Path Planning for Lunar Rover

Overall Path Planning scheme

DEM GENERATION

- Process NAC images in USGS ISIS and Ames Stereo Pipeline.
- Generate point cloud and obtain the .las format.
- Convert .las format into .csv using liblas library for importing to MATLAB.

GLOBAL PATH PLANNING USING COURSE MAP DATA GENERATED WITH DEMs.

- Identify the all the cost functions.
- Plan the starting point.
- Plan the end point.
- Choose an algorithm for cost minimization from starting to end point and implementing.

LOCAL PATH PLANNING USING STEREO IMAGES FROM ROVER

- Pan the area using the stereo imaging.
- Map the ROI based on the imaging function. (cam2map)
- Identify the obstacles in image of minimum threshold and generate a field map based on obstacles.
- Identify the end point of traversal.
- Choose an algorithm for minimizing identified cost and implement for generating way points.
- Drive to the way points.

Constraints to rover traversal

- # Mast Shadow
  - Facing the sun and solar panel at the back, shadow of the mast on solar panel can reduce the power generation efficiency.
- # Slopes
  - Higher the slope, higher power requirement for performing the maneuver. Overall power consumption is directly proportional to the sigma of path traversed, boundary condition being **18 degrees for slopes and 33 for internal friction angle** of regolith
- # Distance
  - Distance between target and starting point has to be put to zero. The most important cost constraint.

Algorithms Implemented

A* (ASTAR)

- Grid based path determining algorithm.
- Determines next path point by evaluating neighbor points.
- Pros: Easy implementation. Path determination in least time for small grids.
- Cons: Takes time for finer grids. Interpolation schemes like cell decomposition makes implementation complex.

FMINCON SOLVER

- Determines best path by evaluating overall grid.
- Pros: Easy to implement even with finer interpolation. Directly generates way points.
- Cons: Path generated may not be the best as algo terminates if max iterations completes.

Chandrabhraman

Landing site DEM analysis for rovers

INDEX

> Criteria of selection for landing site

The landing site which satisfies all the constraints based on capability and limitations of structure, power, communication and descent strategy of the lander, is considered the most ideal landing site. However in actual practice there is a cost associated with every constraint, be it topography or the sun elevation angle, and our aim is to minimize that by thoroughly researching the landing site using various tools available. Some of the major landing site constraints are summarized below.

Communication constraint:
The spacecraft should land between the East of 80^o West to West of 80^o East for maintaining a permanent link to the ground stations. Only the near side of the moon is directly available for line of sight communication with earth. Therefore, we have already reduced our search by 41 percent by eliminating the far side.

Gentler slope for landing stability:
With the current configurations of the lander, the maximum slope taken into account in design is 18^o keeping in mind a local instability due to rock fragments and boulders. Beyond 18^o slope can create critical conditions for lander's descent, for instance, in terms of rover's deployment and in worst case, toppling. Our current statistics says that the lunar surface has 95% of the area with less than 16^o slope and thus 18^o is incorporated as worst case scenario in the structural design of the lander.

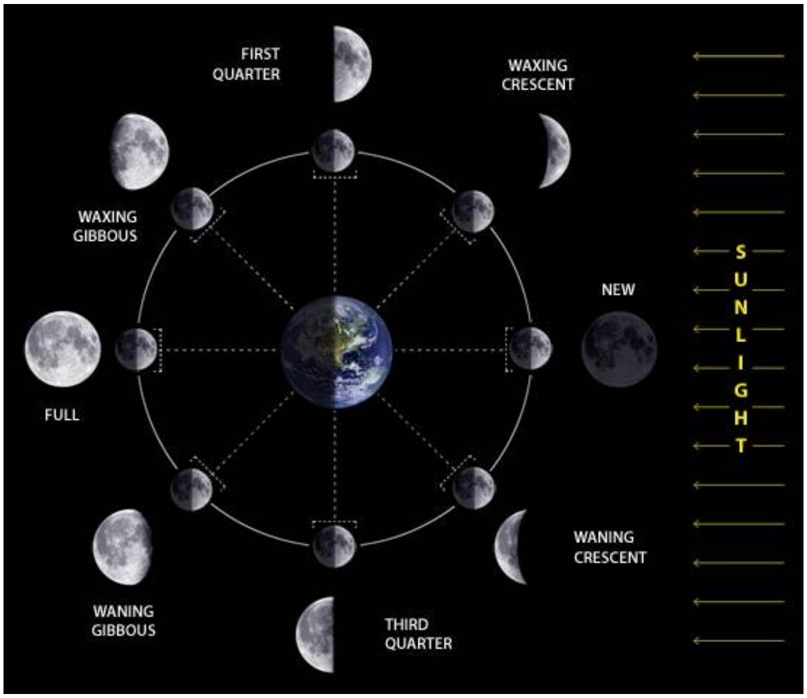
Lower concentration of hazardous surface feature:
Features like craters, boulders and fragmented rocks are seen as a threat to the mission. With respect to the fact that craters have a deep parabolic profiles and steep interior and exterior slopes along their edges, they are a potential hazard to the lander during the descent phase. Similarly large size boulders and rocks on the lunar surface can create surfaces of steep local slopes and thus, generating the conditions of toppling. These features are seen as the primary concerns in the landing site analysis and is dealt in detail further in the document.

No shadow regions
In the absence of a lunar atmosphere, the shadows formed due to various terrain features like mountains are pitch black. Electronics are primarily operating on the solar power thus landing in the region of darkness can quickly drain the battery making it a major cause of concern.

Region of lunar dawn
The period of rotation of moon is slow and lunar day and nights extend to about 14 earth days, so the most suitable region to land should be the region of lunar dawn in order to look forward for the power requirements of the spacecraft. On contrary, lunar nights exhibits very extreme temperature (as low as -150^o Celsius) and enduring periods of darkness thus making it impossible for the spacecraft to survive. So we would like to extend the availability of solar energy as long as possible so as to get enough time and energy for completing our mission.

Non polar regions
Due to surface, topography, roughness, temperature and shadow regions of lunar poles, they are completely excluded from the landing site analysis. Just for an instance, many places on the lunar poles are the regions of permanent darkness and temperatures remains as low as -247^o Celsius there. These constraints are very well minimized if we take in account the north western lunar sphere. The western hemisphere is the region of lunar dawn and some degrees above the equator towards north, we have

flat planes of basaltic lava flow with low density of shallow craters, called the Mare, thus making that place suitable for landing. Our landing site is located near Mare Imbrium therefore our study will be restricted to that region.

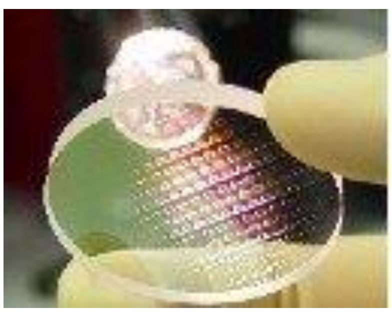


Moon Phase Diagram
[courtesy: api.ning.com]

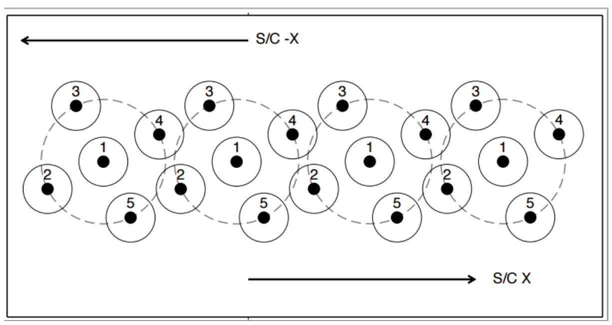
> Theory for calculation of slope distribution

A brief about LOLA

The lunar orbiter laser altimeter (LOLA) is currently orbiting the moon as a part of LRO mission. The dataset delivered by the altimeter is the most precise topographic data available till date, with so much accuracy that it can act as a base map for correcting the misalignments in other datasets.
Slope distribution has been taken into account based on the reading of LOLA data. It works by propagating a single laser pulse through Diffractive optical element that splits it into five beams. These beams then strike and are backscattered from the lunar surface. For each beam, LOLA measures time of flight (range), pulse spreading (surface roughness), and transmit/return energy (surface reflectance). With its two-dimensional spot pattern, it determines slopes along-track and across-track with respect to velocity of the spacecraft. For more details check out the link here.



Diffractive optical element

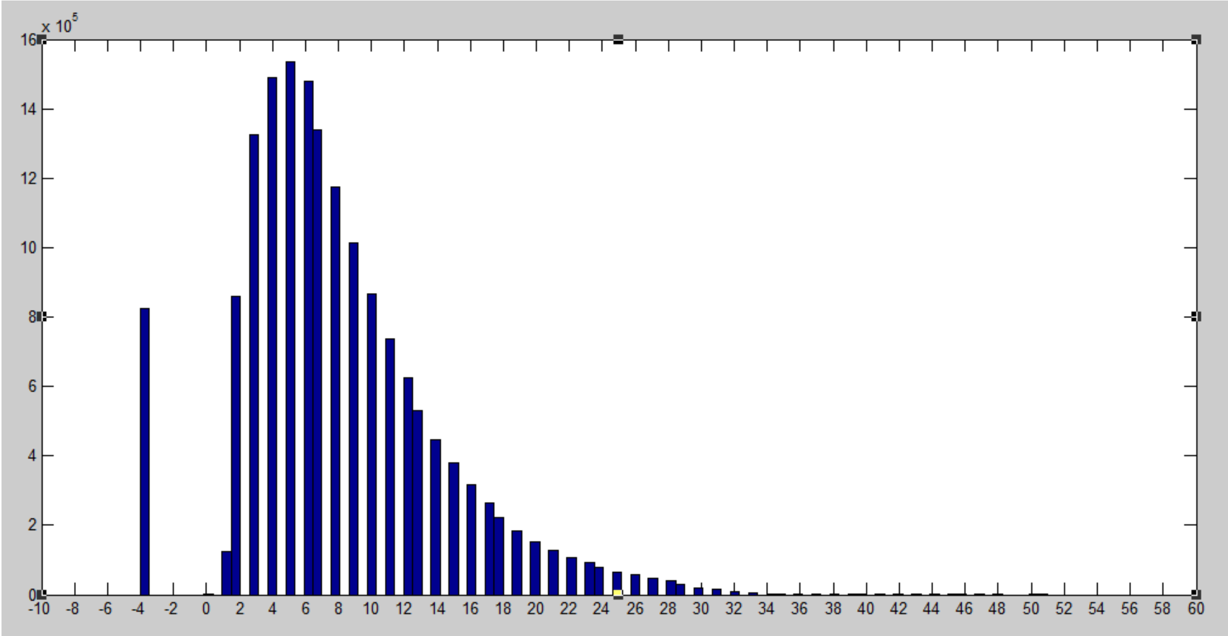


Cross pattern beams

[Courtesy: http://lunar.gsfc.nasa.gov/lola/]

LOLA fires at a fixed, 28-Hz rate, so that for a nominal 1600 m/s ground track speed there is one shot approximately every 57 m. At a nominal 50-km altitude in polar orbit, each spot within the five-spot pattern has a surface diameter of 5 m while each detector field of view has a diameter of 20 m; the surface spread is 25 meters, and the spots form a cross pattern canted by 26 degrees counterclockwise to provide five adjacent profiles.
[source: http://imbrium.mit.edu/DOCUMENT/RDRSIS.PDF]

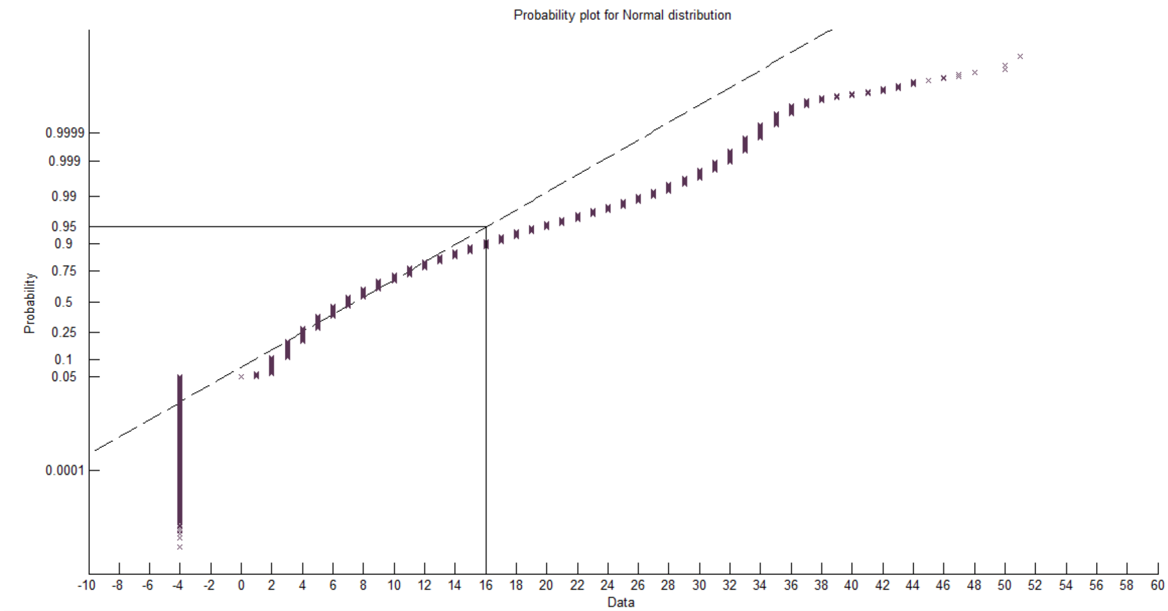
The data values plotted below are the slope distribution diagram has been taken from PDS Geosciences Node. Total number of observations taken are 2880 x 5760 = 16588800 and are plotted using MATLAB.



The data depicted above is a frequency distribution of surface slope pixel with a resolution of 16m/pixel by 16m/pixel and is called **Gridded data record slope maps**. Here, the Y axis represents the frequency of occurrence of a slope value (on a scale of 10^6 as shown) and X axis represents cross track and along the track values of slope, with respect to velocity of the LRO in the polar orbit. The raw data obtained is a digital number and has been multiplied with a scaling factor (as specified in the dataset label) and plotted in matlab. The formula is as follows

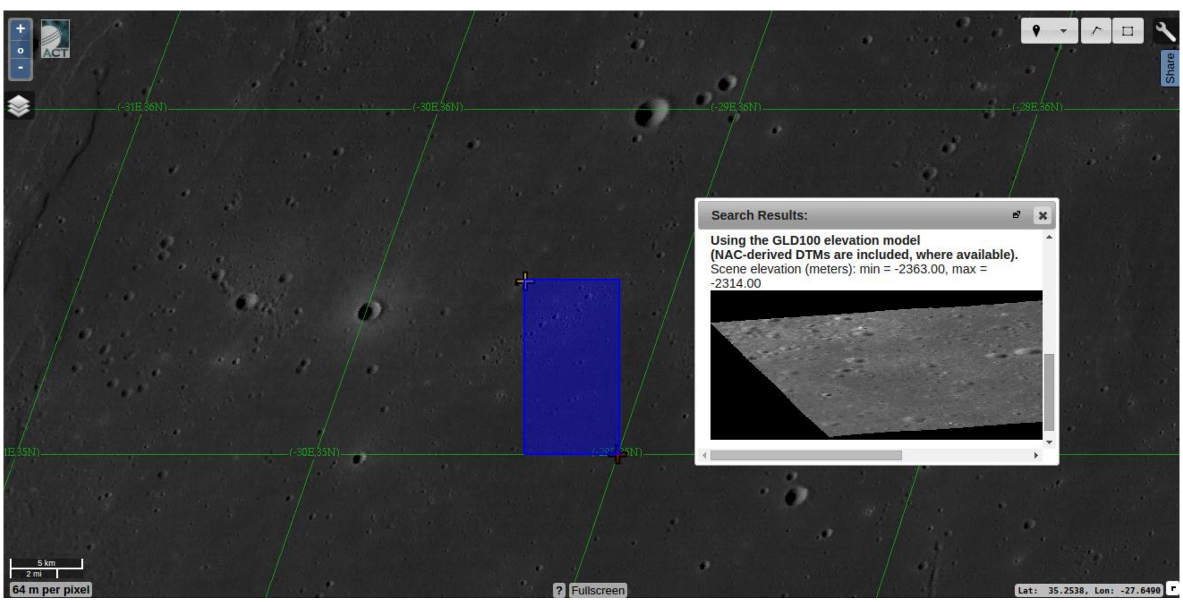
`slope value = digital number * scaling factor + offset`

A missing constant has been assigned to the ambiguous data values equal to = -32768 and when multiplied with the scaling factor, gives a value equal to ~= -4, thus justifying the high frequency of negative slope between -2 to -4.
If we pick any point on the lunar surface in between the specified coordinates, the probability of encountering a particular slope value can be well established using the above frequency distribution. Consider the probability plot on the basis of that data:



The above distribution is a geographical probability which states that "the probability of occurrence of more than 16 degree slope, is less than 5%". However, this does not mean that the probzability of encountering the surface with more than 16 degrees of slope during the descent is less than 5%. This is because the probability of spacecraft, drifting away from the targeted landing site, decreases with the drift.

A square region of approximately 14x14 km containing the current landing site has a maximum elevation of -2214m and minimum elevation of -2363m.



[Source: target LROC]

The most closest missions associated with TI's landing site is the Lunakhod 1, which dropped on the coordinates 34.99°W, 38.23°N followed by Change's 3 with landing coordinates as 44.12°N 19.51°W. Our studies and assumption regarding the lunar regolith is mostly based on the data of these missions.

[Source: https://en.wikipedia.org/wiki/Chang'e_3]

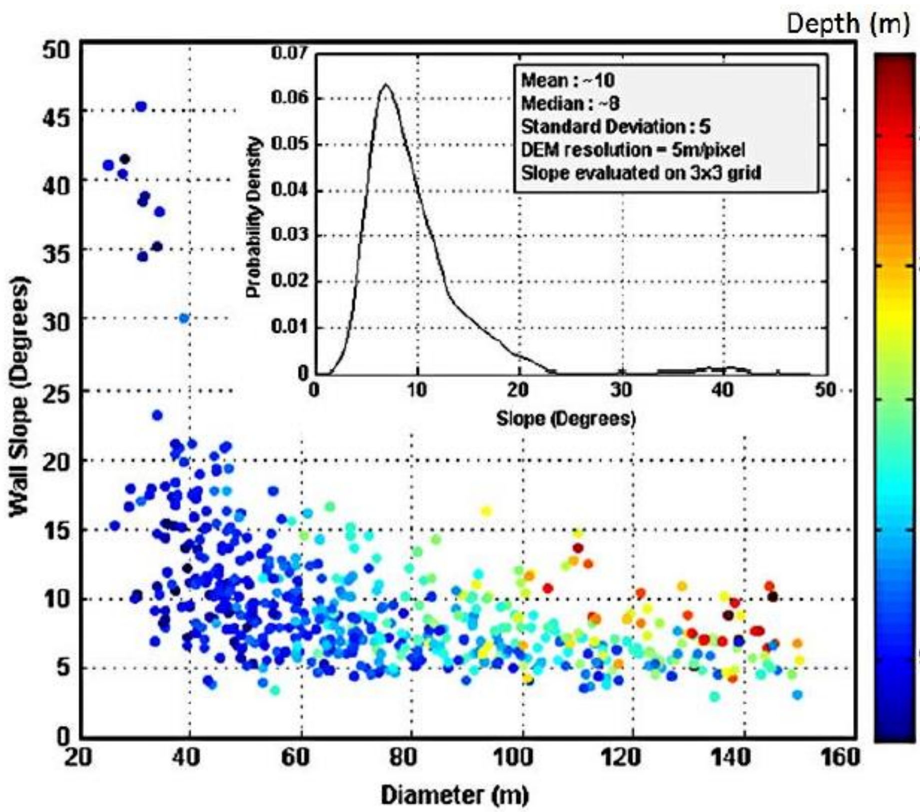
[Source: www.lpi.usra.edu/meetings/lpsc2012/pdf/1750.pdf]

> Crater characteristics

Morphology of small lunar craters (SLC)

Data regarding the large lunar craters has been well established by recent lunar missions such as Lunar Reconnaissance orbiter NAC images. These are the craters with more than 200m of diameter. Craters below this size comes under the category of small lunar craters and are the major cause of concern for the success of mission. They are not very well understood but some statistics and relationships have been derived regarding their distribution and behaviour over time. These are summarized below.

SLC are transient on geological timescale. They quickly get degraded compared to the bigger ones. The observed (depth vs diameter) d vs D from SLC is well fit by both, a power law ($d = aD$ b ; R 2 = 0.98) and a simple linear relationship ($d = mD + c$; R 2 = 0.86). SLC are found to be shallower than the overall crater profile with a median d/D of 0.13. With increase in D, the median D decreases, likely due to a larger asymmetrical spread of depth. Lower mean wall slopes (MWS) are associated with large diameter craters. Wall slopes are relatively shallow (median ~8^o) with 95% of the data at slopes less than 18^o. Depth decreases with crater size with a sharp drop and beyond 35m the rate of change is small. A MWS of 15^o is observed for craters with D < 100m and 11^o for larger ones (100m < D < 200m).

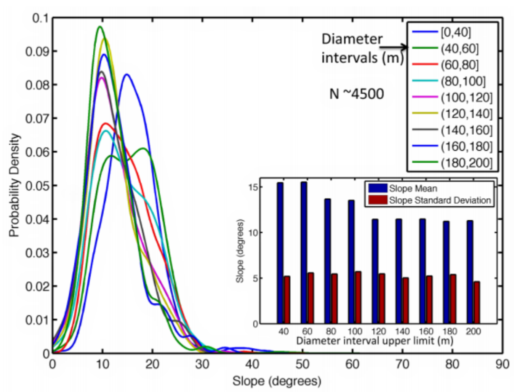


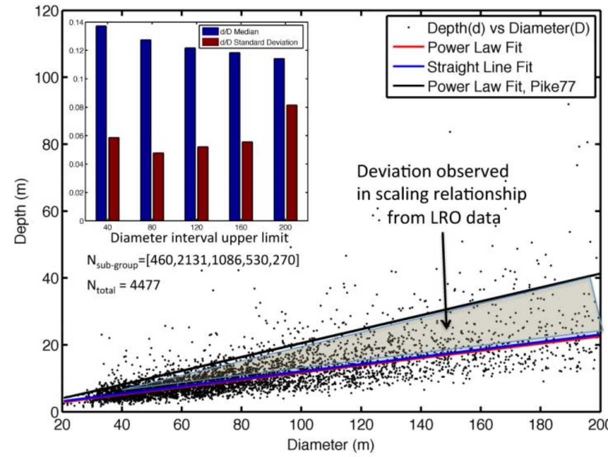
[source: AGU_2012_Mahanti_Robinson_Stelling.pdf]

The peak slope in diameter bins decrease as diameter increases and MWS distribution among craters is also observed to change from a bimodal distribution to a more unimodal nature as diameter increases. Steeper wall slopes are observed in the Highland regions rather than the mare. From the lunar exploration point-of-view, the distribution of MWS suggest that rover designs with stable locomotion performance up to 30 ◦ slope will be effective for all over the lunar surface. This result further reduces down to 18^o when talking in account of TI's landing site. It was found that the crater population of

Lunakhod 1 area have a typical d/D ratio of ~0.02 to 0.2 - 0.026. The measured maximum slope was found to be 7° to 28°.
[Source: http://www.hou.usra.edu/meetings/lpsc2014/pdf/1584.pdf]

Figure 1: Depth vs Diameter from observed SLC. Fit to observations is shown as solid lines. Bar chart (inset) shows diameter range median and standard deviation values





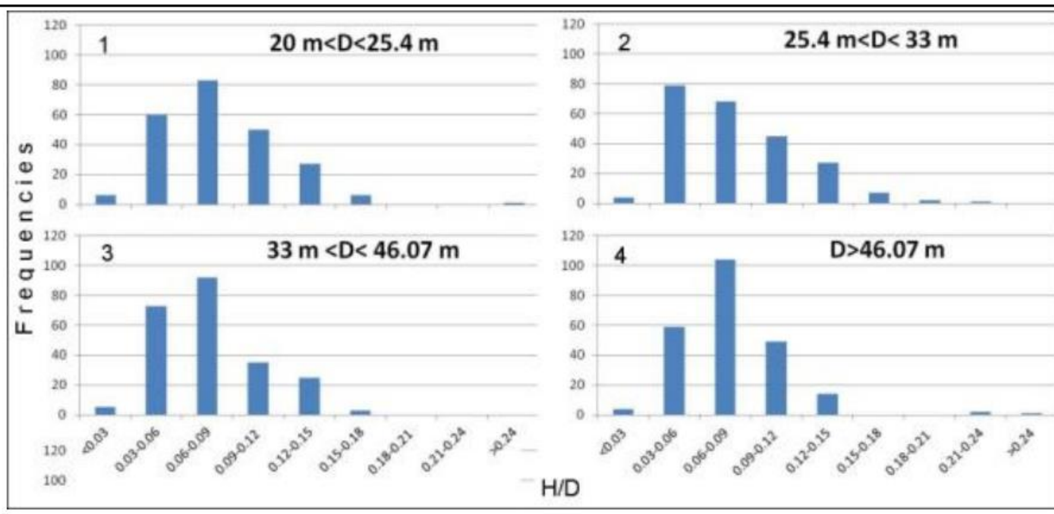


Figure 3. Distribution of H/D in four different in crater diameters parts of the studied crater subpopulation.

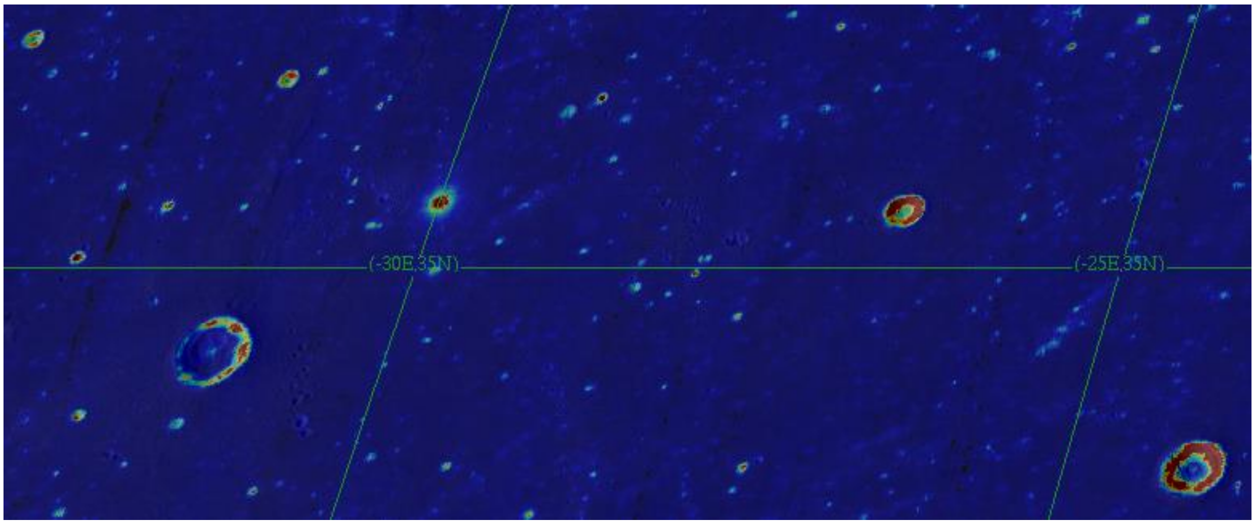
[Source: basilevky morphology of small impact craters]

> Boulder Characteristics

The analysis was done on 91 panoramic images captured by Lunokhod 1, 2 and 17 for Apollo 11-15 astronauts along with 6 LRO NAC images. The average height by visible diameter ratio (h/d) and average height by maximum diameter ratio (h/D) of Mare imbrium was found to be 0.6 +- 0.03 and 0.53 +- 0.03 with a standard deviation of 0.28 respectively. An h/D ratio of 0.5 is recommended for engineering models. This result is based on the analysis of 445 rocks for 6 sites covered by Lunakhod 1 and 2

missions and was found to be astonishingly indistinguishable by 95%. Therefore this study can be generalized for whole lunar surface and is more closely associated to our landing site.
[source: Height-to-diameter ratios of moon rocks from analysis of Lunokhod-1 and -2 and Apollo 11–17 panoramas and LROC NAC images]

During lunar night boulders and fragmented rocks stay warmer than regolith. LRO diviner data maps the temperature of terrain and able to locate rock abundance. It has been observed that rock abundance is maximum in craters than on plane lands.



The regions of high rock abundance RED = HIGH, BLUE = LOW
[Courtesy: target LROC]

Summary of probability scenarios for encountering big boulder sizes and steep slopes together:
Probability for encountering small boulders 2 to 5cm ~= 0.25 to 0.90
Probability for encountering 15cm or bigger boulder ~= 0.05 to 0.15
[Source: Lunar sourcebook]

During crater formation assuming all the boulders gain homogeneously equal kinetic energy with respect to the proportion of their masses, the boulders with small mass are likely to travel farther than Boulder with big mass, which gets localized around the crater. So the places with higher chances of finding big boulders, chances of finding smaller ones decreases and vice versa.
Therefore from set theory:
XUY = X+Y-XnY
Probability for encountering 15cm or bigger boulder (upper limit) ~= 1-(0.25+0.15-0.25*0.15) = 0.6375
Probability for encountering 15cm or bigger boulder (lower limit) ~= 1-(0.90+0.05-0.90*0.05) = 0.095
Probability for encountering slopes greater than 18cm = 1-0.97 = 0.03
Probability of encountering 18 degree or greater slope with 18 cm or bigger boulder (upper limit) = 0.6375*0.03 = 0.01913
Probability of encountering 18 degree or greater slope with 18 cm or bigger boulder (lower limit) = 0.095*0.03 = 0.00285

> Soil Characteristics

Mares are predominantly flat, dark planes composed mainly of basalts. Only 16% of the lunar surface is covered with lunar mare. They are relatively flat because of their history of large flows of basaltic lava. That makes them a better landing site choice compared to other places on the moon.
Results from the Apollo 14 and 15 missions together indicate that the Imbrium Basin formed about 3.85 b.y. ago. Two surprising volcanic materials were discovered at the Apollo 15 site: an aluminous, non-Mare basalt, rich in KREEP (Potassium, Rare Earth Elements and Phosphorous), and an emerald-green glass of ultramafic composition and pyroclastic origin.
[Source: LunarSourceBook.pdf]

Mechanical properties of soil:

The Lunakhod 1 data are the best estimates of lunar regolith till date.

Table 1. Estimates of lunar soil density.

Bulk density ρ (g/cm^3)	Investigator	Mission
0.3	Jaffe (1964, 1965)	
0.4	Halajian (1964)	
1.5	Christensen *et al.* (1967)	Surveyor I
0.8	Cherkasov *et al.* (1968)	Luna 13
1.5	Scott and Roberson (1967, 1968) and Scott (1968)	Surveyor III & VII
1.54 to 1.75	Costes and Mitchell (1970)	Apollo 11
0.74 to > 1.75	Scott *et al.* (1971)	Apollo 11
1.81 to 1.92*	Costes *et al.* (1971)	Apollo 11
1.6 to 2.0	Scott *et al.* (1971)	Apollo 11
1.80 to 1.84*	Costes *et al.* (1971)	Apollo 12
1.55 to 1.90	Houston and Mitchell (1971)	Apollo 12
1.7 to 1.9	Carrier *et al.* (1971)	Apollo 12
1.2	Vinogradov (1971)	Luna 16
1.5 to 1.7	Leonovich *et al.* (1971)	Lunokhod 1
1.45 to 1.6	Carrier *et al.* (1972)	Apollo 14
1.35 to 2.15	Mitchell *et al.* (1972)	Apollo 15

*Upper bound estimates.

[Source: http://articles.adsabs.harvard.edu//full/1972LPSC....3.3235M/0003236.000.html]

Properties of Regolith:

>> Surface regolith
Density: 1.5 g/cm3
Cohesion: 0.01 Mpa
Friction angle: 33 degrees
Porosity: 40 %

>> Hard soil/ Soft rock
Density: 2.1 g/cm3
Cohesion: 1 Mpa
Friction angle: 45 degrees
Porosity: 0 %

>> Hard rock

```
Density: 3.2 g/cm3
Cohesion: 10 Mpa
Friction angle: 45 degrees
Porosity: 0 %
```
[Source: http://www.lpi.usra.edu/lunar/tools/lunarcratercalc/]

Boulder size estimations of different missions:

Table 1.8 Recommended values of lunar soil cohesion and friction angle (Carrier *et al.* 1991).

Depth Range (cm)	Cohesion (kPa)		Friction Angle (°)	
	Average	Range	Average	Range
0-15	0.52	0.44-0.62	42	41-43
0-30	0.9	0.74-1.1	46	44-47
30-60	3	2.4-3.8	54	52-55
0-60	1.6	1.3-1.9	49	48-51

Table 1.7 Porosity and void ratio of the lunar soil (Carrier *et al.* 1991).

Depth Range (cm)	Average Porosity, n	Average Void Ratio, e
0-15	52 ± 2	1.07 ± 0.07
0-30	49 ± 2	0.96 ± 0.07
30-60	44 ± 2	0.78 ± 0.07
0-60	46 ± 2	0.87 ± 0.07

Table 1.5 Bulk density of lunar soil (Carrier *et al.* 1991).

Average Bulk Density (g/cc)	Depth (cm)
1.50 ± 0.05	0 – 15
1.58 ± 0.05	0 – 30
1.74 ± 0.05	30 – 60
1.66 ± 0.05	0 – 60

[Source: UNDERSTANDING MECHANICAL BEHAVIOR OF LUNAR SOILS FOR THE STUDY OF VEHICLE MOBILITY]

> DEM Generation:

This is perhaps the most important part of the landing site analysis. DEM (data elevation model) generation aims at building a 3D digital model of lunar surface of the landing site using stereo bundle of LRO NAC images. DEM generation is done using AMES stereo pipeline and ISIS.

What is a DEM?

The Lunar Reconnaissance Orbiter is a NASA robotic spacecraft currently orbiting the Moon in an eccentric 30 by 180 km polar mapping orbit at a speed of 1.6km/s. During this orbiting period, it takes stereo images of the lunar surface using two of its cameras, a wide angle camera and two narrow angle camera (NAC - Left and NAC - Right). WAC has a low resolution 100m/pixel ideal for capturing a vast area in one go (60 km swath), on the other hand NACs have a resolution of 0.5m / pixel, making their application more suitable for capturing details of lunar surface, craters, boulders etc (5 km swath).

Originally NACs and WAC were introduced keeping in mind the objectives of finding potential landing sites, mapping the regions of permanent shadow or illumination, creating high resolution maps of polar massifs with permanent or near-permanent illumination, Observing regions from multiple angles to derive high-resolution topography etc. NACs are of importance to us.
[Reference: LRO NACs]

Many overlapping images with sufficient ROI window, of same area gives us a bundle of images perceived from different angles. This enables us to see the third axis (depth) from two plane flat pictures and thus the elevation details of topography and other features. This is what exactly a DEM is. It's the elevation data about every point mapped on the ground control net (detailed below) correlated in the ROI of two overlapped images. Not necessarily the overlapping of left and right NACs. It can be an overlap of any image to the any other.

Summary of the procedures and an inside to the algorithms used:
NAC images taken by the spacecraft are first processed in ISIS. Raw images taken from orbiter are post processed for calibration and alignment with respect to the lunar map. These include the following steps:

Level 0 - Preparing Raw image data for ISIS processing.
>> Removal of image artifacts and radiometric corrections.
They appear as regular black dots on an image in regular pattern.
Program corrects them replacing them with a weighted average of the neighborhood pixels.

>> Removal of image blemishes
Causes due to telemetry data dropouts or transmission error, malfunctioning detectors, dust specks located in optical path or coherent noise by electronic signals.
Resolved by taking weighted average of unaffected neighborhood pixels.

>> Camera shading correction
Occurs due to non-uniform brightness sensitivity across field of view of instrument.
Correction of non-identical DN value is done by-

$$O_{i,j} = Wo * [\{ G_{i,j} * (R_{i,j} - DC_{i,j}) \}] / Exp\_t$$

where,
$O_{i,j}$ = pixel value at i,j of output corrected image.
$R_{i,j}$ = pixel value i,j of input raw image.
$G_{i,j}$ = multiplicative correction at pixel i,j
$DC_{i,j}$ = Additive correction at i,j
Exp_t = Exposure duration of observation
W_o = normalization coefficient

Level - 2 Geometric processing
>> Transformation from image coordinate to map coordinate
Attaching SPICE kernel data which include the 5 elements for transformation. These are position of spacecraft, position of planet body and its rotational elements, camera mounting angle, Initially referenced attitude for imaging system and time ordered events, instrument temperature along with duration of exposure of observation.
Errors in camera pointing causes images to be improperly mapped. Removing these camera pointing errors by adjusting the C matrix.
Making ground control net by using locations of features (LAT - LON) and improving C matrix of image.

Tying images to each other using the ground control net and thus forming the image mosaic. The regions of overlap are later used for improving the C matrix.
Optical distortion correction. It occurs due to scanning pattern of image vidicon is barrel shaped than rectangular.
Applying the geometric transformation. The program written for geometric transformation is divided into two parts. The first part maps the input pixel camera pixel to output map pixels and the second part executes this mapping. The default projection adopted by USGS for mapping is sinusoidal projection.

Level - 3 Photometric Normalization
If the stereo images are of different brightness images then photometric normalization is employed for normalizing the brightness level.

Level - 4 Seam removal and Image mosaicking
Seam removal removes the small discrepancies in image brightness. After applying correction factors resulting brightness difference in minimized. Images are now ready for accurate digital mosaic. Images are overlapped on an initially blank projection. Order is important and we prefer to keep the lowest quality images on the bottom so that we can have the best quality DIM (digital image model). Although this step is not taken in consideration in TI landing site's DEM generation.
[Reference: http://isis.astrogeology.usgs.gov/Isis2/isis-bin/intro_digi_mosaic.cgi]

Photometrically normalized images are ideal for feeding in ASP. The DEM can be obtained in .csv format (comma delimited) with rows containing point information of height above datum at correlated LAT LON. This information can further be used in MATLAB for complex mathematical purposes, for rover path planning and can be used for hazard detection and avoidance during descent.

> Brief introduction to various components necessary for DEM generation process.

A brief about the following topics is necessary so as to the develop clarity of development of DEM process.

LRO NACs
The basics of NAC's and its objectives have already been introduced to you in the previous section (introduction to DEM). Let’s dig down further. TI's landing site lies at the coordinates 35.2504 N, 29.2376 W. During descent the velocities in three directions namely cross track (perpendicular to the descent trajectory, ideally should be zero), in track (the direction in which the spacecraft is moving) and Radial (velocity perpendicular to in track velocity) contains some errors in their measurements. These errors form an error ellipsoid around the lander, projection of which on the ground (after multiplying the error velocities with estimated decent time) is an ellipse out of which the probability of landing the spacecraft is negligible. So the site under inspection is not a point but an area, kilometers wide. For this reason, our landing area for inspection would be 29 - 29.5 W and 35 - 35.5 N which is a window of nearly 14 x 14 km square. All the data and raw images of the concerned landing site are available on Lunar Orbital Data Explorer. Only raw images must be used for processing.

USGS ISIS
ISIS stands for integrated software for imagers and spectrometers, which is a planetary image processing open source developed by USGS for NASA. It can process 2D images from camera and 3D cubes derived from imaging spectrometers. It has functionalities of a standard image processing software as well as for

various missions such as Lunar Orbiter, Apollo, Voyager, Mariner 10, Viking, Galileo, Magellan, Clementine, Mars Global Surveyor, Cassini, Mars Odyssey, Mars Reconnaissance Orbiter etc.
[Reference: http://isis.astrogeology.usgs.gov/documents/Overview/Overview.html]

ISIS is a Linux based software and is divided into various sub programs with GUIs of their own. It does not support windows. For our considerations, we would be using ISIS 3. One thing about ISIS to nail down is that it takes all the coordinate values in terms of north and east with 360 degree domain. Therefore, for instance, the coordinates 29 W longitude and 35 North Latitude would be written be written in ISIS as 331 East longitude and 35 North Longitude.

Ames Stereo Pipeline
Ames stereo pipeline, also known as ASP in short, is an open source for producing cartographic products, including digital elevation models (DEMs), ortho-projected imagery, and 3D models. Credit for its development goes to intelligent robotics group, intelligent robotics division, Ames research center, NASA. Though initially developed for ground control and scientific visualization application, stereo pipeline has evolved in recent years to address orbital stereogramametry and cartographic applications. For processing non terrestrial applications, ASP must be installed with ISIS.

ASP is also Linux based and is a CLI. Technical support staff for ASP and ISIS are completely separate as these are developed by two different organizations. For ISIS it is available at
https://isis.astrogeology.usgs.gov/IsisSupport/.
and for ASP
http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/ngt/stereo/#Mailing.

Installation procedures
>> <u>ISIS</u>
Details about the installation procedures are given on the following link, but certain prerequisites needed to be intact while following various steps. Downloading ISIS using installation script requires `OpenJDK Java x runtime` package, for running the installation GUI. It is available at the software package manager. Things work just fine if you choose to use CLI instead. You may not prefer to download the kernel data as things work just fine with the spice web server. However, it won't, if 'translations' group is missing or incomplete in your base directory or your ISIS is not updated to the latest version. If you're using ubuntu then command line for updating ISIS is

```
rsync -azv --delete --partial isisdist.astrogeology.usgs.gov::x86-
64_linux_UBUNTU/isis .
```

and for updating or fixing incomplete files in your base directory, the command line is

```
rsync -azv --delete --partial
isisdist.astrogeology.usgs.gov::isis3data/data/base data/
```

for other distros, check the section 'Downloading the ISIS Binaries'.
After downloading the binaries you must set your environment variables. For that you will point ISISROOT variable to your isis directory at the end of your `bashrc` file. Execute the following commands in your terminal

```
gedit ~/.bashrc
```

and add the following lines

```
ISISROOT=~/Isis/isis
. $ISISROOT/scripts/isis3Startup.sh
```

>> ASP
If you're sure about all the dependencies required for running ASP is installed in your system then you can directly download the ASP binaries and set the environment variables in the `bashrc` just like we did for ISIS. For doing that execute the following commands in your command prompt after downloading and saving the binaries in your home directory.

```
gedit ~/.bashrc
PATH = "/path/to/StereoPipeline/bin:${PATH}"
```

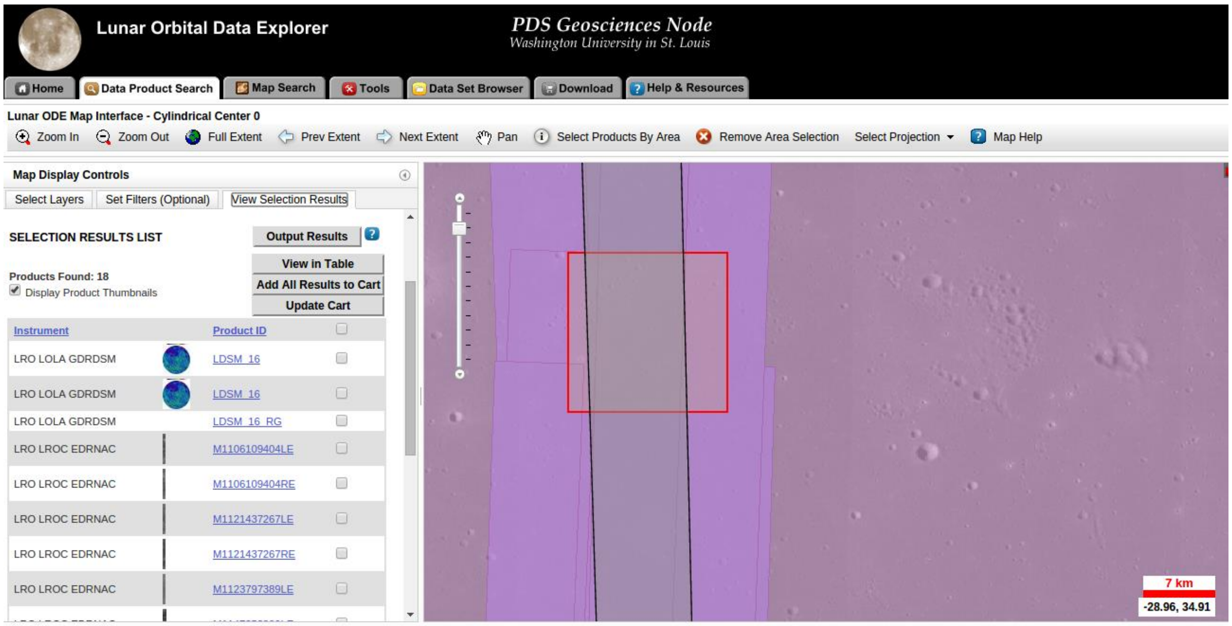
However, if you want to build those binaries manually and want to automatically download and install the dependencies, you may need to do that using binary builder. Details of the installation procedure is given in the ASP book. Additionally, for installing the requisite packages you might want to refer to lunakhod.org. One important thing to be nailed down is that ASP has a dependency on the *base directory of ISIS*.

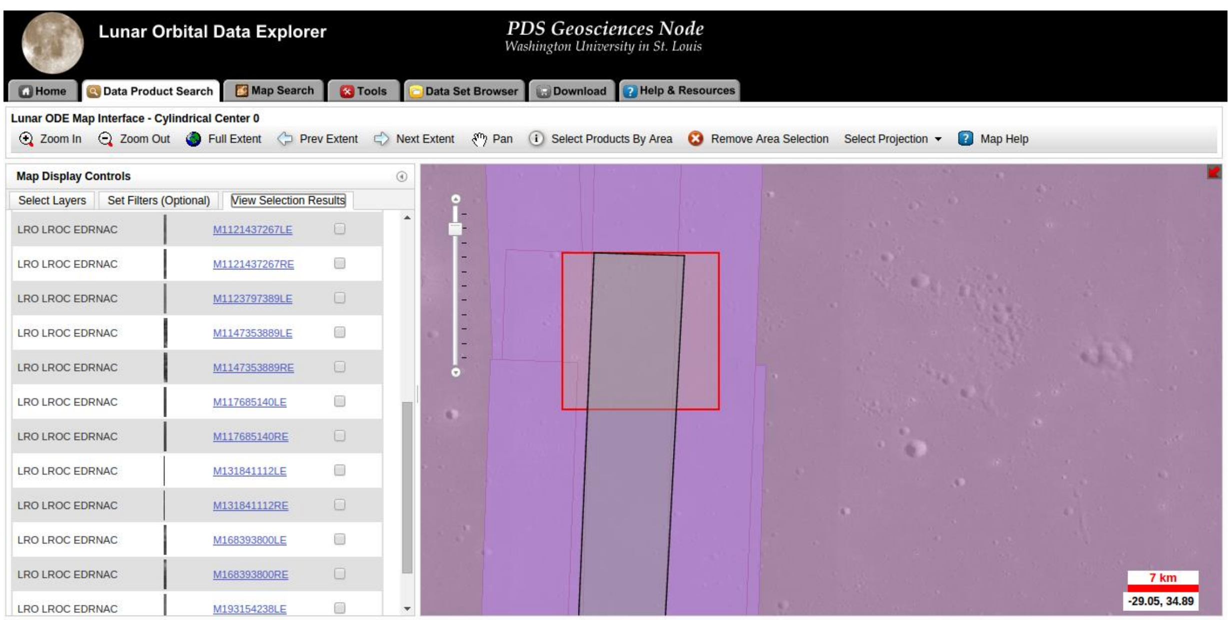
DEM generation procedure
The first step always starts with analyzing the area in question and gathering the exact data set required for that area. For performing calibration and other standard ISIS's image processing, you will be need the 'raw images' (NAC EDR) from PDS geosciences node. Stating again, our current area of investigation is 29 - 29.5 N and 35 – 35.5 W which is approximately a window of 14 x 14 km square.
A summary of the commands employed (in sequence) for generating DEM is stated below:
>> ISIS based commands
`lronac2isis` – It imports an LRO NAC image as an ISIS cube. An ISIS cube definition.
`lronaccal` – Radiometrically calibrates and LRONAC image.
`lronacecho` – Removes echo effects from lronac image.
`spiceinit` – for attaching kernel, pointing, position information to the camera cube files.
`cam2map` – Converts camera image to a map projection. Default sinusoidal projection with min max lat lon as 29 – 29.5, 35 – 35.5 respectively with trimming option enabled.
`demprep` – ISIS also have a stereo building command using ISIS's SMTK toolkit. `demprep` is the post processing command of for the DEMs generated by ISIS. However, better map project result of images can be obtained by projecting the resulting .cub on a higher resolution elevation source like the WAC – DTM (surface terrain map). This can be achieved by using the command `demprep`. Visual analysis with respect to the lunar map on product search of Lunar orbiter data explorer, suggests a better orientation but it is not used in the DEM generation process. However there is an ambiguity in the usage of this command on .cub files. Please refer to the documentation of ISIS and ASP for more details.
`cubenorm` – Different lightening condition in overlapping images affects the feature finding algorithm and thus needed to be rectified. Normalization of the image is a solution done using `cubenorm`.

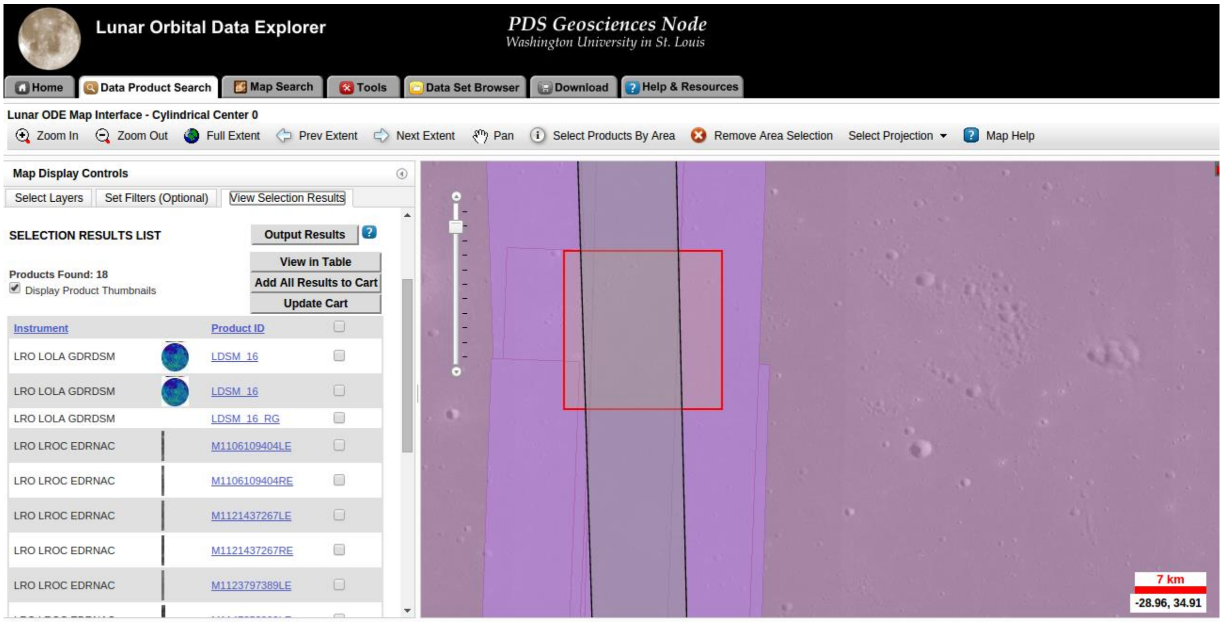
>> ASP based commands
In ASP, our goal is to use the above processed images in ISIS for making DEMs. So it is better to identify the overlapping combinations first. Visual inspection in `qview` is not recommended because it is

exceedingly difficult to identify the overlapping region as there is no prominent topographic feature but rather plane fields.

LODE (lunar orbital data explorer) is the directory that maintains the archive of most of the lunar missions made public. It has an option to display the selected lunar products on the map. One can identify the pairs by visually analyzing the mapped products. The analyzed stereo bundle for TI's landing site in investigation are summarized below.







The overlapping images

[Source: Lunar orbital data explorer]

Codes

M1106L – M1106109404LE
M1106R – M1106109404RE
M1121L – M1121437267LE
M1121R – M1121437267RE
M1147L – M1147353889LE
M1147R – M1147353889RE
M1683L – M168393800LE
M1931L – M193154238LE

BUNDLE WITH 2 IMAGES	BUNDLE WITH 3 IMAGES
M1106R – M1147L	**M1106L – M1121R – M1147R**
M1147L - M1683L	
M1931L – M1121L	
M1147R – M1121R	

A brief about `stereo.default` file:

You can find an instance of `stereo.default` file in the examples folder of StereoPipeline. `stereo.default` file contains the control parameters for stereo command. These parameters needed to be adjusted for getting a nice DEM. Though it is not necessary to manually adjust those parameters and we can simply use the default values, but it is recommended to do so in order to reduce the computation time with little difference in final result. The `stereo.default` parameters used here are listed below.

```
# -*- mode: sh -*-
# Pre-Processing / stereo_pprc
#################################################################
```

```
# Pre-alignment options
# Available choices are (however not all are supported by all
sessions):
#    NONE           (Recommended for anything map projected)
#    EPIPOLAR       (Recommended for Pinhole Sessions)
#    HOMOGRAPHY     (Recommended for ISIS wide-angle shots)
#    AFFINEEPIPOLAR (Recommended for ISIS narrow-angle and DG
sessions)
alignment-method none

# Intensity Normalization
force-use-entire-range       # Use entire input range
# Select a preprocessing filter:
#
# 0 - None
# 1 - Subtracted Mean
# 2 - Laplacian of Gaussian (recommended)
prefilter-mode 2

# Kernel size (1-sigma) for pre-processing
#
# Recommend 1.4 px for Laplacian of Gaussian
# Recommend 25 px for Subtracted Mean
prefilter-kernel-width 1.4

# Integer Correlation / stereo_corr
################################################################

# Select a cost function to use for initialization:
#
# 0 - absolute difference (fast)
# 1 - squared difference  (faster .. but usually bad)
# 2 - normalized cross correlation (recommended)
cost-mode 2
# Initialization step: correlation kernel size
corr-kernel 13 13
# Initializaion step: correlation window size
# corr-search -80 -2 20 2
#corr-search -500 -175 275 175
# Subpixel Refinement / stereo_rfne
################################################################

# Subpixel step: subpixel modes
#
# 0 - disable subpixel correlation (fastest)
# 1 - parabola fitting (draft mode - not as accurate)
# 2 - affine adaptive window, bayes EM weighting (slower, but much
more accurate)
# 3 - affine window, (intermediate speed, results similar to bayes EM)
subpixel-mode 2
```

```
# Subpixel step: correlation kernel size
subpixel-kernel  17 17

# Post Filtering / stereo_fltr
################################################################

# Fill in holes up to 100,000 pixels in size with an inpainting method
#enable-fill-holes

# Automatic "erode" low confidence pixels
filter-mode 1
rm-half-kernel 5 5
max-mean-diff 3
rm-min-matches 60
rm-threshold 3
rm-cleanup-passes 1

# Triangulation / stereo_tri
################################################################

# Size max of the universe in meters and altitude off the ground.
# Setting both values to zero turns this post-processing step off.
near-universe-radius 0.0
far-universe-radius 0.0
```

An important note to keep in mind is that for camera projected images the '`pre-alignment`' option must be set to "`AFFINEEPIPOLAR`" and for map projected images it should be "`NONE`". It has been observed that map projected images give much better results than AFFINEEPIPOLAR. Also `corr-kernel` size must not exceed 21 as computation times starts increasing exponentially as we exceed corr-size 7 with little or no difference in the final results after corr-size 13.

For fine tuning these parameters, you need to run stereo command repeatedly as of trail and error. To combat enormous time consuming processes, it is highly recommended to use the ISIS's '`maptrim`' command (if using map projected images). maptrim reduces the pixels outside of a given lat lon range as null (trimming) and can also crop the image outside that range (cropping). Selecting both the options is recommended. Also, we can employ the 'reduce' command for scaling down the size of images but it is not recommended as the generated DEM might not match the actual one after scaling down. A reduction by a scale of 8 or reducing down the image size to nearly 10 MB is feasible. Another way to reduce the dimension of the image is to manually cropping the image using 'crop' command. For using the crop command, use the '`qview`' tool's tracking option for accurately determining the area to be cropped.

After determining the overlapping stereo bundles, run the following commands for generating the point cloud file. Make sure to reduce the dimensions using any of the above listed procedure if you're using the stereo tool fist for determining `stereo.default` file's control parameters:

```
stereo –stereo-file ~/stereo.default img1.cub img2.cub img3.cub ….
output_directory/output_prefix
point2dem –r moon –orthoimage output_prefix-L.tif output_prefix-PC.tif
```

For judging the results after stereo command is completed, viewing the goodpixelmap is handy. The red pixels in the map shows the regions of failed correlation and similarly the grey are the regions of successful correlation. Another tool for viewing the DEM is ‘`osgviewer`’. For using that tool, run the following command.

```
point2mesh –l –s 5 output_prefix-PC.tif output_prefix-L.tif
```

switch –l is the option for toggling additional illumination and –s is the sampling rate. The default sampling rate is 10. That means it samples every 10th pixel and depending on your DEM you might want a higher sampling rate like 5. To see the prespective views use:

```
osgviewer outputprefix.osgb
```

In the `osgviewer` you can fly around using mouse, but the following keys are useful:
`c`: takes a screenshot (the same thing as hitting print screen)
`h`: allegedly displays help, but in practice they are tiny hieroglyphs on the screen
`l`: lighting on/off (if -l switch was used)
`t`: texture on/off (i.e. only the DEM-based surface model (mesh), or with the left image)
`w`: wire model on/off
`z`: starts recording the camera path
`Z`: stops recording the camera path and saves it to saved_animation.path

To playback the recorded camera path:

```
osgviewer outputprefix.osgb -p saved_animation.path
```

While in the playback, hitting space bar sets you back to the beginning of the path, and hitting c produces a short pause.
Also you can make a colormap and hillshade, the commands are as follows:

```
colormap outputprefix-DEM.tif -o outputprefix-DEM_color.tif
hillshade outputprefix-DEM.tif -o outputprefix-DEM_shade.tif -e 30
```

The -e switch defines the elevation of the light source, in this case 30º (0º being logically the horizon). You can also colourise your hillshade:

```
colormap outputprefix-DEM.tif -s outputprefix-DEM_shade.tif -o
outputprefix-DEM_colorshade.tif
```

[Reference: Ohman_2013_ISIS-ASP-ArcMap_workflow]

These tools are just for visualisation and almost all the times, It is desired to use the stereo generated point cloud outside of ASP. For that it can be converted to Las file format, which is a public file format for the interchange of 3-Dimensional point cloud data. The tool point2las can be used for this purpose.

```
point2las -r moon results/output-PC.tif
```

A Brief description of control parameters `stereo.default`

`>> alignment-method`
Homography - Uses tie points between images to prealign using feature based image matching technique.
Affineepipolar - Uses the above mentioned tie point algorithm but in addition, transform the images such that pair of conjugate epipolar line become collinear and parallel to one of the image axes.
epipolar - Stereo will apply 3D transform to images so that epipolar lines are horizontal.

`>> force-use-entire-range`
Normalize the images and use the channel value that are +-2 standard deviation from mean 1.0. An equivalent command is '`histeq`' in ISIS. You can use `histeq` and disable force-use-entire-range.

`>> prefilter mode`
subtracted mean - Takes large Gaussian kernel and subtracts from input image. Useful with different image intensities.
LoG filter - takes laplacian of Gaussian if input image. Matches blob thus immune to difference in lightening conditions.

`>> prefilter-kernel-width`
size of Gaussian kernel to used for the above pre-processing steps.

`>> corr-seed-mode`
selects a strategy for solving low resolution integer correlation.
0 - None
1 - Low resolution disparity from stereo
2 - Low resolution disparity from DEM
3 - Disparity from full resolution at sparse number of points.

`>> subpixel-kernel`
specify the size of subpixel correlation kernel.

`>> filter-mode`
0 - none
1 - discard pixel at which disparity differs from mean disparity
2 - filter by discarding pixels at which percentage of neighboring disparities are within rm-threshold of current disparity is less than rm-min-match.

rm-half-kernel
erodes isolated regions that are in disagreement with neighbors

max-min-diff
sets max difference between mean current and neighbor pixel disparities and chooses which disparity values to be retained.

rm-min-matches
Percentage of neighboring disparity value that must lie within inliner threshold.
[Reference: ASP_book]

>Scope of improvement

Possibilities of improvement in the quality of the data set cannot be ignored with the commands `bundle_adjust` and `pc_align`. At the time of documentation of this report the testing of these commands were on very initial phases and could not be generalised for all the stereo image bundles and that's why not included in the DEM generation process. However their testing and analysis must not be overlooked, especially `pc_align`. A summary of these commands is noted below:

pc_align

Often the point cloud information generated by the stereo can be quite accurate. But as the positional and velocity errors of the spacecraft cannot be ignored it can have quite significant impact on the actual position of the images on the terrain, which can be misaligned from certain meters to kilometres. Therefore, using a more accurate dataset such as LOLA (lunar orbiter laser altimeter), we can realign our dataset to more precise lat lon values. `pc_align` uses iterative closest point algorithm and both point to point and point to plane configurations are supported. It has a prerequisite as an initial guess for maximum displacement. The command line follows the format

```
pc_align –max-displacement 200 –datum D_MOON --save-inv-transformed-referenc-points –csv-format '1:lon 2:lat 3:radius_m' stereo-PC.tif lola.csv
```

The LOLA dataset used here is publicly available on PDS geosciences in the format Gridded Data Record GDRDEM.

bundle_adjust

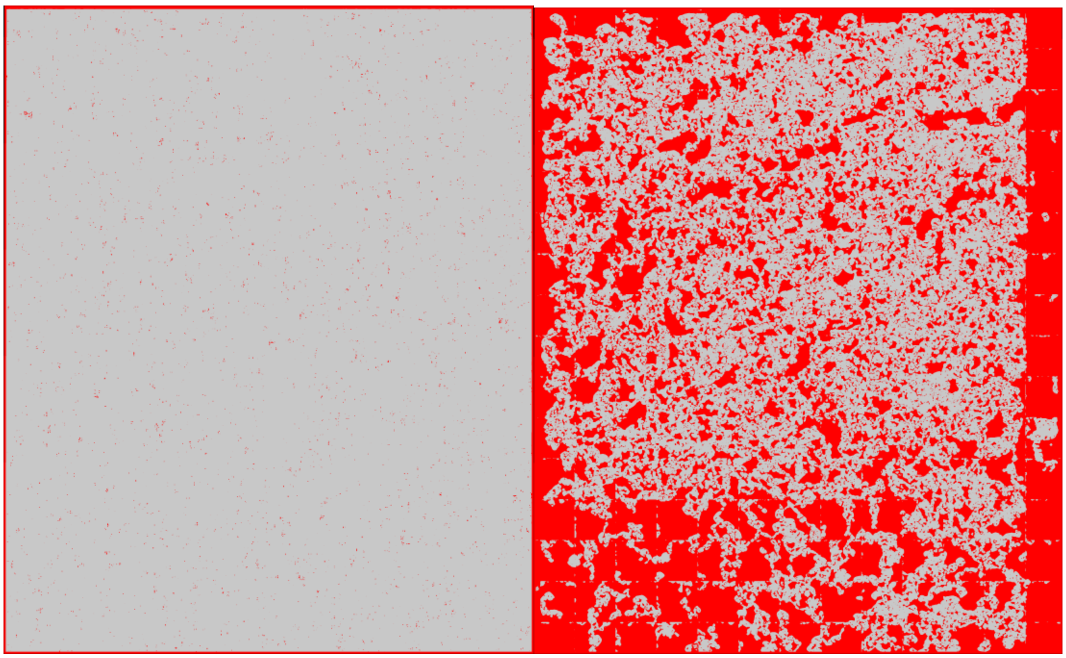
Speaking about the design of ISIS software, the errors generated due to positional and orientation of spacecraft propagate throughout the processes and finally get accumulated in the DEM. Severe distortions can occur resulting in twisted or 'taco shaped' DEM. But in most cases, they are quite subtle and hard to detect. This problem is significant in older datasets such as Voyager and Apollo datasets. Bundle adjust work by matching features of 3D objects in stereo bundle of images. Therefore, bundle adjust is not dependent on an outside source such as an altimeter data for its accuracy, so could prove to be dangerous and consistency of the results in reality must be verified before the DEM is used. However it is recommended to use this tool along with `pc_align` as it makes DEMs more internally consistent and is a correct way of co-registering our DEM with an existing model such as LOLA.
Both ASP and ISIS supports bundle adjust. In ISIS it's known by the command 'jigsaw'. Though ASP's `bundle_adjust` command corrects the geometric shape of the proble, but cannot guarantee that the solution will have correct scale and translation. This is fixed in jigsaw by using two more cost functions. One, which constraints camera parameters to stay relatively close to their initial values and second, a small handful of 3D control points can be user defined and added to the error metric to constrain these points as known locations. However using the `pc_align` command along with `bundle_adjust` can certainly eliminate the limitations of ASP's ambiguities.
Jigsaw has inputs in the form of CNET (input control net) and a list containing the names of the cubes along with their address. For obtaining the control net use the ISIS command 'autoseed'. You can get more information about autoseed on the following link.
http://isis.astrogeology.usgs.gov/IsisWorkshop/index.php/Autoseed
ASP's `bundle_adjust` is more intuitive and easier to use. The command line is as follows.

```
bundle_adjust img1.cub img2.cub img3.cub….. –o run_ba/run
```

```
stereo –stereo-file ~/stereo.default  img1.cub img2.cub img3.cub…..
stereo/test –bundle-adjust-prefix run_ba/run
```

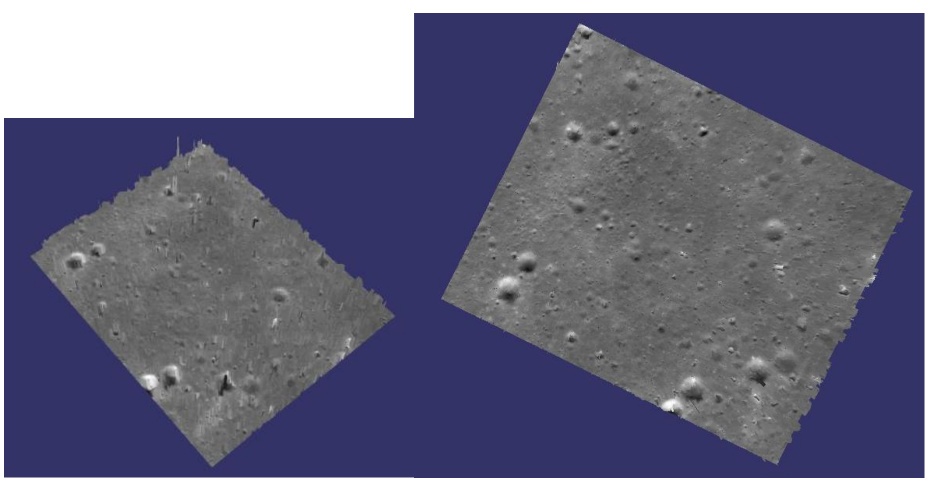
> Generated DEM characteristics

At the time of documentation of this report, I was able to successfully build one DEM for the image sets M1106L – M1121R – M1147R which cover more than 50 percent of the area in inspection with dimensions roughly 7km x 7km (330.6 - 330.8 East, 35.1 - 35.3 West). The generated DEM very well contains the actual landing site. The good pixel map of these DEMs shows the areas of successful correlation. Red for failed correlation (holes) and grey for successful correlation.



Good pixel Map of 1st and 2nd pair of correlation showing partially failed correlation in 2nd pair

The good pixel map of first correlation pair shows excellent overlapping between 2 images. However, certain disparities in the second correlation pair has created noise in the final product. This can be corrected using `pc_align` and `bundle_adjust` as discussed earlier.



Two perspective of generated DEM with little noise but no distortion.

> Characteristics of landing site

The analysis of the landing site using DEM was in initial phase. By visual inspection in the `osgviewer` tool we can say that the landing site has some shallow craters. The color map of the generated DEM suggest a very flat surface with almost no local variations. LOLA data set suggest 95% slopes less than 16^o and lunar regolith is intact with maximum slopes lying in the range 4 to 8 degrees. There is no topographic feature creating shadows and all of this is in good agreement with the landing site constraints we checked.

> Citations and references

- http://api.ning.com/files/kF*sijNveAfQshuXFZrDI60iaRSPVyjIBI7xnNxmc2OcL8DsR9V7P9XV3pn9pGtt81vcy9*zugXaWQziMWvs31xPAyFPM3Mv/moon_phases_diagram.jpg
- http://lunar.gsfc.nasa.gov/lola/
- http://lunar.gsfc.nasa.gov/lola/
- http://imbrium.mit.edu/DOCUMENT/RDRSIS.PDF
- http://target.lroc.asu.edu/q3/
- basilevky morphology of small impact craters
- Height-to-diameter ratios of moon rocks from analysis of Lunokhod-1 and -2 and Apollo 11–17 panoramas and LROC NAC images
- http://www.lpi.usra.edu/publications/books/lunar_sourcebook/pdf/LunarSourceBook.pdf
- http://articles.adsabs.harvard.edu//full/1972LPSC....3.3235M/0003236.000.html
- http://www.lpi.usra.edu/lunar/tools/lunarcratercalc/
- https://etd.ohiolink.edu/rws_etd/document/get/case1233521118/inline
- http://lroc.sese.asu.edu/about
- target.lroc.asu.edu/da/qmap.html
- http://isis.astrogeology.usgs.gov/Isis2/isis-bin/intro_digi_mosaic.cgi
- http://ode.rsl.wustl.edu/moon/
- http://www.lpi.usra.edu/lunar/tools/dems/Ohman_2013_ISIS-ASP-ArcMap_workflow.pdf

Conversion from .las to .csv format

`.las` is the public lidar format containing the lat long and elevation (planetary radius – datum of moon) in binary. Obtaining the DEM in `.las` format might need a conversion to `.csv` for importing the data to other processing softwares, most important MATLAB or OCTAVE. This can be achieved using `las2txt` command available in `liblas` library which is freely available for linux based distros and can be downloaded from Ubuntu software center. For conversion, command line is as follows -

```
las2txt -i input_file.las -o output_file.csv
```

This will generate the required `.csv` file.

[link-www.lilas.org]

GLOBAL PATH PLANNING

ASTAR

Implementation

```
%%variables to divide the DEM into various subsections. Picking 10000
%%points at a time from the file.
R1 = 0;
C1 = 0;
C2 = 2;
R2 = 9999;
```

```
%12435501 is the number of data points avaiable in the DEM file.
while(R2<12435501)
```

```
    %taking only 4 iterations for publishing the document.
    %reading a 10000 values at a time and rearranging and interpolating.
    map_triplet = csvread('dem.csv',R1,C1,[R1 C1 R2 C2]);
    num = size(map_triplet);
    R1 = R2;
    R2 = R2 + num(:,1);
    unit_block = 150; %unit block in cms
```

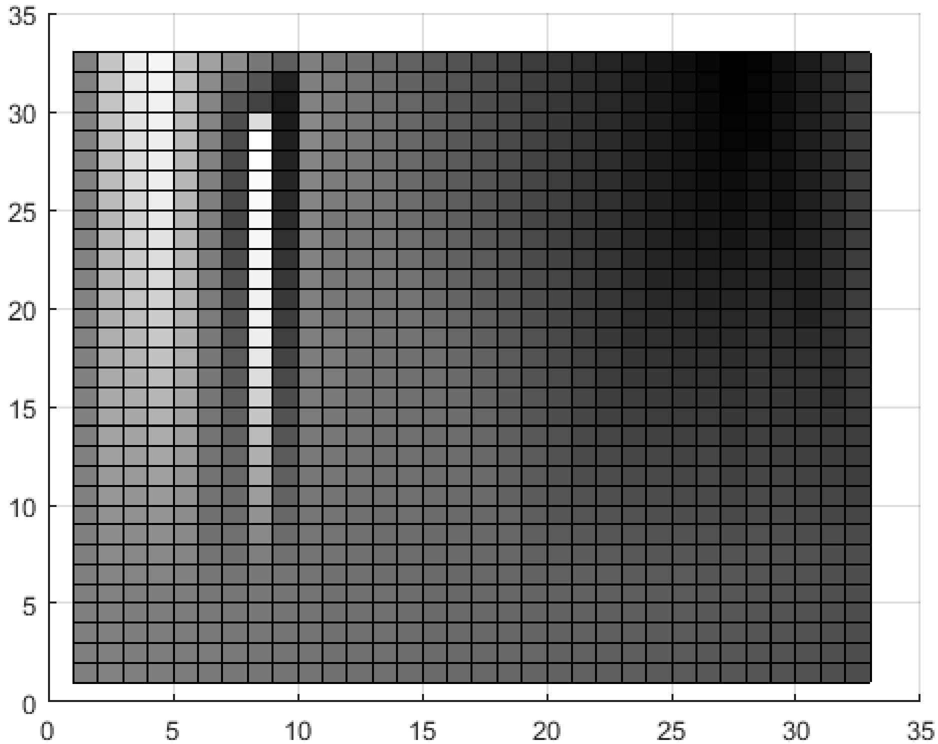
```
    %for converting to easting, northing coordinates.
    for a = 1:num(:,1)
        map_triplet(a,1) = 360 + map_triplet(a,1);
    end
    F = scatteredInterpolant(map_triplet(:,1), map_triplet(:,2), map_triplet(:,3));
    max_Long = max(map_triplet(:,1));
    min_Long = min(map_triplet(:,1));
    max_Lat = max(map_triplet(:,2));
    min_Lat = min(map_triplet(:,2));
```

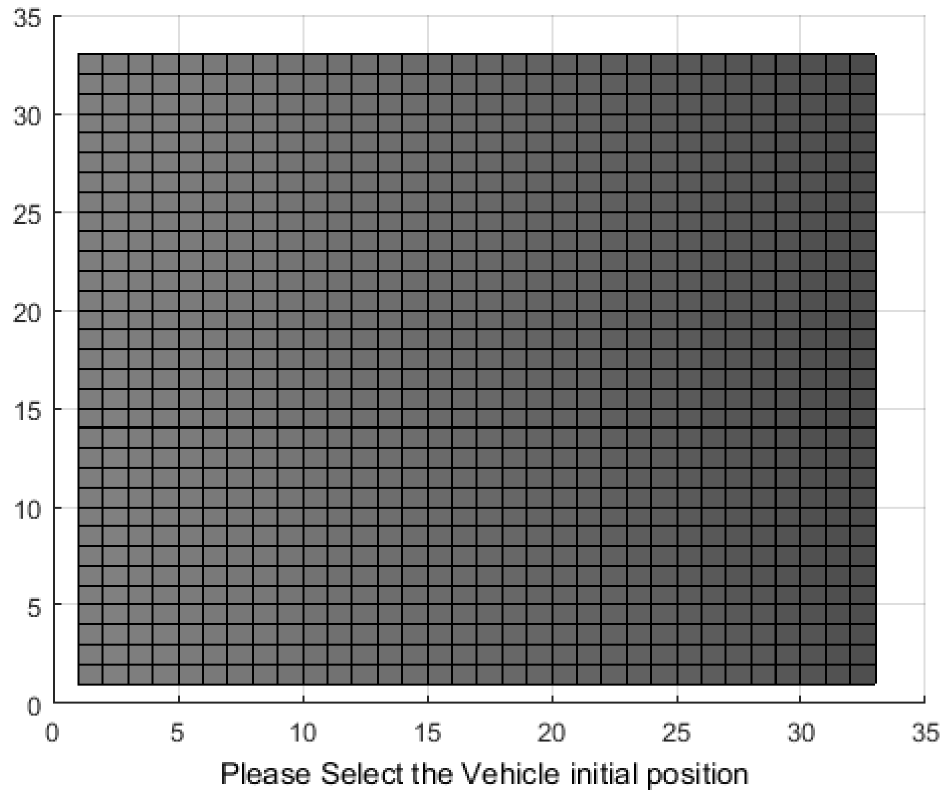
```
    %using the haversine function for interpolating the extracted
    %information.
    x = min_Long:((max_Long-min_Long)/haversine(max_Long, min_Long, min_Lat, min_Lat)*unit_block):max_Long;
    y = min_Lat:((max_Lat-min_Lat)/haversine(max_Long, min_Long, min_Lat, min_Lat)*unit_block):max_Lat;
    [~, MAX_X] = size(x);
    [~, MAX_Y] = size(y);
    %x = reshape(xq, MAX_X*MAX_Y, 1);
    %y = reshape(yq, MAX_X*MAX_Y, 1);
```

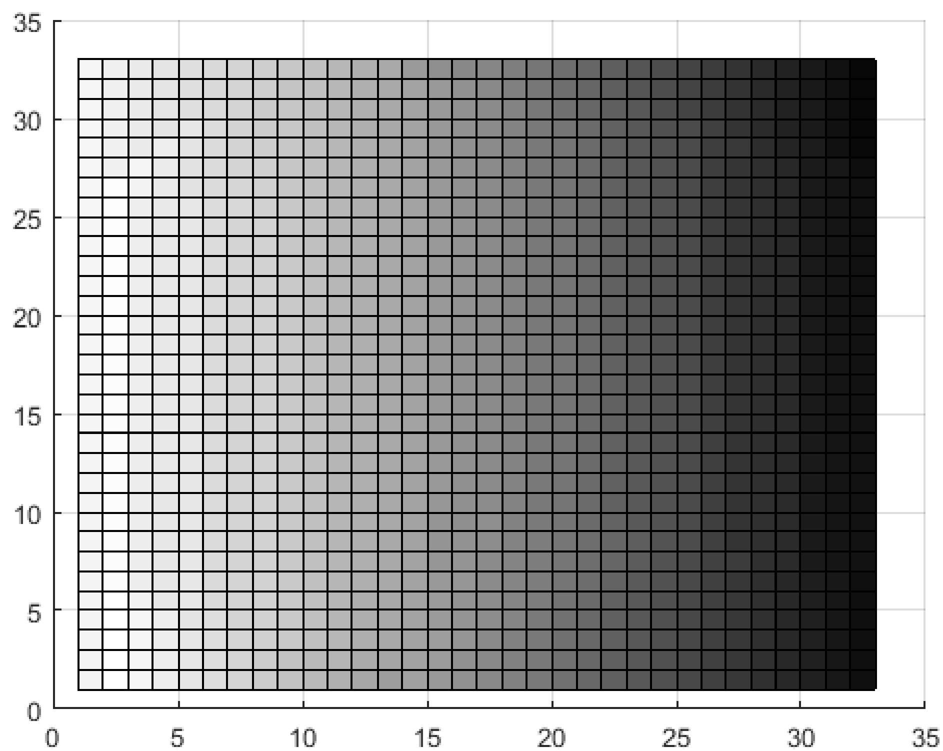
```
    %creating map and putting topography values for evaluation.
    map = ones(MAX_X, MAX_Y);
    for a = 1:MAX_X
        for b = 1:MAX_Y
            map(a,b) = F(x(1,a), y(1,b));
            coor_map(a,1) = x(1,a);
            coor_map(b,2) = y(1,b);
        end
    end
```

```
    %Defining the variables for cost determination.
    phi = 45;   %azimuth
```

```
theta = 30; %elevation

grid on;
hold on;
pcolor(map')
colormap(gray(2000))
```

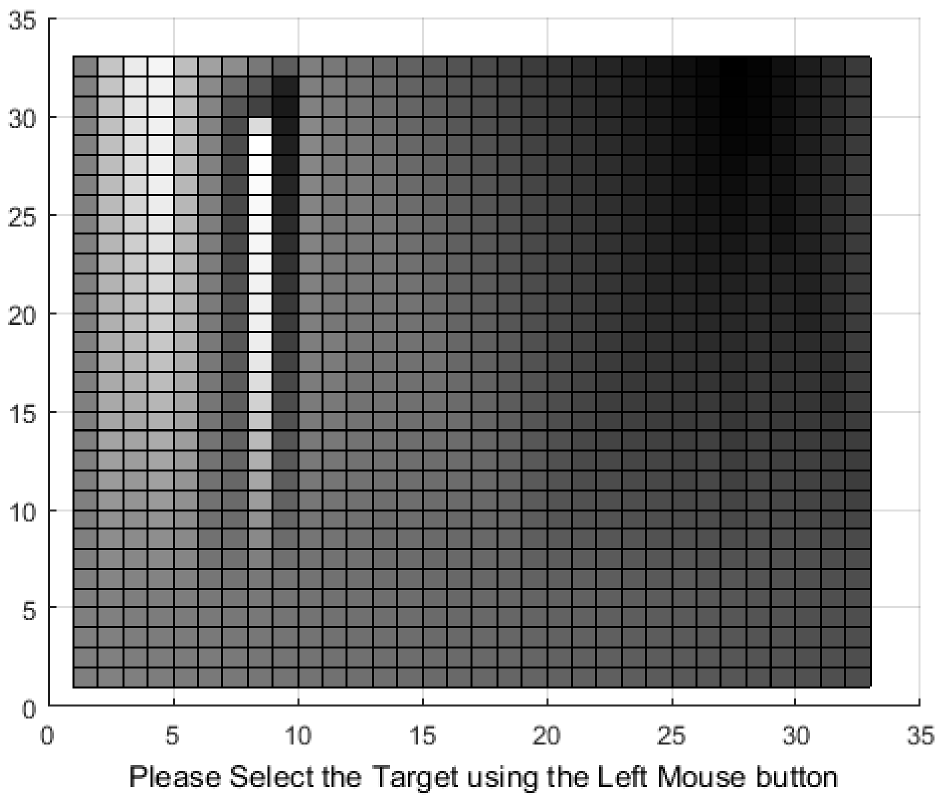


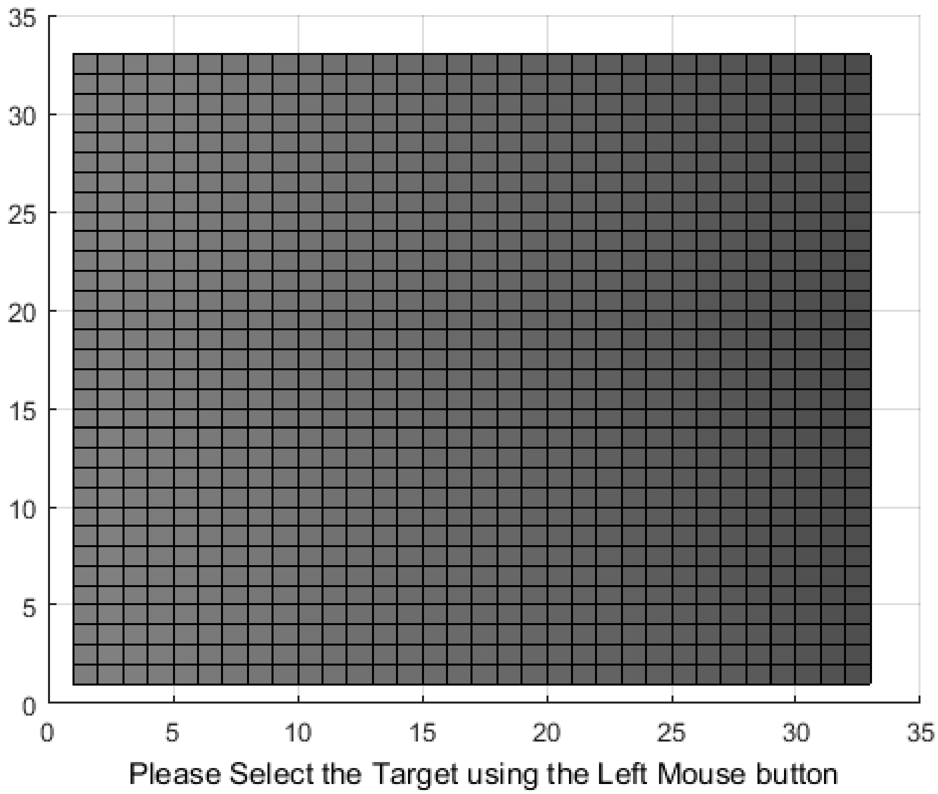




```
%begin interactive start and target
pause(1);
h=msgbox('Please Select the Target using the Left Mouse button');
if ishandle(h) == 1
    delete(h);
```

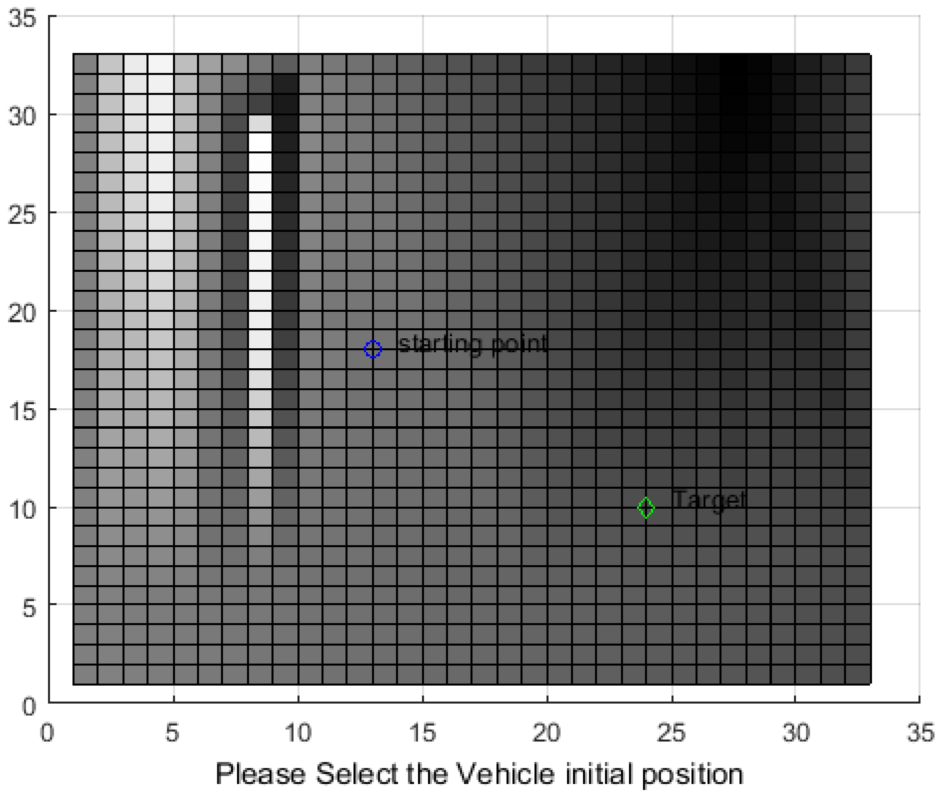
```
end
xlabel('Please Select the Target using the Left Mouse button','Color','black');
but=0;
while (but ~= 1) %Repeat until the Left button is not clicked
    [xval,yval,but]=ginput(1);
end

%getting the values from pointer.
xval=floor(xval);
yval=floor(yval);
xTarget=xval;%X Coordinate of the Target
yTarget=yval;%Y Coordinate of the Target
```

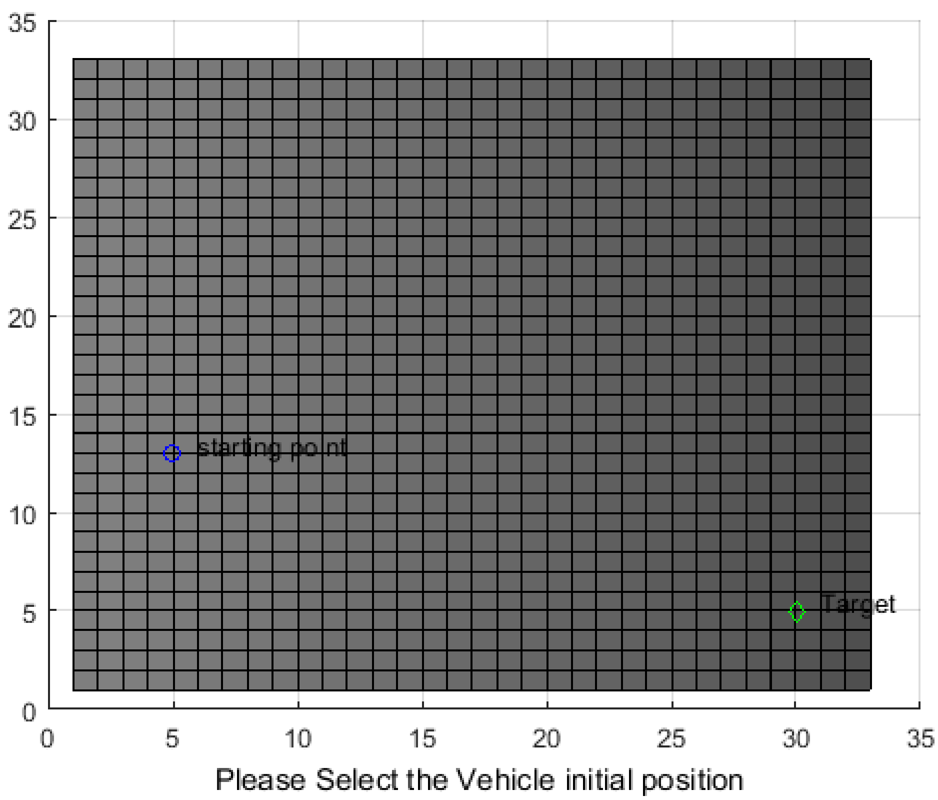





```
Error using ginput (line 84)
Interrupted by figure deletion

Error in main (line 71)
        [xval,yval,but]=ginput(1);
```

```
    %not initializing map with target location cause map should be untouched.
    plot(xval,yval,'gd');
    text(xval+1,yval+.5,'Target')
    pause(1);
    h=msgbox('Please Select the vehicle initial position using the Left Mouse button');
    uiwait(h,5);
    if ishandle(h) == 1
        delete(h);
    end
    xlabel('Please Select the Vehicle initial position ','Color','black');
    but=0;
    while (but ~= 1) %Repeat until the Left button is not clicked
        [xval,yval,but]=ginput(1);
        xval=floor(xval);
        yval=floor(yval);
    end
    xStart=xval;%Starting Position
    yStart=yval;%Starting Position

    plot(xval,yval,'bo');
    text(xval+1,yval+.5,'starting point');
```






```
%End of vehicle-Target pickup

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LISTS USED FOR ALGORITHM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

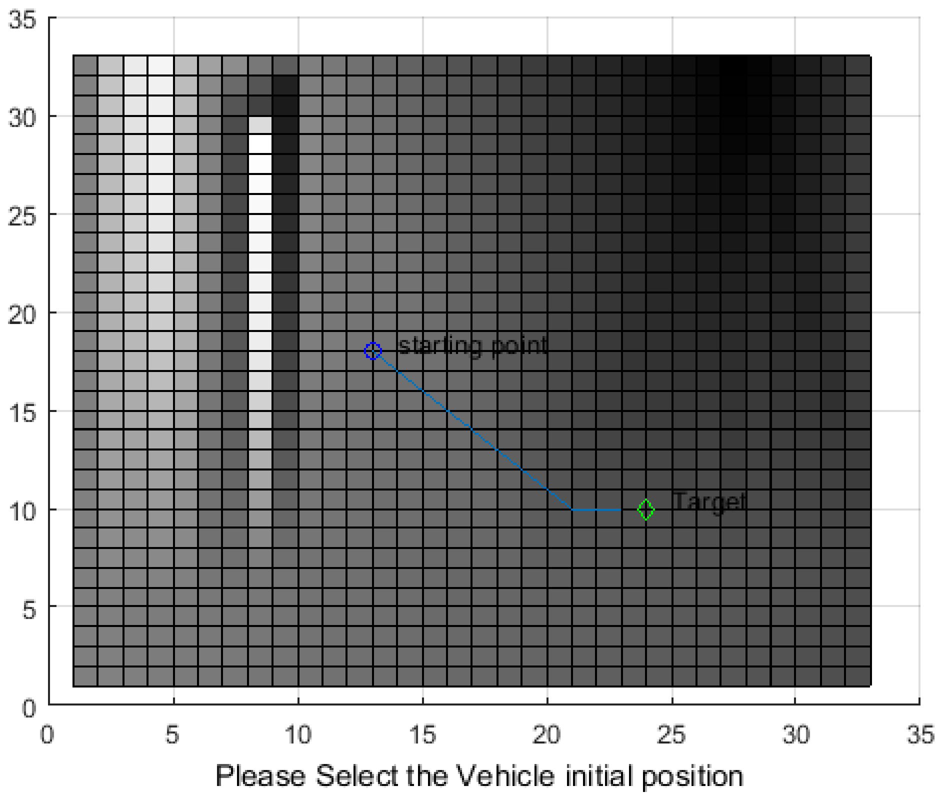
```
RESTRICTED=[]; %restricted points with more deviation than std(map).
OPENList=[];
TRAVERSED=[]; %for keeping track of traversed path (USED FOR PLOTTING). analogous to CLOSED[].
OPTIMUM=[]; %could be used for building hueristics. for keeping track of most optimum path
%(avoiding deadlocks).

%set the starting node as the first node
Node=[xStart, yStart];
Prev=Node;
Raised = [-1,-1];
num_trav = 0;
num_opt = 0;
deadlock = 0;
Target = [xTarget, yTarget];
```

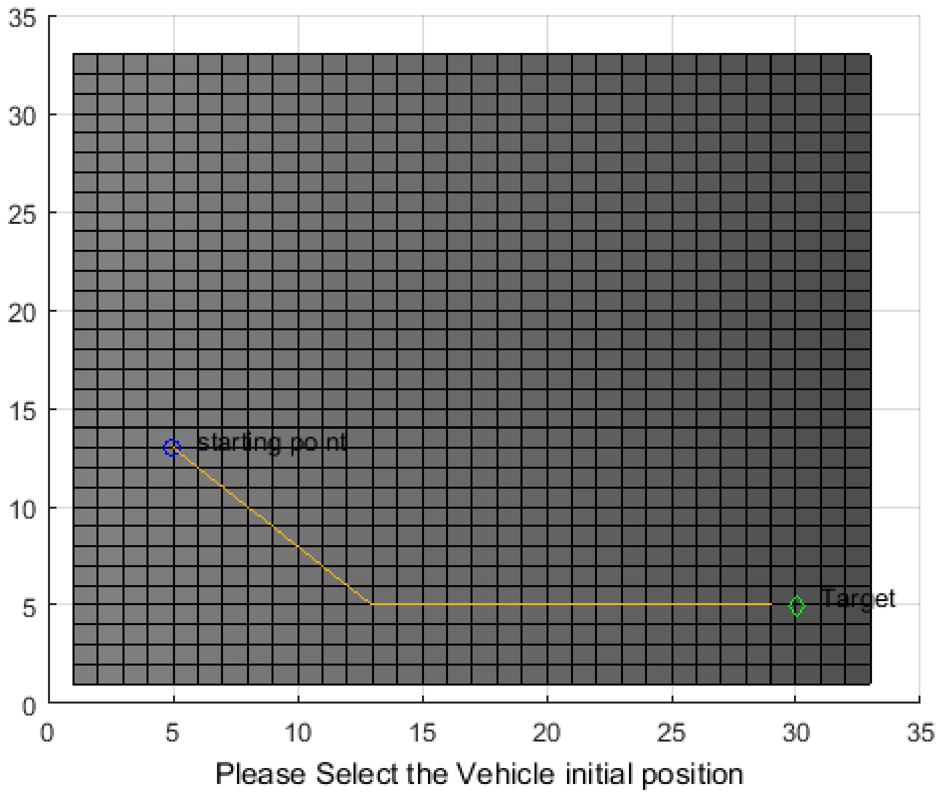
```
%Satisfying the least cost criteria, the ASTAR would get the Target
%node starting from the start point to the end point. While loop starts
%here
while (Node(:,1)~=Target(:,1)) || (Node(:,2)~=Target(:,2))
    OPENList = openNodes(Node, Prev, Raised, RESTRICTED, TRAVERSED, MAX_X, MAX_Y);
    sz = size(OPENList);
    Prev = Node;
    if sz(:,1)~=0
        num_opt = num_opt+1;
        OPTIMUM(num_opt,:) = Node;
        Node = nextPoint(OPENList, Node, Target, coor_map, phi, map);
        OPENList=[];

    %condition when deadlock occurs. No points available for traversal so no points are open
    elseif num_trav>0 && sz(:,1)==0

        %If an element is traversed again, it can create a deadlock
        %situation as the least cost node will be traversed again.
        %DELETING THE LAST ELEMENT FROM OPTIMUM BEACAUSE THAT IS A
        %DEADLOCK.
        Node = OPTIMUM(num_opt,:);
        OPTIMUM(num_opt,:) = [];
        num_opt = num_opt - 1;
    else
        h=msgbox('Sorry, No path exists to the Target!','warn');
        uiwait(h,5);
        break;
    end
    num_trav = num_trav +1;
    TRAVERSED(num_trav,:) = Node;
end
```

```
%Code for plotting the most optimum path.
j=size(OPTIMUM,1);
if ( (Node(:,1) == Target(:,1)) && (Node(:,2) == Target(:,2)) )
```

```
%code for verification of path traversed, calculating costs for
%all grids and plotting it on map.
%Plot the OPTIMUM Path!
 j=size(OPTIMUM,1);
```

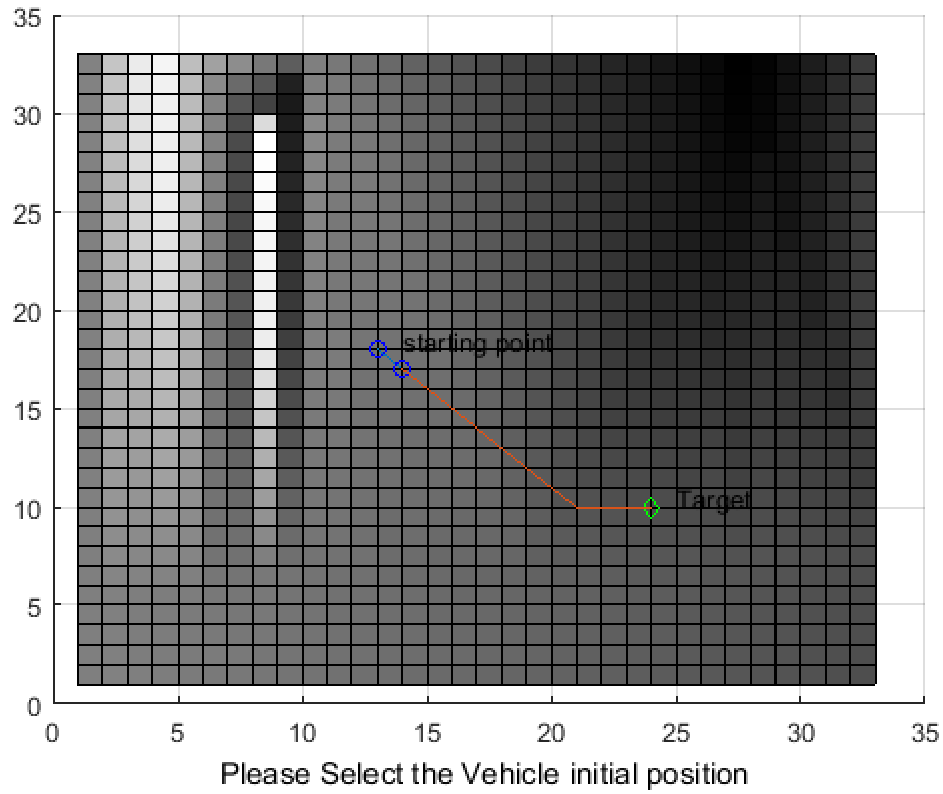
```
p=plot(OPTIMUM(j,1),OPTIMUM(j,2),'bo');
j=j-1;
for i=j:-1:1
    pause(.005);
    set(p,'XData',OPTIMUM(i,1),'YData',OPTIMUM(i,2));
    drawnow;
end;
plot(OPTIMUM(:,1),OPTIMUM(:,2));
hold on;
```

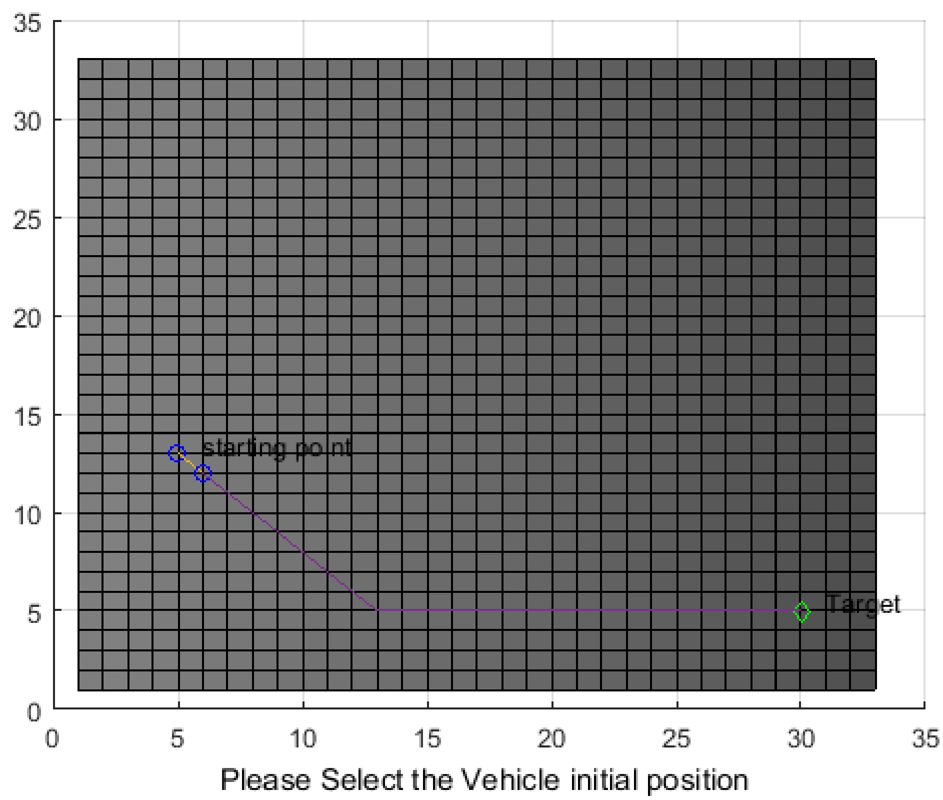





```
%PLOTTING TRAVERSED PATH FOR COMPARISON.
p=plot(TRAVERSED(j,1),TRAVERSED(j,2),'bo');
j=j-1;
for i=j:-1:1
    pause(.005);
    set(p,'XData',TRAVERSED(i,1),'YData',TRAVERSED(i,2));
    drawnow ;
end;
plot(TRAVERSED(:,1),TRAVERSED(:,2));
```

```
 else
     pause(1);
     h=msgbox('Sorry, No path exists to the Target!','warn');
     uiwait(h,5);
end
```


starting point
Target
Please Select the Vehicle initial position


starting point
Target
Please Select the Vehicle initial position

```
end
```

GLOBAL PATH PLANNING

FMINCON

Implementation

Contents


```
%Defining parameters for DEM file division and intial cost parameters.
R1 = 0;
C1 = 0;
C2 = 2;
R2 = 9999;
%Sun parameters
azimuth = 45;   %azimuth
elevation = 30; %elevation
unit_block = 100; %unit block in cms
%A standard input to fmincon solver for determing the number of points
%number of interpolated points between waypoints.
fineness = 101;
```

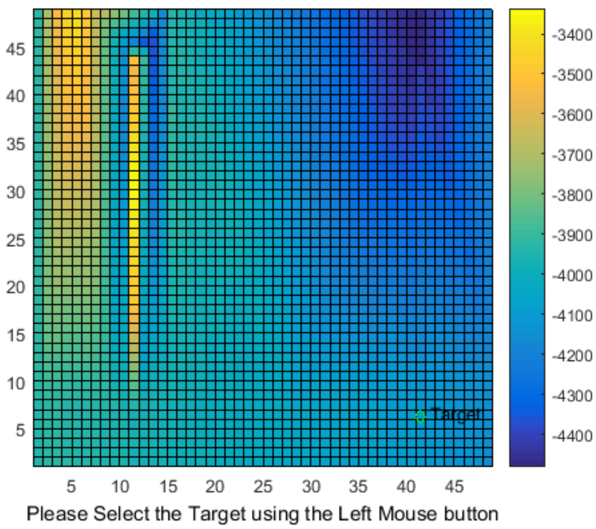
```
while(R2<12435501)
```

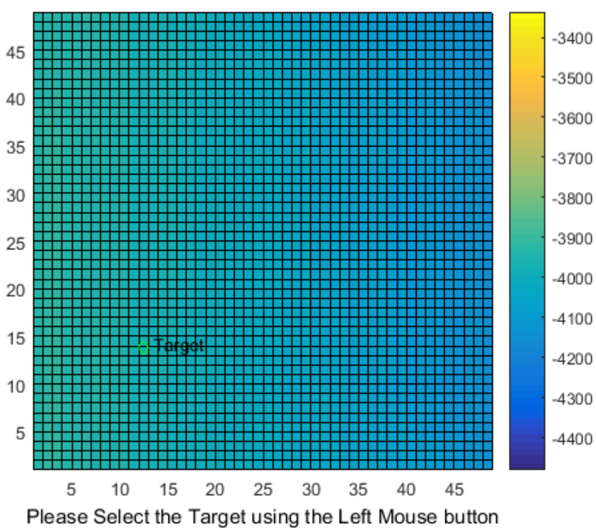
```
    %reading a 1000 values at a time.
    map_triplet = csvread('dem.csv',R1,C1,[R1 C1 R2 C2]);
    num = size(map_triplet);
    R1 = R2;
    R2 = R2 + num(:,1);

    %for converting to easting, northing coordinates.
    for a = 1:num(:,1)
        map_triplet(a,1) = 360 + map_triplet(a,1);
    end

    F = scatteredInterpolant(map_triplet(:,1), map_triplet(:,2), map_triplet(:,3));
    max_Long = max(map_triplet(:,1));
    min_Long = min(map_triplet(:,1));
    max_Lat = max(map_triplet(:,2));
    min_Lat = min(map_triplet(:,2));
    %[xq, yq] = meshgrid(min_Long:((max_Long-min_Long)/haversine(max_Long, min_Long, min_Lat, min_Lat)*unit_block):max_Long, min_Lat:((max_Lat-min_Lat)/haversine(ma

    x = min_Long:((max_Long-min_Long)/haversine(max_Long, min_Long, min_Lat, min_Lat)*unit_block):max_Long;
    y = min_Lat:((max_Lat-min_Lat)/haversine(max_Long, min_Long, min_Lat, min_Lat)*unit_block):max_Lat;
```

```
    [~, MAX_X] = size(x);
    [~, MAX_Y] = size(y);

    %creating topography map in accordance with coordinate map.
    map = ones(MAX_X, MAX_Y);
    for a = 1:MAX_X
        for b = 1:MAX_Y
            map(a,b) = F(x(1,a), y(1,b));
            coor_map(a,1) = x(1,a);
            coor_map(b,2) = y(1,b);
        end
    end
```

```
    grid on;
    hold on;
    axis equal tight
    pcolor(map')
    %begin interactive start and target
    pause(1);
    h=msgbox('Please Select the Target using the Left Mouse button');
    if ishandle(h) == 1
        delete(h);
    end

    xlabel('Please Select the Target using the Left Mouse button','Color','black');
    but=0;
    while (but ~= 1) %Repeat until the Left button is not clicked
        [xval,yval,but]=ginput(1);
    end

    %xval=floor(xval);
    %yval=floor(yval);
    xTarget=xval;%X Coordinate of the Target
    yTarget=yval;%Y Coordinate of the Target

    %not initializing map with target location cause map should be untouched.
```

```
    plot(xval,yval,'gd');
    %%why, 0.5?
    text(xval+1,yval+.5,'Target')
```

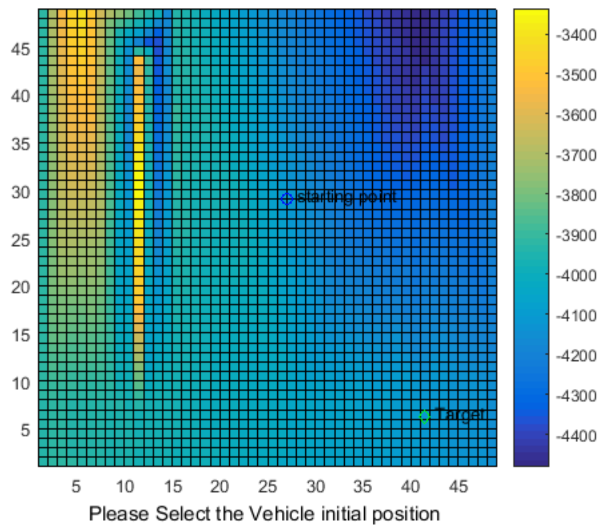


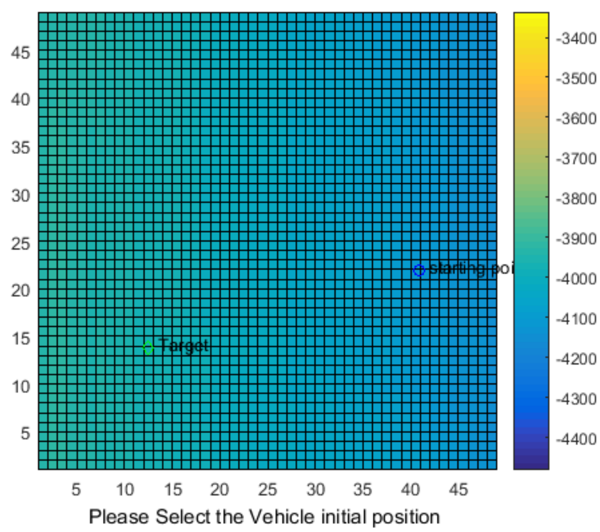



```
Error using ginput (line 84)
Interrupted by figure deletion

Error in main (line 67)
        [xval,yval,but]=ginput(1);
```

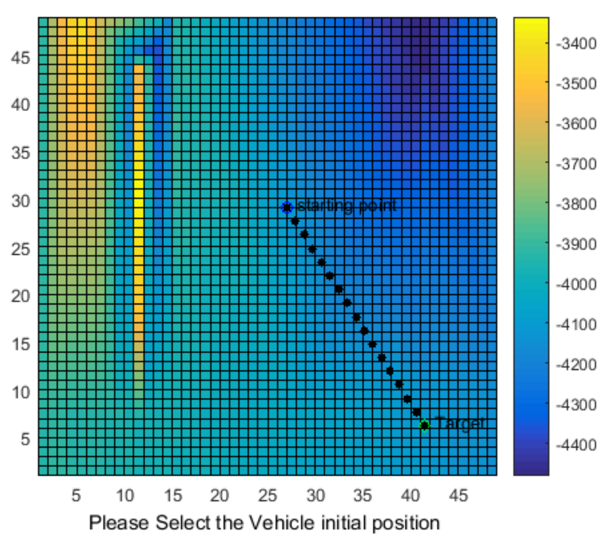
vehicle's initial position

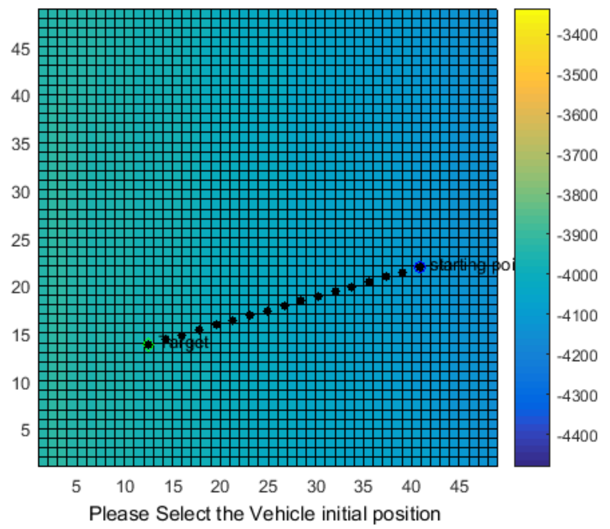
```
    pause(1);
    h=msgbox('Please Select the vehicle initial position using the Left Mouse button');
    uiwait(h,5);
    if ishandle(h) == 1
        delete(h);
    end
    xlabel('Please Select the Vehicle initial position ','Color','black');
    but=0;
    while (but ~= 1) %Repeat until the Left button is not clicked
        [xval,yval,but]=ginput(1);
        %xval=floor(xval);
        %yval=floor(yval);
    end
    xStart=xval;%Starting Position
    yStart=yval;%Starting Position
    plot(xval,yval,'bo');
    text(xval+1,yval+.5,'starting point');
```





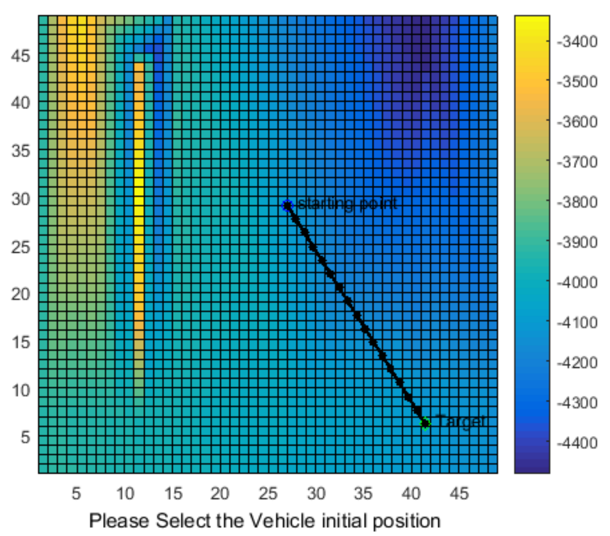

```
numWayPoints = 15;
xWayPoints = linspace(xStart,xTarget,numWayPoints+2)';
yWayPoints = linspace(yStart,yTarget,numWayPoints+2)';

h_wp = plot(xWayPoints,yWayPoints,'color','k','linestyle','none','marker','.','markersize',16);
```

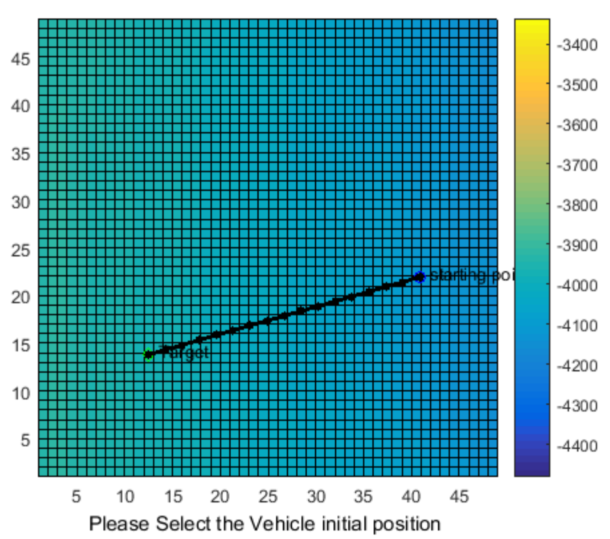




Generate a continuous path from the waypoints

```
PathPoints = WayPoints_To_Path([xWayPoints,yWayPoints],'linear',MAX_X,MAX_Y,fineness);
h_path = plot(PathPoints(:,1),PathPoints(:,2),'k','linewidth',2);
```





Calculate the cost taken

```
StraightLineCost = solverCost( PathPoints, map, coor_map, azimuth, xTarget, yTarget, xStart, yStart, MAX_X, MAX_Y, 'linear', fineness);
```

Find an optimal path using FMINCON

Define Objective Function

```
objectiveFun = @(P) solverCost(P,map,coor_map,azimuth,xTarget, yTarget, xStart, yStart,MAX_X, MAX_Y, 'linear', fineness);

% Set optimization options
opts = optimset('fmincon');
opts.Display = 'iter';
opts.Algorithm = 'active-set';
opts.MaxFunEvals = 10000;

% Initial Conditions
xWayPoints = linspace(xStart,xTarget,numWayPoints+2)';
yWayPoints = linspace(yStart,yTarget,numWayPoints+2)';
ic = [xWayPoints(2:end-1)'; yWayPoints(2:end-1)'];
ic = ic(:);

% Bounds
lb = ones(size(ic(:)));
ub = reshape([MAX_X*ones(1,numWayPoints); MAX_Y*ones(1,numWayPoints)],[],1);

%Do the optimizaiton
optimalWayPoints = fmincon(objectiveFun, ic(:), [],[],[],[],lb,ub,[],opts);
```

```
                                Max     Line search  Directional  First-order 
 Iter F-count        f(x)   constraint   steplength   derivative   optimality Procedure 
    0     31      6.59334       -6.673                                         
    1     63      6.53995       -6.673          0.5        -1.01        0.472   
    2     95      6.37104       -6.699          0.5       -0.808          1.9   
    3    128      6.05322       -6.754         0.25        -1.09         1.21   
    4    163       5.9918       -6.781       0.0625        -1.75         2.15   
    5    196      5.75668       -6.658         0.25        -1.62         1.48   
    6    230        5.697       -6.655        0.125        -1.09         1.32   
    7    264      5.67976       -6.617        0.125       -0.895         1.17   
    8    297      5.38227       -6.498         0.25       -0.966         1.58   
    9    329      5.29679       -6.533          0.5        -1.31         5.34   
   10    362      5.17647       -6.575         0.25        -1.75         2.84   
   11    395       5.1135       -6.537         0.25        -1.01         2.27   
   12    428      5.04664       -6.443         0.25       -0.641         1.44   
   13    459      4.91745        -6.21            1       -0.579         1.55   
   14    492      4.88893       -5.887         0.25       -0.488         3.61   
   15    526      4.83201       -5.838        0.125       -0.515         5.12   
   16    559      4.78637       -5.842         0.25       -0.706         6.72   
   17    593      4.70628       -5.879        0.125       -0.715         20.8   
   18    628       4.6339       -5.896       0.0625       -0.826         26.7   
   19    662      4.62608       -5.913        0.125        -1.08         7.26   
   20    698      4.61886       -5.922       0.0313         -1.2         5.44   
   21    731      4.48192       -5.946         0.25        -1.42         5.75   
   22    763      4.30615       -5.784          0.5        -1.41         3.99   
   23    797      4.25409       -5.763        0.125       -0.806         3.98   
   24    832      4.23827       -5.752       0.0625       -0.754         3.98   
   25    865       4.1864       -5.691         0.25       -0.637         6.74   
   26    900      4.18545       -5.693       0.0625        -0.79         4.78   
   27    934      3.97188        -5.72        0.125       -0.949         6.44   
   28    968      3.92175       -5.714        0.125       -0.853         7.69   
   29   1001       3.8526       -5.648         0.25        -1.02         12.4   
   30   1035      3.78778       -5.609        0.125         -1.2         9.51   
   31   1070      3.74593       -5.575       0.0625        -1.11         8.15   
   32   1103      3.74472       -5.549         0.25       -0.745         7.98   
   33   1138      3.68928       -5.527       0.0625       -0.898         7.95   
   34   1169      3.54645       -5.476            1       -0.957         7.12   
   35   1201      3.44595       -5.456          0.5        -2.22         8.77   
   36   1236      3.42916       -5.463       0.0625         -1.6         6.55   
   37   1268      3.35884       -5.435          0.5        -1.72          9.3   
   38   1299      3.30983       -5.396            1       -0.843          9.7   
   39   1333       3.2981       -5.391        0.125       -0.821         9.81   
   40   1366      3.20328       -5.391         0.25        -1.41         12.2   
   41   1400      3.18164       -5.399        0.125        -1.73         11.5   
   42   1433      3.15598       -5.407         0.25        -1.58         7.47   
   43   1465      3.13804       -5.405          0.5        -1.45         8.28   
   44   1500      3.13426       -5.403       0.0625        -1.18         9.98   
   45   1533      3.08935       -5.385         0.25       -0.778         10.6   
   46   1565      3.02841       -5.373          0.5       -0.636         9.36   
   47   1603      3.02598       -5.372      0.00781       -0.512         9.32   
   48   1635      2.96229       -5.353          0.5       -0.585         10.2   
   49   1668      2.94885       -5.333         0.25       -0.917         12.6   
   50   1700      2.91533       -5.321          0.5        -1.16         11.4   
   51   1734      2.91044        -5.33        0.125       -0.983         13.7   
   52   1767      2.89811       -5.339         0.25       -0.683         13.7   
   53   1801      2.89741       -5.352        0.125       -0.603         13.4   
   54   1834      2.85904       -5.349         0.25       -0.525         22.6   
   55   1869      2.85225       -5.348       0.0625       -0.668         35.1   
   56   1904      2.84341       -5.344       0.0625       -0.662         15.6   
   57   1938      2.83538       -5.332        0.125       -0.474         17.1   
```

58	1971	2.79835	-5.308	0.25	-0.378	38	
59	2009	2.79623	-5.306	0.00781	-0.341	35.6	
60	2045	2.79163	-5.301	0.0313	-0.237	71.8	
61	2079	2.76447	-5.272	0.125	-0.318	55	
62	2114	2.75592	-5.262	0.0625	-0.302	99.9	
63	2149	2.75586	-5.258	0.0625	-0.292	1.01e+03	
64	2187	2.74989	-5.253	0.00781	-0.32	515	
65	2223	2.7423	-5.255	0.0313	-0.368	672	
66	2257	2.72783	-5.279	0.125	-0.344	164	
67	2291	2.71674	-5.28	0.125	-0.249	160	
68	2324	2.69691	-5.286	0.25	-0.86	126	
69	2357	2.68927	-5.293	0.25	-0.645	86.9	
70	2388	2.65172	-5.305	1	-0.477	67.6	
71	2419	2.64556	-5.288	1	-0.238	78.6	
72	2450	2.63851	-5.261	1	-0.244	93.4	
73	2481	2.61187	-5.257	1	-0.343	61.9	
74	2512	2.52837	-5.394	1	-0.401	27.3	
75	2543	2.52105	-5.425	1	-0.144	22.9	
76	2574	2.43263	-5.382	1	-0.923	25.6	
77	2607	2.4252	-5.373	0.25	-0.197	26.1	
78	2639	2.39263	-5.339	0.5	-0.129	26.6	
79	2671	2.38107	-5.319	0.5	-0.253	27.1	
80	2706	2.35402	-5.313	0.0625	-0.354	27.3	
81	2746	2.35069	-5.312	0.00195	-0.607	27.4	Hessian modified
82	2781	2.3466	-5.311	0.0625	-0.252	27.2	
83	2815	2.33862	-5.31	0.125	-0.141	26.7	
84	2849	2.31514	-5.304	0.125	-0.462	42.1	
85	2889	2.31444	-5.302	0.00195	-1.18	80.6	Hessian modified
86	2921	2.28346	-5.268	0.5	-1	42.7	
87	2952	2.26979	-5.284	1	-0.721	32.9	
88	2983	2.26939	-5.287	1	-0.466	31.4	
89	3014	2.25538	-5.29	1	-0.45	27.2	
90	3047	2.24805	-5.272	0.25	-0.3	27.8	
91	3079	2.24037	-5.259	0.5	-0.233	66.8	
92	3111	2.23495	-5.274	0.5	-0.301	55.1	
93	3142	2.21296	-5.275	1	-0.386	68.7	
94	3175	2.20848	-5.27	0.25	-0.315	35.4	
95	3209	2.20729	-5.268	0.125	-0.261	27	
96	3241	2.20009	-5.269	0.5	-0.204	65.1	
97	3273	2.19596	-5.265	0.5	-0.224	51.3	
98	3305	2.19177	-5.264	0.5	-0.228	33	
99	3338	2.17372	-5.266	0.25	-0.231	70.1	
100	3369	2.16705	-5.272	1	-0.229	45	
101	3403	2.16281	-5.271	0.125	-0.0956	46.4	
102	3434	2.1582	-5.27	1	-0.464	44.7	
103	3465	2.13686	-5.267	1	-0.477	58.1	
104	3498	2.12898	-5.264	0.25	-0.224	53.7	
105	3532	2.12864	-5.263	0.125	-0.169	100	
106	3563	2.12545	-5.26	1	-0.13	158	
107	3595	2.12168	-5.259	0.5	-0.29	89	
108	3627	2.11416	-5.26	0.5	-0.413	152	
109	3659	2.11303	-5.268	0.5	-0.573	52.1	
110	3690	2.10022	-5.264	1	-0.126	75.7	
111	3721	2.09816	-5.26	1	-0.0735	66.9	
112	3753	2.09248	-5.26	0.5	-0.0853	136	
113	3784	2.07858	-5.27	1	-0.18	63.8	
114	3815	2.04878	-5.292	1	-0.245	33.4	
115	3846	2.01359	-5.298	1	-0.0806	30.3	
116	3877	1.98508	-5.315	1	-0.0452	38.5	
117	3908	1.92688	-5.324	1	-0.0892	34.6	
118	3939	1.8935	-5.345	1	-0.0964	28.4	
119	3971	1.88259	-5.339	0.5	-0.118	20.9	
120	4002	1.83176	-5.338	1	-0.0699	22.5	
121	4034	1.81952	-5.338	0.5	-0.137	22.7	
122	4065	1.80019	-5.341	1	-0.0799	25	
123	4096	1.79286	-5.342	1	-0.0581	27.7	
124	4130	1.79045	-5.342	0.125	-0.203	27.6	
125	4163	1.78435	-5.341	0.25	-0.116	27.8	
126	4202	1.78397	-5.341	0.00391	-0.25	27.8	
127	4233	1.77086	-5.34	1	-0.374	38	
128	4264	1.76316	-5.341	1	-0.215	40	
129	4297	1.75836	-5.341	0.25	-0.0984	38.1	
130	4333	1.75499	-5.341	0.0313	-0.166	108	
131	4365	1.75195	-5.34	0.5	-0.481	30.4	
132	4396	1.74603	-5.34	1	-0.338	30.3	
133	4429	1.74587	-5.339	0.25	-0.584	30.8	
134	4462	1.7454	-5.338	0.25	-0.208	32.5	
135	4493	1.74469	-5.334	1	-0.216	42	
136	4524	1.74271	-5.332	1	-0.31	43.4	
137	4556	1.7404	-5.331	0.5	-0.176	75.3	
138	4604	1.7404	-5.331	7.63e-06	-6.75	75.7	Hessian modified
139	4637	1.73908	-5.33	0.25	-1.03	73.6	
140	4671	1.73785	-5.33	0.125	-0.742	58.3	
141	4705	1.73776	-5.329	0.125	-0.335	56.6	
142	4738	1.73742	-5.329	0.25	-0.238	54.8	
143	4783	1.73742	-5.329	-6.1e-05	-0.29	54.8	
144	4828	1.73742	-5.329	-6.1e-05	-0.37	54.8	Hessian modified twice
145	4873	1.73742	-5.329	-6.1e-05	-0.278	54.8	Hessian modified twice
146	4910	1.73737	-5.329	0.0156	-0.306	54.2	Hessian modified twice
147	4958	1.73736	-5.329	7.63e-06	-1.91	54.1	Hessian modified twice
148	5003	1.73736	-5.329	-6.1e-05	-2.45	54.1	Hessian modified twice

149	5040	1.73732	-5.329	0.0156	-2.18	53.9	
150	5074	1.73699	-5.329	0.125	-1.15	54.2	
151	5111	1.73695	-5.329	0.0156	-0.812	54.3	
152	5154	1.73693	-5.329	0.000244	-1.36	50.8	Hessian modified
153	5204	1.73693	-5.329	1.91e-06	-4.41	50.8	Hessian modified
154	5254	1.73693	-5.329	1.91e-06	-8.3	50.8	Hessian modified twice
155	5299	1.73693	-5.329	-6.1e-05	-5.83	50.8	Hessian modified twice
156	5344	1.73693	-5.329	-6.1e-05	-6.06	50.8	Hessian modified twice
157	5389	1.73692	-5.329	-6.1e-05	-5.91	50.8	Hessian modified twice
158	5434	1.73692	-5.329	-6.1e-05	-5.27	50.8	Hessian modified twice
159	5479	1.73688	-5.329	-6.1e-05	-5.28	50.7	Hessian modified twice
160	5524	1.73686	-5.329	-6.1e-05	-3.66	50.6	Hessian modified
161	5569	1.73686	-5.329	-6.1e-05	-3.89	50.6	
162	5606	1.73685	-5.329	0.0156	-1.58	49.7	Hessian modified twice
163	5642	1.73684	-5.329	0.0313	-1.18	53.3	
164	5675	1.73676	-5.329	0.25	-0.897	61.4	
165	5707	1.73657	-5.329	0.5	-2.92	53	
166	5738	1.73605	-5.329	1	-2.53	55	
167	5769	1.73558	-5.329	1	-1.1	53.4	Hessian modified
168	5800	1.73509	-5.329	1	-1.23	54.1	
169	5848	1.73505	-5.329	7.63e-06	-15.8	469	Hessian modified
170	5883	1.73504	-5.329	0.0625	-2.3	52.1	
171	5914	1.73436	-5.329	1	-2.27	53.9	
172	5945	1.7327	-5.329	1	-1.77	51.6	Hessian modified
173	5976	1.73243	-5.329	1	-2.03	51.5	
174	6009	1.73174	-5.329	0.25	-3.21	52.3	Hessian modified
175	6040	1.73139	-5.329	1	-3.09	52.4	
176	6074	1.7312	-5.329	0.125	-2.73	53.5	Hessian modified
177	6105	1.73035	-5.329	1	-1.25	53.6	
178	6146	1.73035	-5.329	0.000977	-1.67	53.6	Hessian modified
179	6191	1.73035	-5.329	-6.1e-05	-1.2	53.6	
180	6223	1.73032	-5.329	0.5	-1.36	51.7	
181	6254	1.72944	-5.329	1	-3.7	59.1	
182	6285	1.72911	-5.329	1	-2.41	49	
183	6321	1.72902	-5.329	0.0313	-2.15	53.7	Hessian modified
184	6352	1.72834	-5.329	1	-1.75	53.8	
185	6383	1.72822	-5.329	1	-3.02	53.7	
186	6415	1.72818	-5.329	0.5	-2.48	51.7	Hessian modified
187	6446	1.72793	-5.329	1	-2.29	60	
188	6477	1.72777	-5.329	1	-2.24	59	
189	6517	1.72777	-5.329	0.00195	-3.45	59	Hessian modified
190	6565	1.72777	-5.329	7.63e-06	-5.16	59	Hessian modified twice

Local minimum possible. Constraints satisfied.

fmincon stopped because the predicted change in the objective function
is less than the selected value of the function tolerance and constraints
are satisfied to within the selected value of the constraint tolerance.

No active inequalities.

Iter	F-count	f(x)	Max constraint	Line search steplength	Directional derivative	First-order optimality	Procedure
0	31	32.7692	-9.892				
1	62	32.7692	-9.892	1	-0.00195	0.00139	
2	93	32.4036	-9.935	1	-0.00252	0.455	Hessian modified twice
3	124	30.0475	1.776e-15	1	-0.61	1.37	Hessian modified
4	155	33.8897	8.882e-16	1	-0.117	0.392	
5	186	26.8481	0	1	-0.143	1.09	
6	217	25.1871	0	1	-0.403	256	
7	253	24.7502	-0.0402	0.0313	-68.4	3.13e+03	Hessian modified
8	287	23.9358	-6.035	0.125	-0.367	7.44	
9	322	23.8628	-5.658	0.0625	-0.423	1.04	
10	357	23.2172	-8.304	0.0625	-0.176	4.86	
11	393	22.8194	-9.545	0.0313	-0.206	8.23	
12	430	22.6528	-10.04	0.0156	-0.339	14.1	
13	466	21.7144	-9.727	0.0313	-0.824	6.57	
14	504	21.2656	-9.651	0.00781	-3.33	101	Hessian modified
15	540	21.219	-9.35	0.0313	-2.9	8.52	
16	575	21.0149	-9.778	0.0625	-0.842	4.8	
17	608	19.9722	-10.25	0.25	-0.542	13.2	
18	642	19.9387	-9.145	0.125	-2.04	4.08	
19	678	19.2639	-10.16	0.0313	-0.443	3.78	
20	717	19.1996	-10.3	0.00391	-0.734	12.6	Hessian modified
21	748	18.8861	-10.59	1	-0.438	1.93	
22	779	18.1301	-10.59	1	-1.47	3	
23	811	17.781	-10.59	0.5	-1.4	1.86	
24	844	17.635	-10.57	0.25	-0.602	1.76	
25	875	17.3366	-10.15	1	-1.24	1.77	
26	906	16.9275	-10.11	1	-1.03	1.48	
27	937	16.032	-10.35	1	-1.73	1.72	
28	968	15.7919	-10.58	1	-0.851	2.71	
29	1002	15.7138	-10.58	0.125	-1.13	3.6	
30	1034	15.1124	-10.59	0.5	-1.34	1.78	
31	1071	15.086	-10.59	0.0156	-1.16	1.77	
32	1104	14.788	-10.56	0.25	-1.13	1.82	
33	1140	14.7615	-10.53	0.0313	-0.906	1.83	
34	1172	14.6161	-10.54	0.5	-1.39	1.8	

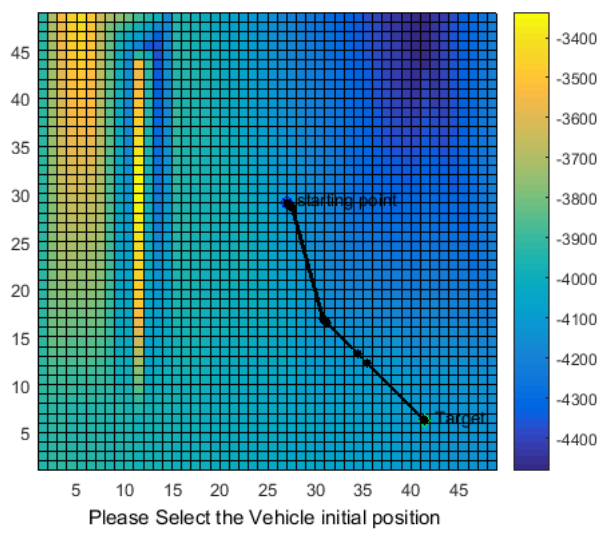
35	1203	14.4868	-10.58	1	-1.33	2.11	
36	1235	14.4341	-10.58	0.5	-1.2	3.21	
37	1266	14.208	-10.58	1	-1.48	4.23	
38	1297	13.6449	-10.58	1	-1.68	29.6	
39	1337	13.6202	-10.58	0.00195	-3.02	88.4	Hessian modified
40	1375	13.5258	-10.58	0.00781	-1.07	30.9	
41	1409	13.4242	-10.58	0.125	-1.62	6.01	
42	1446	13.4233	-10.58	0.0156	-1.37	6.54	
43	1480	13.206	-10.58	0.125	-0.715	7.87	
44	1517	13.1289	-10.58	0.0156	-1.03	12.1	Hessian modified
45	1553	13.1096	-10.58	0.0313	-0.807	11.6	
46	1587	12.9872	-10.58	0.125	-1.22	27.2	
47	1622	12.9547	-10.58	0.0625	-1.15	23	
48	1656	12.9486	-10.58	0.125	-1.36	12.5	
49	1689	12.9378	-10.58	0.25	-1.36	9.51	
50	1721	12.7742	-10.58	0.5	-1.07	14.5	
51	1761	12.4998	-10.58	0.00195	-1.07	49.4	
52	1804	12.4326	-10.58	0.000244	-6.02	336	Hessian modified
53	1836	12.3415	-10.58	0.5	-0.752	15.5	
54	1867	12.2696	-10.57	1	-0.211	19.4	
55	1901	12.2599	-10.57	0.125	-0.636	33.3	
56	1935	12.1214	-10.57	0.125	-1.72	16.7	
57	1969	12.1189	-10.57	0.125	-2.01	20.2	
58	2001	12.1094	-10.57	0.5	-0.681	17.9	
59	2033	12.0998	-10.57	0.5	-0.707	23.3	
60	2065	12.0879	-10.57	0.5	-0.665	25.2	
61	2096	12.0538	-10.57	1	-0.293	24.5	
62	2130	12.0258	-10.56	0.125	-0.288	28.3	
63	2164	11.9721	-10.56	0.125	-0.439	29.9	
64	2204	11.9584	-10.56	0.00195	-0.404	30.3	
65	2242	11.9553	-10.56	0.00781	-0.463	30.5	
66	2273	11.9095	-10.56	1	-0.383	38.3	
67	2307	11.9028	-10.55	0.125	-0.299	47.6	
68	2338	11.8733	-10.55	1	-0.435	64.8	
69	2370	11.8574	-10.54	0.5	-0.442	91.5	
70	2401	11.7896	-10.54	1	-0.677	72.6	
71	2433	11.7794	-10.54	0.5	-0.599	56.5	
72	2464	11.7411	-10.53	1	-0.273	82.8	
73	2496	11.7074	-10.52	0.5	-0.276	105	
74	2527	11.6531	-10.5	1	-0.23	260	
75	2558	11.6109	-10.51	1	-1.78	66.2	
76	2589	11.5855	-10.51	1	-1.02	61.5	
77	2620	11.4668	-10.49	1	-0.276	59.3	
78	2654	11.3719	-10.48	0.125	-0.226	102	
79	2692	11.3551	-10.48	0.00781	-0.214	239	
80	2726	11.3203	-10.47	0.125	-0.531	173	
81	2759	11.3057	-10.47	0.25	-0.519	76.7	
82	2791	11.1147	-10.45	0.5	-0.573	195	
83	2825	11.0108	-10.45	0.125	-0.285	171	
84	2861	10.9067	-10.44	0.0313	-0.26	162	
85	2900	10.857	-10.44	0.00391	-0.268	223	
86	2938	10.8272	-10.43	0.00781	-0.25	663	
87	2973	10.296	-10.39	0.0625	-0.277	111	
88	3008	10.0524	-10.35	0.0625	-0.247	58.7	
89	3039	9.86503	-10.37	1	-1.08	58.1	
90	3070	9.84012	-10.36	1	-0.421	234	
91	3103	9.82792	-10.36	0.25	-1.02	73.6	
92	3134	9.76381	-10.37	1	-0.253	92.6	
93	3165	9.71324	-10.37	1	-0.454	190	
94	3198	9.6305	-10.36	0.25	-0.36	212	
95	3239	9.62905	-10.36	0.000977	-0.367	209	
96	3270	9.54212	-10.36	1	-0.292	82.4	
97	3301	9.41475	-10.36	1	-0.138	76.1	
98	3333	9.38685	-10.36	0.5	-0.104	118	
99	3365	9.35032	-10.36	0.5	-0.411	120	
100	3397	9.33032	-10.36	0.5	-1.47	68.6	
101	3429	9.32629	-10.36	0.5	-0.829	65.5	
102	3460	9.32319	-10.36	1	-0.705	64.9	
103	3492	9.30484	-10.36	0.5	-0.546	73.6	
104	3525	9.29147	-10.36	0.25	-0.531	104	
105	3557	9.28658	-10.36	0.5	-0.291	79.6	
106	3589	9.25061	-10.36	0.5	-0.252	77.7	
107	3624	9.25012	-10.36	0.0625	-0.348	75.1	
108	3657	9.23755	-10.35	0.25	-0.758	78.1	
109	3688	9.20108	-10.35	1	-0.588	89.3	
110	3733	9.20108	-10.35	-6.1e-05	-0.63	89.3	
111	3765	9.18853	-10.35	0.5	-0.222	88	
112	3802	9.18316	-10.35	0.0156	-1.24	88	
113	3850	9.18259	-10.35	7.63e-06	-1.2	88	
114	3896	9.18227	-10.35	3.05e-05	-1.36	88	Hessian modified
115	3928	9.17447	-10.35	0.5	-1.27	85.2	
116	3959	9.16386	-10.35	1	-0.837	89.3	
117	3990	9.13367	-10.34	1	-0.337	88.9	
118	4024	9.128	-10.34	0.125	-0.261	87	
119	4057	9.12162	-10.34	0.25	-0.218	85.8	
120	4091	9.11541	-10.34	0.125	-0.183	86	
121	4127	9.11243	-10.34	0.0313	-0.182	137	
122	4162	9.10791	-10.34	0.0625	-0.179	458	
123	4198	9.10198	-10.34	0.0313	-0.277	571	
124	4236	9.09857	-10.34	0.00781	-0.339	967	
125	4276	9.09703	-10.34	0.00195	-0.336	1.49e+03	

```
  126   4312      9.09499       -10.34        0.0313       -0.295     1.18e+03
  127   4353      9.09339       -10.34      0.000977        -0.31     1.07e+03
  128   4394      9.09316       -10.34      0.000977        -1.31     1.05e+03  Hessian modified
  129   4445      9.09313       -10.34     -9.54e-07        -1.32     1.05e+03
  130   4493      9.09312       -10.34      7.63e-06         -1.3     1.05e+03
  131   4548      9.09312       -10.34     -5.96e-08        -1.32     1.05e+03
  132   4601      9.09312       -10.34     -2.38e-07        -1.31     1.05e+03

Local minimum possible. Constraints satisfied.

fmincon stopped because the predicted change in the objective function
is less than the selected value of the function tolerance and constraints
are satisfied to within the selected value of the constraint tolerance.



No active inequalities.
```

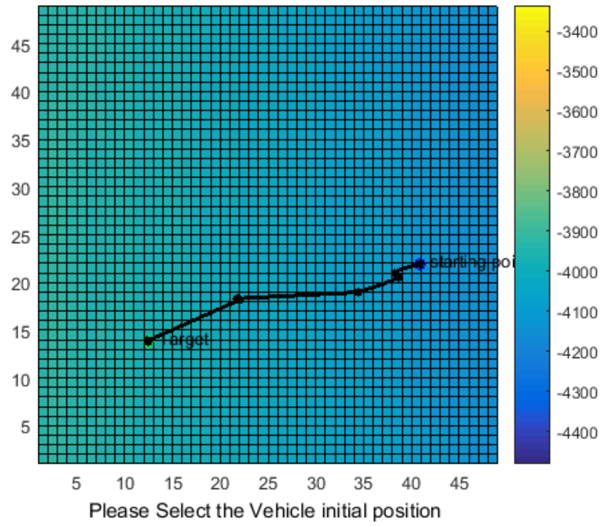
Plot the optimal solution:

```
delete([h_wp h_path]);
optimalWayPoints = [xStart yStart; reshape(optimalWayPoints,2,[])'; xTarget yTarget];

xWayPoints = optimalWayPoints(:,1);
yWayPoints = optimalWayPoints(:,2);
h_wp = plot(xWayPoints,yWayPoints,'color','k','linestyle','none','marker','.','markersize',16);

PathPoints = WayPoints_To_Path([xWayPoints,yWayPoints],'cubic',MAX_X,MAX_Y,fineness);
h_path = plot(PathPoints(:,1),PathPoints(:,2),'k','linewidth',2);
LineCost = solverCost(PathPoints, map, coor_map, azimuth,xTarget, yTarget, xStart, yStart,MAX_X, MAX_Y, 'linear', fineness);
fprintf('total Cost: %.1f\n', LineCost);
pause(2);
```

```
Warning: INTERP1(...,'CUBIC') will change in a future release. Use
INTERP1(...,'PCHIP') instead.
Warning: INTERP1(...,'CUBIC') will change in a future release. Use
INTERP1(...,'PCHIP') instead.
total Cost: 1.8
```




```
Warning: INTERP1(...,'CUBIC') will change in a future release. Use
INTERP1(...,'PCHIP') instead.
Warning: INTERP1(...,'CUBIC') will change in a future release. Use
INTERP1(...,'PCHIP') instead.
total Cost: 10.2
```


starting poi
Target
Please Select the Vehicle initial position

```
end
```

LOCAL PATH PLANNING

FMINCON

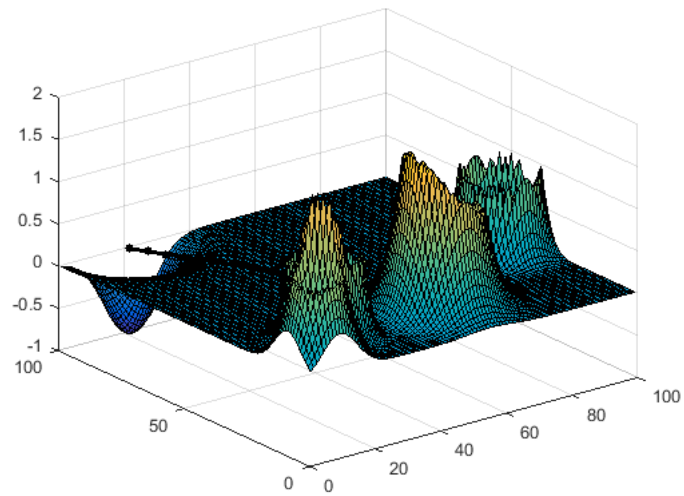
Implementation

Contents


```
function PathPoints = map2path( obstacle, xTarget, yTarget, xStart, yStart, azimuth, elevation, MAX_X, MAX_Y)
```

```
%UNTITLED3 Summary of this function goes here
%   based on potential simple exponential potential field in path planning.

%obstacle contain 1.coordinate_X 2. coordinate_Y 3.Height 4. radius(width)
%creating positive potential field near obstacle.

%These function parameters are for testing. Original parameters would be
%obtained from the camera by stereo imaging the obstacles and plotting
%them. This function performs path planning based on the obstacles defined
%by stereo imaging, a predefined target and starting point.
fieldMap = zeros(100,100);
xTarget = 21;
yTarget = 99;
xStart = 70;
yStart = 10;
azimuth = 45;
elevation = 45;
obstacle = [10,10,54,39; 21,21,37,88; 57,21,89,100; 91,43,45,100; 53,21, 84,32];

%GENERAATING FEILD MAP BASED ON THE DERIVED INFORMATION FROM CAMAERA.
sz = size(fieldMap);
fieldMap = zeros(sz);
targetAperture = min(sz(:,1),sz(:,2)); %define the thickness of exponential function of target.
%should be of the order of smaller dimension of map.
MAX_X = sz(:,1);
MAX_Y = sz(:,2);

%Adding obstacle feild.
for a=1:size(obstacle)
    for b=1:sz(:,1)
        for c=1:sz(:,2)
            y = b;
            x = c;
            fieldMap(b,c) = fieldMap(b,c) + exp(-abs((x-obstacle(a,1)).^2 + (y-obstacle(a,2)).^2 - obstacle(a,4))*1/obstacle(a,3)); %obstacle
        end
    end
end

%adding target field
for b = 1:sz(:,1)
    for c=1:sz(:,2)
        y = b;
        x = c;
        fieldMap(b,c) = fieldMap(b,c) - exp(-abs((x-xTarget).^2 + (y-yTarget).^2)*1/targetAperture);
    end
end

%visualizing fieldMap
x = 1:sz(:,1);
y = 1:sz(:,2);
grid on;
surf(x,y,fieldMap(x,y));
hold on;

%generating waypoints to target based on field map based fmincon solver.
numWayPoints = 20;
xWayPoints = linspace(xStart,xTarget,numWayPoints+2)';
yWayPoints = linspace(yStart,yTarget,numWayPoints+2)';
h_wp = plot(xWayPoints,yWayPoints,'color','k','linestyle','none','marker','.','markersize',16);

% Generate a continuous path from the waypoints
PathPoints = WayPoints_To_Path([xWayPoints,yWayPoints],'linear',MAX_X,MAX_Y,201);
h_path = plot(PathPoints(:,1),PathPoints(:,2),'k','linewidth',2);

% Calculate the cost taken
StraightLineCost = solverCostLocal(PathPoints,fieldMap,azimuth,elevation,xTarget,yTarget,xStart,yStart,MAX_X, MAX_Y,'linear');
fprintf('straight line Cost: %.1f\n', StraightLineCost);
```

```
straight line Cost: 3.9
```



Find an optimal path using FMINCON

Define Objective Function

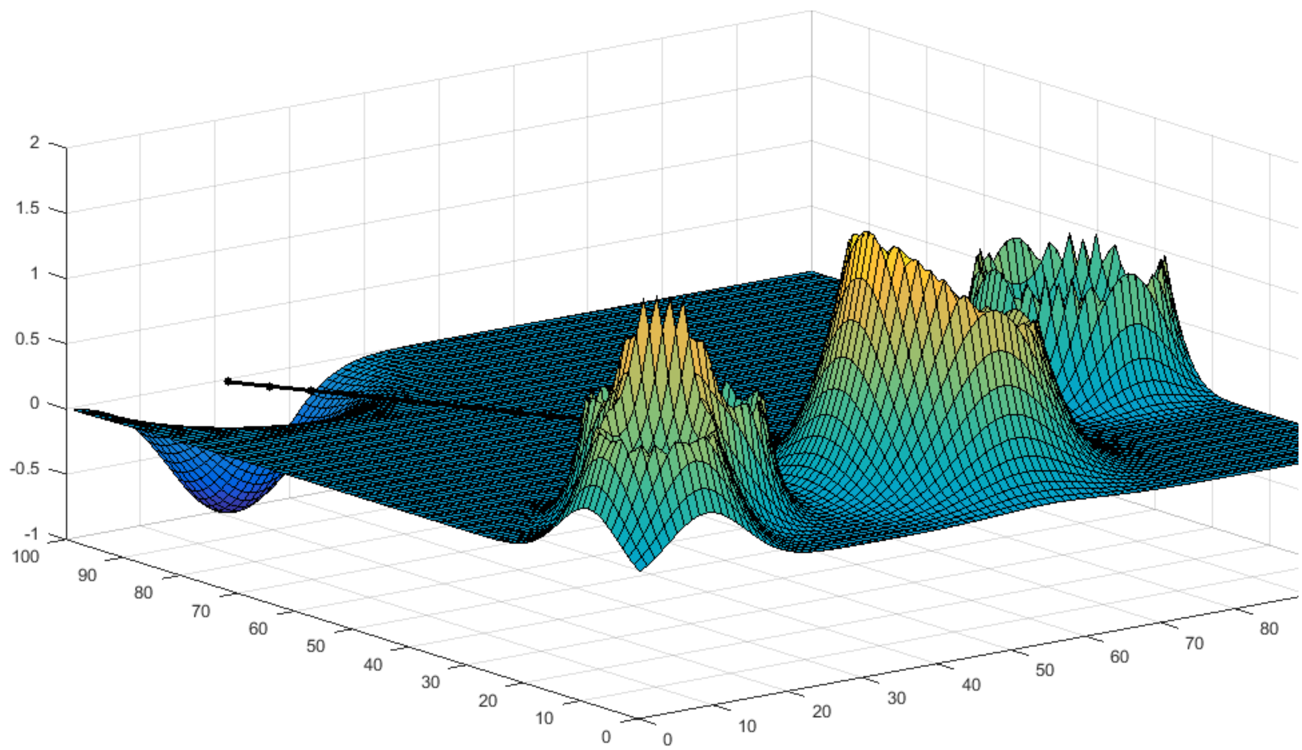
```
objectiveFun = @(P) solverCostLocal(P,fieldMap,azimuth,elevation,xTarget,yTarget,xStart,yStart,MAX_X, MAX_Y,'linear');

% Set optimization options
opts = optimset('fmincon');
opts.Display = 'iter';
opts.Algorithm = 'active-set';
opts.MaxFunEvals = 10000;

% Initial Conditions
xWayPoints = linspace(xStart,xTarget,numWayPoints+2)';
yWayPoints = linspace(yStart,yTarget,numWayPoints+2)';
ic = [xWayPoints(2:end-1)'; yWayPoints(2:end-1)'];
ic = ic(:);

% Bounds
lb = ones(size(ic(:)));
ub = reshape([MAX_X*ones(1,numWayPoints); MAX_Y*ones(1,numWayPoints)],[],1);

%Do the optimizaiton
optimalWayPoints = fmincon(objectiveFun, ic(:), [],[],[],[],lb,ub,[],opts);
```

```
                                Max     Line search  Directional  First-order
 Iter F-count        f(x)   constraint   steplength   derivative   optimality Procedure
    0      41     1.96279       -5.238
    1      82     1.96263       -5.235            1      -0.0125       0.0065
    2     123     1.87368            0            1      -0.0125       0.0668  Hessian modified
    3     164     1.85819       -2.832            1      -0.0148      0.00744
    4     205     1.85383       -2.239            1     -0.00407       0.0127
    5     246      1.8532       -2.338            1     -0.00192       0.0115
    6     287     1.85195       -2.462            1     -0.00314       0.0103  Hessian modified
    7     328     1.84863       -2.653            1     -0.00593      0.00876  Hessian modified
    8     369     1.84007       -2.918            1      -0.0111      0.00916
    9     410     1.82492       -3.328            1      -0.0132        0.013
   10     451     1.81256       -3.437            1     -0.00984       0.0183
   11     492      1.8102       -3.215            1     -0.00489       0.0123
   12     533     1.80935       -3.169            1     -0.00404       0.0115
   13     574     1.80566       -2.996            1      -0.0084       0.0126  Hessian modified twice
   14     615     1.79646       -2.511            1      -0.0101       0.0138
   15     656     1.79041        -2.32            1       -0.012       0.0144
   16     697     1.76888       -1.331            1      -0.0144       0.0149
   17     738     1.75632     -0.04893            1     -0.00595        0.029
   18     780     1.75176      -0.3972          0.5      -0.0281       0.0304
   19     822     1.75122      -0.1986          0.5      -0.0108       0.0252
   20     863     1.74354      -0.1755            1      -0.0136       0.0317
   21     908     1.73978      -0.1646       0.0625      -0.0152        0.252  Hessian modified
   22     949     1.73657      -0.1503            1      -0.0084       0.0287
   23     990     1.73613            0            1      -0.0189       0.0192
   24    1032     1.73255      -0.2208          0.5      -0.0119       0.0311
   25    1073     1.72732      -0.1617            1      -0.0324       0.0233
   26    1116      1.7267       -0.192         0.25      -0.0168       0.0277
   27    1158     1.72487      -0.2309          0.5      -0.0275       0.0658
   28    1200     1.72132       -0.255          0.5      -0.0186        0.117
   29    1251     1.71917      -0.2666     0.000977      -0.0231         1.98  Hessian modified
   30    1296     1.71878      -0.2777       0.0625      -0.0177        0.583
   31    1338     1.71572      -0.2927          0.5      -0.0308        0.188
   32    1380     1.71222      -0.3011          0.5      -0.0273        0.152
   33    1421     1.68465      -0.5615            1      -0.0326       0.0204
```

34	1462	1.67203	-0.4809	1	-0.0147	0.0208	
35	1503	1.66938	-0.4941	1	-0.0113	0.0206	
36	1548	1.66827	-0.5023	0.0625	-0.0134	0.0224	Hessian modified
37	1589	1.66454	-0.4898	1	-0.0157	0.0235	
38	1630	1.64527	-0.425	1	-0.0141	0.0426	
39	1676	1.64091	-0.4118	0.0313	-0.0131	0.105	
40	1723	1.63914	-0.4053	0.0156	-0.0113	0.149	
41	1771	1.63826	-0.4026	0.00781	-0.0121	0.276	
42	1815	1.63818	-0.4023	0.125	-0.0368	0.387	Hessian modified
43	1857	1.63714	-0.4033	0.5	-0.0251	0.793	
44	1902	1.6361	-0.4049	0.0625	-0.034	1.28	Hessian modified
45	1945	1.6352	-0.4076	0.25	-0.019	2.9	
46	1991	1.63341	-0.412	0.0313	-0.0193	6.31	
47	2036	1.63206	-0.4166	0.0625	-0.0233	10.6	Hessian modified
48	2079	1.62439	-0.4943	0.25	-0.0199	0.753	
49	2120	1.62318	-0.4824	1	-0.00569	0.847	
50	2161	1.62243	-0.4794	1	-0.00508	0.971	
51	2202	1.6208	-0.4845	1	-0.00729	1.5	Hessian modified
52	2244	1.61864	-0.5003	0.5	-0.0096	5.61	Hessian modified
53	2285	1.61639	-0.5331	1	-0.015	4.88	
54	2326	1.58409	-0.4015	1	-0.0145	0.747	
55	2368	1.57375	-0.2008	0.5	-0.0178	0.615	
56	2414	1.57347	-0.1945	0.0313	-0.0396	0.609	Hessian modified
57	2455	1.56909	0	1	-0.0128	0.571	
58	2499	1.5661	0	0.125	-0.0254	0.582	Hessian modified
59	2546	1.56542	-0.01817	0.0156	-0.0255	3.44	
60	2591	1.56434	-0.149	0.0625	-0.0113	11.6	
61	2636	1.56389	-0.274	0.0625	-0.0123	3.48	
62	2677	1.56212	-0.08174	1	-0.0107	3.66	
63	2718	1.55568	0	1	-0.0193	5.01	
64	2763	1.55075	0	0.0625	-0.0238	12.5	Hessian modified
65	2806	1.53616	0	0.25	-0.0249	26.3	
66	2852	1.53352	0	0.0313	-0.0276	28.3	
67	2897	1.52999	0	0.0625	-0.0376	27.7	
68	2941	1.52359	0	0.125	-0.0357	29.2	
69	2982	1.51301	0	1	-0.131	11.9	
70	3023	1.51208	0	1	-0.0683	6.92	
71	3069	1.51208	-0.07571	0.0313	-0.00107	7.24	
72	3113	1.50701	-0.2783	0.125	-0.0122	9.24	
73	3158	1.50102	-0.3601	0.0625	-0.0459	16.8	Hessian modified
74	3203	1.50043	-0.3376	0.0625	-0.0291	15.2	
75	3246	1.50032	-0.3428	0.25	-0.219	10.4	
76	3289	1.49838	-0.4473	0.25	-0.0179	10.3	
77	3331	1.49774	-0.433	0.5	-0.146	9.98	
78	3374	1.49761	-0.3984	0.25	-0.0197	9.71	
79	3415	1.49729	-0.4189	1	-0.0257	9.8	
80	3457	1.49727	-0.429	0.5	-0.037	9.93	Hessian modified
81	3499	1.49698	-0.4426	0.5	-0.0165	9.81	
82	3542	1.49643	-0.4529	0.25	-0.0294	9.58	Hessian modified
83	3586	1.49633	-0.4538	0.125	-0.064	9.45	Hessian modified
84	3627	1.49588	-0.4538	1	-0.0337	9.25	
85	3670	1.49588	-0.4537	0.25	-0.0197	9.18	Hessian modified
86	3711	1.49556	-0.454	1	-0.0114	8.92	
87	3752	1.49527	-0.4563	1	-0.0166	9.54	Hessian modified
88	3793	1.49462	-0.4584	1	-0.00667	9.62	
89	3838	1.48483	-0.4878	0.0625	-0.00668	10.8	Hessian modified
90	3879	1.47606	-0.4936	1	-0.00863	11.7	Hessian modified
91	3920	1.45119	-0.5048	1	-0.012	13.4	Hessian modified
92	3962	1.43656	-0.5702	0.5	-0.0174	38.6	Hessian modified
93	4003	1.4338	-0.559	1	-0.00934	25.7	
94	4044	1.41869	-0.5673	1	-0.0156	22.4	
95	4085	1.41122	-0.5755	1	-0.0251	13.4	
96	4126	1.39768	-0.5692	1	-0.0222	11.8	
97	4167	1.39291	-0.5692	1	-0.0387	9.74	
98	4208	1.39099	-0.5666	1	-0.0115	9.09	
99	4249	1.38662	-0.5649	1	-0.00848	8.21	
100	4295	1.36771	-0.5597	0.0313	-0.011	6.25	Hessian modified
101	4343	1.3664	-0.5583	0.00781	-0.00957	5.91	
102	4386	1.36167	-0.5579	0.25	-0.0347	5.82	
103	4439	1.36161	-0.5579	0.000244	-0.0753	5.82	Hessian modified
104	4480	1.35966	-0.5574	1	-0.0205	5.72	
105	4523	1.34625	-0.5505	0.25	-0.0195	4.57	Hessian modified
106	4564	1.33324	-0.542	1	-0.0225	3.52	
107	4605	1.33315	-0.5504	1	-0.0595	4.41	
108	4646	1.32638	-0.5487	1	-0.0187	4.19	
109	4690	1.32094	-0.5422	0.125	-0.0204	3.53	Hessian modified
110	4731	1.31711	-0.5468	1	-0.055	3.98	
111	4772	1.30847	-0.5453	1	-0.0291	3.83	
112	4820	1.30229	-0.5436	0.00781	-0.0365	3.66	Hessian modified
113	4861	1.29909	-0.544	1	-0.0343	3.7	
114	4902	1.29659	-0.5433	1	-0.0402	3.64	
115	4943	1.2887	-0.5419	1	-0.0213	3.51	
116	4986	1.27766	-0.5399	0.25	-0.0195	3.3	
117	5030	1.2702	-0.5395	0.125	-0.0224	3.25	
118	5079	1.2689	-0.5394	0.00391	-0.15	3.25	Hessian modified
119	5120	1.26872	-0.5401	1	-0.0527	3.25	
120	5161	1.2679	-0.5402	1	-0.0151	3.24	
121	5204	1.26359	-0.5404	0.25	-0.0162	3.23	Hessian modified twice
122	5245	1.26013	-0.5412	1	-0.00881	3.27	
123	5286	1.21114	-0.4028	1	-0.00876	3.55	Hessian modified
124	5328	1.20458	-0.5582	0.5	-0.00726	3.59	

125	5369	1.20362	-0.5595	1	-0.00315	3.58	
126	5410	1.20246	-0.5608	1	-0.00495	3.56	Hessian modified
127	5451	1.19765	-0.5443	1	-0.00858	3.47	Hessian modified
128	5492	1.18821	0	1	-0.0147	3.31	
129	5533	1.17173	0	1	-0.0189	3.16	
130	5578	1.14639	0	0.0625	-0.0181	2.94	Hessian modified
131	5622	1.14186	-0.2948	0.125	-0.00839	3.09	
132	5668	1.13751	-0.4272	0.0313	-0.0127	3.06	
133	5720	1.13703	-0.4428	0.000488	-0.0266	3.02	Hessian modified
134	5761	1.1358	-0.4662	1	-0.0302	2.9	
135	5802	1.1348	-0.5554	1	-0.0133	2.87	
136	5843	1.13396	-0.5393	1	-0.0663	2.86	
137	5884	1.13315	-0.5473	1	-0.0402	2.85	
138	5931	1.13283	-0.5511	0.0156	-0.048	2.83	Hessian modified
139	5972	1.13157	-0.5543	1	-0.0225	2.79	
140	6015	1.12744	-0.5705	0.25	-0.0212	2.64	Hessian modified
141	6057	1.12689	-0.5806	0.5	-0.0151	8.48	
142	6101	1.12573	-0.5812	0.125	-0.0257	5.54	
143	6142	1.12353	-0.5723	1	-0.183	2.51	
144	6183	1.12127	-0.5706	1	-0.0328	2.48	
145	6224	1.11549	-0.5626	1	-0.0249	2.16	Hessian modified
146	6265	1.11283	-0.5601	1	-0.0522	2.15	
147	6306	1.08497	-0.5332	1	-0.0181	1.86	
148	6348	1.08342	-0.5118	0.5	-0.0115	2.19	
149	6389	1.08002	-0.5245	1	-0.0234	1.72	
150	6430	1.07811	-0.5219	1	-0.0178	1.7	
151	6472	1.0776	-0.5092	0.5	-0.0202	1.62	Hessian modified
152	6513	1.06947	-0.5168	1	-0.021	1.65	
153	6554	1.06594	-0.519	1	-0.0242	1.67	
154	6596	1.0623	-0.5342	0.5	-0.0157	1.8	
155	6637	1.05906	-0.5268	1	-0.0264	1.72	
156	6678	1.05659	-0.5287	1	-0.0296	1.73	
157	6719	1.05194	-0.5363	1	-0.0135	1.77	
158	6762	1.04723	-0.5382	0.25	-0.0319	1.79	
159	6814	1.04492	-0.5387	0.000488	-0.0814	6.06	Hessian modified
160	6861	1.04491	-0.5387	0.0156	-0.0947	1.79	
161	6902	1.04426	-0.5416	1	-0.0486	1.78	
162	6943	1.04376	-0.5449	1	-0.0252	1.77	Hessian modified
163	6987	1.04019	-0.5718	0.125	-0.0246	1.72	Hessian modified
164	7032	1.0389	-0.584	0.0625	-0.0204	2.94	
165	7074	1.03861	-0.588	0.5	-0.0155	7.48	
166	7115	1.03729	-0.5881	1	-0.0318	2.91	
167	7156	1.03499	-0.5912	1	-0.0133	3	
168	7201	1.03485	-0.5942	0.0625	-0.0162	3	Hessian modified
169	7242	1.03388	-0.5918	1	-0.0686	2.47	
170	7283	1.03237	-0.5828	1	-0.163	1.7	Hessian modified
171	7324	1.03015	-0.5849	1	-0.0258	1.69	
172	7365	1.02834	-0.583	1	-0.0113	2.4	
173	7408	1.0277	-0.5818	0.25	-0.00926	13.3	Hessian modified
174	7449	1.02405	-0.5816	1	-0.0676	3.22	
175	7490	1.01834	-0.5791	1	-0.014	3.06	
176	7533	1.01599	-0.577	0.25	-0.012	2.67	
177	7574	1.01322	-0.5751	1	-0.0265	6.34	
178	7615	1.01188	-0.573	1	-0.113	1.65	
179	7656	1.01014	-0.5728	1	-0.0155	1.65	
180	7697	1.00377	-0.5746	1	-0.0181	1.63	Hessian modified
181	7738	1.00083	-0.5751	1	-0.0193	1.63	
182	7780	0.981399	-0.5816	0.5	-0.0236	1.74	Hessian modified
183	7827	0.978956	-0.5822	0.0156	-0.0186	1.75	
184	7874	0.97795	-0.5828	0.0156	-0.0125	1.76	
185	7918	0.976619	-0.5831	0.125	-0.019	1.76	
186	7967	0.976554	-0.5831	0.00391	-0.134	1.76	Hessian modified
187	8009	0.97613	-0.5833	0.5	-0.0364	1.76	
188	8058	0.975756	-0.5835	0.00391	-0.0404	27.3	Hessian modified
189	8099	0.974889	-0.584	1	-0.154	2.8	
190	8140	0.972889	-0.5853	1	-0.0504	1.72	
191	8184	0.97013	-0.5885	0.125	-0.0474	2.08	Hessian modified
192	8225	0.969966	-0.5869	1	-0.048	1.68	
193	8266	0.96942	-0.587	1	-0.0383	1.68	
194	8308	0.969317	-0.5877	0.5	-0.027	1.71	Hessian modified
195	8349	0.968733	-0.5872	1	-0.062	2.14	
196	8396	0.968416	-0.5862	0.0156	-0.0613	4.52	Hessian modified
197	8437	0.967842	-0.5868	1	-0.028	6.25	
198	8481	0.967243	-0.5879	0.125	-0.0286	9.95	Hessian modified
199	8528	0.967235	-0.5881	0.0156	-0.026	8.49	
200	8569	0.967041	-0.5886	1	-0.029	2.53	
201	8610	0.966396	-0.59	1	-0.0156	2.62	Hessian modified
202	8654	0.965958	-0.5925	0.125	-0.0141	2.62	Hessian modified
203	8695	0.965756	-0.5922	1	-0.0607	3.66	
204	8750	0.965754	-0.5922	-6.1e-05	-0.0656	3.65	
205	8805	0.965749	-0.5922	-6.1e-05	-0.0716	3.64	Hessian modified twice
206	8846	0.965483	-0.5931	1	-0.103	3.09	Hessian modified twice
207	8898	0.965459	-0.5935	0.000488	-0.388	2.87	Hessian modified
208	8939	0.965235	-0.5956	1	-0.0508	2.98	
209	8987	0.965203	-0.5958	0.00781	-0.0753	2.98	Hessian modified
210	9028	0.96497	-0.5964	1	-0.03	3.01	
211	9072	0.96486	-0.5971	0.125	-0.0226	8.75	Hessian modified
212	9114	0.96476	-0.597	0.5	-0.0704	10.4	
213	9158	0.964745	-0.5969	0.125	-0.064	5.72	
214	9199	0.964689	-0.5975	1	-0.0726	3.66	
215	9240	0.964588	-0.5983	1	-0.0321	5.64	Hessian modified

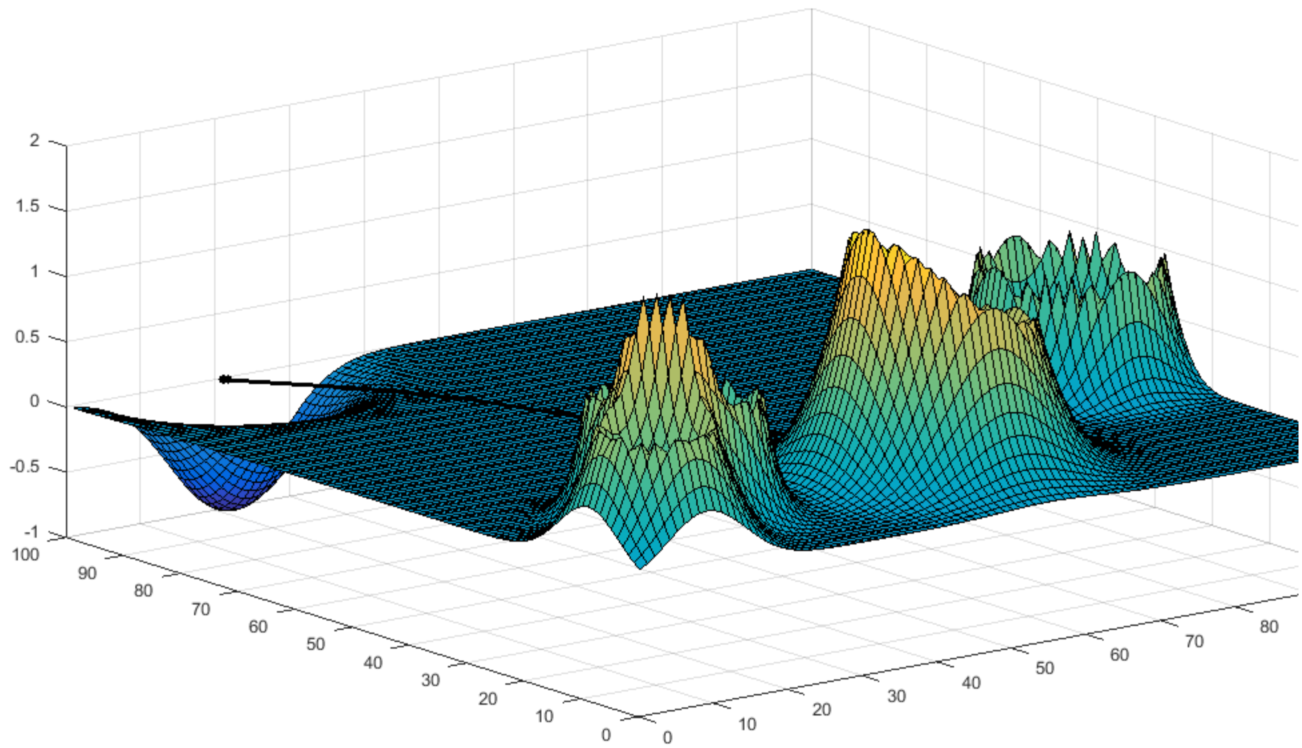
```
  216   9282     0.964521      -0.5991          0.5      -0.0189         23.6  Hessian modified

Local minimum possible. Constraints satisfied.

fmincon stopped because the predicted change in the objective function
is less than the selected value of the function tolerance and constraints
are satisfied to within the selected value of the constraint tolerance.


No active inequalities.
```



Plot the optimal solution:

```
delete([h_wp h_path]);
optimalWayPoints = [xStart yStart; reshape(optimalWayPoints,2,[])'; xTarget yTarget];

xWayPoints = optimalWayPoints(:,1);
yWayPoints = optimalWayPoints(:,2);
h_wp = plot(xWayPoints,yWayPoints,'color','k','linestyle','none','marker','.','markersize',16);

PathPoints = WayPoints_To_Path([xWayPoints,yWayPoints],'linear',MAX_X,MAX_Y,101);
h_path = plot(PathPoints(:,1),PathPoints(:,2),'k','linewidth',2);
LineCost = solverCostLocal(PathPoints,fieldMap,azimuth,elevation,xTarget,yTarget,xStart,yStart,MAX_X, MAX_Y,'linear');
fprintf('total Cost: %.1f\n', LineCost);
```

```
total Cost: 1.0
```




```
end
```

```
ans =

   70.0000   10.0000
   70.0016    9.9984
   70.0031    9.9969
   70.0047    9.9953
   70.0063    9.9937
   70.1704   10.2175
   70.8545   11.1623
   71.5386   12.1071
   72.2228   13.0519
   72.9069   13.9967
   72.9803   16.0732
   72.3818   19.3946
   71.7834   22.7160
   71.1850   26.0373
   70.5866   29.3587
   70.4157   30.3076
   70.4158   30.3075
   70.4159   30.3074
   70.4160   30.3073
   70.4161   30.3071
   67.1880   33.0110
   63.7985   35.8500
   60.4090   38.6891
   57.0195   41.5281
   54.2760   43.8260
   54.2780   43.8240
   54.2800   43.8220
   54.2820   43.8200
   54.2840   43.8181
   54.2891   43.8129
   54.2984   43.8035
   54.3077   43.7941
   54.3170   43.7847
   54.3263   43.7753
   54.3324   43.7691
   54.3370   43.7645
   54.3416   43.7599
   54.3461   43.7553
   54.3507   43.7507
   52.8136   44.3107
   51.1143   44.9301
   49.4149   45.5495
   47.7156   46.1689
```

```
   46.2219   46.8322
   45.9626   47.7590
   45.7034   48.6858
   45.4441   49.6126
   45.1848   50.5394
   44.9239   51.4707
   44.6604   52.4095
   44.3968   53.3482
   44.1333   54.2869
   43.8698   55.2257
   43.5110   56.2466
   43.0937   57.3181
   42.6764   58.3896
   42.2591   59.4611
   41.8417   60.5326
   41.4605   61.5546
   41.0853   62.5683
   40.7101   63.5821
   40.3349   64.5958
   39.9603   65.6106
   39.5909   66.6348
   39.2215   67.6589
   38.8522   68.6831
   38.4828   69.7073
   38.0795   70.8412
   37.6082   72.1945
   37.1369   73.5478
   36.6656   74.9011
   36.1943   76.2544
   35.9928   76.8339
   35.9937   76.8330
   35.9946   76.8321
   35.9955   76.8311
   35.9964   76.8302
   35.0103   78.5750
   33.7919   80.7306
   32.5735   82.8861
   31.3551   85.0417
   30.0990   87.2176
   28.0869   89.8019
   26.0747   92.3863
   24.0626   94.9706
   22.0505   97.5549
   20.6150   99.3992
   20.6209   99.3933
   20.6268   99.3874
   20.6327   99.3815
   20.6386   99.3756
   20.6429   99.3714
   20.6457   99.3685
   20.6486   99.3657
   20.6514   99.3629
   20.6543   99.3601
   20.7101   99.3019
   20.7826   99.2264
   20.8551   99.1509
   20.9275   99.0755
   21.0000   99.0000
```