limited view of the world. Finally, as local plans cannot consider the entire mission at once they are not guaranteed to produce plans that are efficient over the entire course of the mission. Efficient local plans can only be developed for the immediate future, known as "locally efficient."

For UAV mission planning it is possible to provide sufficient information a priori to enable the UAV to develop global plans. This information would include as a minimum the terrain and airspace boundary information. Short-term disturbances (e.g., avoiding a collision with another aircraft) can be accommodated through reflex-style algorithms that make local changes to the global mission plan. This mixed approach of global-planning algorithms and reflexes will enable the IMP to plan efficient paths within civilian airspace. At this stage of the research, consideration is not being given to these reflexive algorithms. The focus is upon developing efficient global-planning algorithms suitable for mission planning in civilian airspace.

A range of methods, for global planning in known environments, has been developed and proven by researchers. These methods range from algorithms that partition the world into grids and then move from cell to cell to artificial-intelligence-based path searches. The most common methods available are as follows: 1) cube space (C-space) algorithms, where the digital world is divided into cubes[21] and where these are a three-dimensional extension of two-dimensional grid worlds[22,23]; 2) octree algorithms, which are a hierarchical representation of C-space that enables the digital world to be viewed at multiple resolutions[15,17]; 3) Voronoi diagrams, which compute a graph of the free paths within the digital world[1–3,24]; and 4) dynamics-based trajectory planning systems, which incorporate the aircraft's dynamics into the planning processes.[25–27]

Voronoi diagrams have previously been employed for UAV mission planning. They are typically combined with artificial-intelligence-based graph search algorithms.[1–3,24] An example Voronoi diagram is shown in Fig. 4.

Voronoi diagrams can be used to incorporate the concept of risk into the mission-planning process by keeping further away from dangerous obstacles (e.g., a military test range). Because of this capability, Voronoi diagrams have often been used for military planning systems.[1–3,24] The primary disadvantage with Voronoi diagrams is that the UAV is required to fly along the lines of the Voronoi graph (except when transiting to/from a waypoint). This prevents the UAV from flying close to obstacles and thus eliminates potentially more efficient plans. Previous UAV implementations[1–3,24] have only used Voronoi diagrams for two-dimensional mission planning. However the civilian airspace environment is three dimensional and therefore demands a three-dimensional planning approach in order to provide a wider operating envelope.

One alternative to Voronoi diagram's are the techniques that incorporate an aircraft's dynamics into the mission planning. A leading example of this is the research underway at the University of Michigan.[25–27] This research has developed trajectory-planning techniques that factor in an aircraft's dynamics and in so doing are capable of adapting the plan based upon changes in the dynamics of the aircraft, for example, damage to control surfaces. Planning techniques such as these are extremely powerful but are also computationally intensive, often being performed offline. As the research is focused upon efficient, online mission planning, mechanisms that incur a lower computational overhead are desired.

A. Cube-Space Mission Planning

Cube-space (C-space) algorithms have been employed frequently within the robotics field to solve the mission-planning problem.[16]

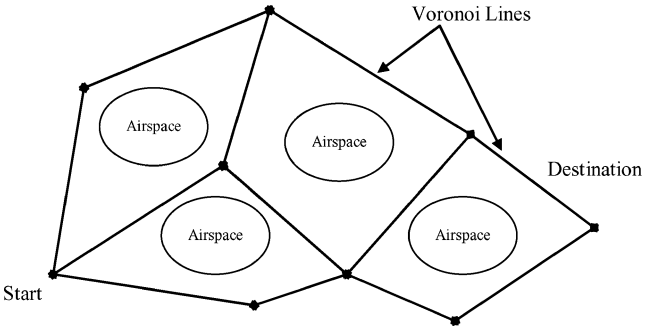

Fig. 4 Voronoi diagram example.

To date C-space algorithms have not been applied for UAV mission planning. The C-space algorithms are based upon dividing the three-dimensional world into cubes. The cubes are marked as either free or occupied. A cube that contains any part of an entity within the world (e.g., a region of airspace) will be marked as occupied. As can be seen in Fig. 5, all cubes (in this case squares because it is a plan view) that overlap with the region of airspace are marked as occupied (shaded).

Additionally each free cube is assigned a value (cost), for example, the distance from the cube to the goal (destination). The path (i.e., mission) is planned by jumping from cube to cube with the direction of the jump based upon the values assigned to the free cubes. This process is shown in Fig. 6.

The critical part of C-space-based mission planning is the algorithm used to map values onto the free cubes. The research utilized a cross section of cost assignment algorithms. This cross section ranged from simple and rapid algorithms through to detailed, but slower, cost assignment algorithms. One commonly used mapping is to propagate a three-dimensional wave from the goal (destination) marking each free cube with the distance to the goal.[13,15] This can be viewed as a sphere expanding from the goal where the edge of the sphere is the wave front being propagated. This is not the

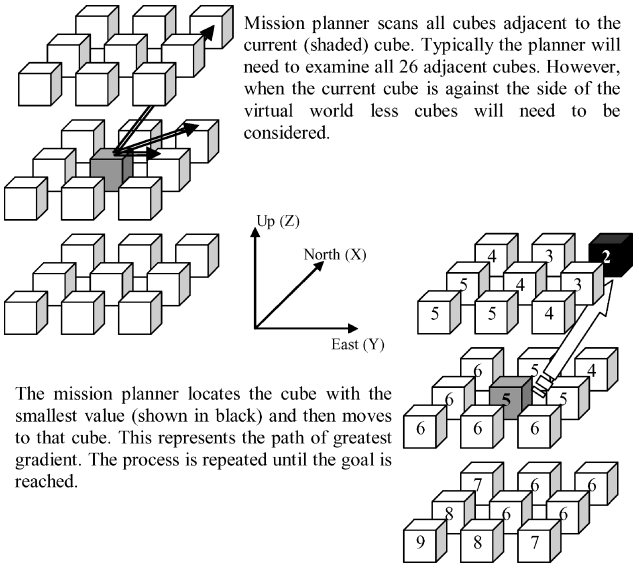

Fig. 6 Example of the scanning process performed by the mission planner.

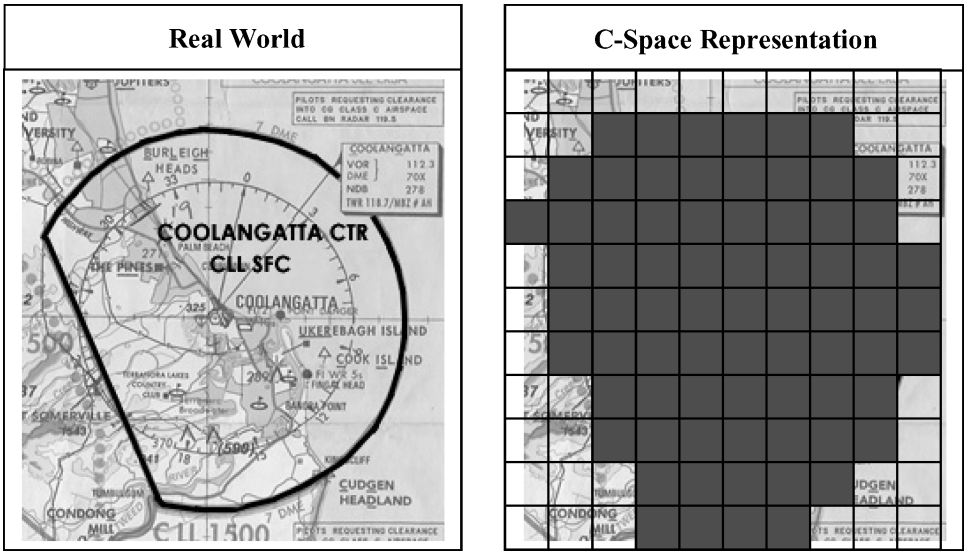

Fig. 5 Plan-view comparison of real world and C-space representations.

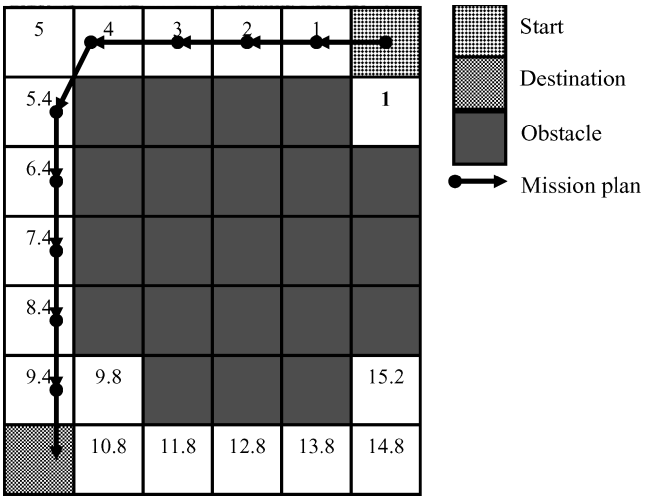

Fig. 7 Path-planning process.

straight-line distance from the cube to the goal. Instead it is the distance that the UAV would need to fly to reach the goal. This requires that the geometric distance wave refract around obstacles so that each cube is accurately marked with the distance required to move from that cube to the goal. The mission planner then starts at the UAV's current location and scans the adjacent cubes. This process is shown in plan view in Fig. 7.

After scanning the adjacent cubes, the mission planner then moves to the cube closest to the goal. This approach is repeated until the goal is reached. The planning process is shown in the pseudocode:

```
Repeat until the goal is reached
Set Best Cube Index to −1
Set Best Cube Value to −1
Loop through all adjacent cubes
   If the cube is occupied (i.e., UAV cannot go there)
Skip to the next cube
   If the cube is free and this is the first cube being tested
      Set the Best Cube Index to the index of this cube
      Set the Best Cube Value to the value of this cube
   If the cube is free and this is not the first cube being tested
      If the value of this cube is less than Best Cube Value
            Update Best Cube Index to the index of this cube
         Update Best Cube Value to the value of this cube
If the Best Cube Index is −1 then no move can be made
   Abort plan generation
Move to the best cube
```

The geometric distance wave mapping has been successfully employed in the past for autonomous ground vehicles.[13,15] However the disadvantage of the geometric distance wave is that it does not factor in the time taken or energy (e.g., fuel) used to traverse the path. For example, there is a significant difference between an aircraft flying a 100-n mile path at a constant altitude and flying a 80-n mile path that requires the aircraft to climb and descend thousands of feet to avoid airspace boundaries. As part of the research, two additional mappings have been developed to enable the generation of mission plans that are more consistent with the practices of a human pilot.

The first mapping estimates the time to fly to the goal. The time wave (basic aircraft model) mapping is applied by first applying the geometric distance wave. Then, based upon the capabilities of the aircraft, the time to fly from each free cube to the goal is estimated. This is calculated by assuming different speeds based upon the direction in which the UAV would fly. The speeds were determined based upon simulated testing, which examined the speed of the aircraft in different configurations (climb, descent, etc.). Figure 8 shows the speeds used to determine the time to fly to each cube within the world.

The result of this mapping is that the decision between climbing, descending, or making no altitude change is weighted appropriately. Very high penalties are assigned to pure vertical manoeuvres in order to prevent the path planner from choosing such a path. Ascending

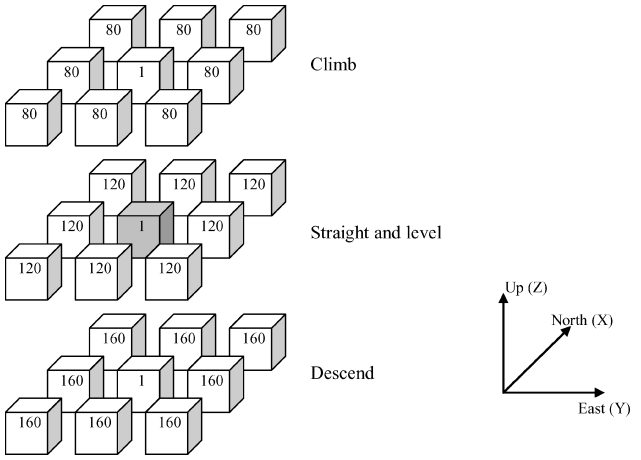

Fig. 8 Speeds (in knots) used for time wave (basic aircraft model) calculation (current cube is shaded).

incurs a penalty in terms of speed, whereas descending incurs an advantage. The consequence of this mapping is that the UAV will fly the path of approximately shortest time. The UAV prioritizes flying straight and level over ascending. Although an advantage is given to descent maneuvers, the UAV will not descend to the lowest level possible and then ascend at the last minute. This is prevented by the penalty for ascending, which makes such a mission plan too costly. Determining the path with the time wave (basic aircraft model) mapping is identical to that for the geometric distance wave mapping. The new mission plan is generated by following the path of greatest gradient.

The final mapping estimates energy required to maneuver the aircraft in terms of the fuel used. Unlike the time wave (basic aircraft model), the fuel wave (detailed aircraft model) mapping does not build on top of the geometric distance wave. The fuel wave (detailed aircraft model) factors in both the distance and the time to move that distance when the wave is propagated. The equation used to determine the fuel usage was based upon the engine model provided by the AeroSim Blockset.[28] First, a basic aircraft model is used to estimate the time the UAV would require to fly from one cube to another. This information is then used by the engine model to estimate the fuel that would be consumed in performing the maneuver. Consequently, the UAV will attempt to fly the most fuel-efficient path. The method for generating the mission plan using the fuel wave (detailed aircraft model) mapping is identical to that for the distance and time wave (basic aircraft model). Any of these mappings could be used to perform onboard mission planning by the IMP. The performance of the different mappings will be compared further on in this paper.

B. Octree-Based Mission Planning

The second type of mission-planning method available is the octree, which is an extension to the C-space methods. An octrees is a hierarchical structure that represents the world at multiple resolutions.[15,17] Figure 9 highlights the difference between C-space and octrees.

The multiple resolutions make crossing large empty spaces more efficient than for a C-space representation. Rather than moving from cube to cube, multiple cubes can be crossed at the same time. This is shown in Fig. 10.

Although this greater efficiency is beneficial, octrees possesses two major disadvantages. Firstly, they consume a larger amount of memory than C-space methods. For example, consider the situation where a cube, with a side length of 16 n miles, is represented at a resolution of 1 n mile. The C-space representation will require $16 \times 16 \times 16 = 4096$ data structures to represent it. An octree representation would require $16 \times 16 \times 16 + 8 \times 8 \times 8 + 4 \times 4 \times 4 + 2 \times 2 \times 2 + 1 \times 1 \times 1 = 4681$ data structures. This is a 14% increase in the storage requirement, and as the dimensions increase (or the resolution decreases) this overhead will grow.

Secondly, the path-planning process is made slower as a result of the greater overhead in traversing the octree structure. This overhead

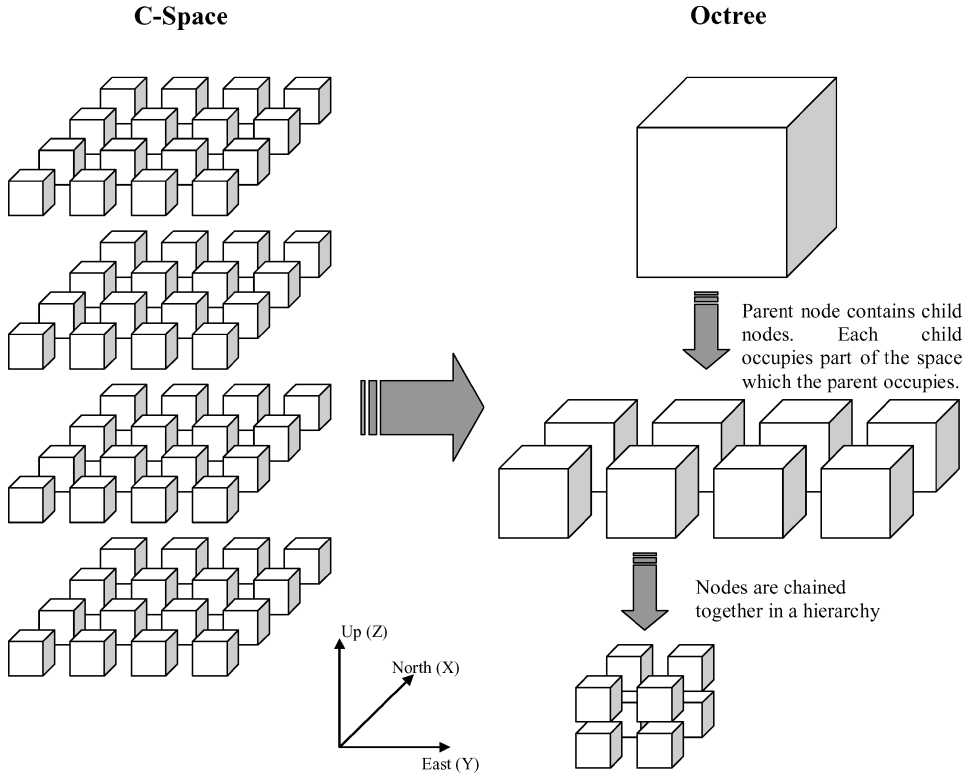

Fig. 9 Difference between C-space and octree representations.

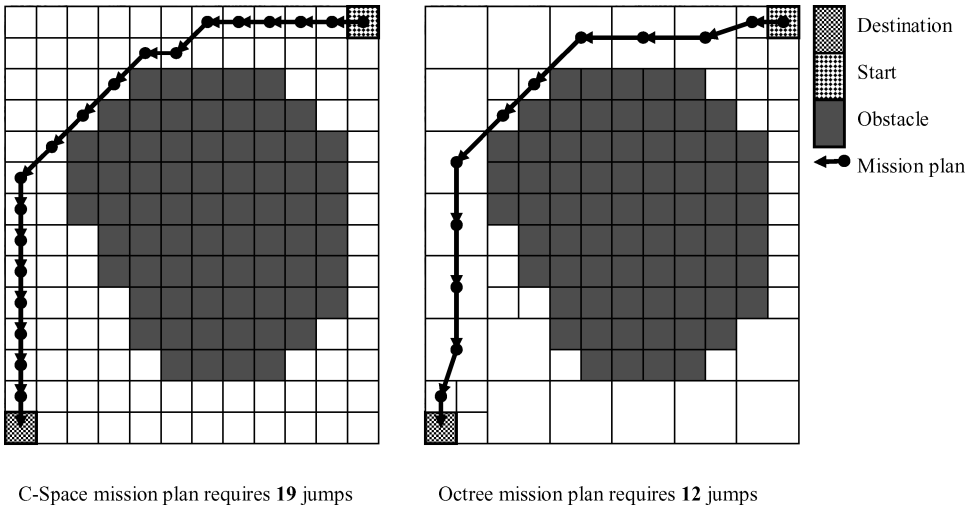

Fig. 10 Efficiency differences between C-space and octree representations.

results from the process of relocating to parent, child, or adjacent nodes. This is slower than for C-space, which can be represented as a multidimensional array.

In the octree method the blocks (called nodes) are marked as either fully free, fully occupied, or partially occupied. If a node is partially occupied, by an entity, then it is possible to change to a finer resolution version of the node and then locate the free nodes within it. The mappings described for C-space can also be applied for octrees in the same manner. The path-planning algorithms are also identical. The only change is that the planner can move to a partially occupied node as well as a fully free node. The octree algorithms can provide more efficient planning; however, the greater overheads frequently outweigh the performance gains, as will be shown in the results.

The C-space and octree algorithms described in this section were implemented into the IMP. Both types of algorithms are capable of meeting the research objectives. The Voronoi diagram methods have not been adopted as they greatly restrict the freedom of the mission planner by forcing flights along predefined paths. The C-space and octree algorithms do not have this limitation and permit operations close to airspace boundaries. As the research is focused upon developing an efficient mission-planning methodology, no penalties have been instituted for operations close to airspace boundaries. However, the resolution of the C-space and octrees was kept at 1 n mile (or higher) providing a minimum buffer of 1 n mile between the UAV and airspace boundaries. This is in keeping with good VFR airmanship. The outlined methods enable the IMP to perform global mission planning onboard the UAV for civilian airspace applications. Results of testing conducted on the planning algorithms will be presented later in the paper.

V. Simulation Environment

Testing the algorithms presented in this paper required the usage of the QUT developed simulation environment known as the aircraft simulation and testing environment (ASATE).[29,30] ASATE provides a real-time simulation environment for an aircraft. In

this case a Cessna 172 was chosen as it provides a proven stable platform.

The flight and sensor model components of ASATE are implemented in MATLAB® Simulink using the AeroSim blockset.[31] The AeroSim blockset provides a full nonlinear, six-degrees-of-freedom simulation of an aircraft. The AeroSim blockset was chosen for its ability to be used in a Simulink model, its support for standard aircraft formats (FlightGear model format), and its low cost (free for academic use). The flight and sensor models are executed in real time on an Intel Pentium II 233-MHz processor with 64 MB RAM. The real-time execution is achieved through the xPC Target component of MATLAB Simulink. xPC Target compiles a Simulink model and executes it in real time on a second (target) computer.

The flight and sensor models (target computer) are connected via RS232 to a computer (host) running Microsoft Windows, which is used to control (e.g., change wind conditions) and to monitor the simulation. The Microsoft Windows computer publishes the flight information (location and orientation) via the Worldwide Web for remote monitoring of simulations.

The IMP is executed on a Pentium II 233-MHz processor with 64 MB RAM under the QNX real-time operating system (RTOS). QNX is a hard real-time operating system chosen for its high reliability. A RS232 link connects the flight hardware running the IMP to the target computer running the flight and sensor models. This link is used to pass flight data to the IMP and to pass control surface deflections to the flight and sensor models. The architecture of ASATE is shown in Fig. 11.

The ASATE system can simulate a range of aircraft and has built-in support (via the AeroSim blockset) for any aircraft models designed for the open-source flight simulation system, FlightGear. ASATE can be connected to either FlightGear or X-Plane for a three-dimensional visual display of the aircraft's current position and attitude. ASATE possesses the capability to simulate the presence of other aircraft as well as various weather conditions (storm fronts, etc.).

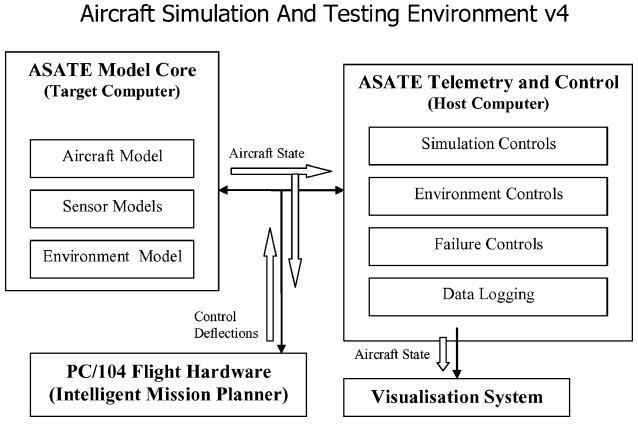

Fig. 11 ASATE system architecture.

The IMP logs all flight (position, attitude, etc.) and algorithm (completion times) parameters in order to assess both the performance of the simulated aircraft and the mission-planning algorithms. The testing regimes conducted are examined in greater detail in the proceeding section.

VI. Testing Regimes

Several testing regimes were designed in order to evaluate the performance of the algorithms described within this paper. The aim of the testing regimes was to assess the ability of the outlined algorithms to meet the research objectives. The testing regimes required the IMP to plan and then fly a given mission. All testing regimes were performed under the QNX RTOS using the ASATE simulation environment. The mission planning itself was performed under Microsoft Windows 2000 Professional on a Pentium 4 1.6-GHz processor computer with 256 MB of RAM. The testing was broken down into four testing regimes as shown in Table 1.

All testing was conducted using real-world airspace and terrain information from southeast Queensland, Australia, centered around Brisbane Airport (YBBN). The mission plan required the aircraft to fly through a complex region of airspace over a total distance of 370 n miles. A picture of the testing scenario, from the software developed for the research, is shown in Fig. 12.

Table 1 List of testing regimes

Testing regime	Aspects tested
1, Plan generation time	Examined the time to generate a mission plan for (horizontal) resolutions of 5, 2, and 1 n miles
2, Generated path length	Analyzed how much longer the generated paths are than the original straight-line path (i.e., measures the efficiency) for resolutions of 5, 2, and 1 n miles
3, Planning algorithm performance	Examined the performance of the individual stages of the mission-planning process for resolutions of 5, 2, and 1 n miles
4, In-flight performance	Compares the results (flight time, distance flown, and fuel consumed) from simulated flights conducted at resolutions of 5, 2, and 1 n miles

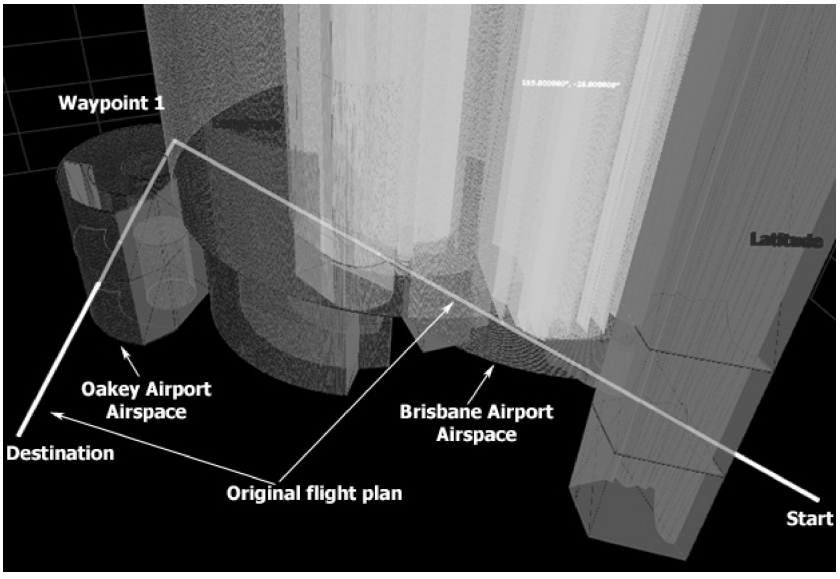

Fig. 12 Original invalid mission plan.

A. Test Results

The first three testing regimes evaluated the planning performance for the individual algorithms. The fourth testing regime was used for confirmation of this evaluation based upon simulated flights. The test results have been broken down into the following four testing regimes: testing regime 1—plan generation time; testing regime 2—generated path length; testing regime 3—planning algorithm performance; and testing regime 4—in-flight performance.

1. Planning Times

Testing regime 1 examined the planning times for each combination of mission planning (C-space or octree) and wave mapping (distance, time, or fuel) algorithms. Table 2 shows the times (in seconds) taken to generate a mission plan for the scenario shown in Table 2.

A graphical representation of the results is provided in Fig. 13.

A number of conclusions can immediately be drawn from these results. As expected, the resolution has a significant impact upon the planning time. Reducing the horizontal resolution from 2 to 1 n mile increases the number of cubes by a factor of four (from horizontal resolution of 2×2 n miles to 1×1 n miles, i.e., factor of four). This results in a corresponding increase in planning time by a similar factor.

As shown in Fig. 14, the octree algorithms are consistently slower than their C-space counterparts. This is expected as the C-space must be constructed prior to the octree representation being constructed. The added overhead from the construction, and navigation (i.e., locating and moving to a new octree node), of the octree representation overshadows any performance gains achieved by skipping large sections of free space. The relative speeds of the different algorithms can also be clearly seen. The fuel wave (detailed aircraft model) algorithms are consistently the slowest of the algorithms; this is because of their greater complexity. The greater complexity of the fuel wave (detailed aircraft model) algorithm results from its usage of both a flight and engine model, which require multiple calculations to be performed to propagate the fuel wave (detailed aircraft model).

The geometric distance wave algorithms are consistently the fastest of the algorithms. This is expected as they are computationally the simplest of the algorithms. The geometric distance wave algorithms only calculate the distance between cubes and do not use any dynamic models. The time wave (basic aircraft model) algorithms add a small extra step on top of the geometric distance wave algorithms. Consequently, the planning times of the geometric distance and time wave (basic aircraft model) algorithms are similar. Based solely upon planning times, the geometric distance wave–C-space algorithm provides the best performance. However, there are other metrics to consider in identifying the optimal algorithm for mission planning in civilian airspace.

2. Generated Path Length

The second testing regime examined the difference in path length between the original plan (shown in Fig. 12) and the generated mission plan. This provides a measure of the efficiency of the planning algorithms. Table 3 shows the increase in path length, relative to the original path, for each of the planning algorithms over the entire

Table 2 Plan generation times (calculated on 1.6-GHz Pentium 4)

Planning algorithm	Horizontal resolution		
	5 n miles	2 n miles	1 n mile
Geometric distance wave–C-space	49.16 s	233.24 s	1013.72 s
Geometric distance wave–octree	49.72 s	235.38 s	1101.16 s
Time wave (basic aircraft model)–C-space	49.17 s	234.14 s	1015.17 s
Time wave (basic aircraft model)–octree	49.80 s	236.19 s	1089.56 s
Fuel wave (detailed aircraft model)–C-space	51.00 s	243.53 s	1067.37 s
Fuel wave (detailed aircraft model)–octree	51.58 s	243.74 s	1112.39 s

Table 3 Percentage increase in path length

Planning algorithm	Horizontal resolution		
	5 n miles	2 n miles	1 n mile
Geometric distance wave–C-space	17.8%	16.5%	4.0%
Geometric distance wave–octree	19.8%	17.0%	4.2%
Time wave (basic aircraft model)–C-space	17.8%	16.5%	4.0%
Time wave (basic aircraft model)–octree	19.8%	17.0%	4.2%
Fuel wave (detailed aircraft model)–C-space	20.9%	13.9%	4.3%
Fuel wave (detailed aircraft model)–octree	21.6%	19.3%	4.7%

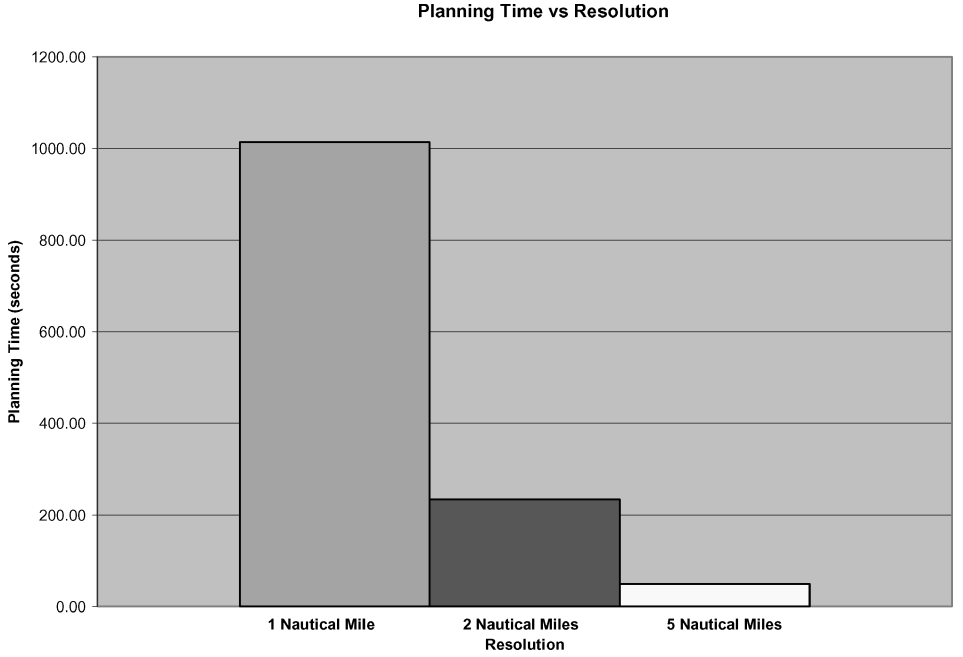

Fig. 13 Graph of planning time against resolution for the C-space (geometric distance wave) algorithm (calculated on 1.6-GHz Pentium 4).

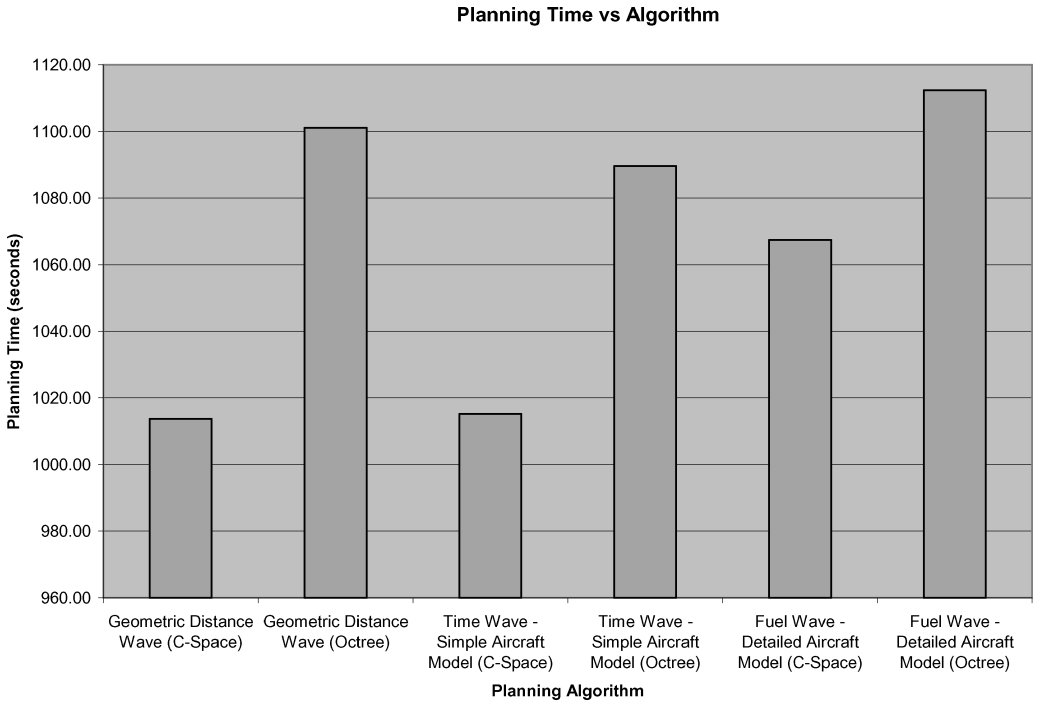

Fig. 14 Graph of planning time against planning algorithm for a resolution of 1 n mile (calculated on 1.6-GHz Pentium 4).

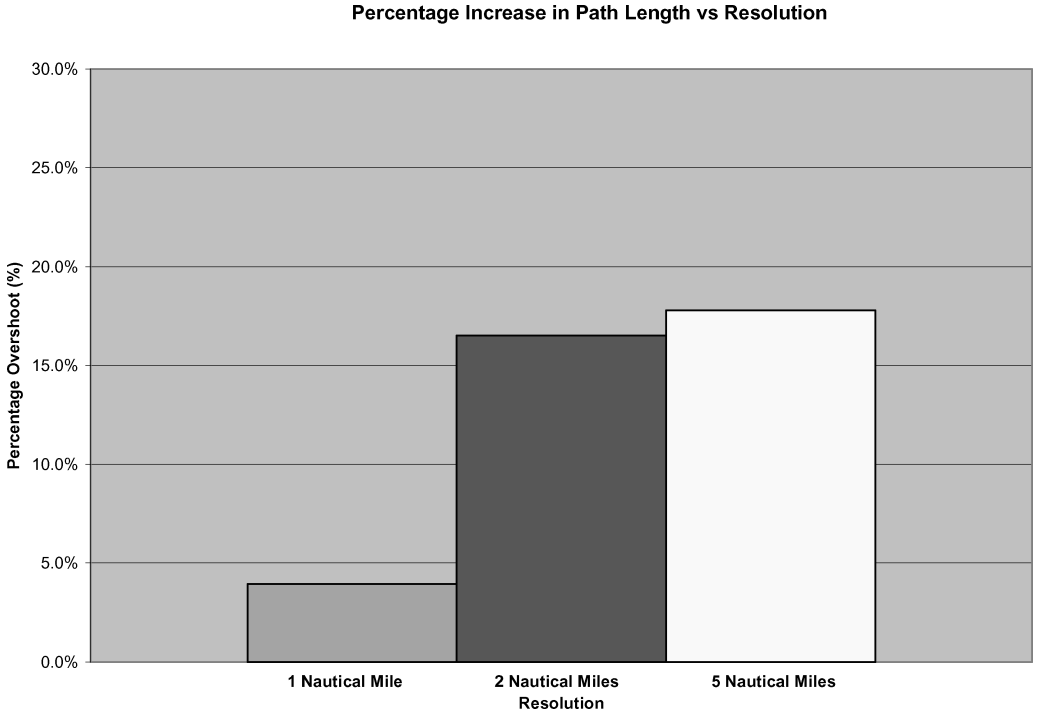

Fig. 15 Graph of increase in path length against resolution for C-space (geometric distance wave) algorithm.

mission plan. The percentage increase measures how much longer the generated plan is than the original plan.

A graphical representation of the results is shown in Figs. 15 and 16.

There is a substantial variation in the increase in path length with the resolution. Finer resolutions (smaller cubes/nodes) result in better efficiencies (lower increase in path length). This is expected as at finer resolution the path-planning algorithms can locate smaller gaps between the entities in the world. Figure 16 provides a comparison of the path length increase between the different planning algorithms for the same resolution.

These results show that planning performed at a resolution of 1 n mile consistently achieves path length increases of less than 5%. The most efficient algorithms are the distance and time wave (basic aircraft model) algorithms using a 1-n mile resolution. These algorithms yield an increase of 4%, which corresponds to a distance of approximately 15 n miles. Horizontal resolutions finer than 1 n mile can produce plans with higher efficiencies; however, these were not able to be tested because of computing limitations.

The C-space and octree algorithms generate plans that are different because of the different ways in which they represent the

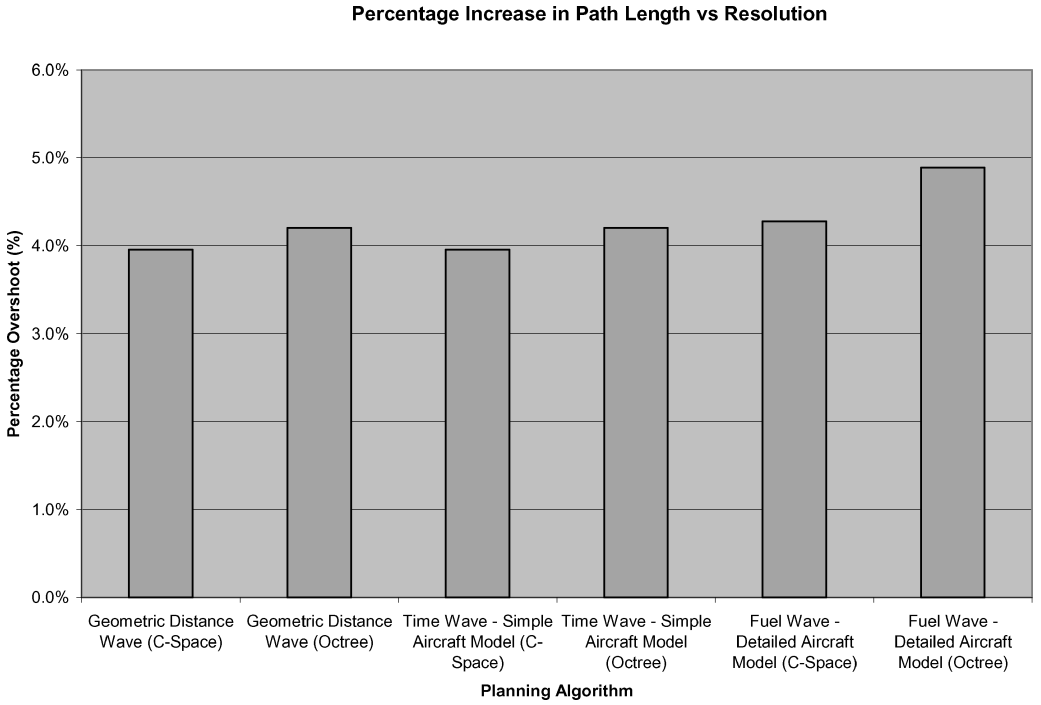

Fig. 16 Graph of path length increase against planning algorithm for a resolution of 1 n mile.

world. This difference becomes evident in the percentage increase in path length measurements. The C-space algorithms consistently find more efficient paths than their octree counterparts. This is a result of the octree algorithms operating at as coarse resolution as possible. Consequently the octree algorithms can fail to see more efficient paths and keep further away from entities in the world, compared to their C-space counterparts.

Next, the performance of the individual algorithms will be examined. The fuel wave (detailed aircraft model) algorithms (specifically the octree algorithms) consistently produce the greatest increase, with up to 21.6% increase. Similarly, the distance and time wave (basic aircraft model) algorithms (specifically the C-space algorithms) produce the least increase except for a resolution of 2 n miles. For this case the fuel wave (detailed aircraft model) (C-space) algorithm provides a 2.6% improvement over the distance and time wave (basic aircraft model) algorithms. This anomaly is a result of performance varying with the resolution and because of differences between the wave mappings.

The wave equations for the distance and time wave (basic aircraft model) expansions are very similar and impose very strict constraints upon the aircraft's movement. The time wave (basic aircraft model) expansion explicitly penalizes the aircraft for any movement that would require the aircraft to climb. These constraints were removed for the fuel wave (detailed aircraft model) expansion as the penalties were implicitly present in the equation used to determine fuel usage. Consequently the fuel wave (detailed aircraft model) expansion results in the aircraft taking paths that are closer to the limits of the aircraft, resulting in more severe maneuvers. In some circumstances this enables the fuel wave (detailed aircraft model) algorithms to locate a more efficient path in terms of distance. It is important to remember however that the fuel-wave (detailed aircraft model) algorithm is not looking for the shortest path in terms of distance; instead, it searches for the path that uses the minimum amount of fuel.

The risk in operating closer to the aircraft limits is that there is a higher likelihood that the mission plan will not be achievable by the aircraft. However, as results in the proceeding sections will show, the fuel-wave (detailed aircraft model)-based algorithms did not take the aircraft beyond its limits. When the resolution is made, finer, all of the algorithms are able to find the more efficient paths.

Based upon increase in path length, the geometric distance wave–C-space algorithm operating at 1 n mile is the optimal algorithm for the mission-planning problem.

3. Mission Planning Performance

The third testing regime assessed the performance of the individual components of the mission-planning algorithms. There are five key components of the algorithms to be examined (four in the case of C-space algorithms). These components are shown in Fig. 17.

The times for the individual stages of the mission-planning algorithms are given in Table 4. All of the times are in seconds. Times could not be determined for the path-planning stage (i.e., construction of the new plan after the cost waves have been propagated) as this stage completed in under 1 ms. Times under 1 ms could not be measured by the computer.

These results present a great deal of information, which will be analyzed progressively. The collision detection time remains consistently around 10 s irrespective of the resolution or planning method. This is to be expected because of the structure of the mission-planning algorithms. The collision detection routines operate on the triangular-mesh representations of the entities rather than the C-space or octree representations. Consequently the collision detection time will complete in the same time irrespective of resolution.

The results clearly identify that the slowest step in the mission-planning process is the construction of the C-space. This process involves determining which cubes in the world are occupied by entities (terrain, airspace, etc.) and which cubes are free. This process consistently occupies more than half of the mission planning time (average of 64%).

The next slowest step is the propagation of the distance, time, or fuel waves (detailed aircraft model). The only exception to this is for a resolution of 5 n miles. For the 5-n miles case the collision detection time is greater. This is simply a result of the resolution being coarse enough that the C-space construction and wave propagation processes take minimal time. For the fine resolution mission plans the wave propagation takes a significant proportion of time, up to almost 40% for the 1-n mile resolution. Future optimizations will also focus strongly on this area.

The final element of timing information presented in the data relates to the conversion from the C-space to the octree representation. As expected, the time taken to perform this process varies with the

Table 4 Planning algorithm component times (calculated on 1.6-GHz Pentium 4)

Planning algorithm	Horizontal resolution 5 n miles	2 n miles	1 n miles	Stage
Geometric distance wave–C-space	10.14 s	9.86 s	9.75 s	Collision detection
	31.06 s	170.25 s	652.05 s	C-space construction
	7.84 s	52.45 s	349.18 s	Wave propagation
	$\ll$ 1 ms	$\ll$ 1 ms	$\ll$ 1 ms	Path planning
	49.16 s	**233.24** s	**1013.72** s	**Total time**
Geometric distance wave–octree	10.16 s	9.91 s	9.83 s	Collision detection
	31.62 s	169.41 s	655.65 s	C-space construction
	5.33 s	54.36 s	369.37 s	Wave propagation
	0.17 s	0.85 s	52.63 s	Octree conversion
	$\ll$ 1 ms	$\ll$ 1 ms	$\ll$ 1 ms	Path planning
	49.72 s	**235.38** s	**1101.16** s	**Total time**
Time wave (basic aircraft model)–C-space	10.16 s	9.95 s	9.80 s	Collision detection
	31.00 s	170.56 s	650.93 s	C-space construction
	7.89 s	52.94 s	351.79 s	Wave propagation
	$\ll$ 1 ms	$\ll$ 1 ms	$\ll$ 1 ms	Path Planning
	49.17 s	**234.14** s	**1015.17** s	**Total time**
Time wave (basic aircraft model)–octree	10.14 s	9.91 s	9.80 s	Collision detection
	30.94 s	169.42 s	655.87 s	C-space construction
	8.39 s	55.14 s	367.46 s	Wave propagation
	0.16 s	0.84 s	46.28 s	Octree conversion
	$\ll$ 1 ms	$\ll$ 1 ms	$\ll$ 1 ms	Path planning
	49.98 s	**236.19** s	**1089.56** s	**Total time**
Fuel wave (detailed aircraft model)–C-space	10.13 s	9.89 s	9.92 s	Collision detection
	31.09 s	170.45 s	653.17 s	C-space construction
	9.66 s	62.61 s	401.63 s	Wave propagation
	$\ll$ 1 ms	$\ll$ 1 ms	$\ll$ 1 ms	Path planning
	51.00 s	**243.53** s	**1067.37** s	**Total time**
Fuel wave (detailed aircraft model)–octree	10.13 s	9.83 s	9.89 s	Collision detection
	31.55 s	172.41 s	665.31 s	C-space construction
	9.59 s	63.50 s	398.86 s	Wave propagation
	0.14 s	0.84 s	28.06 s	Octree conversion
	$\ll$ 1 ms	$\ll$ 1 ms	$\ll$ 1 ms	Path planning
	51.58 s	**243.74** s	**1112.39** s	**Total time**

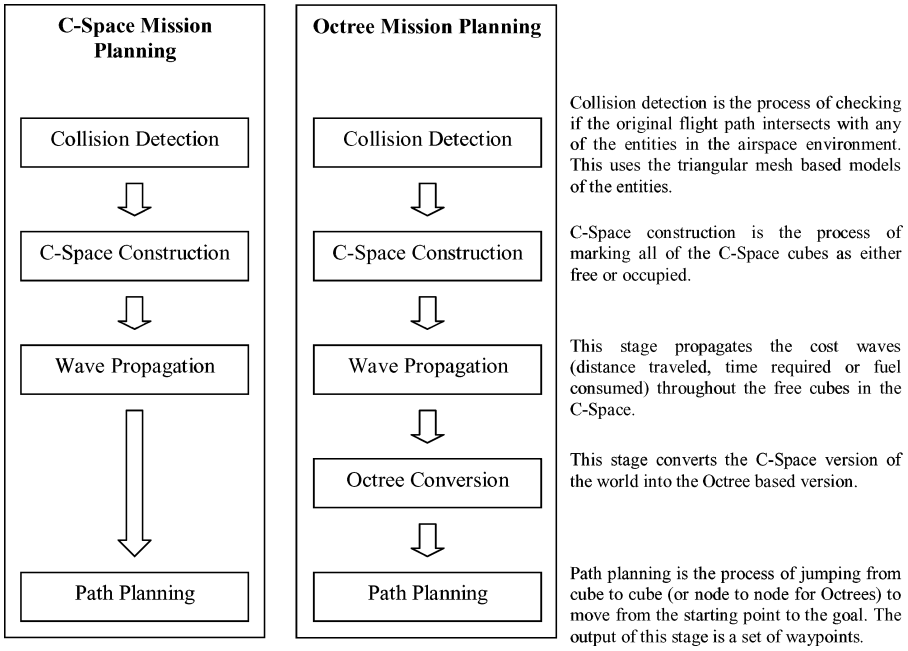

Fig. 17 Mission-planning stages.

resolution of the data space. Finer resolutions result in a larger C-space, which correspondingly takes longer to convert to an octree representation.

The third testing regime has identified the two main areas upon which optimizations should focus: C-space construction and wave propagation. These two components of the mission planning consistently consume the majority of the planning time and as such are ideal candidates for optimization.

4. In-Flight Performance

The final testing regime analyzed the flight time and distance to fly the generated mission plan. This provides verification of the previous efficiency results and an assessment of the quality of the generated plan. If large deviations are observed between the generated plan and flown plan, then it indicates that the mission planner did not produce a plan within the aircraft's capabilities. The flight time provides a measure of the efficiency of the generated plan. The results were

Table 5 Simulated flight results

Planning algorithm	Horizontal resolution			Metric
	5 n miles	2 n miles	1 n mile	
Geometric distance wave–C-space	3:40:09	3:36:10	3:14:32	Flight time
	24.2%	22.6%	9.9%	Distance increase
	208.40 lb	205.38 lb	183.95 lb	Fuel consumed
Geometric distance wave–octree	3:41:13	3:39:6	3:16:19	Flight time
	25.9%	22.4%	9.6%	Distance increase
	210.67 lb	207.04 lb	185.45 lb	Fuel consumed
Time wave (basic aircraft model)–C-space	3:40:06	3:36:06	3:14:15	Flight time
	23.3%	22.2%	9.3%	Distance increase
	208.36 lb	205.25 lb	183.80 lb	Fuel consumed
Time wave (basic aircraft model)–octree	3:41:12	3:39:39	3:15:16	Flight time
	24.6%	22.6%	9.5%	Distance increase
	210.65 lb	207.37 lb	184.84 lb	Fuel consumed
Fuel wave (detailed aircraft model)–C-space	3:46:53	3:38:46	3:10:38	Flight time
	26.5%	18.8%	8.9%	Distance increase
	214.07 lb	203.73 lb	182.65 lb	Fuel consumed
Fuel wave (detailed aircraft model)–octree	3:48:15	3:45:37	3:12:40	Flight time
	27.3%	23.6%	10.4%	Distance increase
	215.35 lb	211.75 lb	184.44 lb	Fuel consumed

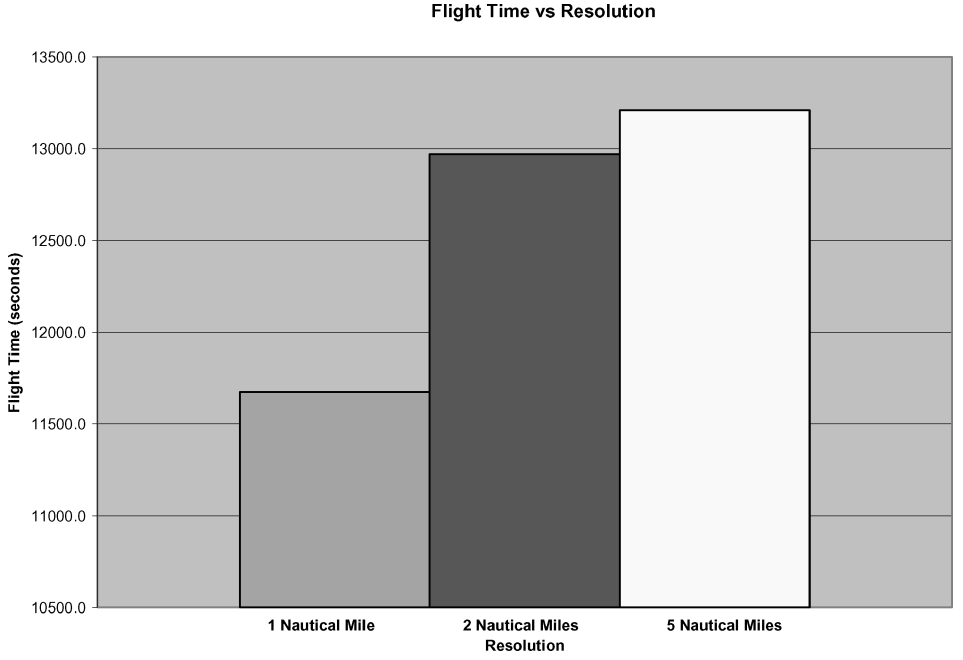

Fig. 18 Graph of flight time against resolution for C-space (geometric distance wave) algorithm.

obtained using the real-time simulation environment ASATE. Both the flight time (in hours, minutes, and seconds), the flown distance (in percentage increase from the original path), and fuel usage are shown in Table 5.

These results demonstrate that the general trends of finer resolutions and C-space algorithms providing better performance have held true as shown in Figs. 18 and 19.

The key differences between the algorithms become clear upon an examination of the flight time and fuel usage. When these factors are examined, the geometric distance wave does not yield the best performance. In fact, it provides the worst performance. This is because the geometric distance wave looks only for the shortest path; it does not consider how long it might take to fly the paths. Consequently, although the paths found using the geometric distance wave are shorter because they have not factored in aircraft performance they take longer to fly.

As shown in Figs. 20 and 21, the fuel wave (detailed aircraft model) provides the best performance in terms of both flight time and fuel usage for this test case. The time wave (basic aircraft model) provides improved performance over the geometric distance wave; however, it does not perform as well as the fuel wave (detailed aircraft model). The better performance of the fuel wave (detailed aircraft model) algorithm occurs even in cases where the path flown is longer than that of the other algorithms. However, although the path flown is longer it is more efficient providing benefits to time and fuel efficiency. These results show that the fuel wave (detailed aircraft model) mapping in C-space is the most efficient of the algorithms in terms of in flight performance. This contradicts the previous findings where the fuel wave (detailed aircraft model) was typically the least efficient algorithm. The reason for this contradiction is that the fuel wave (detailed aircraft model) algorithm has been designed to generate paths, which, though they can be longer than for the distance or time wave (basic aircraft model), are more fuel efficient.

The results also show that although the time wave (basic aircraft model) yields faster flight times than the geometric distance wave it is not faster than the fuel wave (detailed aircraft model). This is a result of the time wave (basic aircraft model) using a very

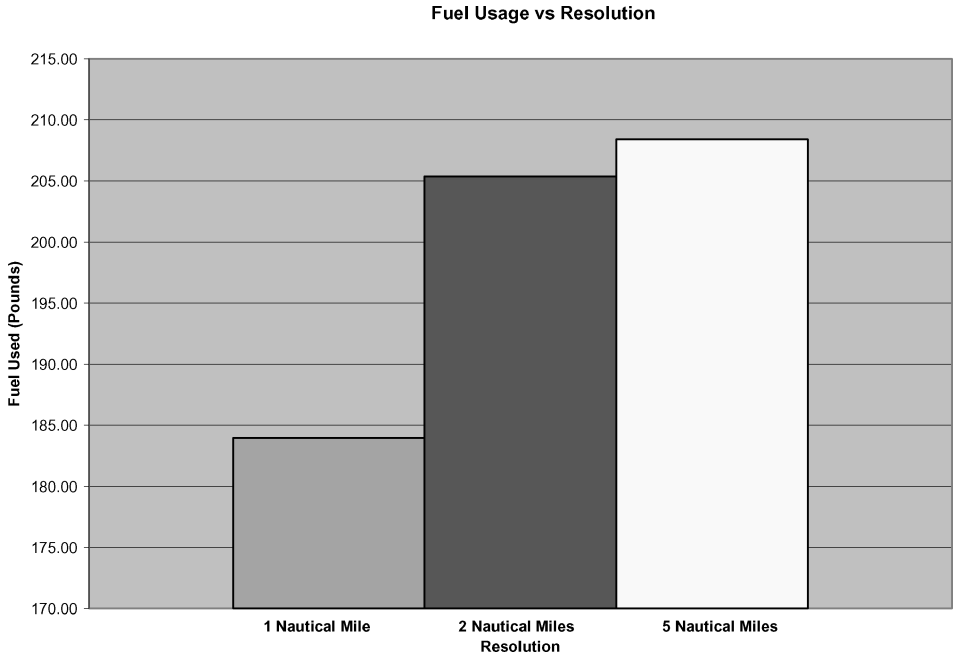

Fig. 19 Graph of fuel usage against resolution for C-space (geometric distance wave) algorithm.

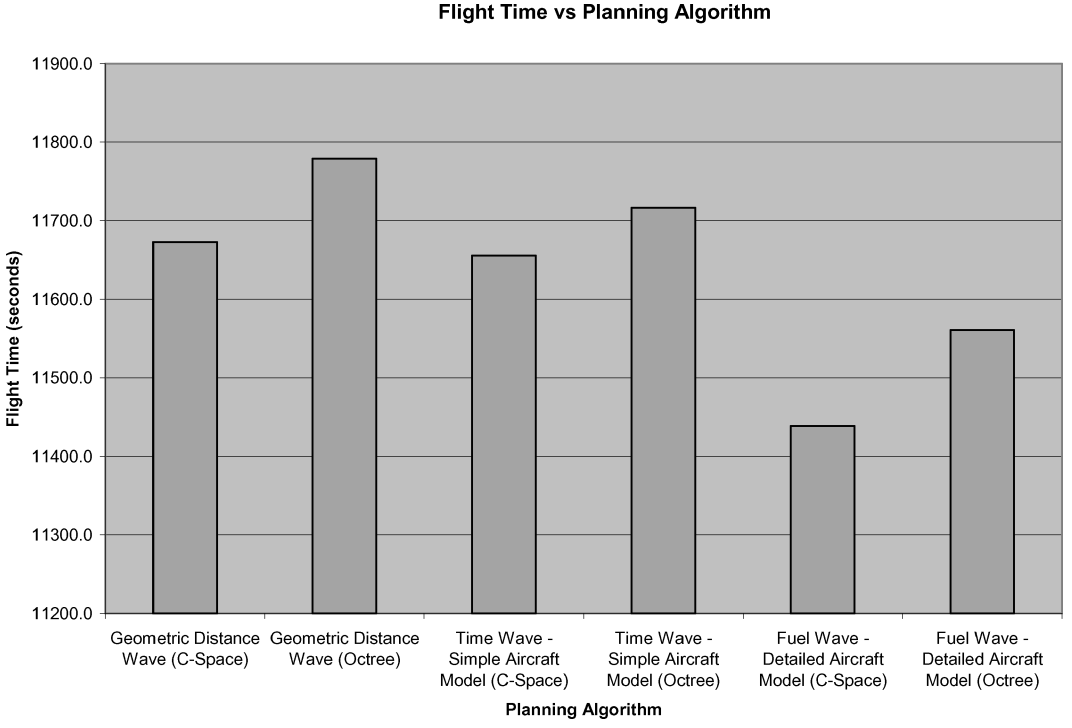

Fig. 20 Graph of flight time against planning algorithm for a resolution of 1 n mile.

coarse model of the speed of the aircraft. As the fuel wave (detailed aircraft model) algorithm uses a more accurate, but slower, model of the aircraft's dynamics, it is able to achieve better in-flight performance.

Visual inspection of the path flown by the aircraft indicates that the aircraft is readily able to fly the generated paths. There were no cases where the aircraft needed to loop around to reach a waypoint. This indicates the generated path did not exceed the limits of the aircraft. This does not guarantee that the mission-planning algorithms will work for every possibly situation. However, at this stage the research is not aimed at developing an aviation-certified mission-planning system. The general methodologies of aviation certification standards such as DO-178B have been considering during the design of the mission-planning system. For example, once a plan has been generated the plan is verified using an independent set of algorithms to ensure it does not pass through any known entities.

Future research will be conducted to examine other factors such as the turn/climb rates required by the aircraft to gain a more detailed estimate of how achievable the mission plans are. The behavior of the aircraft can be seen in Figs. 22 and 23, which show the original invalid desired mission plan, the planned path, and the flown path. Note that in these figures the class C and D airspace have been removed for clarity. The figures shown were generated using the

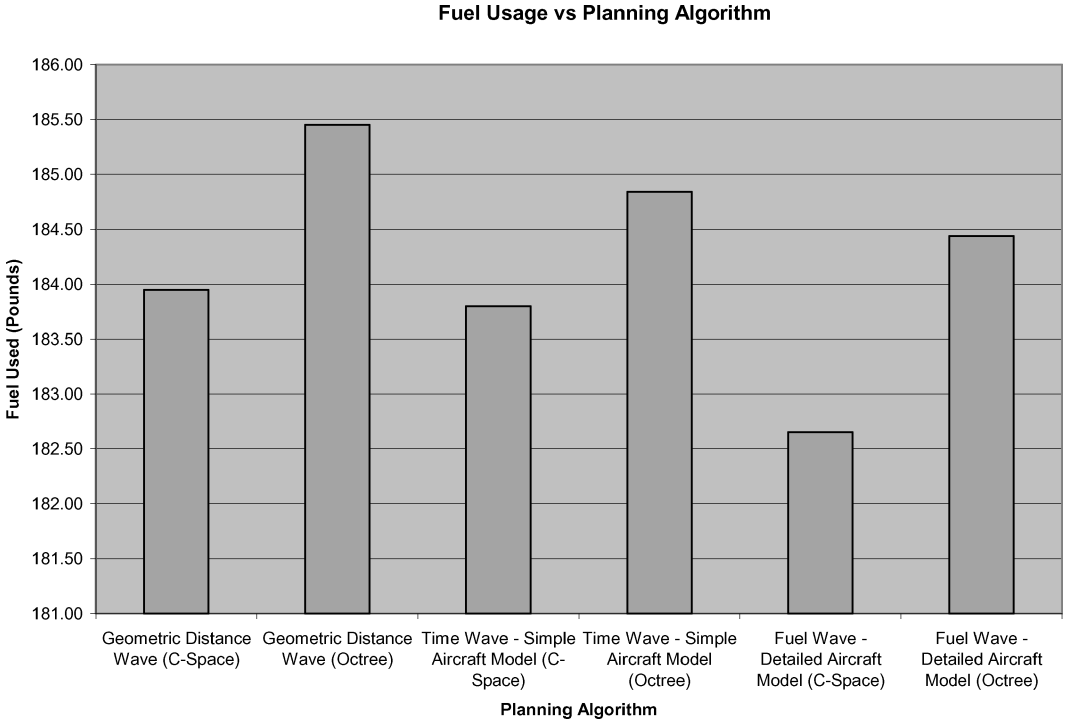

Fig. 21 Graph of fuel usage against planning algorithm for a resolution of 1 n mile.

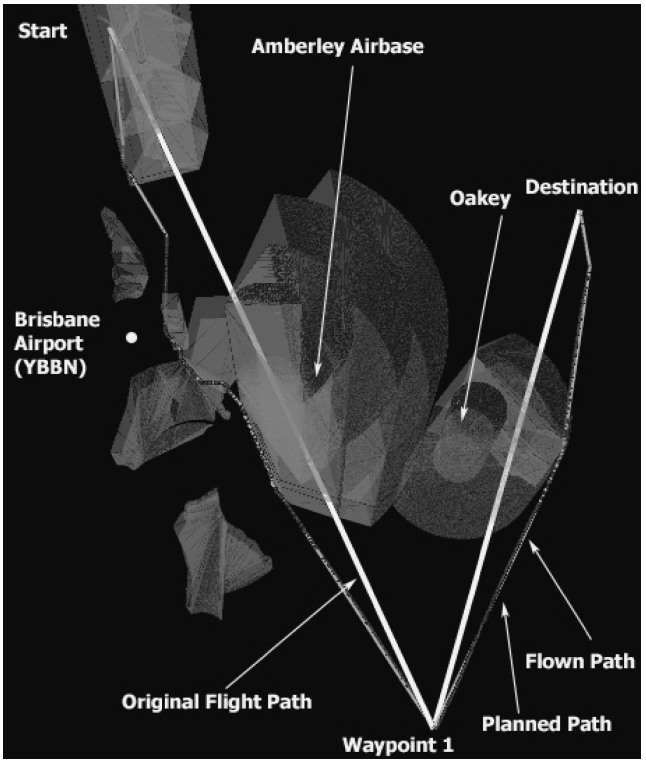

Fig. 22 View of mission planning results.

geometric distance wave (C-space) planning algorithm operating at a resolution of 1 n mile.

This figure shows that the aircraft has found a small gap above a region of controlled airspace. The path found by the mission planner requires minimal altitude changes except where required. This is in line with the behavior of a human pilot operating under the same conditions. The figure shows that the aircraft performs the planned ascents and descents faster than the planned path requires. This results from the design of the aircraft flight controllers. The controllers work to capture the next required altitude as soon as possible. Consequently the aircraft is capturing the required altitude sooner than intended. This is most obvious where large altitude change occurs. This has the potential to cause the aircraft to enter into controlled airspace or collide with another obstacle. This area will be addressed in the near future by modifying the flight controllers to fly precisely along the path laid out in the mission plan rather than attempting to capture the desired altitude immediately.

Throughout the flights the aircraft did not collide with any obstacles within the world as can be seen from the figure. This was verified based upon the flight data collected throughout the simulated flight. This fact combined with the proven ability of the aircraft to fly all of the given plans demonstrates that any of the planning algorithms can meet the research objectives.

VII. Analysis

Based upon the testing performed, any of the algorithms presented could be used for on board mission planning within civilian airspace. However, in terms of performance the testing has demonstrated that the C-space algorithms provide higher performance than their octree counterparts. octrees should be capable of providing better performance as they can skip large sections of free space in a single jump rather than the multiple jumps required by C-space algorithms. However, the extra overhead from the construction of and navigation within the octrees outweighs the advantages gained when navigating large areas of free space.

Finer resolutions also provide better performance, at the cost of longer planning times. The fuel wave (detailed aircraft model) algorithm provides the best in-flight performance, whereas the geometric distance wave algorithm takes less time to compute. The overall best performance is provided by the C-space [fuel wave (Detailed aircraft model)] algorithm. This algorithm provides the shortest flight times and the least fuel usage for the given testing scenario.

These results have identified the areas where additional performance gains can be obtained. The two slowest operations are the construction of the C-space and the propagation of the distance, time, or fuel wave (detailed aircraft model). The C-space construction is the slowest operation, and substantial performance gains can be achievable through precomputation of the C-space.

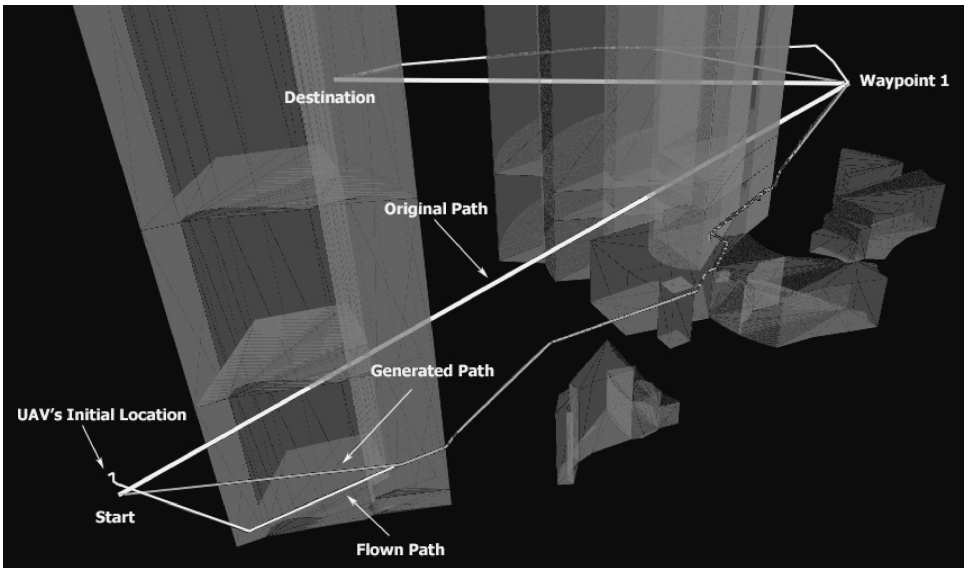

Fig. 23 Side view of mission-planning results.

Table 6 C-space library sizes

Resolution, n miles	Library size, MB
1	87.8
2	23.2
5	4.4

The C-space representations for airspace, terrain, and buildings will not change over the course of a mission. This makes it possible to precompute and store the C-space representations for each entity. The precomputed representations can be imported into the digital world when needed rather than being generated on board.

To test this theory, a library of precomputed C-space was constructed at varying resolutions. The mission-planning algorithms were modified so that rather than constructing the C-space online the C-space representations of the entities were retrieved from the library. Table 6 shows the sizes of the C-space libraries for a range of different resolutions.

The precomputed C-space libraries are sufficiently small enough to be readily stored onboard a UAV. The availability of affordable and high-capacity solid-state storage devices (e.g., compact flash cards) makes the storage of these precomputed libraries onboard a UAV readily achievable.

In addition to modifying the C-space construction, changes were also made to the propagation of the different cost waves. In the preceding results the cubes/nodes were marked with their true distance from the goal [for the distance and time wave (basic aircraft model)s]. Propagating the geometric distance wave therefore requires that the following equation shown be used numerous times:

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{1}$$

Logically, if this equation is being called a multiple times any optimizations made to the equation will speed the entire path-planning process. It is in fact not necessary for the true distance to be propagated; instead, the square of the distance can be used. The square of the distance still provides an indication of how far a cube is from the goal. However, it removes the need to perform a square-root operation for every calculation, which has the potential to improve the mission-planning performance.

These optimizations were implemented into the intelligent mission planner and were subjected to the same testing scenario as the preceding results. As the C-space class of algorithms yielded the best performance previously (in terms of planning time and efficiency), the tests focused solely upon the C-space algorithms. This decision was driven by the intent to identify the best performing algorithm. The performance of the optimized mission-planning algorithms (relative to the preceding algorithms) are shown in Tables 7 and 8.

Table 7 Nonoptimized planning times (calculated on 1.6-GHz Pentium 4)

	Horizontal resolution		
Planning algorithm	5 n miles	2 n miles	1 n mile
Geometric distance wave–C-space	49.16 s	233.24 s	1013.72 s
Time wave (basic aircraft model)–C-space	49.17 s	234.14 s	1015.17 s
Fuel wave (detailed aircraft model)–C-space	51.00 s	243.53 s	1067.37 s

Table 8 Optimized planning times (calculated on 1.6-GHz Pentium 4)

	Horizontal resolution		
Planning algorithm	5 n miles	2 n miles	1 n mile
Geometric distance wave–C-space	8.58 s	13.25 s	144.08 s
Time wave (basic aircraft model)–C-space	8.63 s	13.67 s	146.89 s
Fuel wave (detailed aircraft model)–C-space	17.80 s	69.50 s	378.32 s

As can be seen from these results, the implemented optimizations provide a substantial improvement in the performance. Performance improvements between 60 to 80% were achieved for the different combinations of algorithms and resolutions. The addition of using the precomputed C-space information and modification of the distance and time-wave (basic aircraft model) propagation has clearly had a dramatic impact upon the performance. Distance and time-wave (basic aircraft model)-based mission plans can be performed at a 1-n mile resolution in under 2.5 min, compared to over 16 min previously. Fuel wave plans can be produced in under 6.5 min at 1-n mile resolution, compared to the previous time of over 17 min.

Based upon these results, the optimal solution for mission planning in civilian airspace is the fuel wave (detailed aircraft model)

algorithm, in C-space at a resolution of 1 n mile. This provides the best in-flight performance of the different algorithms considered, with an acceptable planning time.

VIII. Conclusions

New trends are emerging in the field of uninhabited airborne vehicles. These trends are based around increasing the onboard intelligence of UAVs in order to more fully exploit their capabilities. This paper has focused upon increasing the onboard intelligence in one specific area: mission planning. A novel multidisciplinary algorithm was presented designed to satisfy the following objectives: 1) to perform three-dimensional onboard mission planning for civilian airspace operations, where the UAV will be granted the maximum freedom of movement within the environment; and 2) to plan efficient paths in terms of distance traveled, time required, or fuel consumed.

The unique approach presented in this paper has been proven to satisfy the research objectives through simulated testing. These results have demonstrated that the approach outlined can be used to enable a UAV to plan its own missions onboard. The mission plans can be optimized based upon distance traveled, time required, or fuel consumed. Furthermore, when planning a mission the aircraft is given greater freedom of movement than previous approaches. This novel multidisciplinary algorithm therefore meets the established research objectives.

The novel multidisciplinary approach uses a combination of algorithms from the three-dimensional graphics and robotics field. Three-dimensional graphics routines are used to construct and maintain situational awareness of the operating environment. The situational awareness provides the foundation on which higher level actions such as mission planning can be based. The mission planning itself is achieved through algorithms drawn from the robotics field. These algorithms were implemented and tested using a scenario constructed from real-world data. Optimizations were then made to the algorithms in order to enhance the performance.

Based upon simulation testing, the optimal algorithm for UAV onboard mission planning in civilian airspace was identified. This algorithm was a cube-space algorithm using the fuel wave (detailed aircraft model) cost function and a resolution of 1 n mile. Using this algorithm, missions were able to be planned in under 10 min. This algorithm required approximately 88 MB of storage space to store the library of precomputed civilian airspace entities. A further 2 MB was required to store the digital representation of the world used for collision detection (between desired path and entities) purposes.

The novel multidisciplinary approach presented in this paper enables a UAV to perform its own mission planning. Human operators are only required to provide the UAV with its mission objectives, and then the onboard systems will takeover. This therefore provides benefits in terms of increasing the onboard intelligence while also reducing the operator workload. The capabilities of the UAV can be more fully exploited as the UAV can replan its mission during a flight based upon a change in the environment (e.g., storm front, directive from air traffic control, etc.). This in turn provides us with UAVs that are more capable and that require less "babysitting" from their human operators.

The novel research presented here represents one step toward highly intelligent UAV platforms.

Acknowledgments

This work was carried out in the Cooperative Research Center for Satellite Systems with financial support from the Commonwealth of Australia through the CRC Program. This research was supported in part by a grant of computer software from QNX Software Systems, Ltd.

References

[1]Beard, R. W., and McLain, T. W., "Trajectory Planning for Coordinated Rendezvous of Unmanned Air Vehicles," AIAA Paper 2000-4369, Aug. 2000.

[2]Beard, R. W., McLain, T. W., and Goodrich, M., "Coordinated Target Assignment and Intercept for Unmanned Air Vehicles," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 6, 2002, pp. 911–922.

[3]Bortoff, S. A., "Path Planning for UAVs," *Proceedings of the American Control Conference*, Vol. 1, 2000, pp. 364–368.

[4]Mitra, T., and Chiueh, T.-C., "An FPGA Implementation of Triangle Mesh Decompression," *IEEE Symposium on Field-Programmable Custom Computing Machines*, Inst. of Electrical and Electronics Engineers, Computer Society, Napa, CA, 2002, pp. 22–34.

[5]Evans, F., Skiena, S. S., and Varshney, A., "Optimizing Triangle Strips for Fast Rendering," *IEEE Visualisation '96*, Vol. 1, Inst. of Electrical and Electronics Engineers Computer Society Press, Los Alamitos, CA, 1996, pp. 319–326.

[6]Schroeder, W. J., Zarge, J. A., and Lorensen, W. E., "Decimation of Triangle Meshes," *SIGGRAPH '92*, edited by J. J. Thomas, Vol. 26, ACM Press, New York, 1992, pp. 65–70.

[7]Hugues, H., "Progressive Meshes," *SIGGRAPH '96*, ACM Press, New York, 1996, pp. 99–108.

[8]Carr, N. A., Hall, J. D., and Hart, J. C., "The Ray Engine," *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, Eurographics Association, Saarbrucken, Germany, 2002, pp. 37–46.

[9]Möller, T., and Trumbore, B., "Fast, Minimum Storage Ray-Triangle Intersection," *Journal of Graphics Tools*, Vol. 2, No. 1, 1997, pp. 21–28.

[10]Segura, R. J., and Feito, F. R., "Algorithms to Test Ray-Triangle Intersection. Comparative Study," *9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2001, pp. 76–81.

[11]Kim, J.-O., and Khosla, P. K., "Real-Time Obstacle Avoidance Using Harmonic Potential Functions," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, 1992, pp. 338–349.

[12]Wu, K.-H., Chen, C.-H., Ko, J.-M., and Lee, K.-D., "Path Planning and Prototype Design of an AGV," *Mathematical and Computer Modelling*, Vol. 30, No. 7–8, 1999, pp. 147–167.

[13]Horng, J.-H., and Li, J. T., "Vehicle Path Planning by Using Adaptive Constrained Distance Transformation," *Pattern Recognition*, Vol. 35, No. 6, 2002, pp. 1327–1337.

[14]Ting, Y., Lei, W. I., and Jar, H. C., "A Path Planning Algorithm for Industrial Robots," *Computers and Industrial Engineering*, Vol. 42, Nos. 2–4, 2002, pp. 299–308.

[15]Vörös, J., "Low-Cost Implementation of Distance Maps for Path Planning Using Matrix Quadtrees and Octrees," *Robotics and Computer Integrated Manufacturing*, Vol. 17, No. 6, 2001, pp. 447–459.

[16]Bandi, S., and Thalmann, D., "Path Finding for Human Motion in Virtual Environments," *Computational Geometry*, Vol. 15, Nos. 1–3, 2000, pp. 103–127.

[17]Kitamura, Y., Tanaka, T., Kishino, F., and Yachida, M., "3-D Path Planning in a Dynamic Environment Using an Octree and an Artificial Potential Field," *IEEE International Conference on Intelligent Robots and Systems*, Vol. 2, Inst. of Electrical and Electronics Engineers Computer Society Press, Pittsburgh, PA, 1995, pp. 2474–2481.

[18]Foskey, M., Garber, M., Lin, M. C., and Manocha, D., "A Voronoi-Based Hybrid Motion Planner," *IEEE International Conference on Intelligent Robots and Systems*, Vol. 1, Inst. of Electrical and Electronics Engineers Computer Society, Maui, HI, 2001, pp. 55–60.

[19]Kassim, A. A., and Kumar, V. V. K. V., "Path Planners Based on the Wave Expansion Neural Network," *Robotics and Autonomous Systems*, Vol. 26, No. 1, 1999, pp. 1–22.

[20]Warren, C. W., "A Technique for Autonomous Underwater Vehicle Route Planning," *Journal of Oceanic Engineering*, Vol. 15, No. 3, 1990, pp. 199–204.

[21]Roth-Tabak, Y., and Jain, R., "Building an Environment Model Using Depth Information," *Computer*, Vol. 22, No. 6, 1989, pp. 85–90.

[22]Matthies, L., and Elfes, A., "Integration of Sonar and Stereo Range Data Using a Grid-Based Representation," *IEEE International Conference on Robotics and Automation*, Vol. 2, Inst. of Electrical and Electronics Engineers Computer Society Press, Washington, DC, 1988, pp. 727–733.

[23]Elfes, A., "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer*, Vol. 22, No. 6, 1989, pp. 46–57.

[24]Li, S.-M., Boskovic, J. D., Seereeram, S., Prasanth, R., Amin, J., Mehra, R. K., Beard, R. W., and McIain, T. W., "Autonomous Heirarchical Control of Multiple Unmanned Combat Air Vehicles (UCAVs)," *American Control Conference*, Vol. 1, Inst. of Electrical and Electronics Engineers, Anchorage, AK, 2002, pp. 274–279.

[25]Atkins, E. M., Durfree, E. H., and Shin, K. G., "Plan Development Using Local Probalistic Models," *Twelfth Conference on Uncertainty in Artificial Intelligence*, edited by E. Hoyvitz and F. V. Jensen, Morgan Kaufman Publishers, Portland, OR, 1996, pp. 49–56.

[26]Atkins, E. M., Durfree, E. H., and Shin, K. G., "Autonomous Flight with CIRCA-II," *Autonomous Agents'99 Workshop on Autonomy Control Software*, edited by O. Etzioni, J. P. Muller, and J. M. Bradshaw, ACM Press, Seattle, WA, 1999.

[27]Alonso-Portillo, I., and Atkins, E. M., "Adaptive Trajectory Planning for Flight Management Systems," AIAA Paper 2002-1073, 2002.

[28]*Aerosim Blockset Version 1.1 User's Guide*, Unmanned Dynamics, LLC, Hood River, OR, Aug. 2003.

[29]McManus, I., and Walker, R., "Intelligent Agents 'Hardware in the Loop' Simulator for UAV and Spacecraft Mission," *Cooperative Research Center for Satellite Systems Conference*, 2003.

[30]McManus, I., Greer, D., and Walker, R., "UAV Avionics 'Hardware in the Loop' Mission Simulator," *Australian International Aerospace Conference*, AIAC 2003-036, July–Aug. 2003.

[31]*Aerosim Blockset Version 1.01 User's Guide*, Unmanned Dynamics LLC, Hood River, OR, Nov. 2002.