{ إِنَّ فِي خَلْقِ السَّمَوَاتِ وَالأَرْضِ وَاخْتِلافِ اللَّيْلِ وَالنَّهَارِ
لآيَاتٍ لأُولِي الأَلْبَابِ الَّذِينَ يَذْكُرُونَ اللَّهَ قِيَامًا وَقُعُودًا
وَعَلَى جُنُوبِهِمْ وَيَتَفَكَّرُونَ فِي خَلْقِ السَّمَوَاتِ وَالأَرْضِ
رَبَّنَا مَا خَلَقْتَ هَذَا بَاطِلاً سُبْحَانَكَ فَقِنَا عَذَابَ النَّارِ }

(آل عمران ١٩٠-١٩١)

إهداء

،، إلى من كلله الله بالهيبة والوقار ،، و علمني معنى العطاء ،، إلى من أحمل إسمه بكل الفخر ،، ستبقى كلماتك نجوما أهتدي بها اليوم وفي الغد وإلى الأبد.. أرجو من الله أن يمد في عمرك ،، و رزقني برك ،، والدي العزيز

،، إلى ملاكي في الحياة .. إلى معنى الحب والتفاني .. إلى بسمةالحياة وسر الوجود ،، إلى من كان دعائها سر نجاحي ،، أدامك الله سراجا في حياتي ،، أمي الحبيبة

،، إلى من أرى التفاؤل بأعينهم .. والسعادةفي ضحكاتهم ،، فأنا منهم و هم مني ،، إخوتي الأعزاء

،، إلى توأم الروح ورفيقة الدرب ،، صاحبة القلب الطيب والنوايا الصادقة ،، زوجتي الغالية

أهدي هذا العمل ،،

شكر و عرفان

،، إلهي ،، لايطيب الليل إلا بشكرك ،، ولايطيب النهارإلا بحمدك ،، ولاتطيب الدنيا إلا بذكرك ،، ولا تطيب الآخرة إلا بعفوك ،، ولا تطيب الجنة إلا برؤيتك ،، فلك الحمد و الشكر حمدا كثيرا كما ينبغي لجلال وجهك و عظيم سلطانك

،، إلى من بلغ الرسالة وأدى الأمانة .. ونصح الأمة .. إلى نبي الرحمة ونور العالمين..
خاتم النبيين و سيد المرسلين ،، سيدنا محمد ،، عليك أفضل الصلوات و أتم التسليم

أتوجه بجزيل الشكر والامتنان إلى كل من ساعدني من قريب أو من بعيد على انجاز هذا العمل ، وأخص بالذكر ،، من وقفت على المنبر وأعطت من حصيلة فكرها ،، ثم تفضلت بالإشراف على هذا البحث:

.. د/رانيا عبداللطيف

فجزاها الله عني خيرا ولها كل التقدير والاحترام

،، من وضع البذرة الأولى لهذا البحث ، و لولا فضل الله أولا ،، ثم مجهودك الكبير لم يكن ليكتمل ،، فلك كل التقدير ،،

م/ عبدالله الرشيد

،، من وقف بجانبي و لم يبخل ،، و كان نعم العون و السند ،، أخي و صديقي

م/حسن البصري الأمين

فلكم جزيل الشكر و الثناء

ABSTRACT

Target Localization using small unmanned aerial vehicles (UAVs) has emerged in the recent decades as an effective and essential option in surveillance, military and other fields. Often, UAVs holds camera and with the aid of different sensors it can determine the target location. This research represents mathematical method for calculating the distance between the UAV and the target using UAV coordinates and the tilt angle of the camera when posing to the target and then with knowing the heading of the UAV it obtains the coordinates of the target. This proposed method assumes that the target altitude is known or estimated from previous background about the geographical nature of the field or provided momentarily to the system.

المستخلص

يعتبر تحديد مواقع الأهداف و إحداثياتها باستخدام الطائرات غير المأهولة من التطبيقات التي ظهرت في العقود الأخيرة كخيار مهم و مجدي لما يتميز به من قلة التكلفة نسبيا و إمكانية استخدامه في مناطق ذات خطورة طبيعية أو عسكرية أو غير ذلك كما يمكن استخدامه في مجالات المساحة الجوية. غالبا ما تستخدم هذه الأنظمة كاميرا يتم تثبيتها على الطائرة غير المأهولة و بمساعدة مجموعة من الأجهزة و الحساسات تقوم بحساب موقع الهدف.هذا البحث يقدم طريقة رياضية لحساب إحداثيات الهدف بمعلومية إحداثيات الطائرة و زاوية رؤية الكاميرا للهدف ، و من ثم حساب موقع الهدف بمعلومية زاوية انحراف الطائرة عن الشمال الجغرافي. هذه الطريقة تفترض أن ارتفاع الهدف عن مستوى سطح البحر معلوم أو مقدر إما عن طريق معلومة بواسطة الخرائط أو أي طريقة أخرى لحساب ارتفاع الهدف عن سطح البحر أو يتم تزويد النظام به لحظيا.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

d - Angular distance.

R - Earth radius at target position.

θ - The angle between the perpendicular line from the UAV to the center of the earth and the real target position above the Mean Sea Level.

θ_o - The angle between the line from the UAV to the center of the earth and the image of the target on the Mean Sea Level.

θ_e - The angle between the image of the target on the Mean Sea Level and the real position of the target above the Mean Sea Level.

Ψ - The angle between the target to the earth center and the target image to the center.

Ω - The angle between the extension of the line from the target to the target image and the radius (R) line.

LIST OF ABBREVIATIONS

UGV -	Unmanned Ground Vehicle
UAV-	Unmanned Aerial Vehicle
MAV-	Micro Aerial Vehicle
INS-	Inertial Navigation System
IMU-	Inertial Measurement System
GPS-	Global Positioning System
DGPS-	Differential Global Positioning System
GNSS-	Global Navigation Satellite System
DEM-	Digital Elevation Map
NED-	North East Down
MICE-	Multiple Image Coordinate Extraction
CEP-	Circular Error Probable
nmi-	Nautical Mile

Chapter One: Introduction

1.1- Introduction:

In many surveillance systems and military applications, there is an essential need to rapidly process imagery data from vision systems (typically a camera) and extract the coordinates of a detected location or target seen by the camera. This may requires mathematical calculations and intensive data processing to resolve for position determination.

In recent years, odometry sensors have been widely used for estimating the motion of vehicles moving in a 3D space environment such as Unmanned Aerial Vehicle (UAVs) and Unmanned Ground Vehicles (UGVs). For instance, Inertial Navigation Systems (INSs) are applied to measure linear acceleration and rotational velocity, and capable of tracking the position, velocity and altitude of a vehicle by integrating these signals [4] or mobile vehicles use the Global Position System (GPS) and Inertial Measurement Units (IMUs) for land vehicle applications [5].

Besides the additional sensors, a camera allows a robot to perform a variety of tasks autonomously. The use of computer vision for localization has been investigated for several decades. The camera has not been at the center of robot localization while most of researchers have more attention to other sensors such as laser range-finders and sonar. However, it is surprising that vision is still an attractive choice for sensors because cameras are compact, cheaper, well understood, and ubiquitous [1].

In such systems, laser range finder is still used as a sensor to know the direct distance between the camera and the target.

1.2- Problem statement:

In case of the absence of the laser range finder, another method is needed to determine the great circle distance between the target and the projection of the camera in the surface of the earth, in order to extract the coordinates of the target.

1.3- Proposed Solution:

To derive a mathematical Model that is able to calculate the great circle distance between two points on the earth surface.

1.4- Aims and Objectives:

The aim of this project is calculating the coordinates of a user specified target using the self coordinates of the vision system, the heading from the geometric north and mathematical equations to find the distance to the target.

1.5- Scope:

A generic vision based system is taken that’s composed of a Global Positioning System (GPS) device to identify the first point location, digital compass to find the bearing, a camera that rotates in pan and tilt and encoders to determine the tilt angle of the camera. Based on the information taken from these parts, a formula is derived to calculate the distance.

In this research the system is considered as aligned to the horizon so the affecting factor that comes from the pitch angle of the UAV is not included in the calculations.

Also, even though this formula is applicable for different systems, it is basically meant to be used for small UAVs and for short range distances.

1.6- Methodology:

In order to determine the coordinates of a destination point knowing the starting point coordinates two tasks will be done:

- Calculating the great circle distance between the projection point of the system on the surface of the earth and the position of the target.
- Calculating the coordinates (longitude and latitude and heading of the target) using well known standard equations derived from the law of cosines and the law of tangents in the math applied to the spherical shape of the earth.

1.7- Thesis Outlines:

The first chapter is an introduction about the research. The second chapter provides a review about different researches and previous work that has been done related to the topic of this research. Chapter Three explains the methodology that has been followed to derive the new method of calculating the coordinates of the target and the derived equations. In chapter an analysis is done to the results and compared to another method. Finally, in chapter five there are the suggestions and the recommendations.

Chapter Two: Literature Review

2.1- Introduction:

Target Acquisition systems are systems designed for different purposes like tracking, reconnaissance and map creation systems, in order to help different surveillance, navigation and military applications. These systems generally consist of a number of sensors and devices and able to integrate the information from all those sensors in order to build a precise awareness of target's positions. For example the correct position of a certain target can be obtained using information from a Laser rangefinder sensor, with digital compass and *Global Navigation Satellite System* (GNSS) information.

2.2- Geo-registration Techniques:

Historically, sensors have relied on geo-registration techniques using ground truth to derive target coordinates. Geo-registration is the alignment of an unreferenced image with a geodetically (latitude, longitude, and elevation based) calibrated reference image. Such alignment allows each observed image pixel to inherit the coordinates and elevation of the reference pixel it is aligned to. The reference imagery is usually a wide area, high-resolution ortho-image. Each pixel in the reference image has a longitude, latitude and elevation associated with it (in the form of a DEM - Digital Elevation Map). Since the reference image is usually dated by the time it is used for geo-registration, it contains significant dissimilarities with respect to the aerial video data. The aerial video data is captured from a camera mounted on an aircraft. The orientation and position of the camera are recorded, per-frame, in the telemetry information. Since each frame has this telemetry information associated with it, geo-registration would seem to be a trivial task of projecting the image onto the reference image coordinates [1]. In general, this process aligns two images (satellite is the most common, but is not limited to satellite images) and correlating the images to a physical location by examining a set of distinguishable points.

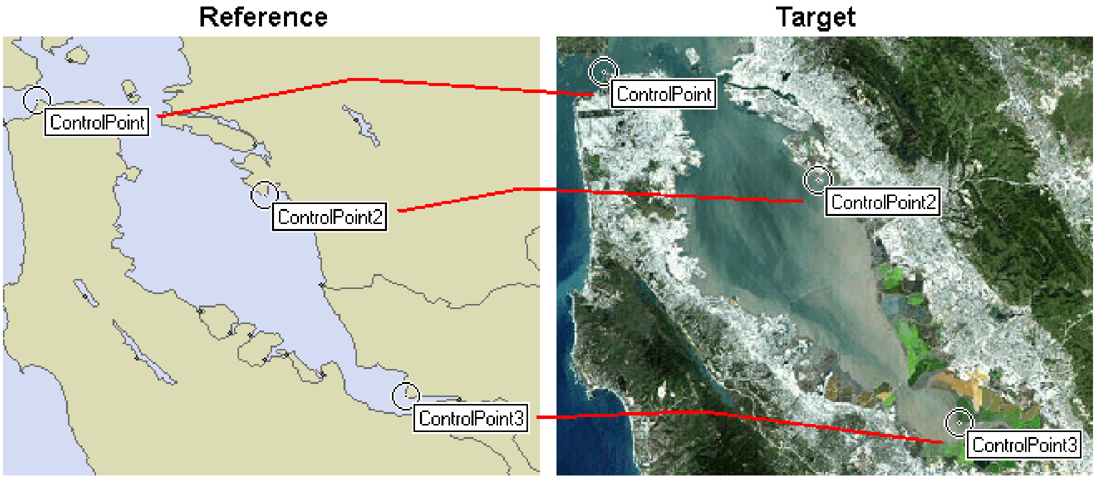


Figure2.1: Example of Image Geo-registration

It has been a rather recent area of study. Coarse geo-registration was fundamental to early surveillance images taken during the cold war era for military use. By the 1970s this process had been developed for non-military purposes, allowing for the development of GIS geo-registration. This method is time consuming and can be unreliable in poor visibility conditions when ground reference data is hard to observe.

Yu Jiaxiang [2] presented an approach to estimate the geodetic coordinates that requires several images and associated camera locations. The mathematical models are constructed by linearization of the collinearity equations to iteratively compute the pose angles and focal length of the camera. At least three images of the target, along with at least three identifiable common points among the images, are needed for reckoning camera pose angles and focal length. The three dimensional (3D) coordinates of ground target are calculated using forward intersection. Neither digital elevation model (DEM) nor measured pose angles of the camera was used. Camera positioning error was the only main factor that degrades the geo-location accuracy. The three dimensional (3D) coordinates of ground target are calculated using forward intersection. That method could get the target coordinates with no dependence on digital elevation model (DEM) and the measured values of camera pose angles, therefore two of the three primary error sources in the traditional UAV target location approach were eliminated. Simulation and real image experiment results showed that the accuracy of the estimated target location was close to that of the UAV position, and that target location error was within 5m circular error probable (CEP) on condition that the UAV was navigated by differential global positioning systems (DGPS).

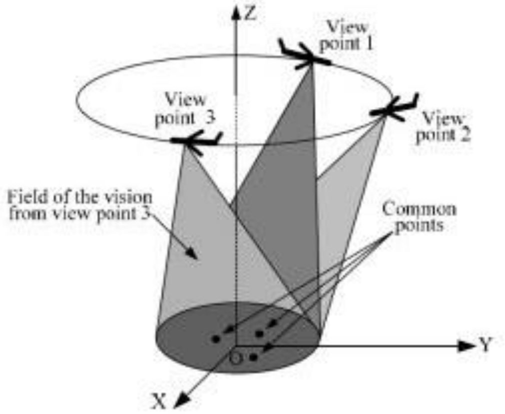


Figure2.2: Modified Image Geo-registration that takes more than one image

Also, Gianpaolo Conte, Maria Hempel, Piotr Rudol, David Lundstrom, Simone Duranti, Mariusz Wzorek, Patrick Doherty presented a method for high accuracy ground target localization using a Micro Aerial Vehicle (MAV) equipped with a video camera sensor[3]. The proposed method was based on a satellite or aerial image registration technique. The target geo-location was calculated by registering the ground target image taken from an on-board video camera with a geo- referenced satellite image. This method did not require accurate knowledge of the aircraft position and altitude; therefore it was especially suitable for MAV platforms which do not have the capability to carry accurate sensors due to their limited payload weight and power resources.

The paper also presented results of a ground target geo-location experiment based on an image registration technique. The platform used was a MAV prototype which won the 3rd US-European Micro Aerial Vehicle Competition (MAV07). In the experiment a ground object was localized with an accuracy of 2.3 meters from a flight altitude of 70 meters. The main advantages using that method could be summarized in the High accuracy ground target localization, Instantaneous performances In contrast to other approaches which require many measurement samples and a specific flight path around the target is not required for a successful application of the approach. The method is applicable in urban areas where the road network is dominant, or in any other environment with a relevantly structured clutter.

However, these methods and other Geo-registration techniques still lack the precision that is desired to provide precise results due to changing resolutions and image inconsistencies. They used stereo photogrammetric techniques to determine the 3-D relative position of image features to the camera location. These require intensive data processing to resolve for position and rotation angle changes between the stereo images and also rely on known reference points from a database to establish the absolute location of target features.

In addition, with the development of better resolution sensors, and the precision geo-location capability provided by the Global Positioning System (GPS), it is now possible to deliver imaging sensors with embedded geo-registration capability, avoiding the need for extensive image analysis to extract precise target coordinates. The existing methods for registration no longer are accurate enough to cope with the increase of details. Another area of significant focus is the ability to automate the geo-registration process. There has been a great deal of work to make the process more automated, but it still is an open area of work.

2.3- Sensors and self localization techniques:

With the revolution in the electronics and computer technologies researches started to look in another direction that depends on the sensors and self-localization without relying on any reference data.

Bing-FeiWu et al [4] presented a vision-based algorithm for localizing targets in 3D environment. It was achieved by the combination of different types of sensors including optical wheel encoders, an electrical compass, and visual observations with a single camera.

Based on the robot motion model and image sequences, extended Kalman filter was applied to estimate target locations and the robot pose simultaneously. The technique was especially suitable for navigation and target tracing for an indoor robot and had a high potential extension to surveillance and monitoring for Unmanned Aerial Vehicles with aerial odometry sensors. The experimental results presented “cm” level accuracy of the localization of the targets in indoor environment under a high-speed robot movement [4]. In that method the targets are assumed to be stationary landmarks in the proposed algorithm. If the algorithm is modified to track a moving target it will be more efficient.

D. Blake Barber Joshua D. Redding · Timothy W. McLain · RandalW. Beard · Clark N. Taylor [5] introduced a system for vision-based target geo-localization from a fixed wing micro air vehicle. Using the pixel location of the target in an image, measurements of MAV position and altitude, and camera pose angles, the target is localized in world coordinates. The geometry required to produce raw localization estimates was discussed in the paper. The primary contribution of the paper was the description of four key techniques for mitigating the error in the raw estimates. These techniques include RLS filtering, bias estimation, flight path selection, and wind estimation. The algorithms were successfully flight tested on a micro air vehicle using Procerus’ Kestrel autopilot and a designed gimbal system. Geo-location errors below 5 m were repeatedly obtained under a variety of weather conditions. Throughout the paper they have assumed a flat earth model and a stationary target.

Vladimir N.Dobrokhodov, Isaac I.Kaminer, Kevin D.Jones and Reza Ghabcheloo [6] addressed a development of a vision-based target tracking system for a small unmanned air vehicle. The algorithm performed autonomous tracking of a moving target, while simultaneously estimating GPS coordinates, speed and heading of the target.

Tight real-time integration of UAV’s video and telemetry data-streams with geo-referenced database allowed for reliable target identification, increased precision and shortened time of target motion estimation. A low cost off the shelf system was utilized, with a modified radio controlled aircraft airframe, gas engine and servos. Tracking was enabled using a low-cost, miniature pan-tilt gimbal. The control algorithm provides rapid target acquisition and tracking capability. A target motion estimator was designed and shown in multiple flight tests to provide reasonable targeting accuracy. The impact of tracking loss events on the control and estimation algorithms was analyzed in detail.

The system was capable of tracking a moving target and estimating its position and velocity was developed. Straightforward nonlinear analysis was used to motivate a simple control system for coordinated control of a UAV and of gimbaled camera. The algorithm relied on the information obtained from the onboard camera directly for feedback, thereby eliminating any lags caused by introducing an estimator in the feedback loop. In addition, a critical feature of the proposed algorithm was that it could maintain a desired range to target, when actual range is not known. Results of the stability analysis for both stationary and moving target cases provided explicit means of choosing the control gains.

Furthermore, a nonlinear filter for target motion estimation was introduced. The filter performance was analyzed in the presence of tracking loss events. It was shown that the filter exhibited graceful degradation of performance in the presence of these events. The extensive results of multiple flight test for moving targets supported this conclusion.

Having been implemented onboard a low cost (< $10K) generic UAV system and tested in numerous flight experiments the entire system shows remarkable robustness to unpredictable flight conditions and human operator related factors. Overall, the control system and target motion estimator were shown to perform well in both nonlinear simulation and in numerous flight tests. [6].

K.Senthil Kumar, Kavitha.G, Subramanian.R and Marwan [7], presented a method for determining the location of a ground based target when viewed from an Unmanned Aerial Vehicle (UAV). By determining the pixel coordinates on the video frame and by using a range finder the target’s geo-location is determined in the North- East-Down (NED) frame. The contribution of this method is that the target can be localized to within 9m when view from an altitude of 2500m and down to 1m from an altitude of 100m.That method offered a highly versatile tracking and geo-localization technique that has very good number of advantages over the previously suggested methods.

The airborne images obtained from an UAV are analyzed in ground control station. By using the thermal images, all weather and night operation are possible. Visual and thermal image fusion was done and the fused image was given for target tracking. That system had the benefit of enhanced target tracking application wherein only visual or thermal target tracking would not provide sufficient efficiency. Thus the image fusion process augments information leading to an improved system as a whole. The overall system incorporated segmentation, fusion and target tracking principles. By using the tracked coordinates on the image frame, the coordinates can be shifted to the NED frame which will give the GPS coordinates of the target. The method proved it robust application for the military and can prove efficient since it has an accuracy of 0.9m from an altitude of 100m.

On the other hand, precise altitude determination is a key requirement for precision targeting. Current targeting systems use range and bearing to target (relative to the Forward Observer’s GPS coordinates) to derive the target coordinates. The accuracy of these coordinates is currently limited by the accuracy to which the bearing to target can be observed.

Conventional methods of determining altitude include gyro-compassing, GPS interferometry and magnetic sensors. Existing portable targeting systems provide bearing to an accuracy of 10 (mrad) which introduces 10 m of position error in target coordinates at a range of 1 km.

Under contract to the US Navy, NAVSYS Corporation has developed a system that have the capability to determine precise target coordinates without relying on ground truth by using GPS/inertial-aided sensors. Dr. Alison K. Brown, Gengsheng Zhang and Dale Reynolds from NAVSYS *Corporation* In their paper [8] presented an analysis of the system errors with simulation results and field test data for the precision targeting systems being developed by NAVSYS. Work is still continuing at NAVSYS on developing a man-portable targeting system (SPOTS) capable of providing target coordinates with a 2 m TLE and an airborne version of targeting system which is capable of providing this level of performance out to extended ranges from the target.

NAVSYS also have developed a precision GPS/inertial altitude determination method that enables (1 mrad) pointing accuracy to be provided using low cost (missile-grade) inertial instruments. They presented Test data in a paper [9] shows that their method provides better than 30 times improvement in accuracy over conventional gyro-compassing methods.

Rick Zhang and Hugh H.T. Liu [10] proposed a UAV vision-based relative altitude estimation method using a given size (length) information of ground vehicle in conjunction with a well known target localization technique. Through proposed algorithm in relative altitude estimation, opens doors for possible improvement or expansion in small UAV applications in target localization, tracking, and terrain exploration missions. It has been shown that using a given target size to estimate UAV altitude yields reliable results and can be applied to existing methods of target localization and tracking. Through small scale experiments, it was found that even an 8% error in the altitude estimation yielded good localization results without filtering. Furthermore, that method may be applied to target velocity estimation.

William Semke, Jaganathan Ranganathan, and Matthew Buisker[11] developed an end-to-end system that uses closed form analytical expressions to determine gimbal pointing angles based on GPS and IMU information has been demonstrated to be effective. The system allows for accurate pointing of a camera from an UAV using minimal computational time. Many other algorithms require complex solution strategies to solve for the pointing angles; the method presented here has simple algebraic calculations to establish the proper pointing and orientation of the camera.

Thomas B. Criss, Marilyn M. South, and Larry J. Levy [12] debates that current geo-location methods using GPS, pointing angles, and terrain elevation are only accurate to 100 m at best, which is inadequate for precision strike.

They published an article discusses a developed technique called Multiple Image Coordinate Extraction (MICE) that can use unmanned air vehicle (or other equivalent) imagery to pinpoint target coordinates to within about a 5-m CEP by applying principles of photogrammetry to determine the locations and sizes of objects.

MICE, an algorithm for calculating geodetic coordinates and sizes of objects using remotely sensed imagery, requires only several images and associated camera locations. Resultant target location errors will be of about the same magnitude as the errors in the UAV GPS positions.
MICE is part of several Strategic Systems Department programs. The fundamental technology, developed for another application, was first presented in a 1993 department report.1 MICE was applied to data acquired during the SSN/UAV demonstration, in which it was able to locate a target to better than a 10-m accuracy from a standoff range of 3.3 nmi. Work on MICE targeting is continuing as part of the Navy Tactical Control System Program.

Generally different methods and algorithms are developed using image geo-registration, image processing and sensors, all of them seeking better accuracy and performance.

Chapter Three: Methodology

3.1- System Overview:

The proposed system is composed of a gimbal holding a camera which is used to observe and detect targets. And this gimbal has the ability to rotate in the pan and the tilt (2-axis of freedom) using motors. Also, there are two absolute encoders mounted at the center of the pan and tilt axis to calculate the position of the camera and then determine the tilt angle. Beside these, GPS is connected to the system to get the location of itself and digital compass to know the heading angle. All these components are integrated through sort of processor like microcontrollers or digital signal controllers. This gimbals' system is mounted on UAV and used in surveillance and tracking targets and when locating a target on the screen it calculates the coordinates of the target.

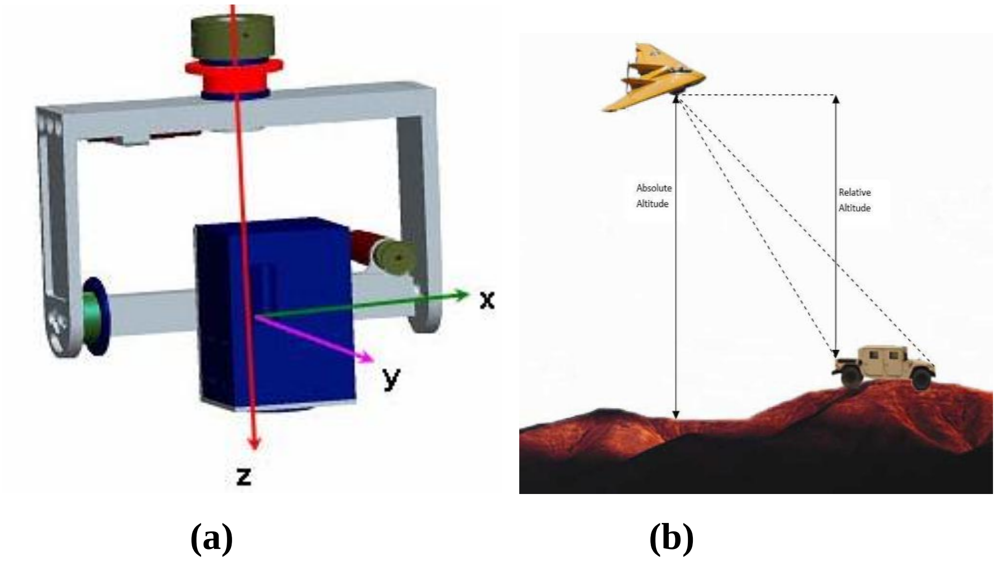


Figure3.1 (a): gimbal system design [16]

(b): target localizing [15]

3.2- Formula Calculations:

A simplified flow diagram for a system that extracts the coordinates of the target based on GPS, digital compass and laser range finder will consists of:

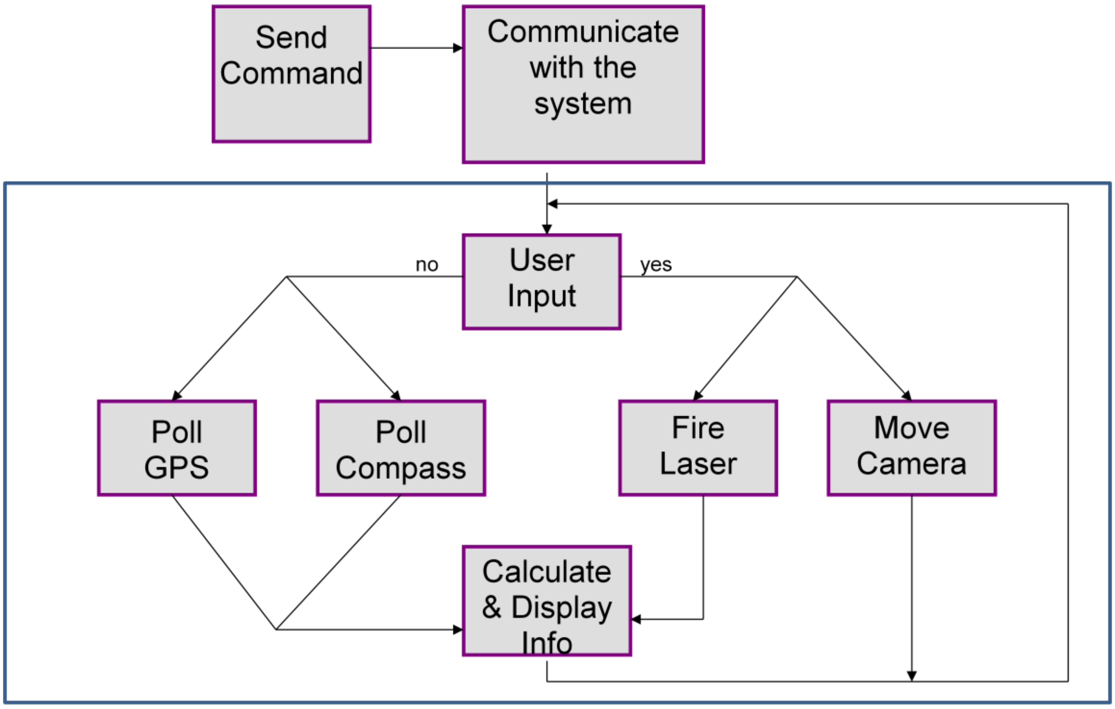


Figure3.2: Flow Diagram of an example system that calculates the coordinates of the target using LRF

In such system, the laser range finder is used to determine the direct distance between the camera on the air and the target on the ground. Then by using the triangulation method, circular distance between the target and the projection point of the camera on the earth surface is calculated. After that, the coordinates of the target is calculated by applying the spherical law of cosines and spherical law of tangents. These laws are already known and used by inertial navigation systems (INS).

If the laser range finder is not available in the system for any reason, the distance to the target cannot be measured directly. For example if the camera is mounted on a small unmanned aerial vehicle (UAV), there will be some limitations on the weight and the mechanical design of the observation system (the camera in our case). Also, in some systems the hardware design cannot be altered because of the limitations in the resources.

For that reason, the proposed method will replace the Laser Range Finder readings with an algorithm to find the circular distance to the target directly without knowing the direct distance from the air.

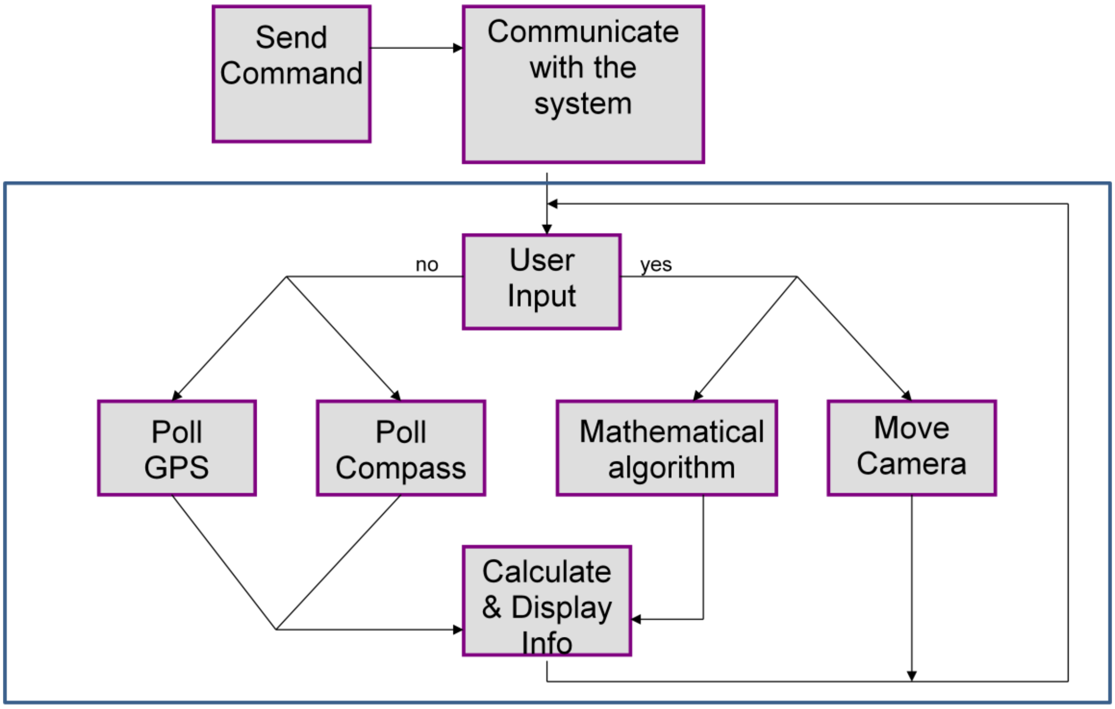


Figure3.3: Flow Diagram of the proposed system that calculates the coordinates of the target with mathematical model

The Mathematical algorithm will consist of two parts:

- The first part will calculate the circular distance between the projection point of the system on the surface of the earth and the position of the target.
- The second part is calculating the coordinates (longitude and latitude and heading of the target) using well known standard equations derived from the law of cosines and the law of tangents in the math applied to the spherical shape of the earth.

3.2.1- Calculating the great circle distance to the target:

We have two cases for the target position on the Earth surface:

a. The target is above Mean Sea Level.

b. The target is below Mean Sea Level.

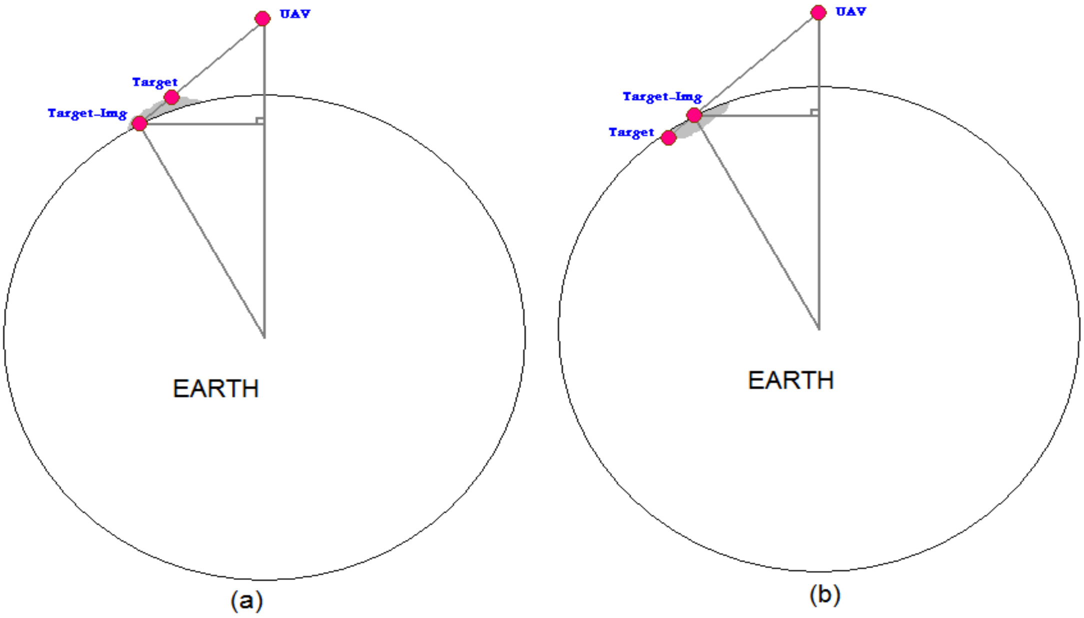


Figure3.4: Target Level

(a) Target is above mean sea level

(b) Target is below mean sea level

The main equation to calculate the angular distance between the projection of the UAV that holds the camera and the target on the earth surface is given as follows:

$$\mathbf{d = R\,\theta} \rightarrow (3.1)$$

Where:

d: Angular distance.

R: earth radius at target position.

θ: the angle between the perpendicular line from the UAV to the center of the earth and the real target position above the Mean Sea Level.

a. Target above Mean Sea Level :

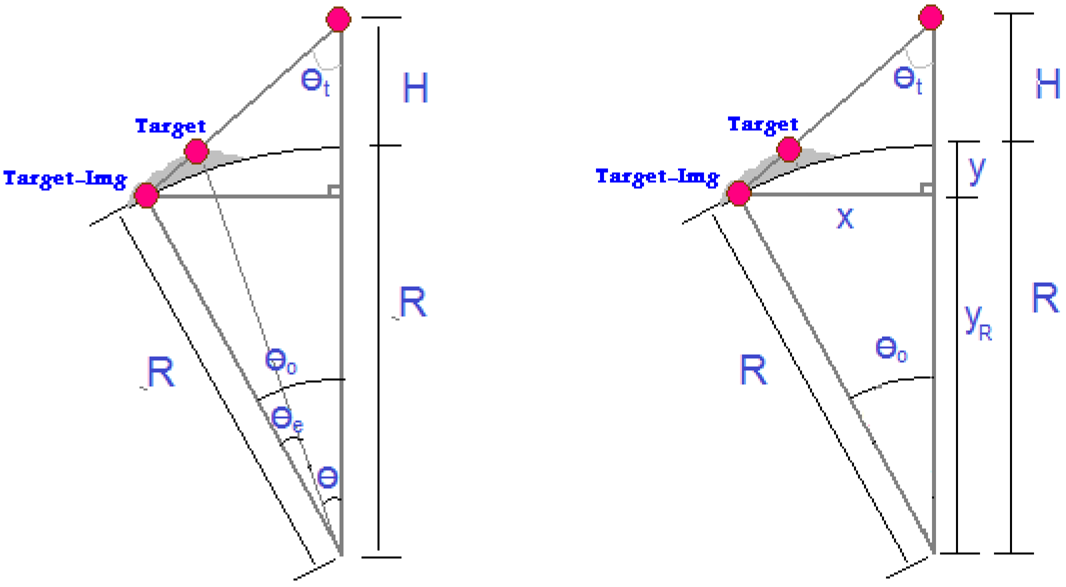


Figure3.5: Angles used in the derivation of $[\theta_o]$ in case [1]

From figure [5], to derive θ angle, we see that:

$$\boldsymbol{\theta} = \boldsymbol{\theta_o} - \boldsymbol{\theta_e} \qquad (3.2)$$

Where:

θ_o : is the angle between the line from the UAV to the center of the earth and the image of the target on the Mean Sea Level.

θ_e : is the angle between the image of the target on the Mean Sea Level and the real position of the target above the Mean Sea Level.

So, we need to derive two equations, for θ_o and θ_e. For that we have to know some parameters initially as given inputs. These parameters are:

- H: is the UAV Altitude and it is given from the GPS.

- θ_t :is the angle between the perpendicular line from the UAV to the center of the earth and the line of sight of the camera to the target (we will name it as Camera tilt angle) .

- H': is the target Altitude. And this value is estimated according to previous history about the nature of the area, because there is no specific way to know the exact altitude of the target before knowing its position firstly.

- R: is the radius of the earth. Because the earth is not exactly spherical, the radius varies slightly from one area to another.

We will take the average as value to the radius of the earth which equals 6371 km.

i- θ_o Equation Derivation:

$$\mathrm{y}_R = R\cos\theta_0$$

$$y = R - \mathrm{y}_R = R(1-\cos\theta_0)$$

$$x = (y + H\ )\tan\theta_t = \ (R(1-\cos\theta_0) + H\ )\tan\theta_t$$

$$\sin\theta_0 = \frac{x}{R} = \frac{(R(1-\cos\theta_0) + H\ )\tan\theta_t}{R}$$

$$\begin{aligned}\mathrm{R}\sin\theta_0 = &\ \ (R(1-\cos\theta_0) + H\ )\tan\theta_t \\ &= (R+H)\tan\theta_t - \mathrm{R}\ \tan\theta_t\cos\theta_0\end{aligned}$$

Or

$$\mathrm{R}\sin\theta_0 + \ \mathrm{R}\ \tan\theta_t\cos\theta_0 = \ (R+H)\tan\theta_t$$

Put $\boldsymbol{R = A}$ **,** $\boldsymbol{B = \ R\ tan\,\theta_t}$ ***and*** $\boldsymbol{C = (R+H)\,tan\,\theta_t}$

$$\mathrm{A}\sin\theta_0 + \ \mathrm{B}\cos\theta_0 = \ C$$

Let $\boldsymbol{\emptyset = tan^{-1}\frac{A}{B}}$

$$\tan \emptyset = \frac{A}{B}$$

$$D = \sqrt{A^2 + B^2}$$

$$\sin \emptyset = \frac{A}{D} \; , \qquad A = D \sin \emptyset$$

$$\cos \emptyset = \frac{B}{D} \; , \qquad B = D \cos \emptyset$$

$$\mathrm{A} \sin \theta_0 + B \cos \theta_0 = D \sin \emptyset \sin \theta_0 + D \cos \emptyset \; \cos \theta_0 = C$$

From sum to Product Formulas in trigonometry, we will use the following formula to rewrite the above equation:

$$\mathbf{cos(a - b) = cos\,(a)\,cos(\,b) + sin(\boldsymbol{a})\;\; sin(\boldsymbol{b})}$$

$$D \sin \emptyset \sin \theta_0 + D \cos \emptyset \; \cos \theta_0 = \mathrm{D} \cos(\theta_0 - \emptyset) = C$$

$$\theta_0 = \cos^{-1}\left(\frac{C}{D}\right) + \emptyset$$

$$\emptyset = \tan^{-1}\frac{A}{B} = \tan^{-1}\frac{R}{R \; \tan \theta_t} = \tan^{-1}\left(\frac{1}{\tan \theta_t}\right)$$

$$\theta_0 = \cos^{-1}\left(\frac{(R + H) \tan \theta_t}{\sqrt{(R)^2 + (\mathrm{R} \; \tan \theta_t)^2}}\right) + \tan^{-1}\left(\frac{1}{\tan \theta_t}\right)$$

Finally we get:

$$\boxed{\theta_0 = \cos^{-1}\left(\frac{\left(\frac{R+H}{R}\right)\tan\theta_t}{\sqrt{1+(\tan\theta_t)^2}}\right) + \tan^{-1}\left(\frac{1}{\tan\theta_t}\right) \rightarrow (3.3)}$$

ii- θ_e Equation Derivation:

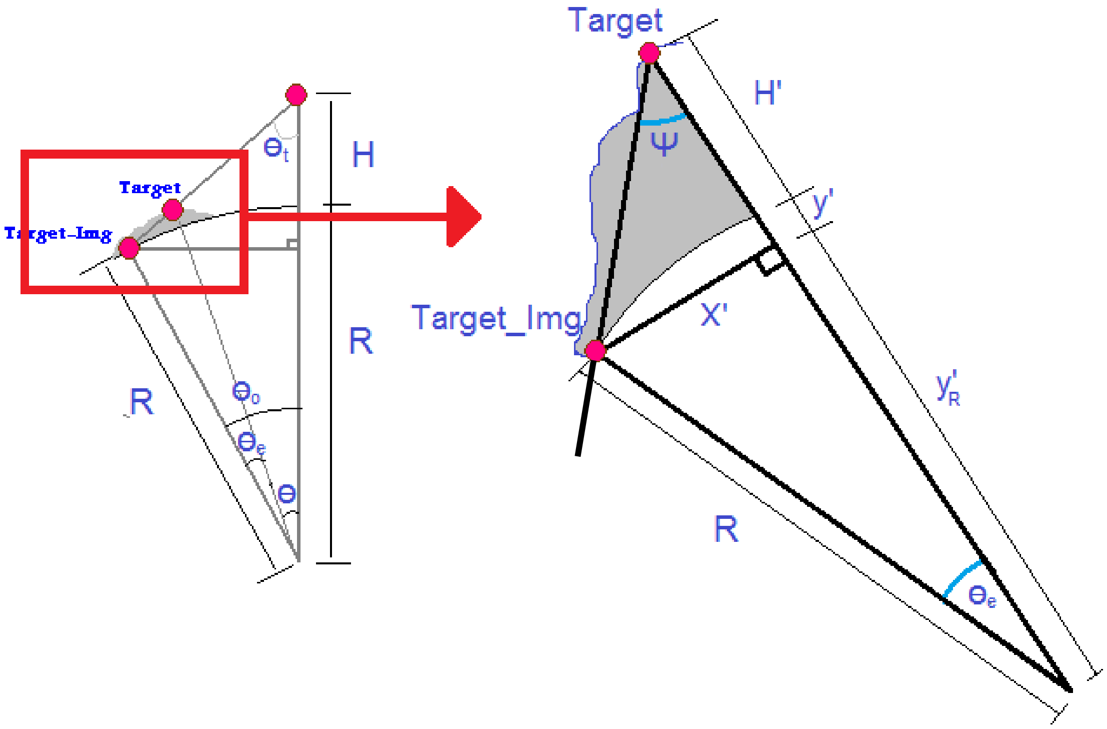


Figure3.6: Angles used in the derivation of [θ_e] in case [1]

To derive θ_e new angles appears:

- Ψ: The angle between the target to the earth center and the target image to the center.
- Ω: The angle between the extension of the line from the target to the target image and the radius (R) line.

$$\theta = \theta_o - \theta_e$$

$$\Omega = \theta_t + \theta_o$$

$$\Psi = \theta_t + \theta = \theta_t + \theta_o - \theta_e = \Omega - \theta_e$$

$$y'_R = R\cos\theta_e$$

$$y' = R - y'_R = R(1-\cos\theta_e)$$

$$x' = (y' + H')\tan\Psi = (R(1-\cos\theta_e) + H')\tan\Psi$$

$$\sin\theta_e = \frac{x'}{R} = \frac{(R(1-\cos\theta_e) + H')\tan\Psi}{R}$$

$$\begin{aligned} \mathrm{R}\sin\theta_e &= (R(1-\cos\theta_e) + H')\tan\Psi \\ &= (R + H')\tan\Psi - \mathrm{R}\cos\theta_e\,\tan\Psi \end{aligned}$$

$$\mathrm{R}\sin\theta_e = \frac{(R+H')\sin\Psi - \mathrm{R}\cos\theta_e\,\sin\Psi}{\cos\Psi} = \frac{(R+H')\sin(\Omega - \theta_e) - \mathrm{R}\cos\theta_e\,\sin(\Omega - \theta_e)}{\cos(\Omega - \theta_e)}$$

$$\mathrm{R}\sin\theta_e\cos(\Omega - \theta_e) = (R + H')\sin(\Omega - \theta_e) - \mathrm{R}\cos\theta_e\,\sin(\Omega - \theta_e) \rightarrow (2\,*)$$

$$\boxed{\Omega = \theta_t + \theta_o \qquad \rightarrow (3.4)}$$

From Product of Sum Formulas in trigonometry, we will use the following "sin(a)cos(b)" product formula to rewrite the above equation:

$$\mathbf{sin(a)\,cos}(\boldsymbol{b}) = \frac{\boldsymbol{sin\,(a+b) + sin\,(a-b)}}{\mathbf{2}}$$

Then we can rewrite Equation (*2) as follows:

$$R\left(\frac{\sin\Omega + \sin(2\theta_e - \Omega)}{2}\right) = (R + H')\sin(\Omega - \theta_e) - R\left(\frac{\sin\Omega - \sin(2\theta_e - \Omega)}{2}\right)$$

$R \sin \Omega + \mathrm{R} \sin(2\theta_e - \Omega) = 2(R + H') \sin(\Omega - \theta_e) - R \sin \Omega + \mathrm{R} \sin(2\theta_e - \Omega)$

$$R \sin \Omega = (R + H') \sin(\Omega - \theta_e)$$

$$\sin(\Omega - \theta_e) = \frac{R}{R + H'} \sin \Omega$$

Or

$$\boxed{\theta_e = \Omega - \sin^{-1}\left(\frac{R}{R + H'} \sin \Omega\right) \rightarrow (3.5)}$$

These four equations shown below are used to calculate the angle **θ**, which is used (in equation [1]) to calculate the distance of range projection from UAV to target on the earth surface (in case of Target above Mean Sea Level).

$$\boxed{\boldsymbol{\theta = \theta_o - \theta_e} \qquad \rightarrow (3.6)}$$

$$\boxed{\boldsymbol{\theta_0 = \cos^{-1}\left(\frac{\left(\frac{R+H}{R}\right) \tan \theta_t}{\sqrt{1 + (\tan \theta_t)^2}}\right) + \tan^{-1}\left(\frac{1}{\tan \theta_t}\right)} \quad \rightarrow (3.7)}$$

$$\boxed{\boldsymbol{\theta_e = \Omega - \sin^{-1}\left(\frac{R}{R+H'} \sin \Omega\right)} \qquad \rightarrow (3.8)}$$

Also equation (5) could be simplified by using the above equations (7, 8).

$$\theta = \theta_o - \theta_e = \theta_o - \Omega + \sin^{-1}\left(\frac{R}{R + H'} \sin \Omega\right)$$

$$
\begin{aligned}
\theta \;=\; \theta_o - \;\; \theta_t + \theta_o + \sin^{-1}\left(\frac{R}{R+H'}\sin\Omega\right) \\
=\; \sin^{-1}\left(\frac{R}{R+H'}\sin\Omega\right) - \theta_t
\end{aligned}
$$

$$\boxed{\boldsymbol{\theta} = \sin^{-1}\left(\frac{\boldsymbol{R}}{\boldsymbol{R+H'}}\sin\boldsymbol{\Omega}\right) - \boldsymbol{\theta_t} \qquad \rightarrow (3.9)}$$

B. Target below Mean Sea Level:

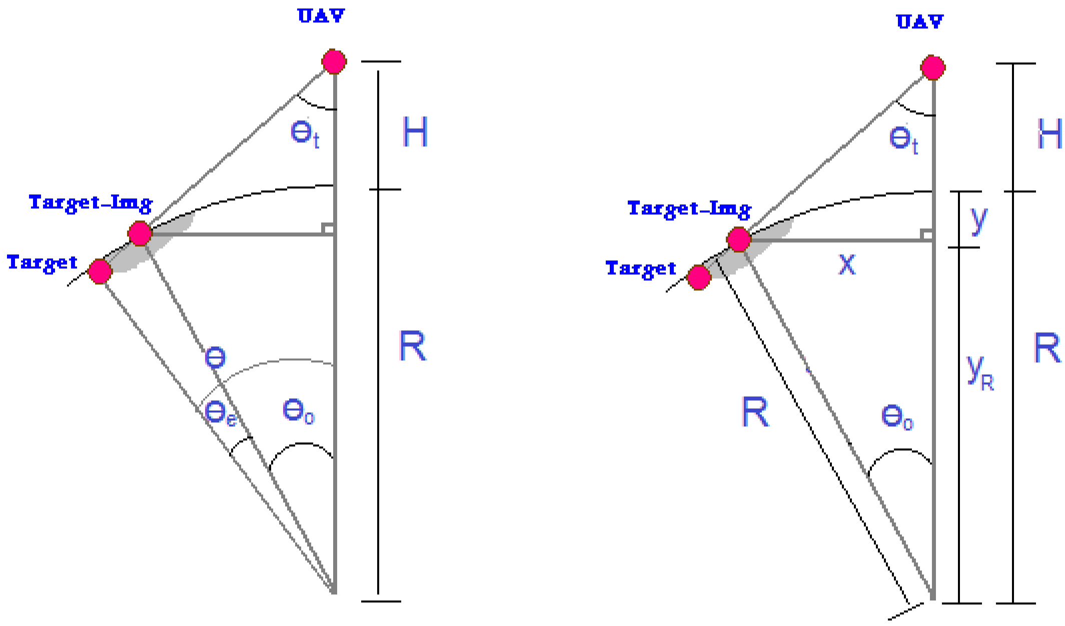


Figure 3.7: Angles used in the derivation of $[\theta_o]$ in case [2]

As we have seen in previous steps in case (a), to drive the angle θ equation there are two derivations for θ angle components, In this case where the target is below the Mean Sea Level we see that:

$$\theta \;=\; \theta_o + \;\; \theta_e$$

So we have to derive the two equations, the first one for θ_o and the other for θ_e.

But as shown in figure [6] above, the derivation for θ_o equation is based on the same assumptions as in the previous case, and then the derivation is same.

$$\boxed{\theta_0 = \cos^{-1}\left(\frac{\left(\frac{R+H}{R}\right)\tan\theta_t}{\sqrt{1\ +(\tan\theta_t)^2}}\right) + \tan^{-1}\left(\frac{1}{\tan\theta_t}\right) \rightarrow (3.10)}$$

- **θ_e Equation Derivation:**

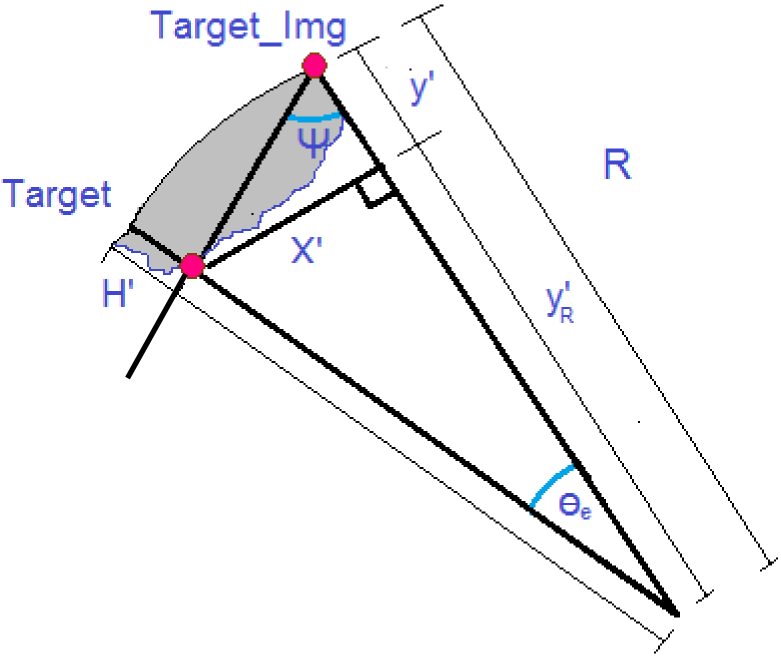


Figure 3.8: Angles used in the derivation of [θ_e] in case [2]

To derive θ_e the same angles are used:

$$\theta = \theta_o + \theta_e$$

$$\Omega = \theta_t + \theta_o$$

$$\Psi = \theta_t + \theta_o = \Omega$$

But because the target is below the Mean Sea Level, we find that:

$$\mathrm{y}'_R = (R - \mathrm{H}')\cos\theta_e$$

Instead of $(R\cos\theta_e)$ as in the previous case.

$$y' = R - \mathrm{y}'_R = R - (R - \mathrm{H}')\cos\theta_e = R(1-\cos\theta_e) + H'\ \cos\theta_e$$

$$x' = y'\tan\mathit{\Psi} = \ (R(1-\cos\theta_e) + H'\ \cos\theta_e)\tan\mathit{\Psi}$$

$$\sin\theta_e = \frac{x'}{R-H'} \ = \frac{(R(1-\cos\theta_e) + H'\ \cos\theta_e)\tan\mathit{\Psi}}{(R-H')}$$

$$(\mathrm{R}-H')\sin\theta_e = \ \ (R(1-\cos\theta_e) + H'\ \cos\theta_e)$$
$$= R\ \tan\mathit{\Omega} - (\ \mathrm{R}-\mathrm{H}')\cos\theta_e\tan\mathit{\Omega}$$

Or

$$(\mathrm{R}-H')\sin\theta_e + \ (\ \mathrm{R}-\mathrm{H}')\ \tan\mathit{\Omega}\ \cos\theta_e = \ R\ \tan\mathit{\Omega}$$

put $\boldsymbol{(\ R - H') = A,\ B = \ (\ R - H')\ \tan\mathit{\Omega},,\ C = R\ \tan\mathit{\Omega}}$

$$\mathrm{A}\sin\theta_e + \ \mathrm{B}\cos\theta_e = \ C$$

Let $\boldsymbol{\emptyset = \tan^{-1}\frac{A}{B}}$

$$\tan\emptyset = \frac{A}{B}$$

$$D = \sqrt{A^2 + B^2}$$

$$\sin\emptyset = \frac{A}{D}\ , \qquad A = \ D\sin\emptyset$$

$$\cos\emptyset = \frac{B}{D}\ , \qquad B = D\cos\emptyset$$

$$\mathrm{A}\sin\theta_e + B\cos\theta_e = D\sin\emptyset\sin\theta_e + D\cos\emptyset\ \cos\theta_e = C$$

From sum to Product Formulas in trigonometry, we will use the following formula to rewrite the above equation:

$$\mathbf{cos(a - b) = cos\ (a)\ cos(\ b) + sin(\boldsymbol{a}\ )\ \ sin(\boldsymbol{b})}$$

$$D\sin\emptyset\sin\theta_e + D\cos\emptyset\ \cos\theta_e = \mathrm{D}\cos(\theta_e - \emptyset) = C$$

$$D\cos(\theta_e - \emptyset) = C$$

$$\theta_e = \cos^{-1}\left(\frac{C}{D}\right) + \emptyset$$

$$\emptyset = \tan^{-1}\frac{A}{B} = \tan^{-1}\frac{(\mathrm{R}-\mathrm{H}')}{(\mathrm{R}-\mathrm{H}')\ \tan\Omega} = \tan^{-1}\frac{1}{\tan\Omega}$$

$$\theta_e = \cos^{-1}\left(\frac{R\ \tan\Omega}{\sqrt{(R-\mathrm{H}')^2 + \left((\mathrm{R}-\mathrm{H}')\ \tan\Omega\right)^2}}\right) + \tan^{-1}\left(\frac{1}{\tan\Omega}\right)$$

These four equations shown below are used to calculate the angle theta, which is used (in equation [1]) to calculate the distance of range projection from UAV to target on the earth surface (in case of Target below Mean Sea Level).

$$\boldsymbol{\theta = \theta_o + \theta_e} \qquad \rightarrow(3.11)$$

$$\boldsymbol{\theta_0 = \cos^{-1}\left(\frac{\left(\frac{R+H}{R}\right)\tan\theta_t}{\sqrt{1+(\tan\theta_t)^2}}\right) + \tan^{-1}\left(\frac{1}{\tan\theta_t}\right)} \qquad \rightarrow(3.12)$$

$$\boldsymbol{\theta_e = \cos^{-1}\left(\frac{R\,\tan\Omega}{\sqrt{(R-H')^2+\left((R-H')\,\tan\Omega\right)^2}}\right) - \tan^{-1}\left(\frac{1}{\tan\Omega}\right)} \rightarrow (3.13)$$

$$\boldsymbol{\Omega = \theta_t + \theta_o} \qquad \rightarrow (3.14)$$

As we have seen, the distance of range projection from UAV to target on the earth surface is calculated by the equations (1) to (14);

The Distance can be calculated directly from Pythagoras' theorem, by ignoring the curvature of the earth.

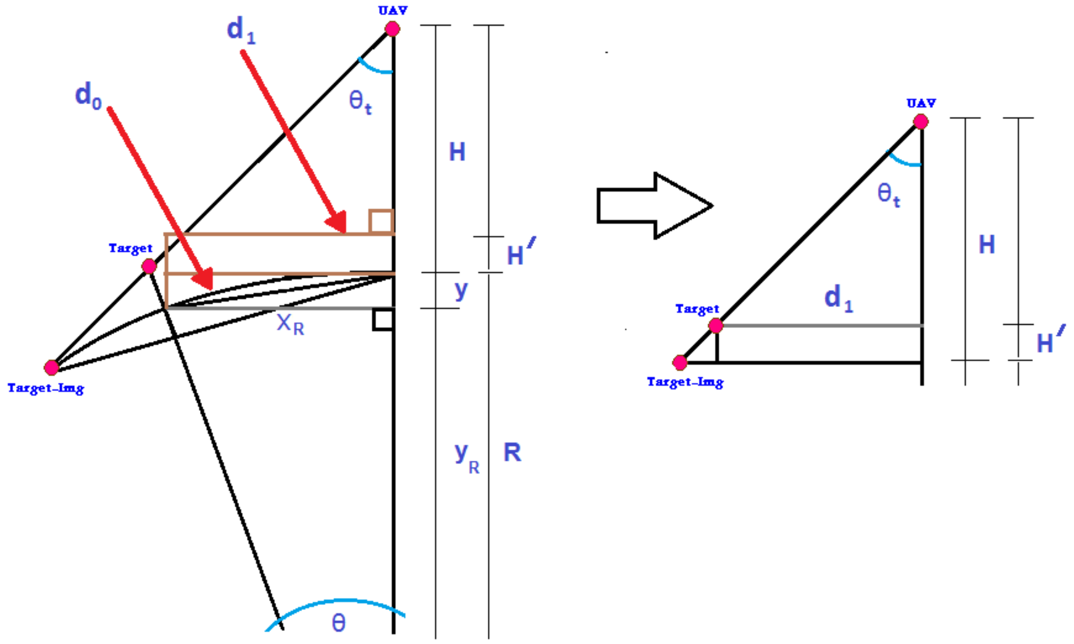


Figure 3.9: Deriving of the distance by ignoring the curvature of the earth

In figure 3.9 the distance is denoted by $d_{1,}$ and given by the following equations:

(a) Target above Mean Sea Level:

$$\mathbf{d_1 = (\mathit{H} - \mathit{H'})\tan\theta_t} \quad \rightarrow (3.15)$$

(b) Target below Mean Sea Level:

$$\mathbf{d_1 = (\mathit{H} + \mathit{H'})\tan\mathit{\theta_t}} \quad \rightarrow (3.16)$$

3.2.2- Calculating the target coordinates:

If a great-circle journey is started from a location L0 in an initial direction A0 and the travelled distance is D, the coordinates of the destination location L1 can be found by solving the spherical triangle for the side "**90°-Lat1**" and the angle "**Lon1-Lon0**".

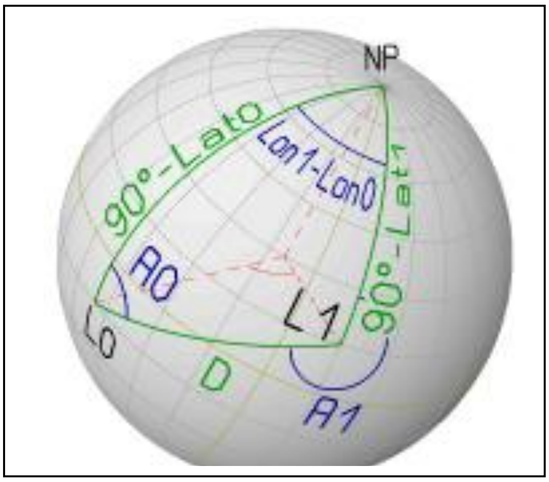


Figure 3.10: Spherical Law of cosines and Tangents applied to the earth [11]

Applying the "Law of Cosines for sides" for "**90°-Lat1**" obtains the latitude of L1:

$$\cos(90^{\circ}\text{-}Lat1) = \cos(90^{\circ}\text{-}Lat0)*\cos(D) + \sin(90^{\circ}\ Lat0)*\sin(D)*\cos(A0)$$

$$\sin(Lat1) = \sin(Lat0)*\cos(D) + \cos(Lat0)*\sin(D)*\cos(A0)$$

$$Lat1 \quad = \mathrm{asin}[\sin(Lat0)*\cos(D) + \cos(Lat0)*\sin(D)*\cos(A0)]$$

The Longitude of L1 is obtained by applying the "Law of Tangents" to "**Lon1-Lon0**":

$$\tan(Lon1\text{-}Lon0) = \sin(D)*\sin(A0) \ / \ [\cos(D)*\sin(90^{\circ}\text{-}Lat0) - \cos(A0)*\sin(D)*\cos(90^{\circ}\text{-}Lat0)]$$

$$= \sin(D)*\sin(A0) \ / \ [\cos(D)*\cos(Lat0) - \cos(A0)*\sin(D)*\sin(Lat0)]$$

The summarized results yielding the coordinates of a destination (Lat1, Lon1) for a given departure (Lat0, Lon0), initial bearing (A0) and distance (D):

$$\boxed{Lat1 = \mathrm{asin}[\sin(Lat0)*\cos(D) + \cos(Lat0)*\sin(D)*\cos(A0)] \rightarrow (3.17)}$$

$$\boxed{Lon1=Lon0+\mathrm{atan2}(\sin(D)*\sin(A0),\cos(D)*\cos(Lat0)\cos(A0)*\sin(D)*\sin(Lat)) \rightarrow (3.18)}$$

The distance (D) in the equations represents what is called the great-circle distance which equals the angular distance divided by the radius of the earth (d/R). By applying these two equations (17) and (18) we will get the latitude and longitude of the target noting that all the angles should be converted to radian.

Chapter Four: Results and Discussion

In the previous chapter a derivation to the circle distance between a projection's point of a camera that observes from the air and a target on the ground. In this chapter the proposed formula for calculating the circle distance is compared to the distance if we used Pythagoras' theorem. Then, we calculate the coordinates of the target for each method and analyze the results. To apply Pythagoras' theorem, pictures taken by a camera mounted on a UAV are used to calculate the target distance and coordinates.

4.1- The compared method Description:

A (Canon Power Shot Sx50 HS) Camera with the following specifications was used in the test:

Table 4.1: specifications of the camera used in the test

Camera detector size (pixels)	Max resolution 5616 x 3744
Camera detector size (m)	36x24mm
Camera optical focal length (m)	35mm

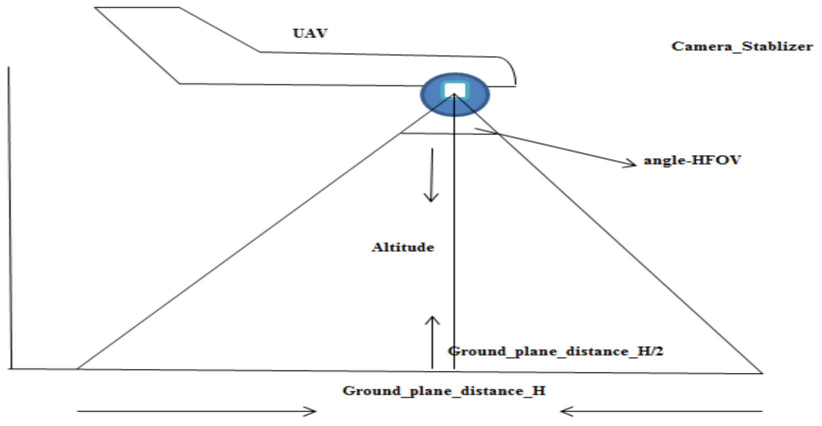


Figure4.1: shows the main parameters in the testing system

At first we need to calculate the following parameters:

i- The Vertical Field of view of the camera above, which is the angle covered by the camera vertically:

$$\text{VFOV} = 2\text{atan}\left(\frac{\frac{detector\_size\_vert}{2}}{camera\_focal\_lens}\right) = 0.1896 radian \qquad (4.1)$$

So the vertical ground plane for a picture taken by this camera at a specific altitude is:

$$\text{Ground\_plane distanceV} = 2\tan\left(\frac{VFOV}{2}\right) xaltitude \qquad (4.2)$$

ii- The horizontal Field of view of the camera above, which is the angle covered by the camera horizontally:

$$\text{HFOV} = 2\text{atan}\left(\frac{\frac{detector\_size\_Hort}{2}}{camera\_focal\_lens}\right) = 0.2556 \text{ radian} \qquad (4.3)$$

So the horizontal ground plane for a picture taken by this camera at a specific altitude is:

$$\text{Ground\_plane\_distance\_H} = 2\tan\left(\frac{HFOV}{2}\right) xaltitude \qquad (4.4)$$

iii- The Vertical instantaneous Field of view :

$$\text{IFOV\_V} = \text{atan}\left(\frac{pix\_size\_V}{camera\_focal\_lens}\right) \qquad (4.5)$$

$$\text{IFOV\_V} = 6.32\text{x}10^{-5} \quad \text{radian}$$

So the vertical ground plane resolution for a picture taken by this camera at a specific altitude is:

$$\text{Ground\_plane\_resolution\_V} = \tan(IFOV)xaltitude \qquad (4.6)$$

iv- The Horizontal Instantaneous Field of View:

$$\text{IFOV\_H} = \text{atan}\left(\frac{pix\_size\_H}{camera\_focal\_lens}\right) \ \text{(radian)} \qquad (4.7)$$

So the vertical ground plane resolution for a picture taken by this camera at a specific altitude is:

$$\text{Ground\_plane\_resolution\_H} = \tan(IFOV)xaltitude \qquad (4.8)$$

By applying these formulas for different altitudes as an example, we get the following table:

Table 4.2: Samples for calculating the camera parameters at different altitudes

Altitude	Ground_plane_distance_H	Ground_plane_distance_V	Ground_plane_resolution_H
540m	555 m	370 m	0.099 m/pixel
755m	776 m	517 m	0.138 m/pixel

These parameters are necessary to calculate different targets on pictures taken of different altitudes.

4.2- The Analysis procedure:

It will be as follows:

i- Firstly, we calculate the distance to the target, given:

- The first point coordinates which is the location of the camera on the earth (longitude, latitude and altitude).
- The heading of the camera.

- The tilt angle of the camera.
- The estimated altitude of the target.

And then we use the formulas (1, 5, 6, 7) as described in the previous chapter if the target is above the Mean Sea Level or (1, 10, 11, 12, 13) if the target is below the Mean Sea Level.

ii- Secondly, we apply the formulas (16) and (17) to find the longitude and the latitude of the target.

iii- Using the same inputs in step (i) to calculate the distance using Pythagoras' theorem (by taking targets on different pictures).

iv- Repeating step (ii) to calculate the coordinates with the new distance.

v- Comparing the distance in the two cases (results of step (i) compared to results of step (iii)).

vi- Comparing the coordinates in the two cases (results of step (ii) compared to results of step (iv)).

4.3 Analysis Result:

The following pictures were taken at Longitude(32.4925756) and Latitude(15.8179897) at three different altitudes (500,700 and 1000) meters respectively. Six targets were chosen on the pictures.the diagonal number of the pixels from the center of the frame is calculated .

Table 4.3: Show the taken targets on the pictures and their location on the frame

Target	Colom x_loct/ pixels	Row y_loct/pixels	Diognal pixles
1	1101	1820	1708
2	860	2476	2039
3	580	3452	2731
4	1029	396	2311
5	540	916	2446

then the distance to the target is calculated by knowing the ratio between the number of pixels and the distance represnted by each one based on the previous calculations of the Field of View,instantaneous Field of view and the plane distance (horizontally anf vertically).

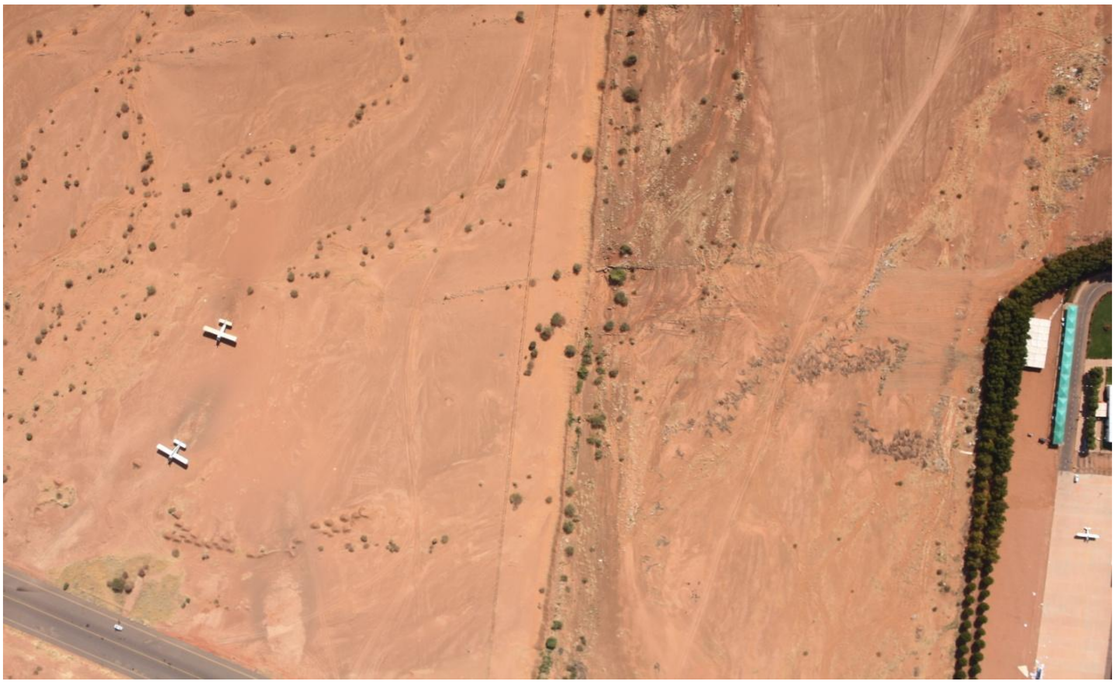
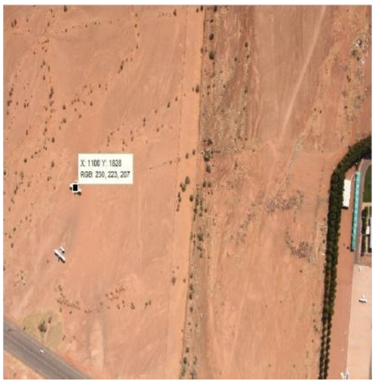
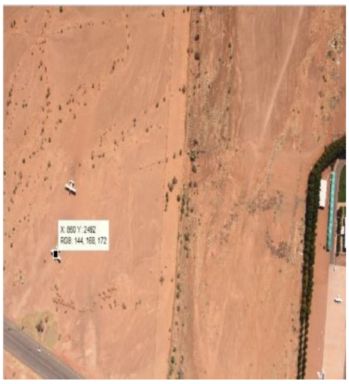


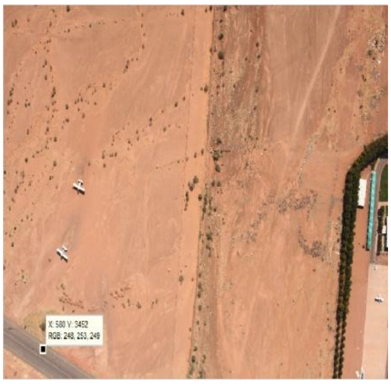
Figure4.2: original picture taken at altitude (500m)



(a)



(b)



(b)

Figure4.3: targets (a) (b) and (c) at the previous picture

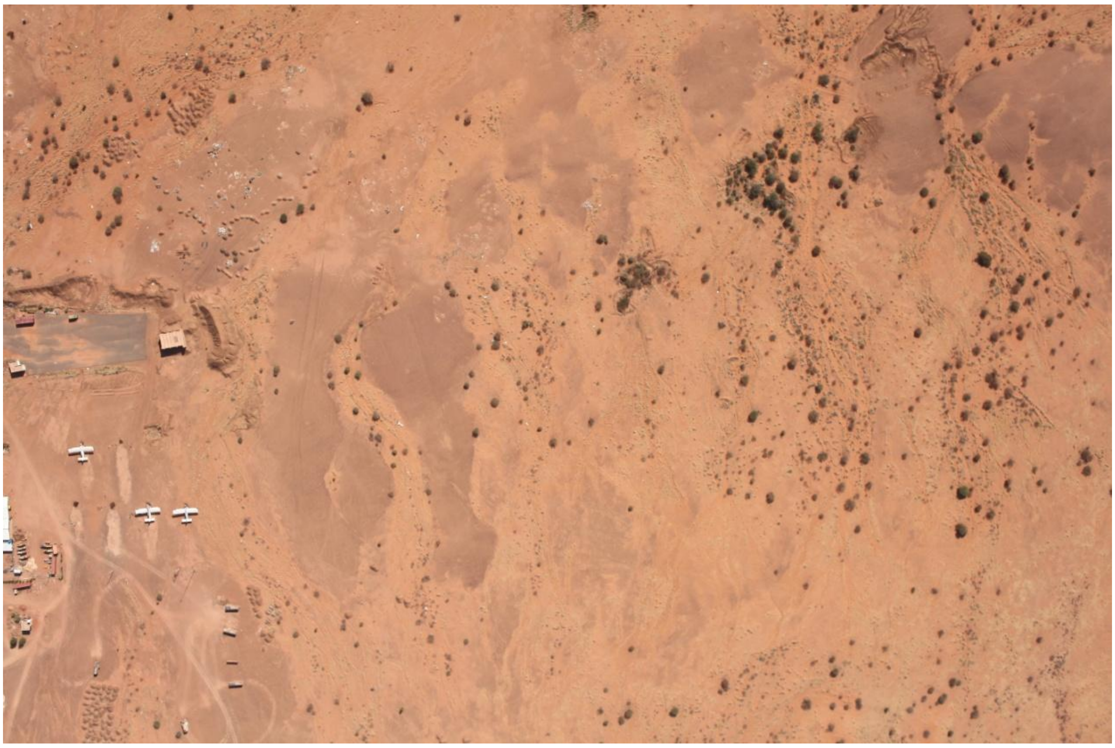
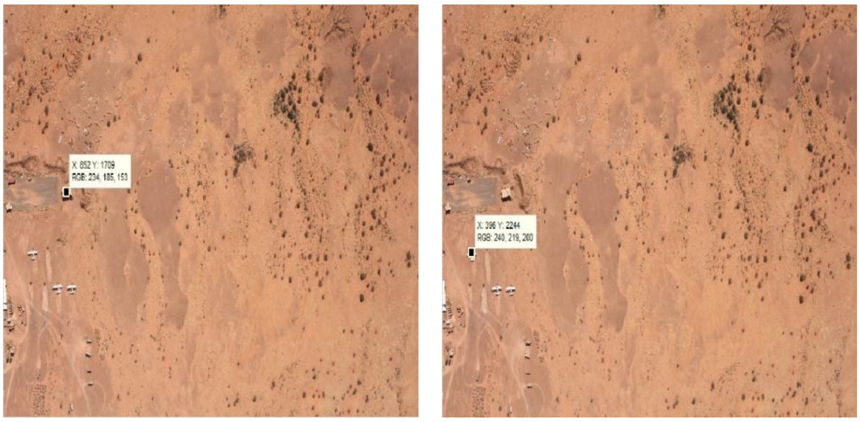


Figure4.4: picture taken at altitude (700m)



(a) (b)

Figure4.5: targets (a) and (b) at the previous picture

- We get the following result :

Table 4.4: Parameters of each target on the picture

Tilte angle /d	CIFOV deg/pixels	Number of pixel	Pixels per me	Uav Alt	Target dist
17.08	0.01	1708	0.099	540	169.09 m
20.39	0.01	2039	0.099	540	201.86 m
27.31	0.01	2731	0.099	540	270.37 m
22.42	0.01	2311	0.138	755	318.92 m
23.57	0.01	2446	0.138	755	337.55 m

-By using the proposed method, we get the following :

Table 4.5: Distance to the targets using both methods

Proposed Method	Camera parameters Method	Distance Errors
167.289 m	169.09 m	1.801 m
220.694 m	201.86 m	18.789 m
287.404 m	270.37 m	17.034 m
309.350 m	318.92 m	9.57 m
332.962 m	337.55 m	4.588 m

Table 4.6: Actual data about the UAV and the chosen targets locations of figure (4.3)

	LAT	LONGT	ALT(m)	HEADING	TILT ANGLE	DISTANCE(m)
UAV1	15.8075251	32.499422	935.555	-	-	-
TARGET A	15.806914	32.500851	395	113.86	17.197	167.3
TARGET B	15.807455	32.501382	395	92.14	22.21	219.89
TARGET C	15.808256	32.502006	395	37.84	28	287.37

Table 4.7: Actual data about the UAV and the chosen targets locations of figure (4.4)

	LAT	LONGT	ALT(m)	HEADING	TILT ANGLE	DISTANCE(m)
UAV2	15.8179897	32.4925756	1164.648	-	-	-
TARGET D	15.81565	32.49417	392	146.38	21.821	309.36
TARGET E	15.81611	32.49501	391	129	23.314	333.41

Table 4.8: The derived distance and coordinates of the chosen targets of figures (4.3) and (4.4) using the proposed method

	Derived Distance	Derived Lat	Derived Longt
TARGET A	167.289	15.806921	32.500854
TARGET B	220.694	15.807451	32.501483
TARGET C	287.404	15.809566	32.50107
TARGET D	309.350	15.815673	32.494177
TARGET E	332.962	15.816105	32.494994

4.4- Direct implementation of Pythagoras theorem:

If we took the location (16.69225, 78.915033) as an example, and fixed the camera altitude to 5 Km and the target altitude to 0.289 Km, by changing the tilt angle:

Table 4.9: Calculation Results after fixing the altitude and changing the tilt angle

Result	Tilt Angle	Derived Distance	
		Proposed Method	Pythagoras' theorem
1	5	0.412141565021	0.412159094
2	13	1.087592151314	1.087620048
3	22	1.903396118238	1.90336755
4	31	2.830903979936	2.830654376
5	38	3.681300788492	3.680636586
6	42	4.242883461491	4.241803453
7	50	5.617048209296	5.614351175
8	59	7.848111855698	7.840420641
9	66	10.600411447234	10.581079242
10	72	14.549452554656	14.498967144
11	78	22.346937029683	22.163512446
12	80	27.041749131639	26.717408652
13	85	56.732442764959	53.846976398
14	87	107.053957198154	89.891234936

We see that the increase in the tilt angle increases the distance to the observed target. This increasing in the distance is not linear because for small angles the difference between their distances is small (for example the difference between 22° and 31° is less than 1 Kilometers) but when it goes high the difference increases (for example the difference between 78° and 80° is near to 5 Kilometers).

Also, the difference between the calculated distance by each method increases when the tilt angle is increased. That is because of the curvature of the earth. In the method that uses Pythagoras' theorem the distance is taken as direct line which is not correct.

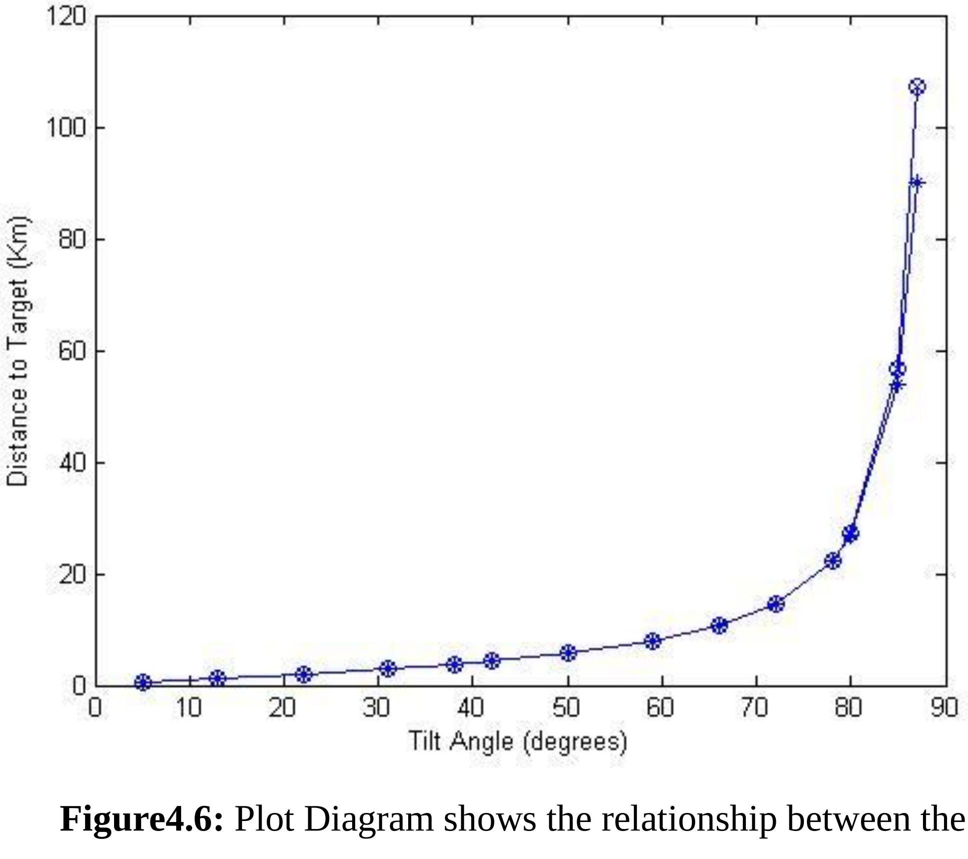


Figure4.6: Plot Diagram shows the relationship between the Tilt angle and the distance to the target

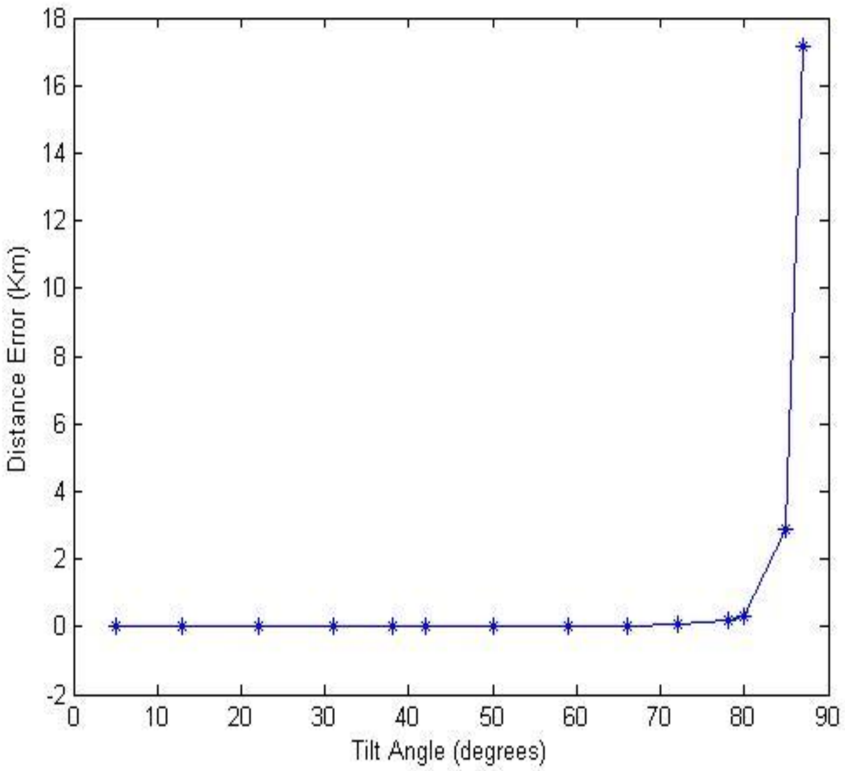


Figure4.7: Plot Diagram shows the relationship between the Tilt angle and the distance Error (off the target)

Again, If we took the first location (16.69225, 78.915033), But this time fixed the tilt angle randomly to (70) degrees and the target altitude remains the same. By changing the height (altitude) of the camera:

We notice the proportional relationship between the Height of the camera (the altitude) and the distance to the target. This time the relationship is linear. This is logical because whether the distance is curve or direct line the calculations will be similar.

But again the difference between the calculated distances in each method is not linear.

Table 4.10: Calculation Results after fixing the tilt angle and changing the altitude

Result	Cam Alt(H). (km	Derived Distance	
		Proposed Method	Pythagoras' theorem
1	0.5	0.579763917241	0.579717736
2	1.7	3.879760718225	3.876690639
3	2.6	6.357849358385	6.349420316
4	3.3	8.287089742866	8.27265451
5	4.2	10.769912316313	10.745384187
6	5.5	14.360937038886	14.317104833
7	6.2	16.296887297699	16.240339026
8	6.9	18.234468158176	18.16357322
9	7.3	19.342390984851	19.262564188
10	7.8	20.728046469654	20.636302898
11	8.4	22.391937998957	22.284789349
12	8.8	23.501869951302	23.383780317
13	9.2	24.612339599207	24.482771285
14	10	26.834895247683	26.68075322

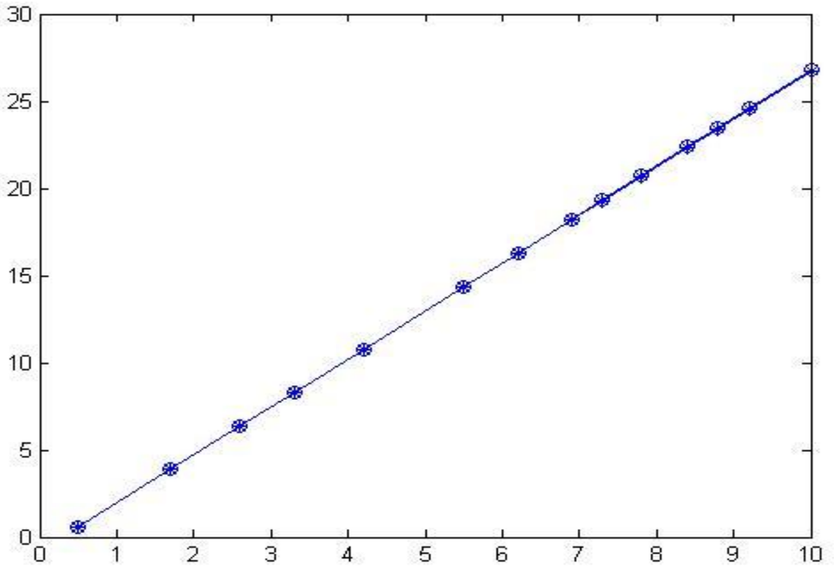


Figure4.8: Plot Diagram shows the relationship between the camera altitude and the distance to the target

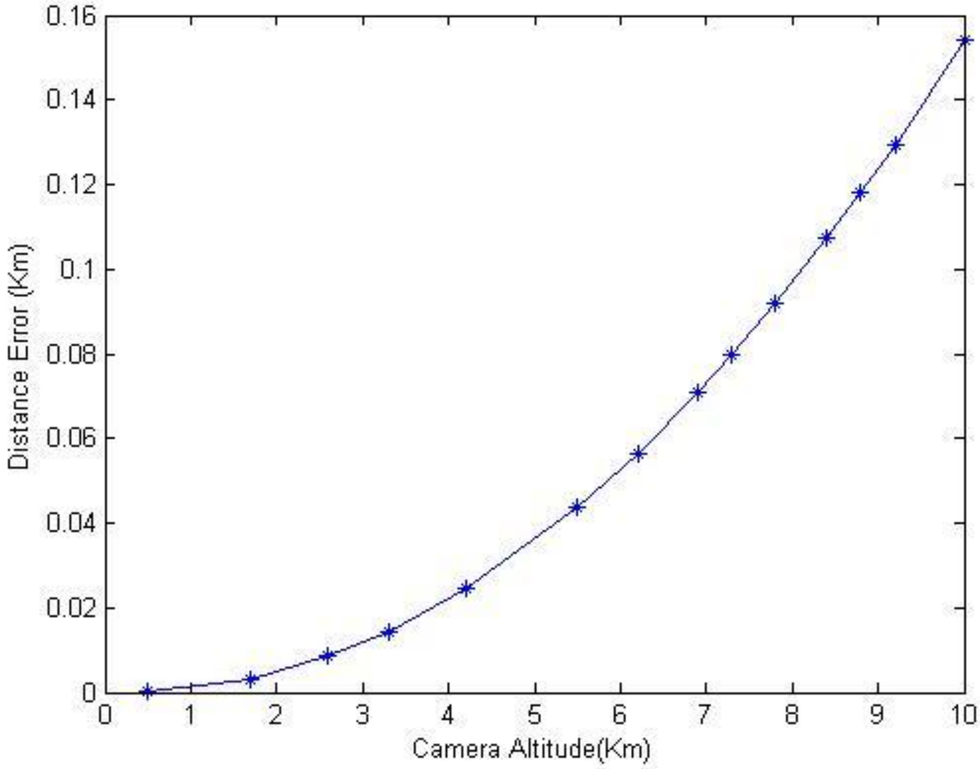
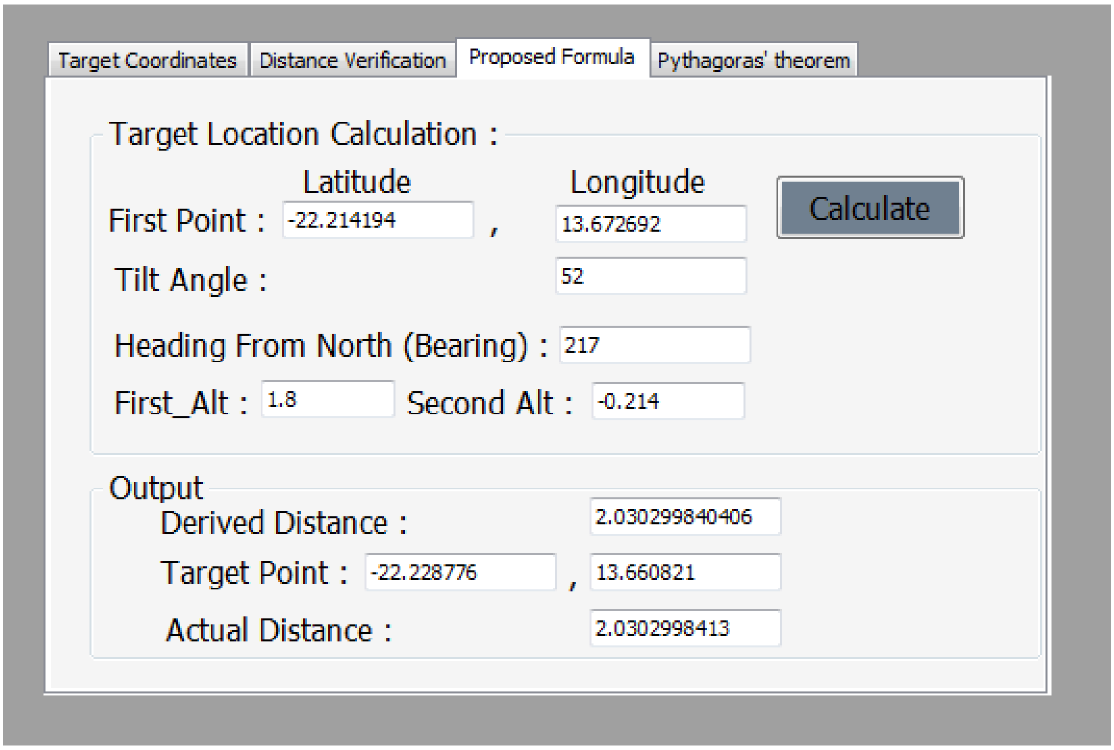


Figure4.9: Plot Diagram shows the relationship between the camera altitude and the distance Error (off the target)

4.5 Software:

In order to implement and analyze these formulas, a graphical user interface (GUI) was designed using C#.net that allows the user to enter the inputs, calculates the results and shows it back. It calculates the distance and the target coordinates by the proposed formula and using Pythagoras' theorem.

It has also the feature of calculating the distance between two points on the earth given the coordinates of them besides calculating the target point given the starting point, the heading and the distance. The last two options use known verified formulas used by most of the navigation systems.



Target Location Calculation :

Latitude Longitude

First Point : -22.214194 , 13.672692

Tilt Angle : 52

Heading From North (Bearing) : 217

First_Alt : 1.8 Second Alt : -0.214

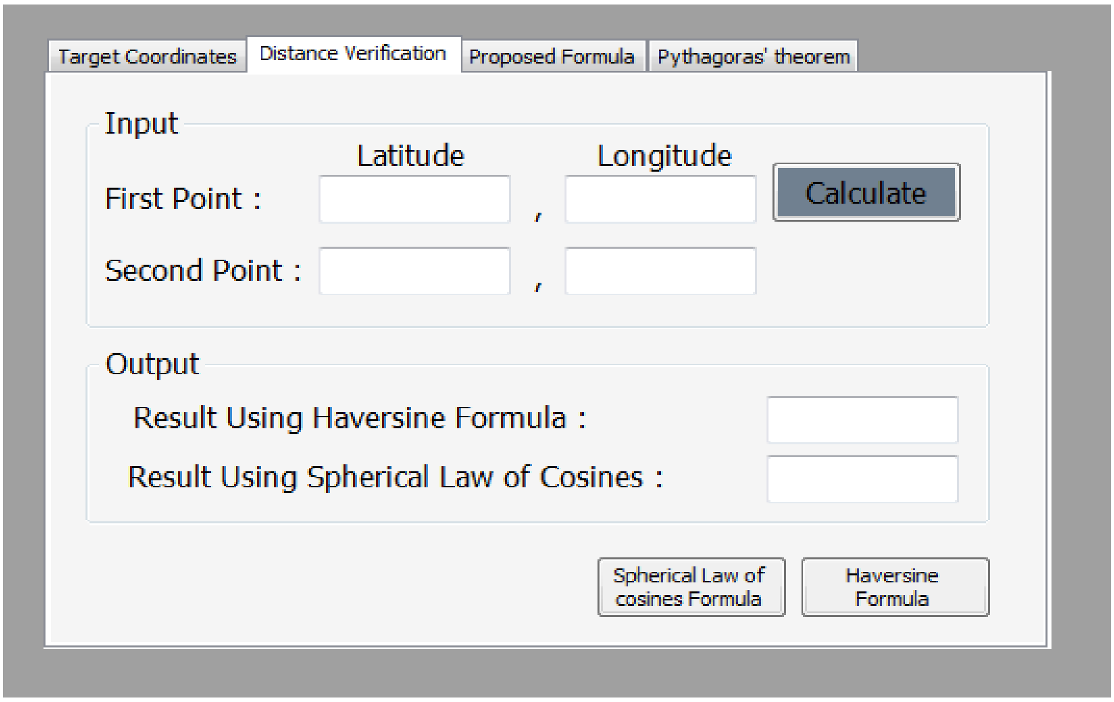
Output

Derived Distance : 2.030299840406

Target Point : -22.228776 , 13.660821

Actual Distance : 2.0302998413

Figure4.10: Calculating the formula on GUI



Input

Latitude Longitude

First Point : ,

Second Point : ,

Output

Result Using Haversine Formula :

Result Using Spherical Law of Cosines :

Figure4.11: Calculating the distance using both Spherical Law of cosines method and Haversine method

By taking the following locations randomly and applying them on the GUI we had the following Results:

Table 4.11: Random Locations taken to test the GUI

No.	Latitude	Longitude	Tilt Angle	Heading	Cam Alt(H). (km)	Target Alt(H'). (km)
1	16.69225	78.915033	60	45	2.6	0.289
2	7.626114	125.539803	23	110	5.7	0.283
3	-25.224267	-56.788336	64	230	3.8	0.072
4	64.360222	-115.776072	10	95	7.2	0.266
5	54.688383	25.280281	58	10	10.5	0.100
6	33.311839	67.205667	87	170	6.9	2.875
7	-20.086944	127.61125	44	308	0.5	0.294
8	-43.590597	172.61125	76	250	25	0.106
9	18.575375	-72.294711	35	80	12	0.030
10	13.15905	-87.974053	7	350	9	0.002
11	61.942553	11.440372	80	63	2	0.855
12	-20.519928	24.79715	15	145	4	0.916
13	-22.214194	13.672692	52	217	1.8	-0.214
14	-51.697658	-59.908697	3	60	2.3	-1.119

Table 4.9: Results taken from GUI

Locations	Outputs					
	Proposed Method			Pythagoras' theorem		
	Derived Dist.	Derived Lat.	Derived Long.	Derived Dist.	Derived Lat.	Derived Long.
Location 1	4.004768222328	16.717715	78.941624	4.002769416	16.717703	78.941611
Location 2	2.299454137706	7.619041	125.559408	2.299380079	7.619041	125.559408
Location 3	7.652872055756	-25.268495	-56.846637	7.643532721	-25.268441	-56.846565
Location 4	1.222620931557	64.359262	-115.750759	1.222651284	64.359261	-115.750759
Location 5	16.678172650139	54.836086	25.325505	16.643479102	54.835779	25.325411
Location 6	88.497795322885	32.527943	67.369579	76.801575168	32.631558	67.348081
Location 7	0.198925707069	-20.085843	127.609749	0.198931888	-20.085843	127.609749
Location 8	103.199486324395	-43.90166	171.400822	99.844380559	-43.891749	171.440371
Location 9	8.385311100729	18.588454	-72.216358	8.381484232	18.588448	-72.216394
Location 10	1.10482690008	13.168835	-87.975825	1.104815479	13.168835	-87.975825
Location 11	6.511619639862	61.969094	11.551401	6.493617683	61.969021	11.551094
Location 12	0.826250874564	20.513841	24.801701	0.826355309	20.51384	24.801701
Location 13	2.030299840406	-22.228776	13.660821	2.029987429	-22.228774	13.660823
Location 14	0.061862097721	-51.69738	-59.90792	0.061893587	-51.69738	-59.907919

Chapter Five: Conclusion and Recommendations

5.1- Conclusion:

Target coordinates calculation is a difficult task due to many reasons. Using small UAVs causes restrictions on the size and weight of electro-optical sensor assemblies. The degree of freedom of the gimbals that carries the camera adds complexity to the system too.

Also, the accuracy of the sensors that gather the data like the GPS, the digital compass and the encoders which are used to know the pan and tilt angle of the camera can cause a severe error on the estimated position of the target. Digital compass must be mounted to the azimuth so it can rotate horizontally but it has to be fixed in the elevation unless the sensor is used to provide some data about the tilt angle of the camera.

Another factor that affects the accuracy of the tilt angle beside the accuracy of the camera encoder itself is the pitch of the plane or the UAV. So to avoid this problem the pitch angle of the UAV should be added to the tilt angle of the camera or the two angles are aligned from the beginning that the tilt angle of the camera and the pitch angle are the same.

Not to mention the altitude of the target which is unknown. It is initially estimated from previous data about the nature of the ground taken from historical images. Because this proposed research is mainly meant for small UAVs, the error factor can be small in some cases. But if we are looking for high precision target acquisition this error is considerable.

5.2- Recommendations:

The future researches can be divided into two parts:

- The first part is optimizing the system by enhancing the sensors accuracy to avoid the errors result in the calculations. Some of the systems nowadays integrate between the visual system(the gimbals in our case) and the autopilot of the UAV so that it can use the coordinates of the UAV instead of having different GPS and the pitch angle to add it to the tilt angle of the gimbal.

- The second part is developing visual system that is GPS independent. This system can use an image processing algorithm that provides real time image geo-registration. Many researches have been done in this field but it is yet promising.

References

[1] SHEIKH, Yaser, Sohaib KHAN, and Mubarak SHAH. Feature-Based Georegistration of Aerial Images. In: A. Stefanidis and S. Nittel, eds. Geosensor Networks. Boca Raton, Florida: CRC Press. 2004. ISSN 0415324041.

[2] Yu Jiaxiang, Xiao Deyun, Jiang Ludong, and Guo Rui, "New Geo-location Approach Based on Camera Coordinates and Common Points on Multiple Images", Defence Science Journal, Vol. 59, No. 1, January 2009, pp. 43-48, 2009.

[3] Gianpaolo Conte, Maria Hempel, Piotr Rudol, David Lundstrom, Simone Duranti, Mariusz Wzorek, Patrick Doherty, "High Accuracy Ground Target Geo-location Using Autonomous Micro Aerial Vehicle Platforms",2007

[4] Bing-FeiWu,Wang-Chuan Lu, and Cheng-Lung Jen , "Monocular Vision-Based Robot Localization and Target Tracking " Journal of Robotics, 2011.

[5] D. Blake Barber · Joshua D. Redding ·Timothy W. McLain ·RandalW. Beard · Clark N. Taylor, "Vision-based Target Geo-location using a Fixed-wing Miniature Air Vehicle" 2006.

[6] Vladimir N.Dobrokhodov, Isaac I.Kaminer, Kevin D.Jones and Reza Ghabcheloo, "Vision-Based Tracking and Motion Estimation for Moving Targets Using Small UAVs", American Institute of Aeronautics and Astronautics 2006.

[7] K.Senthil Kumar[1], Kavitha.G[2], Subramanian.R[3] and Marwan[4], "Image Fusion of Video Images and Geo-localization for UAV Applications", ACEEE Int. J. on Information Technology, Vol. 01, No. 02, Sep 2011

[8] Alison K. Brown, Gengsheng Zhang, Dale Reynolds, "Precision Targeting Using Gps/inertial-aided Sensors" NAVSYS corporation.

[9] Ian Longstaff, Dr. Alison Brown, "Precision Altitude Determination Using a Low Grade GPS-Aided Inertial Measurement Unit", NAVSYS Corporation

[10] Rick Zhang and Hugh H.T. Liu, "Vision-Based Relative Altitude Estimation of Small Unmanned Aerial Vehicles in Target Localization", 2011 American Control Conference, 2011.

[11] William Semke, Jaganathan Ranganathan, and Matthew Buisker, "Active Gimbal Control for Surveillance using Small Unmanned Aircraft Systems",

[12] Thomas B. Criss, Marilyn M. South, and Larry J. Levy, "Multiple Image Coordinate Extraction (MICE) Technique for Rapid Targeting of Precision Guided Munitions", JOHNS HOPKINS APL TECHNICAL DIGEST, VOLUME 19, NUMBER 4 (1998)

[13] F. M. Mirzaei and S. I. Roumeliotis, “A Kalman filter-based algorithm for IMU-camera calibration: observability analysis and performance evaluation,” IEEE Transactions on Robotics, vol. 24, no. 5, pp. 1143–1156, 2008.

[14] S. B. Kim, S. Y. Lee, J. H. Choi, K. H. Choi, and B. T. Jang, “A bimodal approach for GPS and IMU integration for land vehicle applications,” in Proceedings of the 58th IEEE Vehicular Technology Conference (VTC ’03), pp. 2750–2753, October 2003.

[15] Dr. Alison K. Brown, “Precise and Attitude Georegistration of Video Imegery For Target Geolocation”,NAVSYS Corporation 14960 Woodcarver Road, Colorado Springs CO 80921

[16] Redding, J., McLain, T.W., Beard, R.W., Taylor, C“Vision-based target localization from a fixed wing miniature air vehicle”, Proceedings of the American Control Conference, pp. 2862–2867.
Minneapolis, MN (2006)

[17] A. Brown, “High Accuracy Targeting Using A GPS-Aided Inertial Measurement Unit”, ION 54th Annual Meeting, June 1998, Denver, CO

Web sites:
[18] http://www.erikdeman.de/html/sail042e.htm -visited on 20-December2014.

[19] http://en.wikipedia.org/wiki/Spherical_trigonometry -visited on 20-December2014.

Appendix A: Software Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace TargetCoordinatesCalculation
{
    public partial class Form1 : Form
    {
   ////// Pythagoras theorm variables ///////////////////////////////////////////////
        double P_long, P_lat, P_dist;
   //////////distance  variables ////////////////////////////////////////////////////////////
        double R = 6371,   //km
            Lat1, Lat2,
            Longt1, Longt2,
            dLat, dLongt,
            a, c, Distance, Distance2,
             num, denum,
        theta_belw, num2, denum2, theta_e2;
```

```
 //////////////////// coordinates variables  //////////////////////////////////////////

 double target_lat, Target_longt, cam_lat, cam_longt,
heading,target_dist;


////////////////////////////// theta vatiables/////////////////////////////////////////////

        double theta_0, theta_e,
theta_t,theta_above,theta_above2,theta_below, omega,

first_lat,first_long,head,tilt_angle,derived_D,derived_lat,Derived_long,H
,H2,actual_d;

        public Form1()

        {

            InitializeComponent();

        }

        private void calc_Dist_Click(object sender, EventArgs e)

        {

            try

            {

                        if (double.TryParse(P1_lat.Text, out Lat1) &&
                double.TryParse(P2_lat.Text, out Lat2)&&
                double.TryParse(P1_long.Text, out Longt1) &&
                double.TryParse(P2_long.Text, out Longt2))

                {

 //////////////////////////////////////// Haversine formula
////////////////////////////////////

 dLat =(Lat2 - Lat1).ToRadians();

 dLongt=(Longt2-Longt1).ToRadians();

 a = Math.Sin(dLat / 2) * Math.Sin(dLat / 2) + Math.Cos(
Lat1.ToRadians()) * Math.Cos(Lat2.ToRadians())* Math.Sin(dLongt / 2)
* Math.Sin(dLongt / 2);

 c = 2 * Math.Atan2(Math.Sqrt(a), Math.Sqrt(1 - a));
 Distance =Math.Round( R * c,4);

 Result_txt.Text = Distance.ToString();

  ///////////spherical Law of cosines formula///////////////////////////////////////

Distance2 = Math.Acos(Math.Sin(Lat1.ToRadians()) *
Math.Sin(Lat2.ToRadians())

                + Math.Cos(Lat1.ToRadians()) *
                Math.Cos(Lat2.ToRadians())  * Math.Cos((Longt2 -
                Longt1).ToRadians())) * R;

Distance2 = Math.Round(Distance2,4);

Result2_txt.Text = Distance2.ToString();

              }

              else MessageBox.Show("Some of your inputs are incorrect");

           }

  catch (Exception ex)

{ MessageBox.Show("Unable to Calculate the Distance..Please check
your inputs" + e.ToString()); }

       }



       private void button1_Click(object sender, EventArgs e)

       {

           MessageBox.Show("R = earth’s radius (mean radius =
6,371km)\n Δlat = lat2− lat1\n Δlong = long2− long1 \n a = sin²(Δlat/2)
+ cos(lat1).cos(lat2).sin²(Δlong/2) \n c = 2.atan2(√a, √(1−a)) \n d =
R.c");

       }

       private void button2_Click(object sender, EventArgs e)

       {

           MessageBox.Show("d = acos (\n\tsin(lat1)*sin(lat2)+\n\tcos(lat1
)*cos(lat2)*\n\tcos(long2−long1)\n\t) * R");

       }
```

```
        private void button5_Click(object sender, EventArgs e)

        {

            try

            {

 if (double.TryParse(cam_lat_txt.Text, out cam_lat) &&
double.TryParse(cam_longt_txt.Text, out cam_longt) &&
double.TryParse(Heading_txt.Text, out heading) &&
double.TryParse(target_coo_distance_txt.Text, out target_dist))

                {

                    if ((90 < cam_lat) || (cam_lat < -90) || (cam_longt < -180) ||
(cam_longt > 180)) { MessageBox.Show("Out of range"); }

                    else

                    {

                        target_lat = Math.Asin(Math.Sin(cam_lat.ToRadians()) *
Math.Cos((target_dist / R)) +Math.Cos(cam_lat.ToRadians()) *
Math.Sin(target_dist / R) * Math.Cos(heading.ToRadians()));

                        Target_longt = cam_longt.ToRadians() +
Math.Atan2(Math.Sin(heading.ToRadians()) * Math.Sin(target_dist / R)
* Math.Cos(cam_lat.ToRadians()),

                        Math.Cos(target_dist / R) -
Math.Sin(cam_lat.ToRadians()) * Math.Sin(target_lat));



                        target_lat_txt.Text = Math.Round(target_lat.ToDegrees(),
4).ToString();

                        target_longt_txt.Text =
Math.Round(Target_longt.ToDegrees(), 4).ToString();

                    }

                }



                else MessageBox.Show("Some of your inputs are incorrect");

            }
            catch (Exception ex) { MessageBox.Show("Unable to Calculate
the Distance..Please check your inputs"); }

        }

        private void button3_Click(object sender, EventArgs e)

        {

            try

            {

                if (double.TryParse(First_lat_txt.Text, out first_lat) &&
double.TryParse(First_long_txt.Text, out first_long)

                    && double.TryParse(Head_txt.Text, out head) &&
double.TryParse(tilt_angle_txt.Text, out tilt_angle)

                    && double.TryParse(First_Alt_txt.Text, out H) &&
double.TryParse(second_alt_txt.Text, out H2))

                {

                    if ((90 < first_lat) || (first_lat < -90) || (first_long < -180) ||
(first_long > 180) || (head > 360) || (head < 0)

                        || (tilt_angle < -90) || (tilt_angle > 90)) {
MessageBox.Show("Out of range"); }

                    else

                    {

                       //////////// Theta0
calculations///////////////////////////////////////////

                        theta_t = tilt_angle.ToRadians();

                        num = ((R + H)/R) * Math.Tan(theta_t);

                        denum =Math.Sqrt( 1+Math.Pow(Math.Tan(theta_t), 2));

                        theta_0 = Math.Acos(num / denum) - Math.Atan(1 /
Math.Tan(theta_t));

                        theta_0 = Math.Abs(theta_0);
                        //////////// Omega
calculations/////////////////////////////////////////////

                        omega = theta_0 + theta_t;

                        //////////// Thetae
calculations/////////////////////////////////////////////

                        //H2 = Math.Abs(H2);

                        theta_e = omega - Math.Asin((R / (R + Math.Abs( H2)))
* Math.Sin(omega));

                        theta_e = Math.Abs(theta_e);


                        //////////// Thetae2
calculations///////////////////////////////////////////

                        num2 = R * Math.Tan(omega);

                        double denum_temp1 = Math.Pow(R - Math.Abs(H2), 2);

                        double denum_temp2 = (R - Math.Abs(H2) )*
Math.Tan(omega);

                        double denum_temp3 = Math.Pow(denum_temp2,2);

                        denum2 = Math.Sqrt(denum_temp1 + denum_temp3 );

                        theta_e2 = Math.Acos(num2 / denum2) - Math.Atan(1/
Math.Tan(omega));

                        theta_above = theta_0 - theta_e;

                        theta_belw = theta_0 + theta_e2;


                        if( H2 <0)

                        {

                           derived_D = (R * theta_belw);

                         P_dist=(H + H2)* Math.Tan(theta_t);

                        }

                        else if ( H2 >= 0)
            {
              derived_D = (R * theta_above);
              P_dist = (H - H2) * Math.Tan(theta_t);
            }
derived_D_txt.Text =Math.Round( derived_D,12).ToString();

derived_lat = Math.Asin(Math.Sin(first_lat.ToRadians()) *
Math.Cos(derived_D / R) +Math.Cos(first_lat.ToRadians()) *
Math.Sin(derived_D / R) * Math.Cos(head.ToRadians()));

Derived_long = first_long.ToRadians()
+Math.Atan2(Math.Sin(head.ToRadians()) * Math.Sin(derived_D / R) *
Math.Cos(first_lat.ToRadians()),Math.Cos(derived_D / R) -
Math.Sin(first_lat.ToRadians()) * Math.Sin(derived_lat));


////////////Pythagoras theorm
//////////////////////////////////////////////////////////////////

P_lat = Math.Asin(Math.Sin(first_lat.ToRadians()) * Math.Cos(P_dist/
R) +Math.Cos(first_lat.ToRadians()) * Math.Sin(P_dist / R) *
Math.Cos(head.ToRadians()));

P_long = first_long.ToRadians()
+Math.Atan2(Math.Sin(head.ToRadians()) *  Math.Sin(P_dist / R) *
Math.Cos(first_lat.ToRadians()),Math.Cos(P_dist / R)
Math.Sin(first_lat.ToRadians()) * Math.Sin(P_lat));


Py_dist.Text =Math.Round( P_dist,9).ToString();

Py_long.Text = Math.Round(P_long.ToDegrees(), 6).ToString();

Phy_lat.Text = Math.Round(P_lat.ToDegrees(),6).ToString();

 ////////////////////////////////////////////////////////////////////////////////////////////////////////

Derived_lat_txt.Text = Math.Round(derived_lat.ToDegrees(),
6).ToString();

Derived_longt_txt.Text = Math.Round(Derived_long.ToDegrees(),
6).ToString();
```

```
actual_d = Math.Acos(Math.Sin(first_lat.ToRadians()) *
Math.Sin(derived_lat)+ Math.Cos(first_lat.ToRadians()) *
Math.Cos(derived_lat)* Math.Cos((Derived_long -
first_long.ToRadians()))) * R;

actual_d = Math.Round(actual_d, 12);

actual_d_txt.Text = actual_d.ToString();

                    }

                }

    else MessageBox.Show("Some of your inputs are incorrect");

            }

            catch (Exception ex) { MessageBox.Show("Unable to Calculate
the Distance..Please check your inputs"); }

        }

    }

    public static class NumericExtensions

    {

        public static double ToRadians(this double val)

        {

            return (Math.PI / 180) * val;

        }

        public static double ToDegrees(this double val)

        {

            return (180/ Math.PI) * val;

        }

    }

}
```

APPENDIX B: Canon Camera Specifications

Canon power shot Sx50 HS Camera Specifications

Body type	SLR-like (bridge)
Max resolution	4000 x 3000
Effective pixels	12 megapixels
Sensor size	1/2.3" (6.17 x 4.55 mm)
Sensor type	BSI-CMOS
ISO	to, 80, 100, 125, 160, 200, 250, 320, 00, 500, 640, 800, 1000, 1250, 1600, 000, 2500, 3200, 4000, 5000, 6400
Focal length (equiv.)	24–1200 mm
Optical zoom	50×
Articulated LCD	Fully articulated
Screen size	2.8”
Screen dots	461,000
Min shutter speed	15 sec
Max shutter speed	1/2000 sec
Format	H.264
Storage types	SD/SDHC/SDXC
USB	USB 2.0 (480 Mbit/sec)
Weight	nc. batteries) 595 g (1.31 lb / 20.99 oz)
Dimensions	3 x 87 x 106 mm (4.84 x 3.43 x 4.17”)