

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Active Path Planning for Drones in Object Search

Permalink

<https://escholarship.org/uc/item/00s5b7xb>

Author

Wang, Zeyangyi

Publication Date

2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Active Path Planning for Drones in Object Search

A Thesis submitted in partial satisfaction of the requirements

for the degree Master of Science

in

Electrical and Computer Engineering

by

Zeyangyi Wang

Committee in charge:

Professor Tara Javidi, Chair

Professor Farinaz Koshanfar

Professor Truong Nguyen

2017

The Thesis of Zeyangyi Wang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2017

TABLE OF CONTENTS

Signature Page.....	iii
Table of Contents.....	iv
List of Figures.....	vi
List of Tables.....	vii
List of Graphs.....	viii
Abstract of the Thesis.....	ix
Introduction.....	1
Chapter 1 Tree Search Algorithm.....	2
1.1 Search Conditions & Variables.....	2
1.2 Linear Search Method.....	3
1.3 Tree Search Model.....	4
1.4 AZ-net.....	5
1.5 Naive Tree Search Algorithm.....	5
1.6 Modified Tree Search Algorithm.....	7

Chapter 2 Simulations.....	8
2.1 Tree Search Simulations.....	8
2.2 Modeling Processing Time.....	8
2.3 Modeling Zoom Indicator Scores.....	9
2.4 Simulation Results.....	9
 Chapter 3 Implementation & Experiments.....	 11
3.1 Platform Setup.....	11
3.2 Communication Scheme.....	12
3.3 Experiments Results.....	13
 Conclusion.....	 14
 References.....	 15

LIST OF FIGURES

Fig 1: Demo Video Snapshots.....	13
----------------------------------	----

LIST OF TABLES

Table 1: Standard Parameter Table.....	9
--	---

LIST OF GRAPHS

Graph 1: Illustration of Search Area.....	3
Graph 2: Illustration of Tree Search.....	4
Graph 3: Sample Tree Search.....	6
Graph 4: Averaged Simulation Search Time with Standard Parameters.....	10
Graph 5: Performance Comparison by Varying Standard Parameters.....	10
Graph 6: Illustration of Communication Scheme.....	12

ABSTRACT OF THE THESIS

Active Path Planning for Drones in Object Search

by

Zeyangyi Wang

Master of Science in Electrical and Computer Engineering

University of California, San Diego, 2017

Professor Tara Javidi, Chair

Object searching is one of the most popular applications of unmanned aerial vehicles. Low cost small drones are particularly suited for surveying tasks in difficult conditions. With their limited on-board processing power and battery life, there is a need for more efficient search algorithm. The proposed path planning algorithm utilizes AZ-net, a deep learning network to process images captured on drones for adaptive flight path planning. Search simulation based on videos and actual experiments show significant reduction in search time under certain circumstances, compared to traditional linear search method. The thesis will discuss important design tradeoff between performance and battery life.

INTRODUCTION

Commercial drones, usually with a camera on board, are widely used in object searching. Being reliable and efficient, drones perform better than humans in extreme conditions and are relatively low-cost.

There have been plenty of successful prior attempts in implementing an object detector on drones for object search. Many chose to use a cloud server for image processing since drones don't have enough computational power for image neural networks. [1] Because object detectors aren't trained to recognize targets from distance they are inconsistent as detection rate drops when the target is far away from the drone or captured from tilted angles. [2]

The drone must get close to the search target in order to capture high-resolution pictures for a successful detection. Yet it will take too long for drones to find its target by scanning through the whole search area. Instead we'd like to propose a prioritized, hierarchical search algorithm for drones in order to save flight time compared to traditional linear search method. There are multiple scenarios where a such method could help with object search: (1) Search for survivors in an avalanche/open sea. (2) Search for wild animals in prairie. (3) Search for weeds in agricultural crops.

Chapter 1 Tree Search Algorithm

1.1 Search Conditions & Variables

We define the search area to be a square of length 26 meters that can be divided into 16 (4 by 4) grids of equal area. The search area is relatively sparse, containing few other objects of the same size as the search target. The search target is a regular sized basketball that can be present in one of the 16 grids. The drone will hover above the search grid in order to take a picture of search target for object detection. The detection results in either positive or negative of the target object present. We assume the drone is close enough to the target so that the detector is perfect, therefore it always outputs the correct decision if the drone has found the target. The drone's field of view is limited by camera tilt/pan angle of 80 degrees. Based on our measurement, the drone moves at constant speed of 2 meters per second and it takes 2.5 seconds for each detection result. We also define the above variables and constants as standard parameters for later sections.

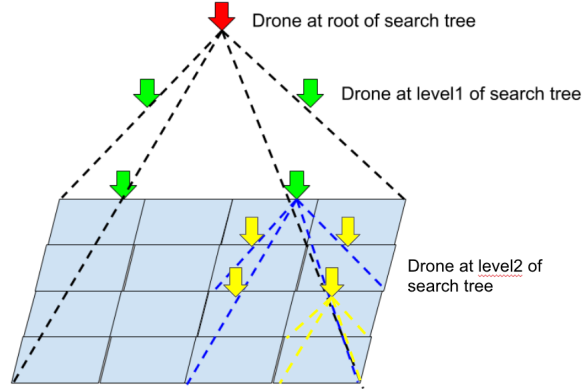
Graph 1: Illustration of Search Area

1	8	9	16
2	7	10	15
3	6	11	14
4	5	12	13

1.2 Linear Search Method

The linear search method will start from any one of the four corners of search area and iterate through every grid until the drone finds the target object. The drone will take a picture of the grid 4 meters above for detection. The expected time for searching one grid T is the horizontal movement time to a neighboring grid and detection time. The expected linear search time is T times expected number of grids searched before finding the target (8.5). Under standard parameters the linear search will take 50 seconds on average. We'll use this as a benchmark for evaluating other algorithms.

Graph 2: Illustration of Tree Search



1.3 Tree Search Model

We propose the tree search algorithm that starts from above the area and drops down gradually. With standard parameters the height of search tree is 16 meters, therefore the height of root (level 0) is 16 meters, the height of level 1 is 8 meters and the height of level 2 (linear search level) is 4 meters. The drone takes a picture of the whole search area at first to decide where it is best to go first. And it detects only when it reaches the last level of search tree, which is 4 meters high. We chose 4 meters since using the latest YOLO2 detector the largest distance from object where the detector still recognizes it is 4 meters. When the drone is flying higher than 4 meters the latest detector stops working and we now need more information to make a decision.

The tree search model can be formulated as following: A node N in a 3-level search tree has anchors $A_1 - A_4$, each with a zoom indicator score Z_k . The search begins in the highest node at L_0 and proceeds downward. At each node in level L_n the drone takes a picture of current search area and receives 4 zoom indicator scores for subsequent nodes

at L_{n-1} . At L_2 the drone detects the target area to verify if the object has been found. The model outputs location of object S when the drone finds the object at L_2 .

1.4 AZ-net

AZ-net [3] is a deep learning neural network trained for 20 common object classes that can't be detected in relatively low resolution images. The input image is divided in four anchors equally. It outputs four zoom indicator scores (normalized between 0 and 1, higher means larger likelihood of finding trained objects) for each input image. The AZ-net tells if an input image is likely to contain trained object classes even images are taken very far from the object. We utilize AZ-net by sending images captured at each node of the search tree to the neural network and rank 4 output scores to decide which anchor to search first. The drone will run AZ-net at root of the search tree then again at level 1. When the drone descends then reaches the bottom of search tree (level 2 when height is 4 meters), it will run detection of one grid for final result.

1.5 Naive Tree Search Algorithm

When zoom indicator scores are wrong and the drone is misled to a negative detection at last, it will instead search other regions until it finds the target. The naive tree search algorithm is a simple method where the drone searches nearest grids first then turns to other grids.

The algorithm is concluded in the following steps:

(1) Starting from root node. Move to highest node in L_1 .

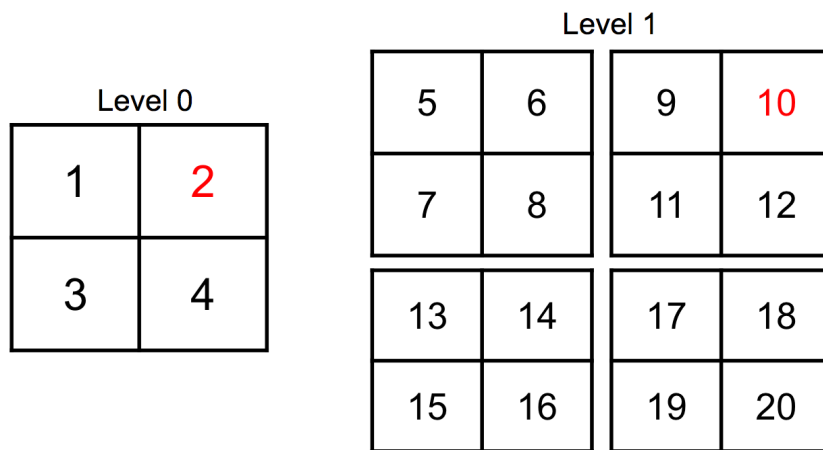
(2) Starting from L_1 . Move to highest node in L_2 .

Run detection. If positive: end search.

(3) Search other L_2 nodes from previous L_1 node. If detection positive: end search.

(4) Move to other L_1 nodes then do step (2) until all L_2 nodes are detected.

Graph 3: Sample Tree Search



An example of naive tree search is presented in Graph 3. At level 0 the largest zoom indicator is marked in red, which leads the drone to level 1 anchor 9,10,11 and 12.

At level 1 the largest anchor score is 10 therefore the initial detection will occur in anchor 10. If detection returns negative the drone will then run detection at level 2 on anchor 9,11 and 12. Next it will return to level 1 and get new zoom indicator scores of sub-anchors of anchor 1,3 or 4 until it locates the target at level 2.

1.6 Modified Tree Search Algorithm

We found the naive tree search algorithm is taking more time than linear search in test runs. When initial detection fails, the drone travels between levels by moving vertically, which is expensive in time. On the other hand, input images to AZ-net at root has much lower accuracy than those taken at level 1 since they are taken further away from the object. The drone therefore is often misled at root, spending too much time searching the wrong anchor.

To compensate the fact that vertical movements are more costly than horizontal movements and zoom indicators perform better at lower heights, we modified the search algorithm as follows:

- (1) Starting from root node. Move to highest node in L_1
- (2) Starting from L_1 . Move to highest node in L_2 with above-threshold score.

Move to the node with highest score if all nodes are below threshold.

- (3) Starting from L_2 . If detection positive: end search.
- (4) Search L_2 nodes that are descent from L_1 node sorted by scores in step (3).

If detection positive: end search.

- (5) Search L_2 nodes that are descent from other L_1 nodes sorted by scores in step (1). If detection positive: end search.

The threshold value based on our training samples is 0.01 for it eliminates most false positives of level 0 zoom indicator scores.

Chapter 2 Simulations

2.1 Tree Search Simulations

To evaluate and compare performance of algorithms we used a simulation test bench. The test bench places a basketball randomly at 1 of 16 grids. In each simulation assume the drone goes to the next node of the search tree according to simulated zoom indicator scores. The simulation finally records total search time used when the drone has found the basketball. The linear tree search time is constant at 50 seconds using standard parameters. The perfect tree search benchmark is defined as the scenario when the zoom indicator always outputs correct results. All simulations are iterated 1000 times for averaged value where the mean search time converges.

2.2 Modeling Processing Time

The drone will stop for processing time T_p when it reaches the next node in the search tree. Processing time includes control response time, transmission time and computing time. Control response time is the time period for drone to receive and respond to the next action command. Transmission time is the time period between pictures taken on drone and server receiving input images. Computing time is the time period for the server to process input images, run on AZ-net or detector, then output zoom indicator scores or detection result. Processing time is modeled as mean of actual measurements of the system.

2.3 Modeling Zoom Indicator Scores

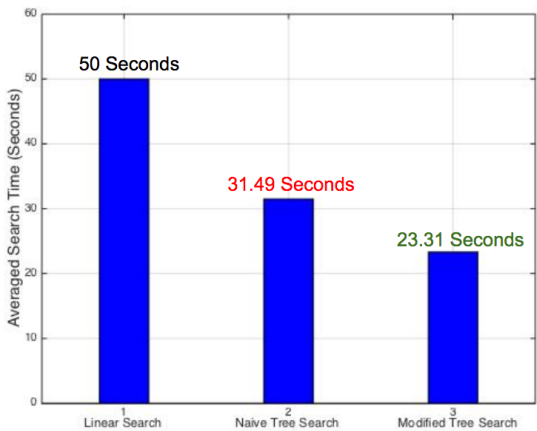
We define 4 zoom indicator scores at level 0 A1,A2,A3 and A4. At level 1 there are 16 zoom indicator scores from A5 to A20. To model zoom indicator output in our simulations, assume Gaussian distributions ($\sim N(\mu, \sigma^2)$) for randomly generated zoom indicator scores at level0/level1. They are classified into 4 categories: level 0 negative, level 0 positive, level 1 negative and level 1 positive. μ and σ modeled using 50 training samples from 8 videos of 4 test scenarios. Then we used sigmoid function to model the output of these variables between 0 and 1, the same range as actual AZ-net output.

2.4 Simulation Results

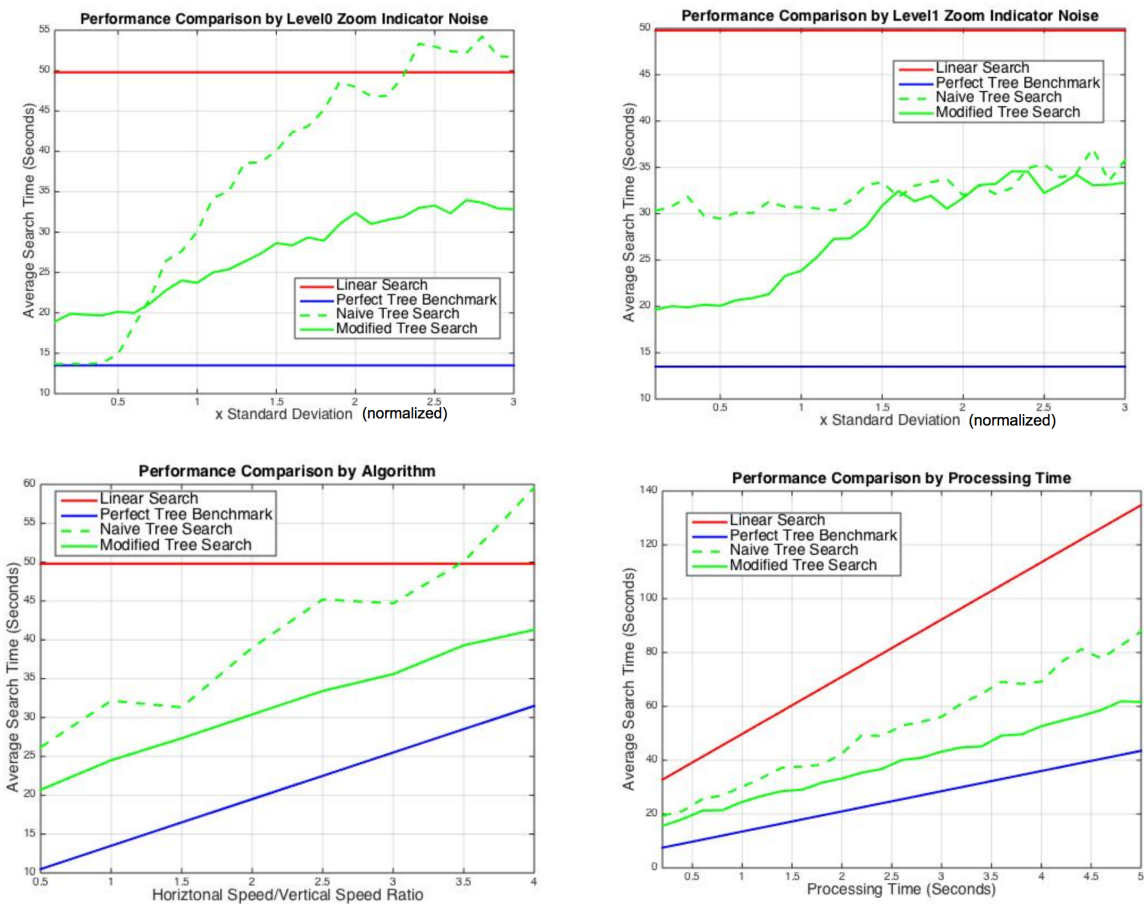
Table 1: Standard Parameter Table

	Standard Parameter
Tree Height	16 (Meters)
Search Area	26*26 (Meters^2)
Drone Speed	2 (Meters/Second)
Processing Time	2.5 (Seconds)
Threshold Value	0.01
Level 0 Positive Mean/Variance	-2.8111/3.4995
Level 0 Negative Mean/Variance	-6.1573/4.2994
Level 1 Positive Mean/Variance	0.8259/4.0570
Level 1 Negative Mean/Variance	-7.4209/6.3449

Graph 4: Averaged Simulation Search Time with Standard Parameters



Graph 5: Performance Comparison by Varying Standard Parameters



Graph 4 shows 37% saving in search time for naive tree search algorithm and 53% saving for modified tree search algorithm compared to linear search benchmark. Graph 5 demonstrates the trend in search time as one of the standard parameters varies and how different search methods are affected. Modified tree search algorithm is less dependent on level 0 zoom indicator accuracy than naive tree search algorithm since the threshold at level 1 corrects some mistakes by previous level 0 zoom indicator. On the other hand it is not averaging worse search time than naive tree search method when level 1 zoom indicator accuracy is bad. We can find how vertical movements is more costly in naive tree search method since it is returning to level 1 after a failed search. When the processing time increases, the modified tree search method is also performing better than naive tree search method since it has averaged less number of stops.

Chapter 3 Implementation & Experiments

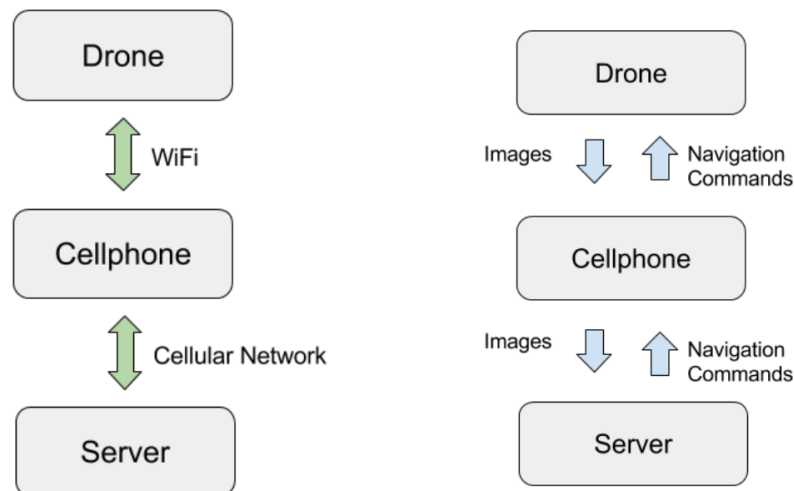
3.1 Platform Setup

We used Parrot Bebop 2 drone for this project, which has a 1080p camera on board and takes pictures of resolution 1280*720 pixels. Since the drone doesn't have enough computing power on board we use a cloud server to process images from AZ-net. The cloud server is set up on a desktop computer equipped with Nvidia GTX 980 Ti graphics card. The server first builds a zoom indicator model on Tensorflow, then waits and accepts incoming image through socket connection, finally outputs zoom indicator scores.

An Android cellphone serves as a relay device between the server and drone. It receives captured images from the drone then sends it to the server. In response the server sends back zoom indicator scores translated into navigation commands to the drone. The control program is built based on official Parrot Software Development Kit. After takeoff the drone will execute a tree search or linear search based on input parameters.

3.2 Communication Scheme

Graph 6: Illustration of Communication Scheme



Since Parrot Bebop 2 drone must be a WiFi access point and can't be used as a client device, there is no official solution to have it connect to an outside network. Therefore there must be a relay device (such as the cellphone we used) connected to the drone's network while communicating with a remote server. Because all Android devices can only connect to either cellular network or WiFi from their default configuration, we used Speedify, a channel bonding software that combines data from both networks

through a VPN. Speedify on the cellphone allows simultaneous networking with the drone and cloud server. However it also causes significant delay in processing time as it directs all network traffic on the phone through the nearest server.

3.3 Experiments Results

We performed two successful modified tree searches on an empty field with standard parameter setup. The drone landed when the detector returns positive. The first demo took 30 seconds and the second took 35 seconds since the zoom indicator returned incorrect anchor at level 0. But the drone quickly recovered since the all level 1 anchors from that were below threshold.

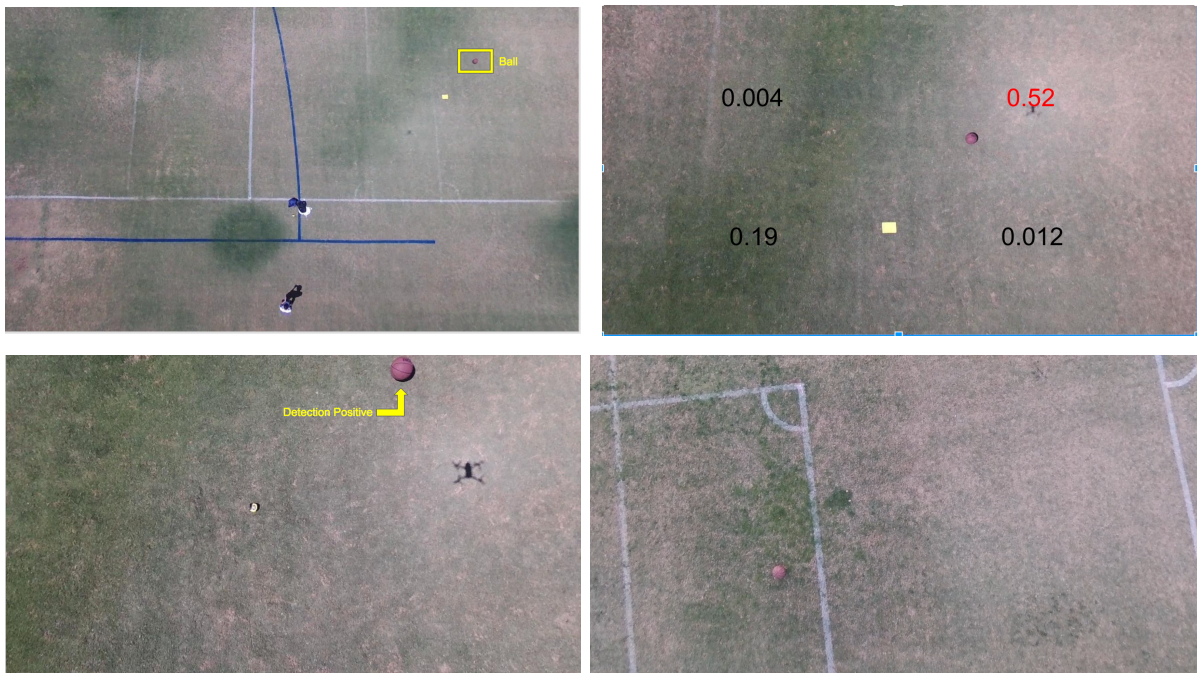


Fig 1: Demo Video Snapshots

Conclusion

We presented a new path planning algorithm to save battery time on drone during object search using a deep learning neural network. We found the performance of modified tree search algorithm is better than linear search under standard parameters. However the algorithm is also limited to viewing angle of search object as most of them appear very differently from height. To improve robustness of tree search algorithm we suggest using an image library of objects in which pictures are taken from wider range of distances and angles.

References

- [1] Jangwon Lee, Jingya Wang, David Crandall, Selma Sabanovic and Geoffrey Fox, "Real-Time Object Detection for Unmanned Aerial Vehicles based on Cloud-based Convolutional Neural Networks"
- [2] Song Han, William Shen, Zuozen Liu, "Deep Drone: Object Detection and Tracking for Smart Drones on Embedded System"
- [3] Yongxi Lu, Tara Javidi, Svetlana Lazebnik, "Adaptive Object Detection Using Adjacency and Zoom Prediction"