NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

PROPAGATING A*: SEARCHING STATE GRAPHS IN ORDER TO FIND A VALID INSTRUMENT APPROACH CONFIGURATION

by

Trent L. Bottin

September 2007

Thesis Advisor:	Christian Darken
Second Reader:	Joseph Sullivan

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Propagating A*: Searching State Graphs in Order to Find a Valid Instrument Approach Configuration			**5. FUNDING NUMBERS**
6. AUTHOR(S) Trent L. Bottin			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The helicopter community has consistently been overlooked in the development of the National Airspace System. The unique flight characteristics of these aircraft make them ideally suited for a wide range of missions that are critical to national defense, medical first response and disaster relief. Full exploitation of these capabilities is limited during inclimate weather because the existing airspace plan was developed around fixed wing aircraft. More specifically, the Federal Aviation Administration lacks the resource to generate terminal area procedures for aircraft not restricted to prepared landing surfaces.

This thesis focuses on the development of a suitable terminal instrument approach procedure generation capability. Artificially intelligent path planning and computer graphics-based collision detection techniques are used to find valid approach procedures that are compliant with the requirements set forth by the Federal Aviation Administration. A variant of the classic A* graph search algorithm is introduced that propagates state change information to successor nodes. The propagation technique allows the algorithm to search the graph in a single pass even though children nodes often impose a state change on their parent nodes.

14. SUBJECT TERMS Terminal Area Procedures, Path Planning, A* search Algorithm, Helicopter Instrument Approach, Global Positioning Navigation, Collision Detection			**15. NUMBER OF PAGES** 85
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

PROPAGATING A*: SEARCHING STATE GRAPHS IN ORDER TO FIND A VALID INSTRUMENT APPROACH CONFIGURATION

Trent L. Bottin
Captain, United States Marine Corps
B.S. The Citadel, 1998

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2007

Author: Trent L. Bottin

Approved by: Christian Darken
Thesis Advisor

Joseph Sullivan
Second Reader

Peter Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The helicopter community has consistently been overlooked in the development of the National Airspace System. The unique flight characteristics of these aircraft make them ideally suited for a wide range of missions that are critical to national defense, medical first response and disaster relief. Full exploitation of these capabilities is limited during inclimate weather because the existing airspace plan was developed around fixed wing aircraft. More specifically, the Federal Aviation Administration lacks the resource to generate terminal area procedures for aircraft not restricted to prepared landing surfaces.

This thesis focuses on the development of a suitable terminal instrument approach procedure generation capability. Artificially intelligent path planning and computer graphics-based collision detection techniques are used to find valid approach procedures that are compliant with the requirements set forth by the Federal Aviation Administration. A variant of the classic A* graph search algorithm is introduced that propagates state change information to successor nodes. The propagation technique allows the algorithm to search the graph in a single pass even though children nodes often impose a state change on their parent nodes.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AGL – Above Ground Level
ATC – Air Traffic Control
CFIT – Controlled Flight Into Terrain
CWO – Collision With an Obstacle
EMS - Emergency Medical Services
FAA – Federal Aviation Administration
FAF – Final Approach Fix
FAR – Federal Aviation Regulation
GPS – Global Positioning System
HEMS – Helicopter Emergency Medical Services
IF – Intermediate Fix
IAF – Initial Approach Fix
IFR – Instrument Flight Rules
IMC – Instrument Meteorological Conditions
IIMC – Inadvertent Instrument Meteorological Conditions
LZ – Landing Zone
MAP – Missed Approach Fix
MSL – Mean Sea Level
NAS – National Airspace System
NTSB – National Transportation Safety Board
NVG – Night Vision Goggles
OIS - Obstacle Identification Surface
ROC - Required Obstacle Clearance
TAWS – Terrain Awareness Warning System
VFR – Visual Flight Rules
VMC – Visual Meteorological Conditions

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to extend the most heart felt thank you to my wife, Mary, whose support as a wife and mother I cherish deeply and relied upon heavily. To Jonathan Towle, your feedback during countless miles ran served to keep things in perspective. Further, despite your choice of airframe and my badgering, I have the utmost respect for you as a pilot and Marine Corps Officer.

Finally, to Professor Darken and Commander Sullivan, your guidance and professionalism maintained a steady course and a focus on the objective. Thank you both.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The work here is inspired by the technological gap between capability and application in the National Airspace System (NAS) with respect to helicopter flight. This thesis proposes the use of artificially intelligent path planning techniques to generate terminal instrument procedures. The purpose of this application is to provide rotorcraft an “anywhere” instrument capability. The approach procedure planning problem (AP3) is presented and an adjusted cost propagation variant of the classic A* algorithm is used to reduce the problem’s dimensionality and search an unknown graph in a single pass.

A. BACKGROUND

Ask an airplane pilot what “IFR” stands for and not surprisingly the answer is “Instrument Flight Rules.” Ask a helicopter pilot the same question and the response given more often than not is “I Follow Roads.” This does not suggest that the rotary wing community lacks basic aeronautical knowledge but rather that there is a fundamental difference in how these two communities view the National Airspace System (NAS).

The NAS was developed as a means to improve and maintain the safety of air traffic. Primary navigation during inclimate weather was accomplished through a network of land-based radio navigation aids (navaids). Pilots could determine their position (or fix) by plotting the heading/distance information from the navaid’s known location or triangulating two heading signals if distance measuring equipment was not available. Radio-based navigation was a significant safety improvement for aviation as a whole, but signal reliability and precision at distance was not suited for low-level flight. Further, navaids were mostly co-located with airports. Thus, the capabilities of helicopters during instrument meteorological conditions[1] (IMC) were restricted to those of fixed wing aircraft.

Unfortunately, rotorcraft operations with the greatest need for instrument flight environment are also the most essential. Crop dusting and commercial logging are important services but they are not time-critical. Flights can easily be limited to day-time only and conducted under visual flight rules (VFR). Emergency medical responders and airborne police do not have this luxury as their duties must be able to be performed day or night and in all weather conditions. Because very few missions are conveniently located near established terminal areas with published approach procedures, ATC services are reduced to basic flight following where aircraft tracking and traffic advisories are provided but primary navigation is left to the pilot.

Flight following maybe sufficient when visibility is good and navigating an aircraft is easy, but at night or during inclimate weather it is simply insufficient. Precise visual navigation during reduced visibility requires a delicate balance between altitude and obstacle separation.

[1] Meteorological conditions defined by visibility of less than 1000 feet vertically and/or 3 miles horizontally.

Fly too high and vital navigation clues are missed[2]. Fly too low and the pilot runs the risk of striking an obstacle (i.e. power lines, cell phone tower, etc.) or controlled flight into the terrain (CFIT).

Poor weather makes things worse by limiting the amount of maneuvering altitude. The pilot is forced to fly just below the cloud layer (referred to as "scud running") risking inadvertent instrument meteorological conditions (I-IMC), in order to achieve maximum obstacle clearance. Double-IMC is extremely dangerous because the sudden and unanticipated transition from flying "outside the cockpit" to relying on instruments is disorienting and often induces vertigo, spatial disorientation and/or lose of situational awareness[3]. All three conditions are exacerbated by the initial close proximity to the ground.

From 1999 to 2002, Rick Frazer published a comprehensive series of articles addressing the high mishap rate of EMS operations [1] [2] [3] [4]. "Air Medical Accidents – A 20-Year Search for information" was the first in the series and presented a comparison between fixed wing and rotary wing communities, categorized the causes and

[2] During visual flight, the pilots primarily fly "by the seat of their pants." Aircraft instruments like the altimeter and airspeed indicator serve as a back up - a reference to gauge the pilot's interpretation of physical and visual clues obtained outside the cockpit.

[3] Vertigo is the erroneous sensation of movement. A pilot will often mistakenly make a control input based on the perceived sensation. Extreme cases of vertigo result in total lose of control of the aircraft. Spatial orientation refers to the ability to conceptualize the aircraft's position, altitude, heading and airspeed with respect to the operating space. Situational awareness refers to the ability to balance the various tasks (aviate, navigate, communicate) required to fly the aircraft. A lose of situational awareness is often referred to as "being behind the aircraft."

identified the phases of flight in which accidents happened. The latter three articles each focused one of the major causes, citing specific examples, updating data and identifying trends. In [1], the author identified 122 accidents associated with dedicated air medical programs from 1978 to 1998 of which 107 (88 percent) were helicopters and the remaining 15 (12 percent) were airplanes.

Frazer established three categorical causes based on details provide in the NTSB reports. A mechanical failure is an instance of a system failure in the aircraft in which no reasonable action on behalf of the pilot could have impacted the outcome. 24 percent (26 of 107) of the HEMS accidents were attributed to mechanical failure and five resulted in at least on fatality. Pilot error is when the actions of the aircrew are identified as the major cause of the mishap. 69 of the 107 HEMS accidents (64 percent) are attributed to pilot error and 32 resulted in at least on fatality. The remaining 12 incidents (five with fatalities) are classified as other because the cause could not be determined or the NTSB's final investigation report had not been released at the time Frazer published his findings[4].

Collision with an obstacle (CWO) and weather are the leading factors associated with pilot error accidents (19 and 24 accidents respectively) but weather related misfortune bore a fatality rate of six times that of CWO (18 and 3 respectively). Ira Blumen suggests this is logical citing that most CWOs occur in the take-off/landing environment with lower airspeeds and altitudes [5].

[4] In each of the subsequent articles, Frazer provides an update to the number of accidents along with a brief overview.

Frazer’s analysis of CWO [3] provides the foundation for this hypothesis stating that 15 incidents took place in the take-off/landing environment but all fatal CWOs occurred during the cruise phase of flight.

Juliana Goh and Dr. Douglas Wiegmann presented a study of general aviation accidents involving weather from the mid-1970s to the mid-1980s [6]. The findings stated that VFR flight into IMC resulted in death 72 percent of the time and accounted for 19 percent of all general aviation fatalities. Using only the 95 accidents with complete NTSB reports, Frazer’s data correlates the fatality rate with respect to I-IMC (75 percent) but fatal HEMS accidents attributed to weather accounted for 49 percent of all HEMS deaths – a difference of 2.5 times. This is because the nature of HEMS operations forces the pilots into marginal weather conditions.

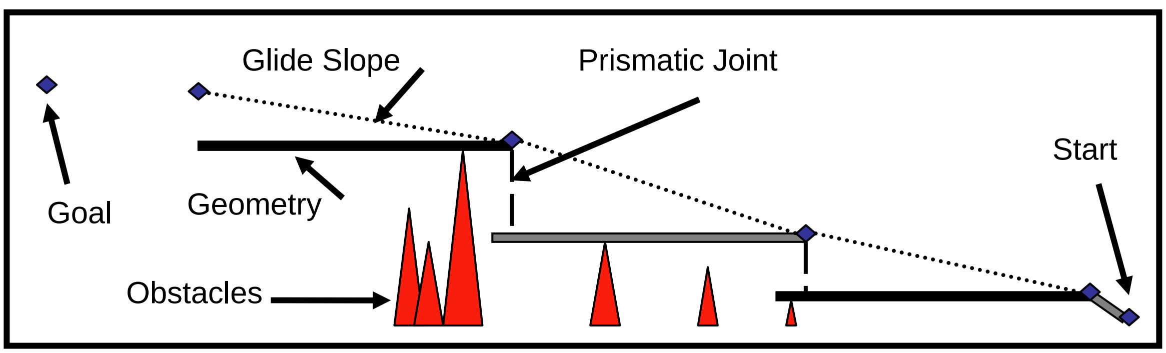
The oversight of weather is not due to a lack of awareness on behalf of the FAA. In 1988, the National Transportation Safety Board (NTSB) conducted a survey of HEMS operations due to its high accident/fatality rate and acknowledged weather as the single greatest threat [7]. The HEMS fatality rate provoked another NTSB special investigation in 2005 [8]. Surprisingly, the report conclusions cited dispatch procedures and risk management as the major causes for accidents but not one recommendation addressed the need to improve the instrument flight services provided to helicopters.

The FAA cannot even claim they lack the technology to make changes. Global Positioning Systems (GPS) brought about major advancements in aviation navigation and spurred the most significant modernization of the NAS since the first instrument landing. In September 1987 – one year before the first NTSB investigation into HEMS operations - the FAA released the *Rotorcraft Master Plan* with the expressed intent of “realiz[ing] the full potential of rotorcraft in meeting the nation's transportation needs” [9]. A cost/benefit analysis study the following year, [10] found that GPS technology and point-in-space approaches presented realistic solutions to airspace deficiencies with respect to helicopter instrument flight. Since then, however, most of the FAA’s vertical flight research has been dedicated to de-conflicting VFR helicopter traffic and fixed wing IFR traffic with what is known as simultaneous non-interfering (SNI) operations [11] [12] [13]. So negligent is their attention to rotorcraft IFR operations, the FAA’s current instrument procedure validation tool does not even include helicopter approach models. The validation process requires “tricking” the software suite into believing the submitted procedure is being executed to an established runway. To make matters worse, the next generation replacement does not nor intends to include the missing models either [14].

B. INSTRUMENT APPROACH PROCEDURE PLANNING PROBLEM

The following is an overview of the problem. An actual instrument approach has more segments but this representation is sufficient to present the complexities of the problem. Exact specifications for the approach geometry and clearance requirements for each segment are presented in Chapter III.

The instrument approach procedure planning problem (AP3) is concerned with finding a valid approach path from the en route phase of flight to the landing environment (see Figure 1). An en route waypoint and intended point of landing are given to be used as the start and goal states[5] and are fixed in 3-space. The approach is defined by five waypoints and four connecting segments. Each segment has a unique geometry and a joint at the waypoint closest to the intended point of landing. The geometry defines a convex hull that must be free of obstacles. The joint is prismatic with one degree of translational freedom (vertical axis) and one degree of rotational axis (vertical axis). A segment has a third degree of freedom in its length. All movement is bounded with minimum, maximum and optimal values.



[5] Construction can occur from either direction or simultaneously. This explanation adopts the convention of having the landing fix serve as the start state.

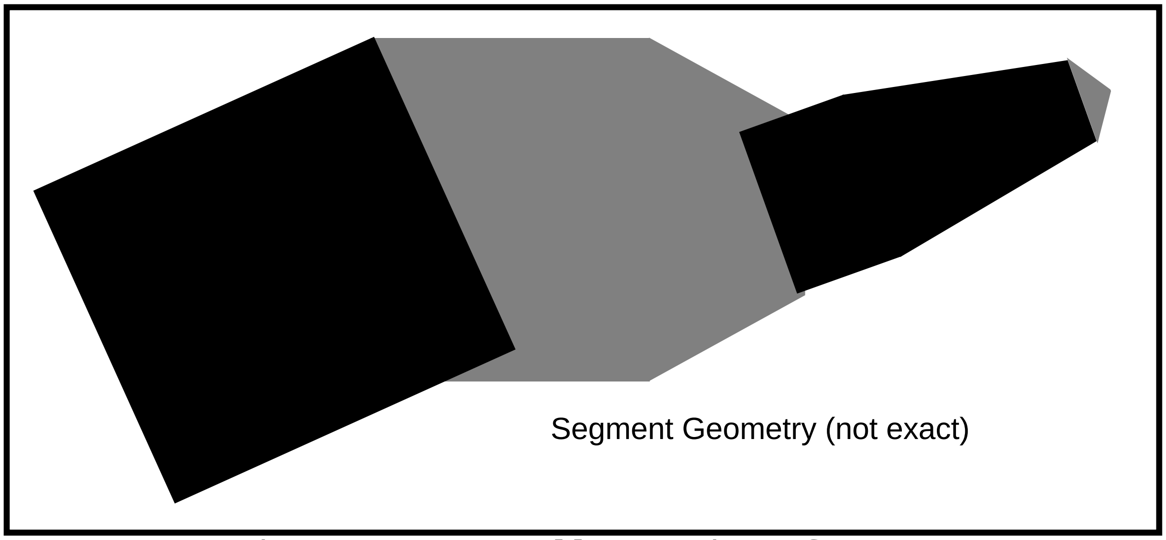


Figure 1. Illustration of AP3.
Top: Side view
Bottom: Top-down view

The best approach path is the least sum of each segment’s cost. A segment’s cost is calculated based on three criteria: the amount of course deviation at the joint, length, and gradient to the next joint (glide slope). The en route waypoint is technically part of the approach. However, its location has an effect on the optimality of an approach and therefore included in the problem.

C. OBJECTIVES

This thesis has two objectives. The first goal is to introduce a cost propagating variant of the A* search algorithm. The second is to demonstrate the potential of using automated planning techniques to generate terminal instrument procedures.

D. SCOPE

The scope of this thesis is limited to discussion on path planning issues related to generating instrument procedures for helicopters. The particular approach model used is the global positioning system (GPS) helicopter point-in-space (PinS) approach. There are many legal and administrative issues involved with fielding a solution along the lines described here but their discussion beyond the scope of research presented.

E. ORGANIZATION

Chapter II. Literature and Technology Review

This chapter presents the current research related to this thesis. Focus is directed at path planning and variations of the A* search algorithm.

Chapter III. Approach Design: Requirements and Specifications

Chapter III covers Federal Aviation Administration orders that govern the design and development of terminal area instrument procedure. Some information addressed chapter is universal but emphasis is on familiarizing the reader with terminology and approach construction details specific to the rotary wing flight.

Chapter IV. System Design

Chapter IV presents the Instrument Procedure Generation Tool developed for this thesis. Specific design considerations are presented with a focus the proposed variant of the A* search algorithm.

Chapter V. Implementation and Testing

This chapter covers specific implementation of the Instrument Procedure Generation Test Environment and results obtained during experimentation.

Chapter VI. Conclusions and Future Works

The final chapter concludes the thesis with a discussion on the feasibility of implementing automated path planning and application domains. Suggestions are presented where additional work can further the research addressed here.

II. LITERATURE REVIEW

In artificial intelligence (AI), path planning is the science of finding a valid transition from a start state to a goal state for some object. When considering a single point representation, a valid solution is a set of points that allows for a continuous and unobstructed path. In the case of rigid bodies, this problem is described as the classic Piano Mover's Problem [15]. When talking about a manipulator (e.g. robotic armature), a path is the set of points that defines a valid configuration, including the positions and/or angles of the robot's joints. The latter definition refers to a single state in the larger motion planning problem as described by the Generalized Mover's Problem. Because the basic concepts apply to both point-objects and manipulators, the term robot will be adopted for both and distinctions made where required.

A. PLANNING OVERVIEW

Path planning can be broken down into three distinct steps that have each been researched extensively. A brief discussion on the first two steps addressing spatial considerations is covered here. Searching for a valid path is the third point and is discussed in the next section.

Spatial representation is determining how to describe the robot and its environment. The most common methods are n-dimensional grids, 2^n-cells and polygonal approximations. [16] [17] and [18] address the strengths and weaknesses of each and introduces other techniques as well.

Spatial reasoning attempts to define the relationship between the robot and its environment. The primary goal in this step is to identify all positions or configurations that result in a collision or intersection. Lozano-Perez’s seminal works, [19] and [20], introduce the concepts of position constraints (point) and configuration obstacles (manipulator).

> Let W be the representation of a problem’s workspace and R be a robot. Let O be a subset of W representing all obstacles. Then, R’s position constraints C are the set of all points in W such that if R occupies a position in C it will intersect with O. The free space (F) to be considered for planning purposes is then defined as F = W – (O + C).

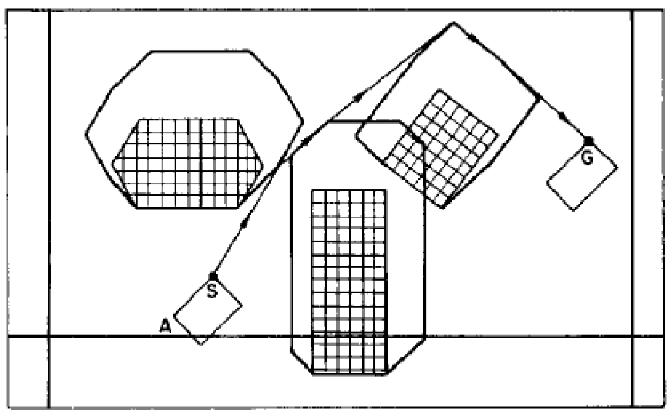


Figure 2. The Position Constraints of Object A (From Lozano, 1983).

Udupa describes a similar method by expanding all obstacles a sufficient amount to ensure adequate separation for an arbitrary position [21]. Hwang and Ahuja provide an extensive survey of other spatial reasoning methods [22].

Regardless of the implementation, the objective of the first two steps is to provide an exact portrayal of the free space to be considered. A directed graph is often used for this purpose but in many problems a discrete representation is not feasible due the size of the problem or the fact that the free space is a continuum. The simple illustration above limits the dimensionality of the problem to 2-space (x and y) but results in an infinite graph due to the problem's continuous nature. Even in a countably finite space, many manipulator problems are computationally intractable due to high dimensionality.

To handle these cases, alternate representation approaches are required. Lozanop-Perez used a visibility graph to implicitly model the free space [19]. Defining the optimal path as the shortest distance, a straight line from the start to goal is used. If the path encounters a forbidden region, a localized optimal path is calculated to circumnavigate the obstacle. Alexander notes that this method has the behavior of staying arbitrarily close to obstacles which may not be desirable for applications [23].

A Voronoi Diagram (VD - introduced by Georgy Vorony) solves this problem by identifying the edge set that is equidistant from the nearest obstacle vertex for all vertices. This can then be used as the graph where any existing path is guaranteed to be maximally clear. Since generating a VD can involve significant overhead, [24] implemented a graph repair algorithm that allow for this structure in a real-time application.

John Canny introduces an approach for complex environments that is conceptually close to the previous method [25]. A low resolution Voroni Diagram is generated globally representing a roadmap of the environment. Then, a local planner is used to map a path from the start state to the closest point on the roadmap (this process is repeated for the goal state). The authors of [26] tackled a high dimensional manipulator problem (8-degrees of freedom) using a probabilistic application to generate a roadmap. Results showed it was unsuitable for dynamic environments since “learning” the roadmap took on the order of 25 minutes.

A completely different spatial technique to reduce the graphical representation is cell decomposition. In this method, the workspace is recursively divided into 2^n cells where n is the dimensionality of the workspace. Quad-trees and Oct-trees are typical algorithms used in two and three dimensions, respectively. If a cell contains an obstacle, it is further decomposed in a recursive fashion. This process is repeated to some pre-defined resolution or until no cells contain obstacles. When the environment has been sufficiently decomposed, a connectivity graph is used to identify adjacent cells that are clear of obstacles. In [27], the author notes that previous work with cell decomposition was restricted to off-line planning due to time constraints as a result of the method’s exponential nature. Linglebach implements Probabilistic Cell Decomposition where the environment is decomposed and then paths are probabilistically sampled from the free space based on sub-goals (intermediate states) [28].

B. SEARCHING FOR A SOLUTION

1. A* Search Algorithm

A* (pronounced A-star) is a deterministic, guided graph search algorithm introduced by [29] that returns the optimal path from a given start state (startNode 0 S) to a given goal state (goalNode 0 S) where S is the set of all possible nodes (see pseudo-code below). It differs from Dijkstra’s more generalized best-first search [30] by augmenting the “cost-to-go” function with an estimated remaining cost value which serves to focus the search to those nodes that offer the most potential of an optimal solution. The algorithm is well suited from many applications but there some characteristics that limit its use.

```
function A_star(startNode,goalNode)
  open = empty;
  closed = empty;
  open.add(startNode);
  node curNode;
  while open != empty
    curNode = open.getFront();
    if closed.contains(curNode)
       continue;
    if curNode == goalNode
       return curNode;
    for all successors of curNode
       successor.parent = curNode;
       successor.setCost(curNode);
       open.add(successor);
    closed.add(curNode)
  return NULL  //No path found
```

Figure 3. A* Psuedo-Code.

2. Reducing Closed List

A* maintains a closed list to track nodes that have already been visited. If the closed list contains a node that is subsequently proposed as a successor to the current node, the potential successor can be omitted. There are application domains where the use of a closed list is either not required or not allowed. The approach path planning problem is an instance where it is not required because the state graph is non-cyclical. However, there has been interesting work in the area it is worth mentioning.

For large search spaces, the overhead required of the closed list can cause a memory limitation. In [31], Korf proposed the Iterative Deepening A* (IDA*) which does not maintain a closed list. Instead, each iteration of IDA* starts a new search from the root node, incrementally searching the graph deeper. The closed list is then replaced with a stack to organize the sequence of searches and search history is retained for the current search only. [32] points out that this introduces duplicate path nodes on the open list and introduce Enhanced IDA* (E-IDA*). The E-IDA* algorithm uses transposition tables to maintain duplicate expansion awareness and pre-sorting methods for recovering the goal path.

Korf and Zhang introduced another scheme to search a graph without requiring the closed list [33]. Divide and Conquer Frontier Search (DCFS) behaves in the same manner but maintains (on each node) it own history in the form of forbidden operations list which restricts the re-expansion of interior node that would have been on the closed list. The terms Divide and Conquer refers to the solution path recovery technique and not to the actual search itself.

Zhou and Hansen point out in [34] that a drawback to the DCFS scheme is that the overhead required by the forbidden operators surpasses that of the closed list for large search spaces. Further, there is susceptibility to duplicate expansion in directed graph searches. The Sparse-

Memory Graph Search (S-MGS) is presented as an alternative. In this approach, a sparse reflection of the closed list is maintained adopting DCFS's frontier concept and solution path recover technique.

3. Reducing Open List

For exponential search problems, DCFS is poorly suited because the growth of the new nodes quickly out paces any growth in the closed list making the open list the limiting factor [33]. Research in the area of reducing the open list of the A* search algorithm is limited because most solutions do not generalize.

Algorithms proposed by [35] and [36] use the well defined properties of a DNA problem to establish an upper bound on the optimal solution cost. Successor nodes with an incurred cost greater than this upper bound are then discarded. Klein and Manning suggest a class of such problems exist citing at least three other application domains in addition the natural language processing problem they researched [37].

Partial Expansion A* presents a generalized solution for bush-like graphs [38]. If during successor discovery new nodes do not show promise toward yielding the solution; the expansion is halted. The parent node's f-value is then increased to a new cost less than the least cost of any of its successors and it is placed back in the open list.

4. Re-Planning

A* is a graph search algorithm with limited path planning capabilities. If the problem is countably finite and static, then A* can be used. The efficiency of A* is greatly reduced in unknown environments or on graphs with partial information. If new information is learned during the search that affects the cost of a node in the closed list, the only option is to correct the graph's information and restart the search.

Karen Trovato explains using Differential A* in such a manner for robot motion planning [39]. A solution is calculated using classic A* with available information and the robot begins along the path. If an unanticipated obstacle is encountered, the robot updates the graph and re-calculates from the new position. [42, 43] traces the progression of Dynamic A* (referred to as D*) using the same concept as Trovato but with advanced techniques to increase performance but restricting repairs to relevant area of the graph.

The Anytime A* algorithm [40] addresses planning in a time constrained environment. The basic concept is to quickly find a realistic sub-optimal path and optimize the solution when time permits. Likhachev et al has combined the D* with the anytime concept (AD*) [42]. In a following on paper, [43], the authors report that results have been mixed and occasionally it is more cost efficient to just re-calculate a fresh, classic A* solution.

III. APPROACH CONSTRUCTION REQUIREMENTS

The National Airspace System is a complex environment but the profile of a flight can be broken into three distinct environments (ground, terminal and en route) each with its own set of procedures. An instrument approach is a terminal procedure intended to provide a safe path of flight during the transition from the en route environment to the landing phase. The following explains the basic components of an approach to familiarize the reader with concepts and terminology. Complete requirements for developing terminal instrument procedures can be found in [44][45][46][47].

A. POINT-IN-SPACE APPROACHES

There are two main types of instrument approaches: precision and non-precision. [46] describes a precision approach as a descent procedure providing course and glide slope information. A non-precision procedure provides course but no glide slope information. A Point-in-Space (PinS) approach is a special type of non-precision procedure because it does not terminate at a landing facility. It is intended to provide a safe IFR let down for aircraft wishing to continue flight visually if possible. This makes it an ideal approach model for this thesis and its use is consistent with The Rotorcraft Master Plan. [10] stated in a supporting cost/benefit study, ”A rotorcraft point-in-space approach, if properly developed, offers a simple and logical means of providing this transition...from IFR to VFR.”

B. SPECIFICATIONS

A PinS approach is constructed in the same fashion as other non-precision procedures. [46] sets forth the guidelines and requirements provided in this section.

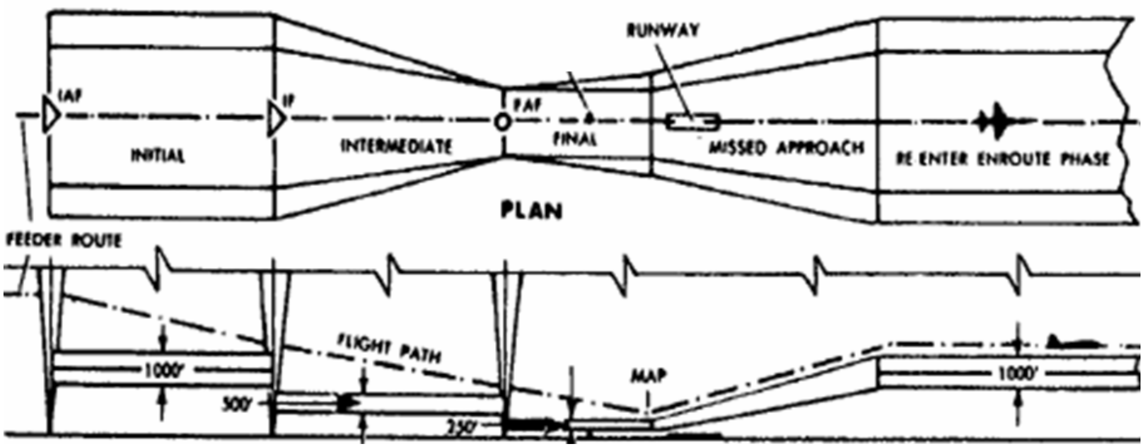


Figure 4. Approach Profile (From FAA Order 8260.3b.).

A basic approach is characterized by five waypoints connected by four segments. The procedure commences with the aircraft at the Initial Approach Fix (IAF) and progresses to the Intermediate Fix (IF), Final Approach Fix (FAF), Missed Approach Point (MAP) and finally to the Missed Approach Holding Fix (MAHF - not depicted) if required (see Figure 6). A Feeder Route providing a transition from the en route environment to the terminal phase of flight may be required but it is not technically part of the approach procedure[6].

Each segment has it own construction specifications but there are some key aspects common to all (see Figures 7 - 9). A segment's length is measured by the horizontal distance between the defining waypoints. There is an

[6] This thesis only addresses the approach from the IAF to the MAP for reasons explained in the next chapter.

invisible surface beneath each segment called the Obstacle Identification Surface (OIS). The Required Obstacle Clearance (ROC) provided by an OIS is different with each segment type but in all cases it is measured from the lowest waypoint in a segment. The OIS is three finite planes. The primary OIS is horizontal, remains at a constant altitude and is flanked by two sloping secondary OISs.

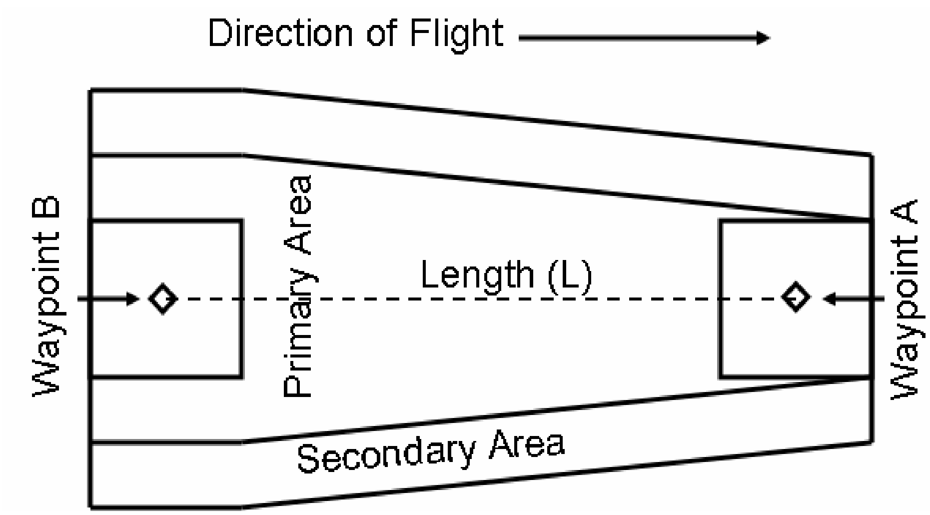


Figure 5. Segment: Top View.

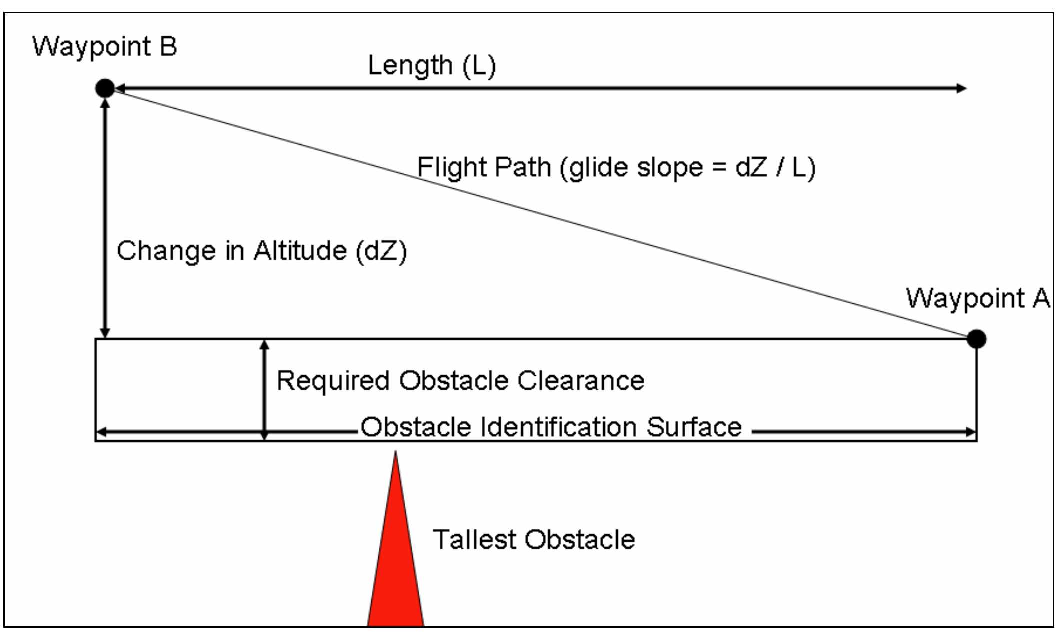


Figure 6. Segment: Side View.

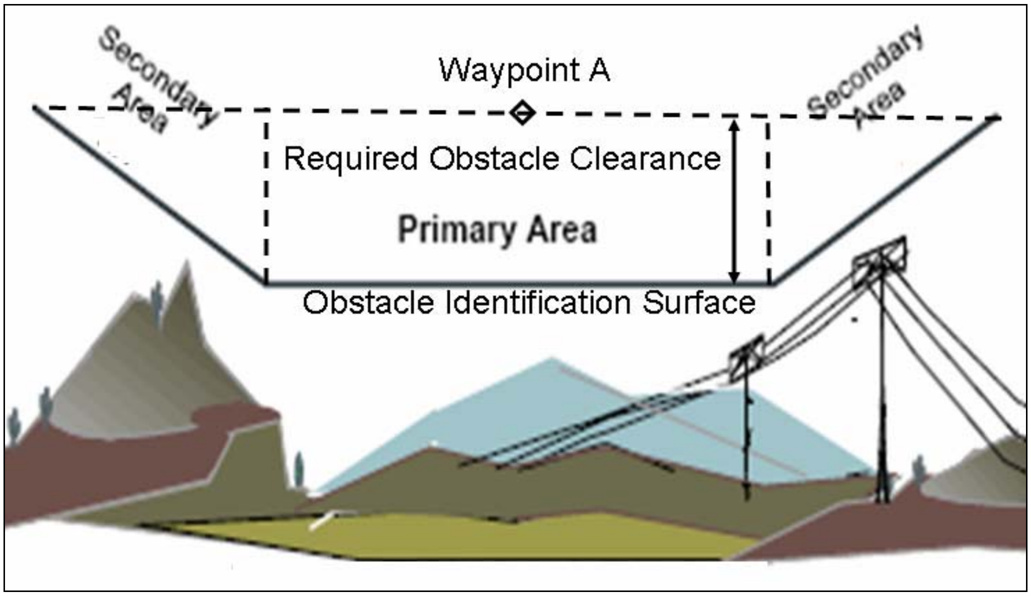


Figure 7. Segment: End View (After FAA Order 8260.3b.).

Each segment has a unique shape that must be considered during construction (see Figure 10). The Initial Segment has a uniform width with whereas the other two segments are tapered. The taper begins 2.0 nm before the FAF in the Intermediate Segment, but immediately after the FAF surface area in the Final Segment. Also note that the Final Segment is the only segment that extends beyond the latest waypoint though all begin before the earliest waypoint.

		s e g m e n t	
	Initial	Intermediate	Final
Length (nm)			
Min	2.0	2.0	2.0
Opt	3.0	3.0	3.0
Max	10.0	5.0	10.0
Gradient (ft/nm)			
Opt	400	400	400
Max[7]	600	600	600
Max Turn[8] (degrees)	120	120	60
Min ROC (ft)	1000	500	250

Table 1. Segment Specifications.

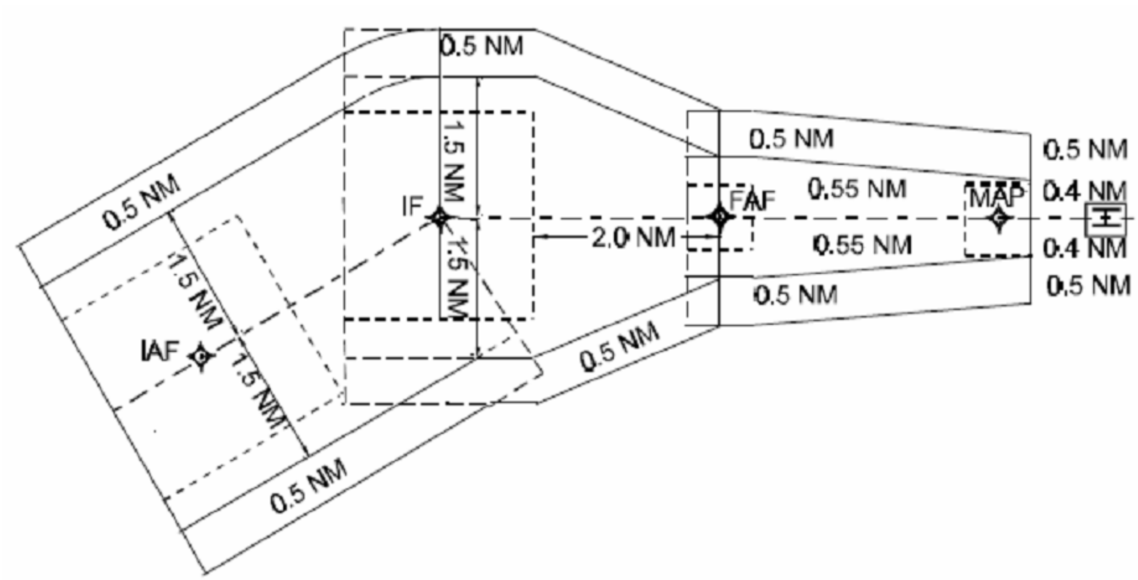


Figure 8. Segment Construction.

[7] The FAA allows for a maximum gradient of 800 ft/nm if required but specific permission is required. This defeats the purpose of automated generation and therefore is not used.

[8] Max turn values are reduced based segment length as given in Table 2 of [Reference 8260.42a]

C. VALIDATION

The process of validating a proposed instrument procedure involves extensive facility surveys and environmental studies. The whole process is expensive and can take as long as 18 months. Instrument automation is intended to reduce this for one-time approaches.

PinS procedures have no associated landing facility so much of the non-administrative overhead[9] is no longer required. The remaining validation process has two steps and is easily automated. The first step is concerned with verifying that the geometry of each segment is constructed correctly as described above. The second step is ensuring that the each segment's OIS is clear of any obstacles. The manner in which this step occurs is strictly an implementation issue but several well known computer graphics techniques are suitable.

[9] As stated in Chapter I, administrative issues are beyond the scope of this work.

IV. DESIGN CONSIDERATIONS

The previous chapter defined optimal states for each segment, implicitly introducing the notion of an optimal configuration. The goal then, is to find the configuration that is closest to optimal given an intended point of landing and the associated terminal area.

Using terminology from robotics and the explanation in Chapter III, the approach model is a revolute and prismatic manipulator in Euclidean three-dimensional space with three degrees of freedom (DoF) for each segment. The dimensionality of the problem produces a configuration graph with exponential growth in the number of increments allowed per DoF and a bush-like appearance. A* is an ideal search algorithm for well defined problems like the one presented here, but rapidly growing structures tend to overwhelm the open list.

This chapter explains an adaptive variant of the classic A* search algorithm and the environment it was tested in. The first section identifies assumptions made with respect to the problem that influenced the development of Propagating A* (PA*) which is introduced in the section following. The last two sections describe the testing environment used and outline specific decision choices.

A. ASSUMPTIONS

As is often the case in applying algorithms, reasonable assumptions are required to define the task at hand and reduce a problem’s complexity. This section outlines those assumptions made with respect the approach planning problem and provides discussion to any implications to the generalization of PA*.

A PinS procedure technically terminates at the missed approach point (MAP), extending to the missed approach procedure if required. By nature, it makes no assumptions about the landing zone (LZ). This is not suitable for an approach generating application because the path between the MAP and the intended point of landing could contain a hazard that would make the procedure unfit. The path planning problem has to assume this responsibility and does so by implementing a check against the area that is traditionally defined by the visual segment of a facility-based procedure. Small increases in terrain elevation near the landing location, however, have the effect of producing excessive approach angles for this segment. A 50 foot vertical displacement of the landing coordinate is used to compensate for this issue. This is a reasonable course of action under the assumption a pilot will be using a “high-hover” technique to identify a safe spot to set down.

The approach is constructed in the opposite direction it is flown - making the landing fix the start node. This is stated as the preferred construction method in [46] and is consistent with tree graphs (i.e. a single root node). The general direction from which an aircraft is approaching the terminal area from is assumed to be known in advance (i.e. a known en route way point or the last point on an en route path search). Minimizing the distance from this point to the Initial Approach Fix (IAF) is part of the planning process thus it used as the goal node for the search. Since we are considering landing approaches only and there are no route points, a hypothetical en route point outside the terminal area radius is used for this purpose.

The next assumption is referred to as the “open sky” policy. This means that there are no overhead obstacles to consider. More formally, for any (x,y) point, there is exactly one value (z) representing the maximum elevation of any obstacle(s) that may exist at that point. Therefore, it is the case that any value z+k, where k is a non-negative real value, is guaranteed to be free of obstacles.

The problem space is assumed to be partially discrete with respect to heading and distance based on requirements set forth by the FAA. Published instrument procedures provide heading and distance information in whole degrees and tenth of a mile increments respectively. To consider any higher degree of precision is pointless because rounding invalidates the procedure’s guarantee of correctness. For example, let a segment at 238.7 degrees be structurally valid and free of obstacles. The same can not be said about the same configuration at 239.0 degrees without checking which would make considering the original configuration a worthless act. The same logic is applied to distance but does not translate to approach gradient. The glide slope of a non-precision procedure is bounded (see Chapter 3) but the value is not “flown” as in a precision approach. It is a measure of how aggressive a procedure’s descent is and therefore must be considered for any attempt to optimize a search.

Accepting the discretization assumption, the depth of the approach’s tree/graph is limited by number of segment types but exponential growth is experienced at each level. The tree’s expansion attributable to any node can be generalized by the following formula:

$$n = \int_{g\,min}^{g\,max} (2\triangle h + 1)[(l_{max} - l_{min}) / l_i + 1] \qquad (4.1)$$

Where g is the gradient, d is the amount of heading change allowed at the given fix (in whole degrees), l_{min} and l_{max} are the minimum and maximum lengths of the segment being constructed and l_i is the discrete distance interval. The scalar value of 2 indicates that the heading change can be either right or left. For example, consider the contribution of a node that is a Final Approach Fix. A heading change of 60 degrees is allowed at a fix of this type and the minimum and maximum lengths for an intermediate segment are 2.0 nm and 5.0 nm respectively. Using the discrete interval of 0.10 nm as previously explained, each Final Approach Fix node adds 3720 Intermediate Approach Fix nodes to the tree. Applying equation 4.1 (with no consideration to the integral over g) results in over 31 million possible approach configurations for the inbound segments (i.e. Initial, Intermediate and Final).

Consider even a discrete application of gradient with a reasonable interval of 10 feet. The minimum gradient is 300 ft/nm and the maximum is 600 ft/nm. A minimum length segment results in a factor of 61 and increases to 301 for segments with a maximum length of 10.0 nm. Such growth is the motivation to derive a method to reduce the number of nodes introduced to the open list.

B. PROPAGATING A* (PA*)

PA* differs from classic A* in its ability to handle unknown information. Traditionally, this implies a dynamic environment or real-world path planning problems (see Chapter II.B). In this context, “unknown” refers to the region of the complete graph between the frontier of the search graph (the explored area of the complete graph) and the goal. The following sub-section introduces a novel approach to reduce to the number of nodes introduced to the open list. The particular area addressed is lines 12 through 15 of the A* code.

```
12          for all successors of curNode
13              successor.parent = curNode;
14              successor.setCost(curNode);
15              open.add(successor);
```

Figure 9. Successor Methods in A*.

1. Successor Generation

At line 12 in the code above, the current node (node A) has been popped off the open list and determined to be a candidate for expansion. Let B be the set of all successors of A and $B_{x,y}$ be the subset of B at a given heading and distance. $B_{x,y}$ then contains all the successors of A sharing a common (x,y) coordinate pair but different gradient values. Regardless of the discrete interval over the gradient, there is only one collision check required for each successor subset $B_{x,y}$ if a feedback collision detection method is used. It is clear from Figure 10 that the required clearance is a function of node A.

This knowledge suggests a check over the segment $AB_{x,y}$ can be used to establish a lower bound on the gradient range for $B_{x,y}$. This is true, but in the worst case (the absence of any obstacles) there will be no reduction in the number of successor nodes added to the open list. An alternative strategy, that is similar to that implemented by Adaptive A*, is to only introduce the node at the lowest gradient. However, problems arises at the successor call of $B_{x,y}'$ (the single node representation of $B_{x,y}$) if there is an obstacle blocking some or all of the successors in the set C.

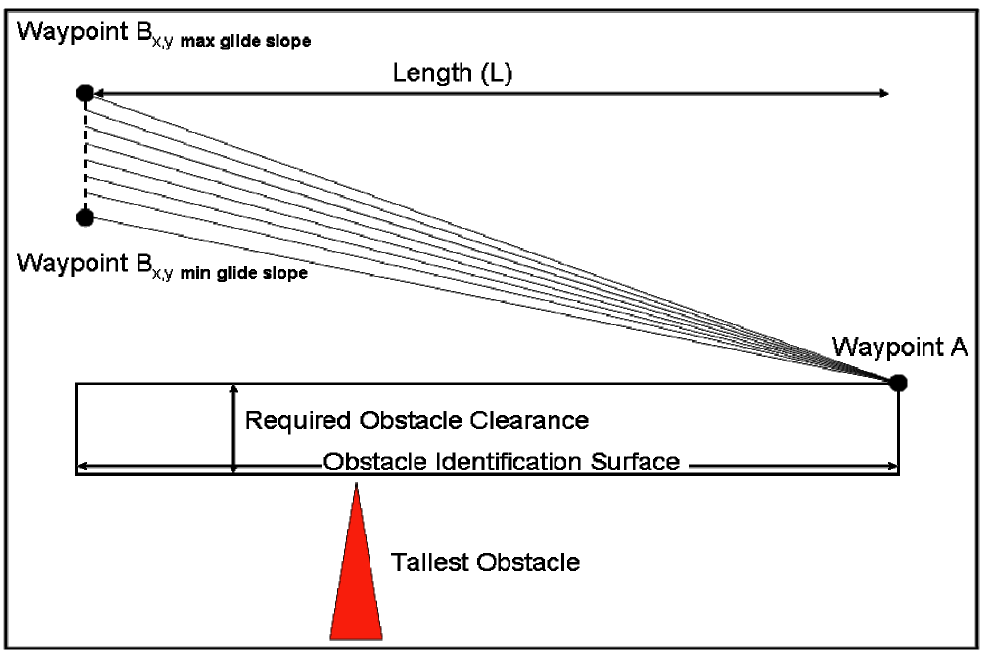


Figure 10. The Discrete Subset $B_{x,y}$.

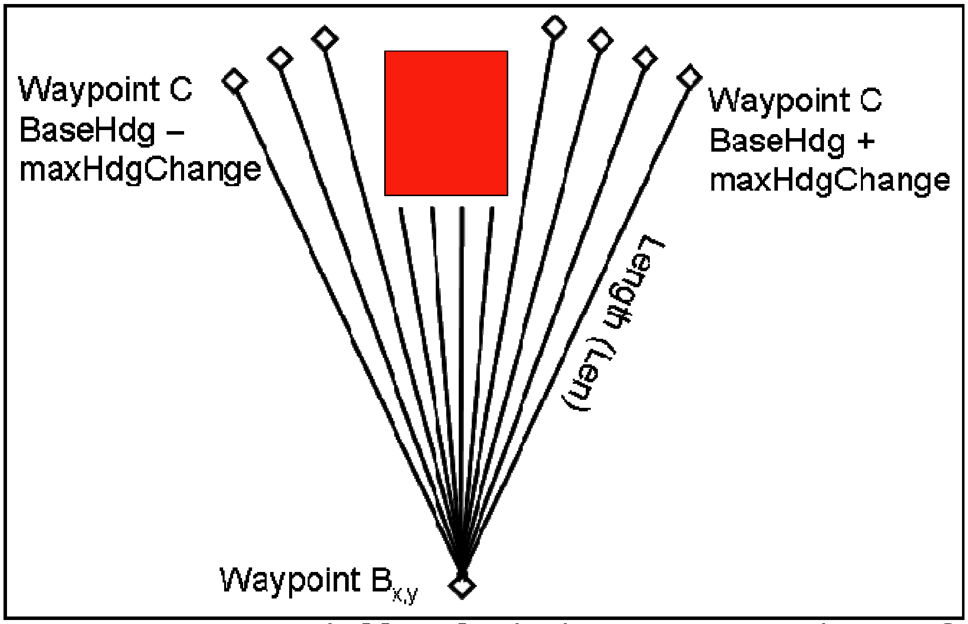


Figure 11. Partially Blocked Successor Subset of C.

The corrective action in a scenario where all successors are blocked is to simply apply the Open Sky policy retroactively to the parent node and increase its altitude to clear any obstacles (see Figure 12). Care must be taken that the new height of $B_{x,y}''$ does not violate the maximum gradient of the previous segment. If this ever happens the node is be removed from consideration as a valid path (see further works for a potential corrective measure).

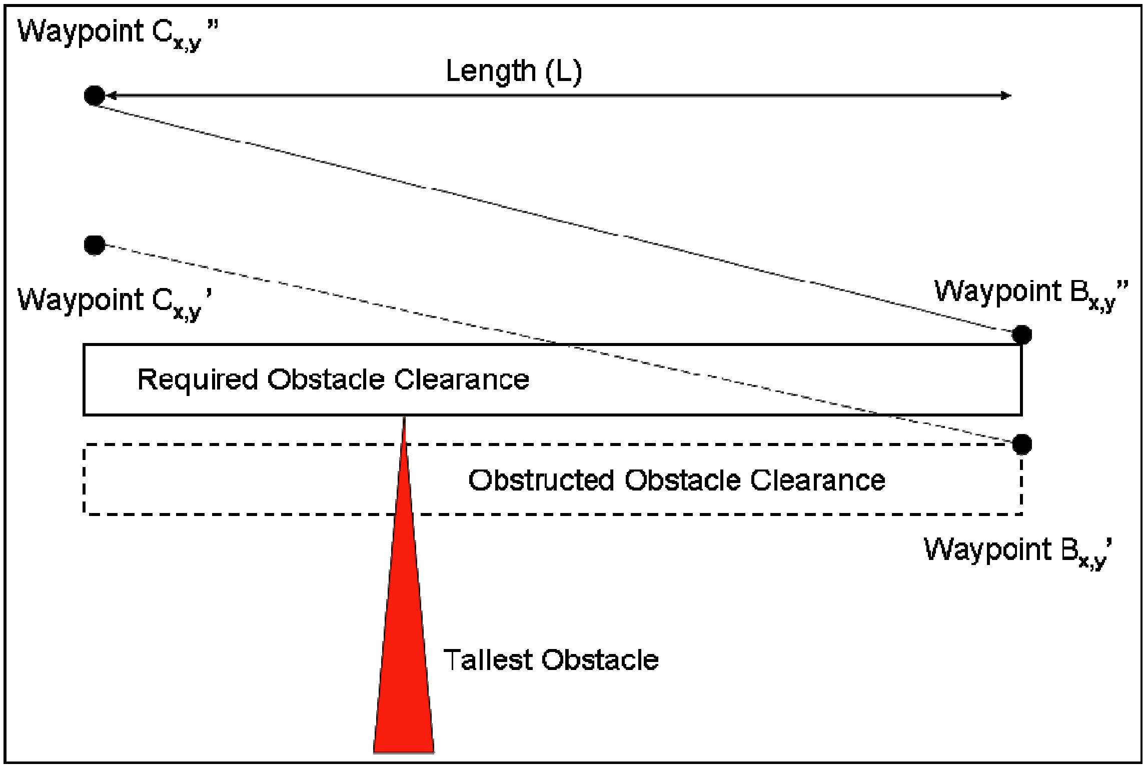


Figure 12. Retroactive Application of the Open Sky Policy.

The partially blocked case is not so easy. The seemingly obvious solution is to place the $B_{x,y}'$ node back in the open queue with a newly adjusted height value. There are three fundamental flaws with this solution that make it unacceptable: redundancy, incompleteness and retrieving the solution path. Assume a systematic left-to-right expansion of Figure 11.

1) $B_{x,y}'$ only failed during the creation of the middle successors. By placing it back on the queue, if it is popped again, all of the valid successors on the left side will be re-created at new height value. The original nodes will have a lower altitude[10] and be popped first, but if

[10] Consideration of the impact of altitude on cost is addressed in the next section.

this occurs multiple times or during the development of an early segment, the ripple effect can cause a reduction in performance.

2) Another problem is that none of the potentially valid successors on the right side will be realized at their best altitude - when $B_{x,y}'$ was popped the first time - and completeness is lost. Just considering obstructed nodes as invalid paths is not satisfactory either for the same reason.

3) Recovering the solution path containing a node that has had its height modified in this fashion will not be possible. Since the parent node was changed, the retrieved path will reflect the adjustment and not be the actual position reflected in the optimal cost value.

A completely different solution can be constructed by combining those already presented. If during expansion a collision is encountered, replicate the parent node with the new height and discard those path nodes that are invalid. This answer solves the completeness and path recovery issues, but actually makes the redundancy problem worse by duplicating both the left and right sides. It also allows the possibility that there might be multiple obstructions experienced in which case the flaw is repeated. It is obvious at this point that neither of these choices satisfactorily fixes the problem.

The PA* algorithm present in this thesis handles this situation by introducing a node variable. During collision detection, feedback is provided indicating the amount of any height adjustment required. A non-zero value indicates a violation was detected and it is stored in the offending successor. The successor, in turn, uses the height node variable in the setCost method call (line 14) and during it own successor generation to reflect the altered height. If the optimal path contains a modified node, the variable is again used to correctly position the affected waypoints during the solution path retrieval process.

2. Cost Function

Line 14 of Figure 9 illustrates where the cost function is used to establish the priority of a node in the open list. Equation 4.2 shows the classic A* cost function in its general form:

$$f = g + h \tag{4.2}$$

The variable *g* reflects the incurred cost of traversing the graph from the startNode to the curNode and *h* is the heuristic-based estimation of the remaining cost to get from curNode to the goalNode.

The edge cost of a node in the approach planning problem can be defined by the three degrees of freedom found in an instrument procedure: 1) change in heading required from the inbound course to the parent node and the outbound course to the successor node, 2) distance between the parent node and the successor node, and 3) the descent gradient between the successor node and the parent node. A fourth is added to indicate any incurred height adjustment described in the previous section and the fifth term reflects the proximity of completion (i.e. a higher value indicates and earlier segment). The behavior of the function is manipulated by adding a scalar constant to each term in the cost function. Using $\triangle x$ as change in heading, l as length, a as approach angle (gradient), z as required adjusted height and n is the number of remaining, g is given in the following equation:

$$(4.3)\quad g = \triangle x + l + a + z + n$$

Thought must be given to the nature of the variables in equation 4.3. The variable x is bounded by 0.0 and 180.0 degrees and the variable a is bounded by approximately 2.8 and 5.6 degrees[11] whereas l can take a value between 2.0 and 10.0 nm depending on the segment being constructed. To make the terms more intuitive to work with they are normalized using the following:

$$(4.4)\quad x_{normalized} = abs[(x_{current} - x_{optimal})/(x_{\max imum} - x_{optimal})]$$

For example, let x be the l-value for an Intermediate segment with a length of 3.9 nm (Note: $X_{optimal}$ and $X_{maximum}$ for the length of an Intermediate segment are 3.0 nm and 5.0 nm respectively). In this example, equation 4.4 gives:

$$Abs[(3.9 - 3.0)/(5.0 - 3.0)] = 0.45$$

Since there is no optimal number of segments, the last term is normalized using the total number of segments required. Applying equation 4.4 has the added benefit of naturally weighting each term so more drastic deviations from the optimal path incur a higher cost and, therefore, are less desirable.

The topic of cost concludes with the admissibility proof of the heuristic function. For a heuristic function

[11] These values represent the minimum and maximum gradients converted to degrees for the sake of discussion.

to be admissible, A* requires it never over-estimate the actual remaining cost from the successor to the goalNode. This poses some problems because the segment to the notional en route waypoint does not have optimal values like the procedure segments. Simply considering the straight line distance is insufficient because this value will always be non-zero. Consider the case of estimating the remaining cost from a MAP on the optimal path. Equation 4.3 will have all but the last term equal to zero, but a straight line estimated remaining cost will never be zero. This is a problem if the procedure is extended along the optimal path to the IAF. The actual cost to the en route point will be the normalized distance to the fix using two times the size of the terminal area as the numerator. But, if this path is the optimal path then the IAF is certainly closer than the MAP and the estimated cost at that point exceeds the total incurred cost because all of the procedure segments had a value of zero and the remaining segments term is unable to over come this.

To overcome this issue, the implemented heuristic function only uses the first and last terms from equation 4.3.

$$h = \vartriangle x + 2(n-1) \tag{4.5}$$

By normalizing $\vartriangle x$ with 180 degrees, the term is guaranteed to never be larger than any segment’s because the maximum deviation is 120 degrees. The larger numerator ensures a smaller term value. A scalar of two is added to the n-term based on performance tests described in the next section. This does not violate the optimistic estimate because:

$$(4.6) \quad 2(n-1) < \sum_{i=1}^{n} i$$

C. SYSTEM DESIGN

The FAA’s Instrument Approach Procedure Automation (IAPA) software is the ideal environment to test the algorithm presented in the previous section. Unfortunately, access to an IAPA workstation was not possible and an analogue, the Instrument Procedure Generation Test Environment (IPGTE), had to be built. The design philosophy used was to only build what would be required of the validation process without impacting the behavior of the planning problem. The strategy for development was to maximize the use of existing software tools and libraries in order to minimize new code generation. A spiral development methodology was employed to facilitate rapid prototyping and an object-oriented software design was used to enable making modifications to individual components. This last decision was critical when iterative testing exposed weakness in a particular implementation decision.

The system design is straightforward. The major components are the user interface, approach model, terrain model and path planners; the latter having been discussed in the previous section.

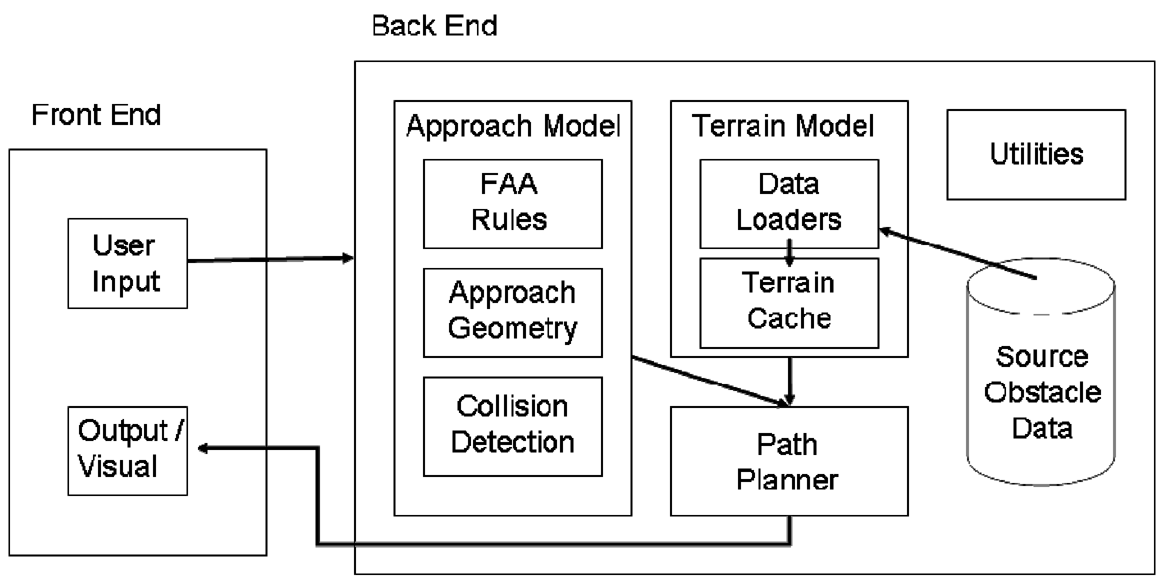


Figure 13. System Diagram.

1. Input/Output

A minimalist approach was taken on the user interface. Input consists of just a desired landing location. The system provides a text-based output containing search status (i.e., segment being evaluated, edge costs, etc.) and a visual representation of the solution and terrain model using the Open Scene Graph library. The visualization is critical for a "birds-eye" appraisal of the procedures quality as well as validating construction and obstacle avoidance.

2. Terrain Model

Deliberation with respect to the physical model was more straight-forward. The model represents the real world and obstacles that must be negotiated while planning the approach path. The decision was made to only use terrain data for two reasons. The first is based on the fact that the nature of an obstacle does not impact the requirement to avoid it. The second reason is that the number of man-made obstacles is negligible compared to the density terrain data.

Digital Terrain Elevation Data (DTED) was used as the source data because it is readily available online and is the same data used by IAPA. [48] provides DTED specifications and file format. The use of this data source presented issues related the coordinate system and data density for the system design. A DTED file provides data in a raster file with evenly spaced posts and is available in different levels of detail. For example, a level one DTED file provides data coverage of one arc degree with a post spacing of 3 arc seconds (approximately 90 meters). The use of the geographic coordinates system is not suited for distance calculations required by the IPGTE because of the longitudinal convergence (i.e. the distance defined a longitudinal interval decreases the closer it gets to the North or South pole). The Universal Transverse Mercator (UTM) coordinate system provides a better alternative due to its regular grid structure and the unit of measure (meters) is consistent with the data post values in a DTED file. The Geo-spatial Data Abstraction Library (GDAL) is used to handle to importing DTED data and source code provided by [49] is used for conversion. [50] provides an in-depth discussion about coordinate reference systems and conversion methods.

The paragraph above states that a level one DTED (level one) file provides data coverage of one arc degree with a post spacing of 3 arc seconds. This translates to 1,442,401 data points[12] (assuming only one DTED file is required which is rarely the case). For this reason, a terrain caching scheme is used that loads all the terrain data for the terminal area and a relative position reference method (with respect to the landing zone) is used to store it for real-time access. When a collision check is required, the relative position of the segment being considered is calculated and data contained in rough bounding box area is returned.

3. Approach Model

As was described in the previous chapter, an approach is a series of segments analogous to a robotic armature and its complexity is defined by the three degrees of freedom. This means details like turn anticipation areas and intra-segment letdown fixes can be omitted since they alter the segment’s physical construction but do not contribute the complexity of the planning problem. The decision was also made not to include the missed approach segment in the approach model. This segment’s contribution to the problem’s complexity is scalar in nature (i.e. an additional segment) so its omission was deemed acceptable for the sake of decreasing system complexity.

The final discussion point concerns the method used for the actual collision detection. The section explaining the algorithm implemented mentions using the feedback from a collision check. This is accomplished by constructing polyhedra and checking the terrain data to see if the

[12] This calculation takes into account the 3 arc seconds of overlap on two sides to provide for continuous coverage.

polytope contains any of the data points. If a point is contained, the amount of penetration is calculated and checked to determine if it exceeds the current maximum value. When all testing is complete the penetration value is then returned indicating the amount of vertical adjustment required to ensure an obstacle free path.

THIS PAGE INTENTIONALLY LEFT BLANK

V. IMPLEMENTATION AND TESTING

A. HARDWARE

All testing was conducted on a Dell Inspiron 9300 laptop computer running Windows XP Professional Edition. The system is configured with a 1.6 GHz Intel Pentium M process, 1.0 G of RAM, ATI Mobility Radeon X300 graphics card and a 30 G internal hard drive.

B. TESTING

Testing was conducted in two stages. The first stage consisted of establishing the performance baseline in an unobstructed environment. The location used for the intended point of landing is near the small town of Turlock in central California at the coordinates north 37 degrees 30 minutes latitude, west 120 degrees 45 minutes longitude. DTED level 0 data was used to reduce the computation time but limited testing showed consistent results with solutions obtained with DTED level 1.

The first step of the baseline testing consisted of exploring increasingly larger state graphs with no coefficient applied to (n-1) term in the heuristic function. The state graph was systematically enlarged by increasing the amount of heading changes that were allowed at each waypoint along the approach path. The length of the Visual Segment was held constant at 0.5 nm and the remaining approach segments were limited to three values: 3.0 nm, 4.0 nm and 5.0 nm. These values represent the range of the Intermediate Segment with one nautical mile intervals. The reduction was done to maintain a graph size that could easily be exhaustively explored in a reasonable amount of time while still providing some variation.

The second step was to apply the coefficient to the nodes remaining term and compare the results against the baseline. In both steps, the search results were compared to a test where the FAF and the IF were only allowed one degree of deviation. This comparison provided insight to the effect of tree branching on the search performance.

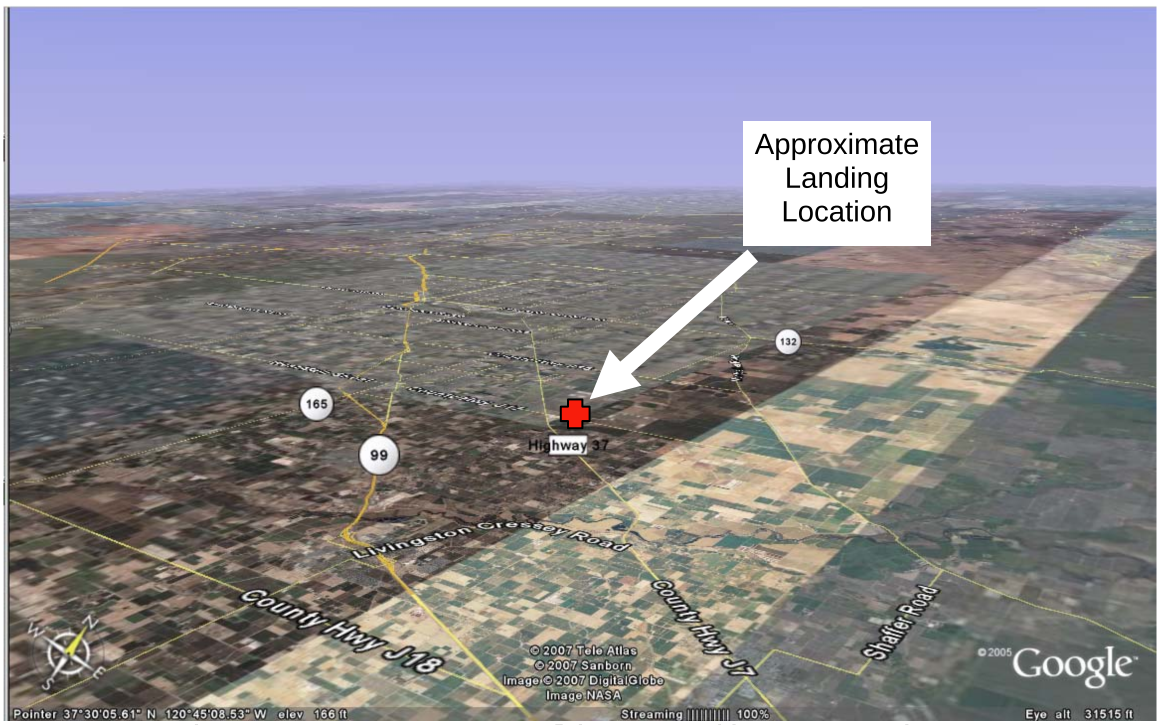


Figure 14. Baseline Landing Location.

The last phase was to evaluate the behavior of the algorithm. The location used was north 36 degrees 24 minutes latitude, west 121 degrees 20 minutes longitude; just off highway 101 near the town of Soledad, California. The area was selected for its channeling terrain which is a good environment to test the algorithm’s ability to find more suitable routes when approached from the side. Further, the presents of a major highway is a makes it a more likely location for a HEMS request.

Testing in the evaluation phase consisted of generating multiple approach procedures to the intended point of landing with the en route node at different positions. The goal was to see if the algorithm could de-conflict the approach and terrain in an acceptable fashion – meaning minimal intra-procedure turning and offset of the Initial Approach Fix.

Figure 15. Evaluation Test Landing Location.

C. RESULTS

1. Baseline Testing

The first table shows the schedule for increasing the size of the graph to be searched followed by the individual test results. The column values reflect the amount of course deviation allowed from the desired heading at each waypoint. The total number of possible nodes that would have to be considered in an exhaustive search can be calculated using the following equation:

$$(5.1)\quad N = \sum_{k=1}^{5} [N_{k-1} * (2\theta_k + 1) * n_{l,k}]$$

Where, $N_0 = 1$, θ_k is the amount of variation the k-th waypoint can be approached from and $n_{l,k}$ is the number of valid lengths for the segment being considered. Remember that $n_{l,1}$ = 1 and $n_{l,2 through 4}$ = 3 as stated in the second paragraph of this section. Further, the IAF only adds one segment, directly to the goal, with no limitation on length making θ_5 = 0 and $n_{l,5}$ = 1.

LZ	MAP	FAF	IF	IAF	Possible Edges
30	0	5	10	0	1061888
60	0	10	20	0	7682048
90	0	15	30	0	25044608
120	0	20	40	0	58333568
150	0	25	50	0	112732928
180	0	60	120	0	758983680

Table 2. Course Deviations and Total Edges.

LZ	MAP	FAF	IF	IAF	Expanded Nodes	Open List
30	0	5	10	0 < 120	385	12538
60	0	10	20	0 < 120	1103	69668
90	0	15	30	0 < 120	2244	195074
120	0	20	40	0 < 120	3784	399668
150	0	25	50	0 < 120	5698	677762
180	0	60	120	0 < 120	5501*	1812313*

Table 3. Expanded Nodes (w = 1.0).
*incomplete search due to memory failure

LZ	MAP	FAF	IF	IAF	Expanded Nodes	Open List
30	0	1	1	0 < 120	293	1758
60	0	1	1	0 < 120	521	3106
90	0	1	1	0 < 120	718	4250
120	0	1	1	0 < 120	858	5118
150	0	1	1	0 < 120	952	5826
180	0	1	1	0 < 120	1066	6414

Table 4. Expanded Nodes (w = 1.0, limited deviation).

LZ	MAP	FAF	IF	IAF	Expanded Nodes	Open List
30	0	5	10	0 < 120	63	956
60	0	10	20	0 < 120	107	2955
90	0	15	30	0 < 120	149	5844
120	0	20	40	0 < 120	191	10164
150	0	25	50	0 < 120	233	15680
180	0	60	120	0 < 120	403	51927

Table 5. Expanded Nodes (w = 2.0).

LZ	MAP	FAF	IF	IAF	Expanded Nodes	Open List
30	0	1	1	0 < 120	63	324
60	0	1	1	0 < 120	105	560
90	0	1	1	0 < 120	141	772
120	0	1	1	0 < 120	169	945
150	0	1	1	0 < 120	195	1113
180	0	1	1	0 < 120	211	1229

Table 6. Expanded Nodes (w = 2.0, limited deviation).

As would be expected, given there were no obstacles, all solutions were the same and were direct paths to the goal. It is clear from the data, the coefficient w = 2.0 out performed the base case with w = 1.0. The reason for this is understood by comparing tables five and six. The last table shows the case where a negligible amount of course deviation is allowed producing only straight configurations. Even though the state graph is reduced from its full potential, the size of the search graph posted in Table 5 is very similar. The close proximity in the results suggests the majority of the search occurred within one degree of the base heading. A major difference, however, is seen in the number of nodes that were introduced to the open list. In Table 5 (row six), less than 0.8 percent of the total number of nodes was used to find the solution. Table six stabilized to approximately 17 percent by the fourth row. It must be noted that the efficiency of the straight-line search scheme comes at the cost of completeness.

2. Evaluation Testing

In this phase, the intended point of landing is approached from four different directions: two from opposing ends along the valley’s axis (340 and 130 degrees) and two from perpendicular headings (060 and 250 degrees). The purpose of the first two tests was to see the growth of the search graph along a path known to contain a solution, but possessing obstacles that would force the algorithm to consider adjusted heights and invalid segments.

The tests were conducted in a similar fashion to the baseline with one exception. The previous findings inspired the concept of smaller but constant ranges for course deviations at the FAF and the IF waypoints. 1, 5 and 10 degrees of course deviation were examined as well as the full range allowed by the procedure specifications. A remaining node coefficient of 2.0 was used for all trials.

LZ	MAP	FAF	IF	IAF	Expanded Nodes	Open List
180	0	1	1	0 < 120	189	1053
180	0	5	5	0 < 120	189	2421
180	0	10	10	0 < 120	189	4131
180	0	60	120	0 < 120	381	43963

Table 7. Expanded Nodes (Heading 340).

LZ	MAP	FAF	IF	IAF	Expanded Nodes	Open List
180	0	1	1	0 < 120	208	1205
180	0	5	5	0 < 120	208	3029
180	0	10	10	0 < 120	208	5309
180	0	60	120	0 < 120	400	50841

Table 8. Expanded Nodes (Heading 130).

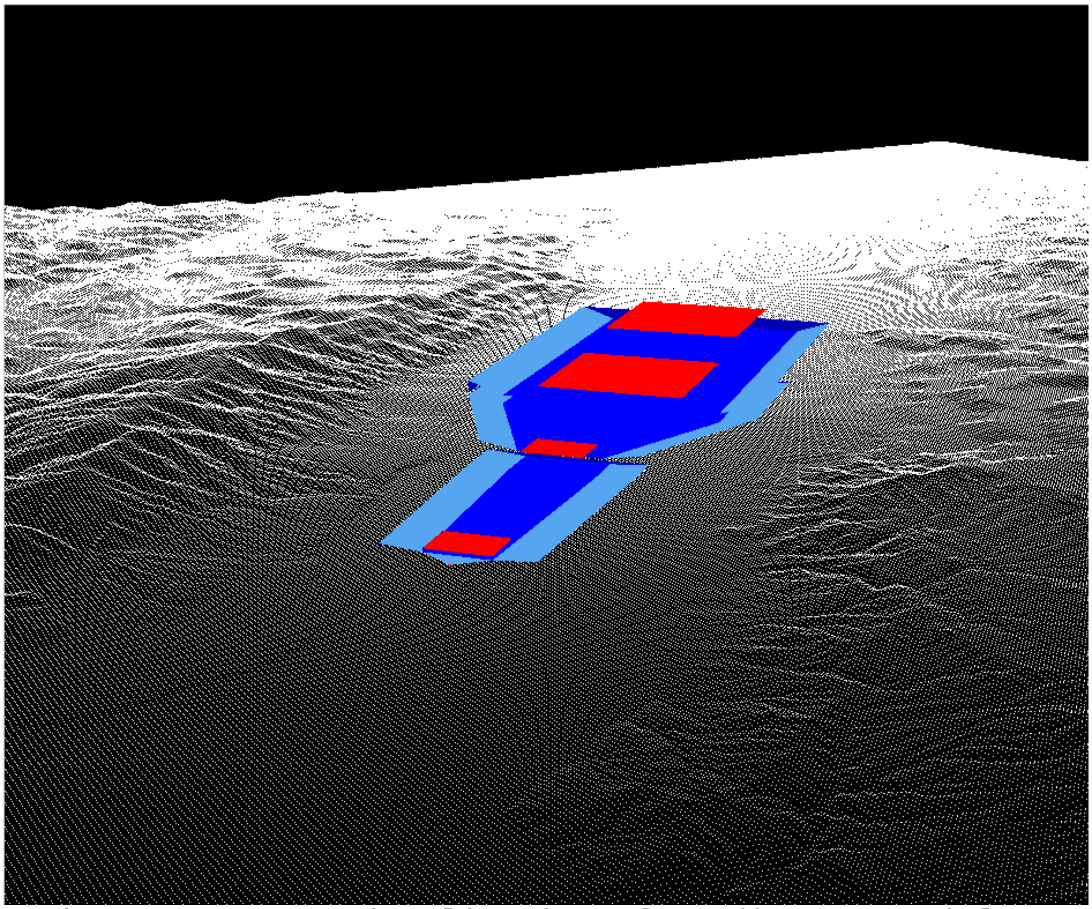


Figure 16. Visualization of Heading 130 Trials.

An interesting thing can bee seen in these tests that gives additional insight to the behavior of the algorithm. Notice that the first three trials in each experiment yielded the same number of nodes expanded before finding the solution. This tells us that the additional nodes expanded in the fourth trial in each case came from expanding FAF and IF waypoints and not from exploring different approach course from the landing fix. The effect of this can be see in the open list growing by ten-fold with only approximately twice the nodes expanded yet the solutions remained the same within each test.

All of the experiments have generated straight procedures so far. This is not all together surprising given the terrain, or absence thereof for the baseline tests. The true assessment of the algorithm's ability to find a suitable approach path is when it must negotiate obstacles. This is considered during the trials with a perpendicular heading.

LZ	MAP	FAF	IF	IAF	Expanded Nodes	Open List
180	0	1	1	0 < 120	495	2081
180	0	5	5	0 < 120	564	6417
180	0	10	10	0 < 120	849	15575
180	0	60	120	0 < 120	22408	476355

Table 9. Expanded Nodes (Heading 250).

LZ	MAP	FAF	IF	IAF	Expanded Nodes	Open List
180	0	1	1	0 < 120	254	983
180	0	5	5	0 < 120	307	3431
180	0	10	10	0 < 120	509	9225
180	0	60	120	0 < 120	17744	375115

Table 10. Expanded Nodes (Heading 060).

The figure below illustrates the solution returned in the experiment with a desired heading of 240 degrees. As can be seen, the terrain is prohibitive for the first three segments and a heading of 197 degrees is held until enough altitude is gained allowing for the last segment to turn to the desired course. The trials on a heading 060 degrees had similar results. These results were only attainable in the last trials when the procedure was afforded full flexibility.

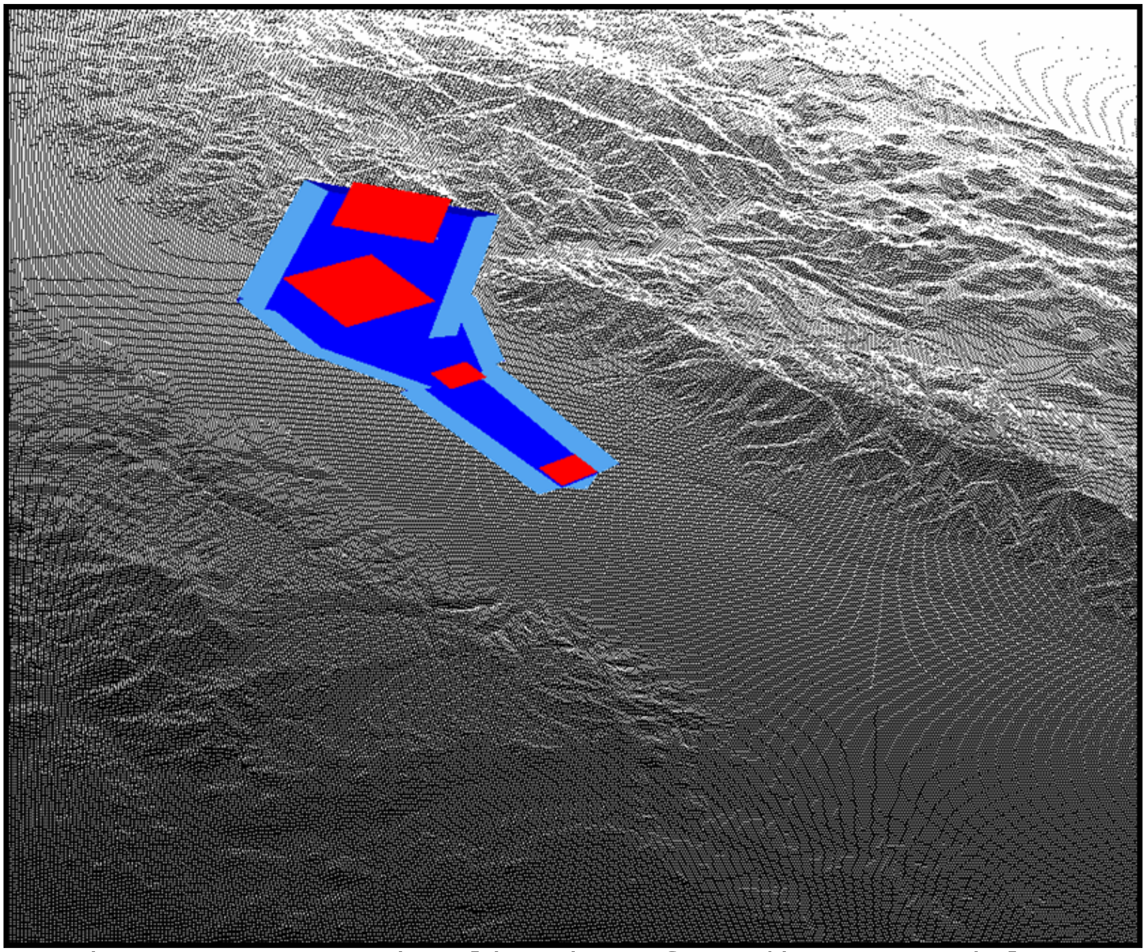


Figure 17. Visualization of Heading 250 Trials.

The data shows a significant increase in the nodes expanded from previous tests; however, the efficiency between restricted and un-restricted (full) expanding is actually more consistent. The following table provides a comparison between the axis-aligned and perpendicular procedures. This suggests the need for some pre-processing of the terrain to determine it nature. If it is determined to be flat or the desired heading is aligned with a liner obstacle feature, the restricted heading planner is more suitable. When this is not the case, full expansion provides the completeness that may be required. The computation time and memory requirement can become a problem but this is not an issue with most realistic landing sites.

Axis Aligned	Expanded / Open List (%)			Growth (Trial 4 / Trial 3)
Heading 340			Heading 340	
Trial 3	0.0458		Expanded Nodes	2.016
Trial 4	0.0087		Open List	10.642
Heading 130			Heading 130	
Trial 3	0.0392		Expanded Nodes	1.923
Trial 4	0.0079		Open List	9.576
Perpendicular	Expanded / Open List (%)			Growth (Trial 4 / Trial 3)
Heading 250			Heading 250	
Trial 3	0.0545		Expanded Nodes	26.393
Trial 4	0.0470		Open List	30.585
Heading 060			Heading 060	
Trial 3	0.0552		Expanded Nodes	34.861
Trial 4	0.0473		Open List	40.663

Table 11. Performance Test Comparisons.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION AND FUTURE WORKS

A. CONCLUSIONS

This thesis brings to light the problem of providing IFR support to rotary wing aircraft. Historically, there was very little to could be done to facilitate the unique capabilities of these aircraft given the operating restrictions of radio navigation aids. Further, the limited use of rotorcraft did not dictate much demand for such specialized support.

Things changed in the mid-1980s. Helicopters became relied upon more heavily for critical operations such as emergency medical services as well as traditional military roles. The absence of suitable IFR support in the NAS became apparent with an alarmingly high fatality rate amongst commercial on-demand rotorcraft services prompting an investigation by the NTSB. In addition to the agency's statement about the dangers posed by weather related operations, studies by Rick Frazer and Ira Blumen presented in the thesis introduction have provided continuing statistical evidence that inadvertent instrument meteorological conditions remains the number one killer amongst the rotary wing community.

Nearly two decades later, the FAA has still failed to properly address the issue despite their awareness of the situation and the implementation of precise GPS navigation. The "anywhere" capability of GPS makes it the ideal technology for point-in-space approach navigation aid, however, IAPA does not even contain the approach models required for the normal processing of helicopter approaches let alone on-demand procedure generation. The work presented here is a good first step towards accomplishing this.

The approach procedure planning problem itself is not unsolvable though there are some domain specific issues that need to be addressed. A reasonable discretizing scheme was introduced to address the continuous nature in two of the three degrees of freedom (length and heading) for each path segment. This technique reduces the any search to being only resolution complete in two dimensions, however, any lose in precision is beyond the capabilities of manned flight and deemed acceptable.

The continuous nature of an approaches glide slope is addressed by the Propagation A* algorithm presented in Chapter IV. The problem space of an instrument procedure allows for the continuum of the joint to be evaluated in a single collision test and represented in a single state graph node. The issue of successor nodes changing the state (and cost) of parent nodes is a problem with the classic A* search algorithm. It is covered here by propagating parental state change information to the successor. Pushing this information forward is essential because each successor may impose a unique state changes on the parent, some of which may invalidate other sibling nodes.

Tests conducted showed sound cost and heuristic functions have been developed and most solutions were identified with fewer than 500 search steps. The normalizing of cost terms provided a natural weighting scheme that focused the PA* search to straight paths. This is critical because the state graph growth is exponential. In a challenging test, the algorithm was still able to find paths that deviated from desired course and required turns but the size of the search graph was considerably larger.

B. FUTURE WORK

The results can only be considered an indication of potential performance because the experiments were conducted with incomplete models. However, valuable insight was gained and areas for future development are bountiful. The most important thing required for the next step in research is to gain access to an IAPA workstation. This is not prohibited by the government and the cost is estimated to be less than ten thousand dollars. Though the system developed for this research was sufficient for a first effort, it is sorely lacking in construction and obstacle completeness.

An evaluation tool for the terminal area is the logical next step. Testing showed there can be significant performance gains if the nature of the local obstacle can be determined. This information can be used to tailor the terminal area search. One such strategy that was considered during development but not implemented was to identify to the most promising heading based on the minimum obstacle height. The terminal environment was broken down into 36 ten-degree slices and each slice was evaluated based on the minimum gradient required to clear all obstacles.

Another area for future research is to develop a method for connecting the approach procedure to the existing airway structure to include considering established en route waypoints as the goal node. The use of a notional en route waypoint is sufficient for the generation of an arbitrary approach. Real world applications will be more constrained and this connection between the terminal and en route environments is a critical link.

The propagating algorithm has room for improvement as well. Given the performance and behavior of PA*, implementation of the partial-expansion scheme introduced by [30] looks very promising. In the tests conducted in the previous chapter, the data showed that the open list still contained about 95 percent of the nodes when a solution was identified. By iteratively expanding a node from the desired heading out to the maximum deviation allowed, the memory constraints of PA* can be drastically reduced. This was demonstrated in the tests that restricted course deviations. Optimality will be lost but performance will be greatly enhanced and such an algorithm will retain resolution completeness.

Retention of optimality may be possible through the use of multi-processor systems. By dividing the problem into approach sectors, the airspace should be able to be exhaustively searched in near real-time. This may be required when complete approach and obstacles models are used. Further, this thesis only considers terminal approaches but departure procedures are required as well. A multi-processor technique can be used to divide the flight profile into the constituent phases and tackle each separately.

Human factors need to be explored as well. Each degree of freedom in an approach procedure is bounded by maximal value (i.e. length, gradient and heading change). The development of complex procedures can be valid but overwhelm the pilot and aircrew. Research is needed to guide procedure development such that human constraints are not exceeded. Procedure displays might be able to be integrated with terrain awareness warning systems to reduce pilot work load.

Required research beyond the scope of this thesis but critical to implementing a resources as described here are the myriad of policy issues. The FAA and the Department of Transportation are understandably strict on their authorization of commercial aviation technologies. In addition to identifying basic administrative procedures that would be required for system implementation, more robust operations research is needed. Specifically, Federal Aviation Regulation part 135 needs to be considered for revision and guidelines need to be established to guarantee integrity of terrain and obstacle data used in procedure generation. Naturally, cost considerations need to be explored as well.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] R. S. Frazer, “Air medical accidents: a 20-year search for information,” Air Medical Journal, vol. 5 No. 5, pp. 34-39, 1999.

[2] R. S. Frazer, “Weather accidents and the air medical industry,” Air Medical Journal, vol. 6 No. 3, pp. 49-53, 2000.

[3] R. S. Frazer, “Air medical accidents involving collisions with objects,” Air Medical Journal, vol. 20 No. 5, pp. 13-16, 2001.

[4] R. S. Frazer, “Air medical accidents attributed to maintenance,” Air Medical Journal, vol. 21 No. 3, pp. 17-21, 2002.

[5] I. J. Blumen, "A Safety Review and Risk assessment in Air Medical Transport," Supplement to the Air Medical Physicians Handbook, Salt Lake City, UT: Air Medical Physician Association, 2002.

[6] J. Goh and D. Wiegmann, "Visual flight rules (VFR) flight into instrument meteorological conditions (IMC): a review of the accident data," presented at the 11th International Symposium on Aviation Psychology, Columbus, OH, 2001.

[7] National Transportation Safety Board, "Commercial emergency medical service helicopter operations," National Transportation Safety Board, Washington, D.C., Tech. Rep. NTSB/SS-88-013, 1988.

[8] National Transportation Safety Board, "Special Investigation Report on Emergency Medical Services," National Transportation Safety Board, Washington, D.C., Tech. Rep. NTSB/SIR-06/01, 2006.

[9] Federal Aviation Administration, "Rotorcraft master plan," Federal Aviation Administration, Washington, D.C., Tech. Rep. FAA-P-8100.1, September 1987.

[10] R. K. Anoll, R. B. Newman, E. D. McConkey, "Rotorcraft low altitude IFR benefit/cost analysis: conclusions and recommendations," Systems Control Technology Inc., Arlington, VA, Tech. Rep. SCT-92RR-8, October 1993.

[11] D. F. Wourms, S. J. Johnson, J. A. Ogden and T. R. Metzler, "Human factors in vertical flight simultaneous non-interfering operations," Human Systems Information Analysis Center (HSIAC), Wright-Patterson AFB, OH, Tech. Rep. HSIAC-SS-2001-003, August 2001.

[12] J. B. Mulligan and Brolly, Xavier L. C., "Pilot behavior and course deviations during precision flight," in Proceedings of the Human Vision and Electronic Imaging, Vol. 5666, pp. 362-373, March 2005.

[13] D. F. Wourms, S. J. Johnson, J. A. Ogden and T. R. Metzler, "Human factors in vertical flight simultaneous non-interfering operations," Human Systems Information Analysis Center (HSIAC), Wright-Patterson AFB, OH, Tech. Rep. HSIAC-SS-2001-003, August 2001.

[14] A. Wass, (private communication) 2006.

[15] J. H. Reif, "Complexity of the Mover's Problem and generalizations," in Proceedings of the 20th Symposium on the Foundations of Computer Science, pp. 421-427, 1979.

[16] Pobil, Angel Pasqual del. and M. A. Serna, *Spatial Representation and Motion Planning*. New York, NY: Springer-Verlag Inc, New York, 1995.

[17] P. Shirley, Fundamentals of Computer Graphics, 2nd Edition ed. Wellesley, MA: A K Peters Ltd., 2005.

[18] C. Ericson, Real Time Collision Detection. San Francisco, CA: Morgan Kaufmann, 2005.

[19] T. Lozano-Perez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," Communications of the ACM, Vol. 22, pp. 560-570, 1979.

[20] T. Lozano-Perez, "Spatial planning: a configuration space approach," IEEE Transactions on Computers, vol. C-32, pp. 108-120, 1983.

[21] S. M. Udupa, "Collision detection and avoidance in computer controlled manipulators," Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1977.

[22] Y. K. Hwang and N. Ahuja, "Gross motion planning - a survey," ACM Computing Surveys, Vol. 24, pp. 219-291, 1992.

[23] R. S. Alexander, "Construction of optimal paths for homogeneous-cost-region path planning problems," Ph.D. thesis, Naval Post Graduate School, Monterey, CA, September 1989.

[24] N. Kalra, D. Ferguson and A. Stentz, "Incremental reconstruction of generalized Voronoi Diagrams on grids," in Proceedings of the Int'l Conference on Intelligent Autonomous Systems, pp. 114-123, IOS Press, 2006.

[25] J. Canny, "The Complexity of Robot Motion Planning" Cambridge, Mass: MIT Press, 1988.

[26] L. Kavraki, P. Svestka, J. Latombe and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Transactions on Robotics and Automation, Vol. 12, No. 4, Aug 1996, pp. 566-579.

[27] B. Faverjon, "Obstacle avoidance using an octree in the configuration space of a manipulator," in Proceedings of the IEEE Int'l Conference on Robotics and Automation, pp. 504-512, 1984.

[28] F. Lingelbach, "Path planning using probabilistic cell decomposition," in Proceedings of the IEEE Int'l Conference on Robotics and Automation, Vol.1, pp. 467-472, 2004.

[29] P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," in IEEE Transactions on Systems Science and Cybernetics, Vol. 4, pp. 100-107, 1968.

[30] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik., vol. 1, pp. 269-271, 1959.

[31] R. E. Korf, "Depth-first iterative-deepening: an optimal admissible tree search," Artificial Intelligence, vol. 27, pp. 97-109, 1985.

[32] A. Reinefeld and T. A. Marsland, "Enhanced Iterative-Deepening Search," in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, pp. 701-710, 1994.

[33] R. E. Korf and W. Zhang, "Divide-and-conquer frontier search applied to optimal sequence alignment," in Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, MIT Press, 2000, pp. 910-916.

[34] R. Zhou and E. A. Hansen, "Sparse-memory graph search," in Proceedings of the 18th Int'l Joint Conference on Artificial Intelligence, 2003, pp. 1259-1266.

[35] T. Ikeda and H. Imai, "Enhanced A* algorithms for multiple alignments: optimal alignments for several sequences and k-opt approximate alignments for large cases," Theoretical Computer Science, vol. 210, pp. 341-374, 1999.

[36] S. Schroedl, "An improved algorithm for optimal multi-sequence alignment," in Journal of Artificial Intelligence Research, vol. 23, pp. 587-623, 2005.

[37] D. Klein and C. Manning, "Factored A* search for models over sequences and trees," in Proceedings of the Int'l Joint Conference on Artificial Intelligence, pp. 1246-1251, 2003.

[38] T. Yoshizumi, T. Miura and T. Ishida, "A* with partial expansion for large branching factor problems," in Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 923-929, 2000.

[39] K. I. Trovato, "Differential A*: An Adaptive Search Method Illustrated with Robot Path Planning for Moving Obstacles and Goals and an Uncertain Environment," Int'l Journal of Pattern Recognition and Artificial Intelligence, vol. 4, 1990.

[40] E. A. Hansen and R. Zhou, "Anytime Heuristic Search," Journal of Artificial Intelligence Research, vol. 28, pp. 267-297, March 2007.

[41] M. Likhachev, G. Gordon and S. Thrun, "ARA*: Anytime A* with provable bounds on sub-optimality," Advances in Neural Information Processing Systems, vol. 16, 2003.

[42] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz and S. Thrun, "Anytime dynamic A*: An anytime replanning algorithm," in Proceedings of the Int'l Conference on Automated Planning and Scheduling, 2005.

[43] D. Ferguson, M. Likhachev and A. Stentz, "A guide to Heuristic-based Path Planning," in Proceedings of the International Workshop on Planning Under Uncertainty for Autonomous Systems, Int'l Conference on Automated Planning and Scheduling, June, 2005.

[44] Federal Aviation Administration, "United States Standard for Terminal Instrument Procedures (TERPS)," FAA Order 8260.3b w/ch 19, May 15, 2002.

[45] Federal Aviation Administration, "Civil Utilization of Global Positioning Systems (GPS)," FAA Order 8260.38a, April 05, 1995.

[46] Federal Aviation Administration, "Helicopter Global Positioning System (GPS) Non-Precision Approach Criteria," FAA Order 8260.42a, June 11, 1997.

[47] Federal Aviation Administration, "Area Navigation (RNAV) Approach Construction Criteria," FAA Order 8260.48, April 08, 1998.

[48] National Imagery and Mapping Agency, "Performance Specification Digital Terrain Elevation Data (DTED)," Tech. Rep. MIL-PRF-89020B, May 23, 2000.

[49] C. Gantz, "Converting Latitude/Longitude to Universal Transverse Mercator (UTM)," June 22, 2000, http://www.gpsy.com/gpsinfo/geotoutm/.

[50] J. P. Snyder, "Map projections - A Working Manual," United States Geological Survey Professional Paper 1395, 1987.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Fort Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education,
   MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center,
   MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity
   (Attn: Operations Officer)
   Camp Pendleton, California