

# Path Planning and Execution for Planetary Exploration Rovers based on 3D Mapping

Andres Mora, Keiji Nagatani and Kazuya Yoshida

*Space Robotics Laboratory, Department of Aerospace Engineering, Tohoku University  
Japan*

## 1. Introduction

The unmanned exploration of the Moon has steadily increased in the past years due to the renewed interest in creating a permanent human settlement on our natural satellite. After detailed remote sensing from orbiters, the focus is now shifting onto autonomous landing and surface locomotion by robotic devices on and around a specific area of highest interest. Especially, robotic exploration is a precursor for future human settlements on the Moon and in-situ resource utilization (ISRU) is one important aspect in the selection process of candidate locations for such settlements. In the early 1960's, Watson (Watson et al., 1961) estimated that it would be possible to find deposits of ice caps at the bottom of craters located at the Moon's polar regions. He argued that the shadows produced on these craters due to the very small deviation of the Moon's equatorial plane position with respect to the Sun create an environment that would present the proper conditions to retain such deposits.

Robotic exploration of these regions becomes a vital initial step on the way to build a permanent base of operations for humans to live on extended periods. The navigation of a vehicle on the Moon is presented with a series of issues such as the trafficability over the lunar soil (called "regolith"), the irregularity of the topography and the poor illumination at the Polar Regions, especially at the craters. In this chapter, attention is centered on the irregularity of the topography and the poor illumination issues. Given that we assume an environment with a very low angle of illumination which results in large shadowed areas, we propose the use of Light Detection and Ranging (LIDAR) systems to perceive the surroundings of the vehicle. These systems are not impaired by the lighting conditions assumed and have been successfully utilized in several outdoor mobile robotic applications (Langer et al., 2000),(Morales et al., 2008),(Skrzypcznski, 2008).

The features of the topography of the surface of the Moon may translate as obstacles for the navigation of the vehicle and also may limit the visual range of the exteroceptive sensors at a given location. These obstacles cause an "occluding" effect on the readings of the sensors therefore decreasing the size of the perceivable area.

Various researchers have approached the occlusion problem presented on indoor and outdoor autonomous navigation of mobile robotic systems. In (Dupuis et al., 2005),(Rekleitis et al., 2009), an irregular triangulated mesh is created based on a LIDAR's sensory data, and then "filtered" in order to find "shadows" or occluding obstacles and eliminate them from the map. A description of the problems that occluding obstacles may present in teleoperated navigation is presented in (Kunii & Kubota, 2006). In (Heckman et al., 2007), a method to classify different voxels based on its location with respect to an occluding obstacle is given, whereas

(Kunii & Kubota, 2006) presents a method to determine the position of a planetary rover by matching a set of 3D maps. An algorithm to calculate the shortest path between two given points within a completely known workspace implementing an art gallery method which selects guards and then connect them is presented in (Danner & Kavraki, 2000).

In this chapter, we present a path planning and navigation system based on various components introduced as follows. The perception of the environment surrounding a mobile robot through LIDAR technology. The recognition of occluded areas in a local map, the selection of a position within the boundaries of the environment known to the vehicle which maximizes the information gained once the vehicle reaches it, a generated path from the initial location of the vehicle to the selected goal and the control procedure implemented to navigate through the environment while following the generated local path.

## 2. Robotic Planetary Exploration

In the last decade the number of robotic space missions whose main goal is the exploration of our Moon and other planets has increased. These missions can be divided into three main phases: *Orbital exploration*, in which an orbiter conducts remote sensing around a target body in order to acquire “global” characteristics of the body such as terrain, geomagnetic, internal constituent data, etc. The second phase is called *Surface exploration with a robotic probe*, where a robotic probe is deployed on the targeted planet to conduct more investigations regarding the surface/internal characteristics of the surveyed planet. In this phase a mobile robotic probe provides a wider range of locations than that of a lander from where different scientific tests can be carried out. The final phase is known as *Sample return*. In this phase of the mission, the samples gathered by the robotic probe are delivered back to the Earth for further analysis. In this chapter, we focus on the second phase of this mission categorization. Due to the broader working field that a mobile robotic probe (or rover) provides to the numerous scientific instruments onboard, this particular type of probe has been implemented successfully in various planetary exploration missions. The first rover to ever land on and traverse the surface of another celestial body was the Soviet Union’s Lunokhod 1 followed by the Lunokhod 2 which reached our Moon and combined covered up to 47[km] of its surface. Their main tasks were to obtain geological data as well as detailed panoramic views of the surface of the Moon. The first space exploration rover to successfully reach another planet was NASA’s Mars Path Finder (Sojourner). The Sojourner was a six-wheeled vehicle weighting 10.6[kg] that traveled about 500[m] from its lander. During its operation, it sent 550 photographs of scenery and terrain close-ups and it analyzed the chemical properties of sixteen locations near the lander. The most recent successfully rover exploration missions have been those carried out by the Mars Exploration Rovers (MER) Spirit and Opportunity. These twin mobile robots landed on January 2004, and up to January 2009 they were still engaging their ongoing robotic missions of exploring the Martian surface and geology. Similarly to the Sojourner, the MERs are six-wheeled vehicles but much larger in size than their predecessor with a weight of 180[kg].

### 2.1 The Moon’s robotic exploration

Most of the scientific experiments performed by robotic missions (satellites, landers and rovers) on the Moon have dealt with mapping its surface, understanding its soil components and several other geological characteristics. We present here some of the characteristics of our natural satellite and the issues a mobile robotic probe would encounter there.

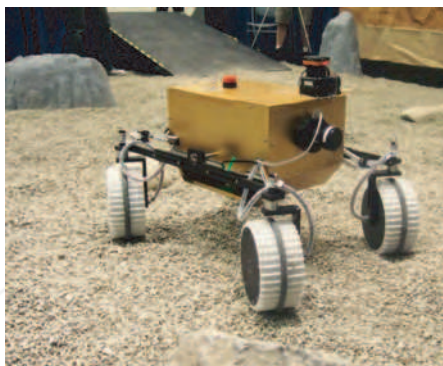


Figure 1. Mobile robotic testbed implemented: El Dorado-II

The distance between the Earth and our natural satellite is 384,403[km], being this the average center-to-center distance due to the change produced by the orbit of the Moon around the Earth. The diameter of the Moon is 3,474[km], its mass is approximately  $7.3477 \times 10^{22}$ [kg] and it has a gravity at the equator of  $1.622[\text{m/s}^2]$  which allows for a relatively easy escape velocity of  $2.38[\text{km/s}]$ . One of the features of the Moon the naked eye is able to look at is its craters.

The comets and interplanetary materials that continuously hit the lunar surface are believed to carry a certain amount of water to the lunar surface. Upon impact against the regolith (dusty, upper layer of the surface), the water would then split into its constituent elements which subsequently, would be liberated to space. One theory (Watson et al., 1961) states, however, that due to the small  $1.5$  [deg] deviation of the axial tilt of the Moon's spin axis to the ecliptic plane, craters located near or at the poles never receive sunlight thus creating a permanent "shadow" over large areas of these polar craters. Inside these shadows, extremely low temperature remain constant allowing the otherwise escaping water constituent elements to be trapped at the bottom of the craters.

### 3. Proposed Scenario

Water and mineral prospecting missions on the surface of the Moon and inside its polar craters are one the interest of major national space agencies around the world. In order to accomplish such task, an initial approach is to send an unmanned robotic mission to our natural satellite and upon arrival initiate its reconnaissance operation. This paper assumes a mission on which a mobile robotic vehicle traverses an area starting at a base close to the landing site and that extends to the locations of interest. The Japanese SELENOlogical and ENgineer Explorer (SELENE) was launched in September 2007 having as main mission objectives: (1) gather information about the lunar geological history and (2) obtain the topographic data required to generate a complete representation of its surface (Kato et al., 2007). The data obtained from the lunar surface can be treated in such manner that a Digital Elevation Map (DEM) can be generated. SELENE's payload allows it to obtain terrain data with a resolution of  $10[\text{m}]$  per pixel and a height detection of objects with a minimum height of  $5[\text{m}]$ . This map could work as a global map and it would correspond to the large areas surrounding the mobile robot.

The illumination conditions in the vicinity of the Moon's polar craters and at its inner walls is the main reason why the authors assume the usage of a Light Detection and Raging (LIDAR)

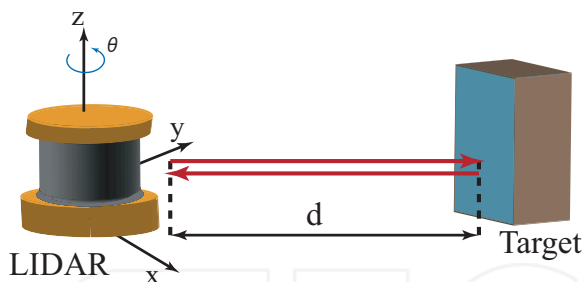


Fig. 2. Time of Flight Concept of laser range sensors. Disparity is avoided by the coaxial characteristic of the emitted and received laser beam.

sensory system on board of the vehicle to gather the information required to build a local map corresponding to the smaller, more detailed areas immediately surrounding the vehicle. The range of compact sensory systems typically goes from 5[m] to up 80[m] and with angular resolutions as small as of 0.25[deg].

In this research, we also assume a wheeled-rover as the mobile robot platform that will navigate from a lunar base to the polar lunar crater. The robotic system used in this research as a model for simulation purposes is a four wheeled-rover called “El Dorado II” and it is shown in Fig. 1. This rover is 0.75[m] long (measured between front and rear axles), 0.56[m] wide (measured from the outer side of its left wheels to the outer side of its right wheels). It is 0.50[m] high without any equipment on it and 0.70[m] when it has a laser range sensor mounted on it.

#### 4. Mapping based on 3D sensing

A laser range sensor is a device that can obtain range measurements through the use of one of two methods: the “Time-of-Flight” (TOF) principle and the determination of the phase-shift between an emitted continuous wave and its reflection (Adams, 1999) on a target’s surface.

##### 4.1 LIDAR principle

The laser range sensor used through out this research, a Hokuyo Top-URG UTM-30LX/LN (Hokuyo, 2008), has being designed under the Time-of-Flight principle. With this method, a laser beam is fired towards an opaque object in order to determine the distance separating the device and the targeted object. It then measures the time taken by the pulse to be reflected off the target and return to the sender. Through the use of elementary physics, the distance,  $d$ , covered in a round-trip is determined by multiplying the velocity,  $v$ , of the energy wave by the time,  $t$ , required to travel that distance,  $d = v \times t$ .

The measured time is representative of travelling twice the separation distance (round-trip), and must therefore be reduced by half to result in the actual range to the target (Everett, 1995). With this in mind, one can deduce that the velocity,  $v$ , is equal to the speed of light,  $c$ , which is considered to be approximately  $3 \times 10^8$ [m/s]. The time that it takes to reach the measured point is then  $d = \frac{c \times t}{2}$ .

In order to take advantage of this principle and to obtain measurements with resolutions in the order of millimeters, it is necessary for the sensor to have a timing circuitry capable of picosecond accuracy. Although nowadays these circuits have reduced their price, various

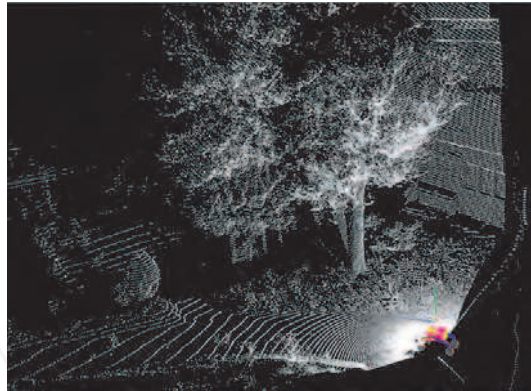


Fig. 3. Scene taken from an outdoors location using a laser range sensor.

other methods have been designed as a way around to overcome this financial issue. A review of the methods that can accomplish such level of accuracy is presented in (Kalisz, 2004).

Laser range sensors are used extensively in 3D object recognition, 3D object modelling, and a wide variety of computer vision related fields. The principle of this technology, which constitutes the heart of the so-called TOF 3D scanners, is shown in Fig. 2, where the inertial coordinate frame is inside the scanner. Laser range sensors offer high-precision scanning abilities, with either single-face or 360[deg] scanning modes.

Sensing systems based on pairs of cameras, or also known as stereo cameras, suffer of a problem caused due to its geometric configuration: disparity. The problem of disparity is produced by the distance separating the two cameras on the  $x$ -axis. This distance between the cameras generates some overlapping of the same object being registered by both cameras which can be surmounted by a number of calibration algorithms (Lucas & Kanade, 1981), (Barnea & Silverman, 1972), (Moravec, 1979), (Sanger, 1988) which eliminate the disparity from one camera's registered object to the other. These procedures are well known in the field of robotics and are of common use, nevertheless they are time and computation power consuming. Laser range sensors eliminate the disparity problem inherent in the stereo cameras systems by keeping the transmitted and received beams coaxial, as can be seen from Fig. 2.

The values a laser range sensor outputs are sets of locii of points given in Polar coordinates,  $(\theta, \phi, \rho)$ . Each scan performed by the sensor is a plane defined by each of those coordinates. That is, a plane with an initial value of  $\theta$ ,  $\theta_{init}$ , equal to 0[deg], is defined by the initial value of  $\phi$  (for example,  $\phi_{init} = 0$ [deg] and its corresponding value of  $\rho$ ) and it ends at the final value of  $\phi$  (for example  $\phi_{final} = 180$ [deg]) with its corresponding  $\rho$ . The sensing information of a given area will be completed when the final value of  $\theta$  (for example  $\theta_{fin} = 180$ [deg]) has been reached.

For computations using these locus of points at each plane, it is necessary to transform the polar coordinates of each point  $(\theta, \phi, \rho)$ , into its corresponding set of cartesian coordinates  $(x, y, z)$ . This transformation is done for every point sensed by the device in order to form what is known as a "cloud of points".

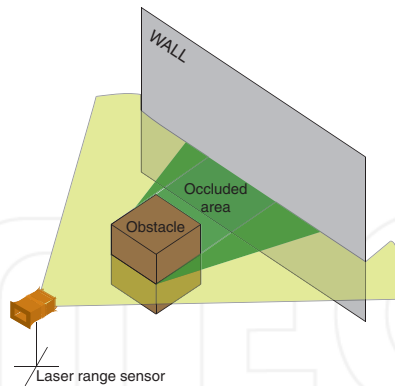


Fig. 4. Occlusion effect conceptual diagram.

After this coordinate transformation, the generated cloud of points can be represented through a graphical user interface (GUI) which can allow a human user to observe the information of the environment retrieved by the sensor. Depending on the range of the sensor, large, very detailed areas can be mapped with a single scanning procedure. An example of such cloud points representation can be seen in Fig. 3. A cloud of points is one of the multiple representations that can be produced from the raw sensor's data, other representations include rectangular meshes, quad-tree representations and triangular grids.

#### 4.2 Occlusion problem

Exteroceptive sensors such as LIDAR-based devices and stereo cameras are capable to retrieve the distance that separates them from an object within its field of view. However, these sensors are not able to "look" behind the objects they perceive. If for example, a laser range sensor is placed in front of a wall and an object moves from one end of the wall to the other, the sensor would register the object as it passes but it would also lose information about the area of the wall immediately behind the object. The effect that the passing object produces onto the laser range sensor while it registers the distance to the wall is known as *occlusion*.

Consider the static environment presented in Fig. 4. The laser range sensor is placed facing a wall and an object is located between the sensor and the wall. Assuming a single plane sweeping action, when the sensor's laser beam reaches the object a discontinuity from the wall reading spans until the occluding object has been completely registered. The occlusion caused by the object can be thought as the shadow the object projects onto the wall once a source of light is set upon the object. When a mobile robot constructs maps in a cluttered environment or in large irregular outdoor environments, the occlusion effect must be considered since the environment information lost in the occluded areas of the map may present hazards for the vehicle such as undetected holes, rocks, etc. A procedure proposed in this chapter to alleviate the occlusion effect in a given map is discussed in Subsection 5.2.



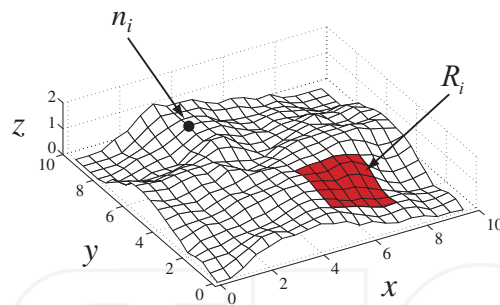


Fig. 5. A Digital Elevation Map as a discrete representation of a cloud of points.

### 4.3 Digital elevation map

The map representation implemented in this research is a commonly used method to display a cloud of points as a set of heights,  $z$ , correspondent to a coordinate pair  $(x, y)$ . This representation is known as a Digital Elevation Map (DEM) as previously introduced in Section 3. A digital elevation map can be considered as a representation of the topography of any given surface in digital format, that is, by coordinates and numerical descriptions of the value of every  $z$  correspondent to a pair  $(x, y)$ . A DEM can be defined by Equation 1;

$$\begin{cases} x = l_{grid} \cdot i & (\text{for } i = 0, 1, 2, \dots, X_{max} - 1) \\ y = l_{grid} \cdot j & (\text{for } j = 0, 1, 2, \dots, Y_{max} - 1) \\ z = (x_i, y_j) \end{cases} \quad (1)$$

In Equation 1,  $X_{max}$  and  $Y_{max}$  define the size of the grid,  $l_{grid}$  defines the length of the smallest possible grid, and the pair coordinate  $(x_i, y_j)$  defines the value of  $z$  for every iteration  $i$  and  $j$ . An example of such representation is given in Fig. 5.

Although ranging measuring systems based on the TOF principle have high-precision measurement characteristics, some potential errors can originate due to various factors, such as (Everett, 1995), (Borenstein et al., 1996):

- Uncertainties in determining the exact time of arrival of the reflected pulse
- Inaccuracies in the timing circuitry used to measure the round-trip time-of-flight
- Interaction of the incident wave with the target surface

these and other sources of errors in the ranging measurement of the laser range sensors have been the motivation to investigate additional characteristics of such sensors (Pascoal et al., 2008), (Ye & Borenstein, 2002) and also to find a model that can describe the likelihood that a given measurement obtained from these devices can be relied as true and not faulty measurements. This analysis is not within the scope of this chapter and shall not be considered. All measurements obtained from the implemented laser range sensor are assumed to be accurate.

## 5. Map Representations

In Section 3, the assumption of having two representations of the surface of the Moon with different levels of detail was introduced. One of them having a wider or *global* coverage of a given area with a lower resolution and the other one covering a smaller or *local* area but with higher resolution.

The wider representation will be referred hereinafter as *Global DEM* or *GDEM*. Based on the imagery information obtained from a lunar orbiter mission such as SELENE, a GDEM can be constructed. This GDEM can be generated with a  $10[\text{m}] \times 10[\text{m}]$  cell resolution and a  $500[\text{m}] \times 500[\text{m}]$  grid area and can be used as an input for a path planner that will generate a path to the specific position where the rover should go and perform the desired science tasks intended during the mission. This path planner is called “Global Path Planner” and will determine a candidate path to the specified goal, however, since the generated path is based on the information provided by the GDEM, the global path can only be used as a “guideline” since the resolution of the GDEM is not detailed enough for a rover with the dimensions presented earlier.

Once the rover leaves the lander, the vehicle will map the environment surrounding it and it will then traverse the targeted area. The map representing this area is called *Local DEM* or *LDEM*. Depending on the LIDAR system used, the resolution of the LDEM can be increased, in this chapter a cell resolution of  $10[\text{cm}] \times 10[\text{cm}]$  and a grid area of  $20[\text{m}] \times 20[\text{m}]$  is used. Throughout the use of this representation, a local path that considers various issues such as obstacle avoidance, visibility and distance to the global goal can be devised. This local path would be generated by what the authors called the “Local Path Planner”. This local path planner will refine the path outlined by the global path planner by getting a better understanding of the local environment given by the local map. The destination where the robot will intend to move to (within its known local map) will be the position from where the vehicle would avoid occluded areas as much as possible in the next sensing or scanning position.

Although the GDEM is an important element in this research, the forthcoming sections will focus on the LDEM, how it is obtained, its characteristics and the relationship it holds with the calculation of the next sensing position planning.

### 5.1 Local DEM

The LDEM’s main purpose is to determine the workspace of the vehicle by recognizing its boundaries, help the vehicle perceive the presence of obstacles within the map and by doing it so provide the information required to calculate the free space and obstacle’s space at a given map. Having determined a connected free space, the proposed planning strategy can be calculated within that specific region.

#### 5.1.1 Characterization of the Local DEM

Let  $n_i$  represent a terrain elevation node of the LDEM as presented in Fig. 5.  $i$  indicates the ubication of the node of the node  $n_i$  in the LDEM and takes values from 0 to  $N$ , where  $N$  is equal to the total number of nodes in the LDEM. Define the map space non-occluded  $\chi_{noc}$  as the set of nodes  $n_i$  of the LDEM that are non-occluded or visible with respect to the location of the laser range sensor and the map space occluded  $\chi_{occ}$  as the set of nodes  $n_i$  of the LDEM that are occluded or invisible with respect to the location of the laser range sensor. The union of the sets  $\chi_{noc}$  and  $\chi_{occ}$ ,  $\{\chi_{noc} \cup \chi_{occ}\}$ , contains all the nodes  $n_i$  within a single LDEM. This union is named the map working space  $\chi_{wspace}$  and can be defined as the semi-circle with a



radial boundary equivalent to the maximum range measurement of the LIDAR as depicted in Fig. 8.

$\chi_{noc}$  is formed by two subsets called the map space free  $\chi_{free}$  and the map space obstacle  $\chi_{obs}$ .  $\chi_{free}$  is defined as that subset of nodes  $n_i$  of the  $\chi_{wspace}$  that are not part of the subset of nodes contained in  $\chi_{occ}$  and that are not part of the subset of nodes contained in the map space obstacle  $\chi_{obs}$  as given by Equation 2;

$$\chi_{free} = \{n_i : n_i \in \chi_{wspace} | n_i \notin \chi_{obs} \text{ and } n_i \notin \chi_{occ}\}, 0 \leq i \leq N \quad (2)$$

$\chi_{obs}$  is defined as the set that contains all nodes  $n_i$  of the  $\chi_{wspace}$  as presented by Equation 3 and that present a problem to the trafficability of the vehicle.

$$\chi_{obs} = \{n_i : n_i \in \chi_{wspace} | n_i \notin \chi_{free} \text{ and } n_i \notin \chi_{occ}\}, 0 \leq i \leq N \quad (3)$$

Having defined the subsets  $\chi_{free}$  and  $\chi_{obs}$ , the map space occluded  $\chi_{occ}$  can be defined as Equation 4.

$$\chi_{occ} = \{n_i : n_i \in \chi_{wspace} | n_i \notin \chi_{free} \text{ and } n_i \notin \chi_{obs}\}, 0 \leq i \leq N \quad (4)$$

The relationship between the set  $\chi_{wspace}$  and its three subsets of nodes  $\chi_{free}$ ,  $\chi_{obs}$  and  $\chi_{occ}$  is given by Equation 5.

$$\chi_{wspace} = \chi_{free} \cup \chi_{obs} \cup \chi_{occ} \quad (5)$$

Finally, it can be noted that there is no intersection between any of the three subsets of the  $\chi_{wspace}$  as given by  $\chi_{free} \cap \chi_{obs} = \emptyset$ ,  $\chi_{free} \cap \chi_{occ} = \emptyset$  and  $\chi_{obs} \cap \chi_{occ} = \emptyset$ .

## 5.2 Visibility index

The visibility of an area of interest viewed from a node  $n_i$ , can be determined by the calculation of an index named by the authors the *visibility index*. This index returns the number of nodes that are within the field of view of the laser range sensor. The calculation of this index is given by Equation 6;

$$Vis = \frac{|\chi_{noc}|}{|\chi_{wspace}|} \quad (6)$$

where  $Vis$  indicates the visibility index within the analyzed given LDEM,  $|\chi_{noc}|$  is the cardinality or number of elements (nodes  $n_i$ ) inside of  $\chi_{noc}$  and  $|\chi_{wspace}|$  is the cardinality or number of elements (nodes  $n_i$ ) contained by the set  $\chi_{wspace}$ . In order to classify a given node as an element of the set given by  $\chi_{free}$  or as an element of the set  $\chi_{occ}$  and to quantify the number of nodes corresponding to each set, the author presents the following procedure Algorithm 1.

In line 2 a single virtual laser beam for each angle of the area to be scanned is traced in the form of a vector  $\vec{L}_i$ . Once this vector has been defined, in line 3 a unitary vector  $\vec{u}_i$  is calculated based on the laser beam's vector  $\vec{L}_i$ . This unitary vector will have a magnitude equivalent to that of the resolution of the LDEM  $d_{Res}$ . Later on, the initial values of the variables of the number of segments  $j$  within each  $\vec{L}_i$  and the maximum value of the slope  $S_{max}$  at any given iteration are set. The maximum number of segments per laser beam  $\vec{L}_i$ , is then calculated. The initial value of a vector that moves along the laser beam's vector  $\vec{L}_i$  in equal displacements of  $\vec{u}_i$  is set to be where the laser range sensor is located, in line 6.

The algorithm then calculates  $\vec{r}_{end_j}$ . This vector represents the location of a node  $n_i$  within the  $\chi_{wspace}$ . Following this calculation, the height of the studied node within the actual  $\vec{L}_i$  represented by  $\vec{r}_{end_j}$  is compared to the height of the node where the laser range sensor is

located  $\vec{p}_{org}$ . If the slope of the current position is higher than the maximum slope so far, the node is determined to be a non-occluded node or an element of the set  $\chi_{noc}$  and the maximum slope value is changed to the current position's one. Otherwise it is classified as an occluded node or an element of the set  $\chi_{occ}$ . The procedure is repeated until the final position of the virtual laser beam is reached and until all the perceivable area has been covered. Once the perceivable area has been covered, the visibility index is calculated and its value returned.

---

**Algorithm 1** Calculate the visibility index,  $Vis$ , in a given LDEM
 

---

**Input:** LDEM, nodes  $n_i \in \chi_{wspace}$

**Output:** Vis index

```

1: while  $i = 0 < i_{max}$  do
2:   Define  $\vec{L}_i: \vec{p}_{org} \rightarrow \vec{p}_{end_i}$ 
3:   Calculate  $\vec{u}_i = \frac{\vec{L}_i}{||\vec{L}_i||} \cdot d_{Res}$ 
4:   Set  $j = 0, S_{max} = 0.0$ 
5:   Calculate  $j_{max} = \frac{||\vec{L}_i||}{d_{Res}}$ 
6:    $\vec{r}_{org} \leftarrow \vec{p}_{org}$ 
7:   while  $j \leq j_{max}$  do
8:      $\vec{r}_{end_j} = \vec{r}_{org_j} + \vec{u}_i$ 
9:     Calculate  $S_j$  between the heights  $\vec{r}_{end_j,z}$  and  $\vec{p}_{org,z}$ 
10:    if  $S_j > S_{max}$  then
11:       $\vec{r}_{end_j} \in \chi_{noc}$ 
12:       $S_{max} \leftarrow S_j$ 
13:    else
14:       $\vec{r}_{end_j} \in \chi_{occ}$ 
15:    end if
16:     $j = j + 1$ 
17:  end while
18:   $i = i + 1$ 
19:   $\theta_i = \theta_{i-1} + \theta_{res}$ 
20: end while
21: Calculate  $Vis = \frac{|\chi_{noc}|}{|\chi_{wspace}|}$ 
22: return  $Vis$ 

```

---

A candidate path, can now be generated by the Local path planner within this map space free  $\chi_{free}$  at the set  $\chi_{wspace}$  (Ishigami et al., 2007). It only remains necessary to select an appropriate location that serves as a goal within the LDEM. The goal at every LDEM is in reality a subgoal of the final path which has as its final goal the destination initially selected for the global path planner.

In order to select each of these subgoals, the visibility index along with other parameters that will be explained later in this chapter are considered. Due to the occlusion effect of the natural obstacles present in the topography the mobile robot traverses, a considerable amount of the environment's characteristics is lost and the parameters utilized to select a goal at a given LDEM may not be used.

## 6. Fractal-based Terrain Generation

Euclidean geometry has been the traditional method to describe man-made objects, it is based on the characteristics of the size or scale of the object and can be calculated through algebraic formulas. A fractal in the other hand, is a fragmented geometrical shape too irregular to be analyzed through traditional Euclidean geometry, it can be split into smaller “self-similar” components independently of the scale, and can be generated by a recursive algorithm. The term “fractal” was first coined by the mathematician Benoit MandelBrot in 1975 (Mandelbrot, 1983) who elaborated one of the nowadays well known fractal sets, the Mandelbrot set.

One of the fundamental characteristics of fractals is the concept of *self-similarity* or *scaling* which is closely related to the Euclidean notion of dimension (Barnsley et al., 1998). This concept is based on the fact that a  $D$ -dimensional self-similar object can be divided into “ $N$ ” smaller copies of itself each of which is scaled down by a factor  $r$  as shown by Equation 7;

$$\begin{aligned} r &= \frac{1}{N^{1/D}} \\ N &= \frac{1}{r^D} \end{aligned} \quad (7)$$

and the fractal dimension of a  $N$ -parts self-similar object scaled by the same ratio as in Equation 7 is given by Equation 8

$$D_f = \frac{\log(N)}{\log(1/r)} \quad (8)$$

and it differs from the Euclidean dimension in the fact that it does not need to be an integer.

### 6.1 Natural terrain generation

An effective mathematical model to simulate natural terrain with topographical accidents such as those of the lunar soil or Mars is the *fractional Brownian motion* or *fBM*. This model is an extension of the random walk or Brownian motion which was initially proposed by R. Brown and further studied by A. Einstein (Mazo, 2002) after observing under the microscope the random movement of particules suspended in a fluid. Brownian motion can be informally defined as the displacement of a particle in one time interval that is independent of the displacement of the particle during another time interval (Sharma & Vishwamittar, 2005).

#### 6.1.1 Terrain generation based on $H$ and $\sigma$

A number of methods can be implemented in order to generate a fractional Brownian motion-based natural terrain. In this research the authors have used the spectral synthesis approach also known as the Fourier filtering method to produce such terrain (Barnsley et al., 1998). A fBm-based curve can be obtain through an appropriate selection of the spectral density  $S_X(f)$  of a random process  $X(t)$ . If a random process  $X(t)$  has a spectral density proportional to  $\frac{1}{f^\beta}$  as shown in Equation 9,

$$S_X(f) \propto \frac{1}{f^\beta} \quad (9)$$

where  $\beta$  is known as the spectral exponent,  $S_X(f)$  will represent a fBM curve with a  $\beta = 2H + 1$ . The scaling behavior exhibited by such curve is characterized by a parameter called the “Hurst Parameter”,  $H$ , after E. H. Hurst who introduced it when studying the statistics of the annual water level fluctuations of the Nile river. The Hurst parameter can take values between  $0 \leq H \leq 1$  and it determines the “roughness” of the curve. When  $H$  tends to 0 the

function's curve becomes rougher, conversely, when  $H$  gets closer to 1 the curve smoothens. Having  $1 < \beta < 3$  will produce a fBm curve with a fractal dimension as given by Equation 10 for a Euclidean dimension  $E = 1$ :

$$D_f = 2 - H = \frac{5 - \beta}{2} \quad (10)$$

In a general form that can be extended to higher dimensions, the spectral density  $S_X(f)$  for a random process  $X(t)$ , can be expressed as Equation 11,

$$S(u_i) \propto \frac{1}{\left(\sum_{i=1}^E u_i^2\right)^{\frac{2H+E}{2}}} \quad (11)$$

where  $u_i$  represents every coordinate axis ( $x, y, z$ ),  $H$  is the Hurst parameter taking values as  $0 < H < 1$  as previously presented and the relationship between  $E$  and  $H$  in the Euclidean space is given by  $D_f = E + 1 - H = E + \frac{3-\beta}{2}$ . Based on this general form, Equation 11 can be written to represent a uni-dimensional space fBM curve equivalent to Equation 9 as described by Equation 12,

$$S(u) \propto \frac{1}{u^{2H+1}} \quad (12)$$

where a value  $H = 1/2$  is called Brownian Noise,  $H = 0$  describes the  $1/f$  noise and  $H = -1/2$  is known as white noise. The general equation for a two-dimensional spectral density is then provided by Equation 13,

$$S(u, v) \propto \frac{1}{(u^2 + v^2)^{H+1}} \quad (13)$$

To build a natural terrain bounded in a DEM the following procedure is performed. Assume that the heights contained in a plane are known and constrained by a two-dimensional grid represented mathematically by a matrix with the corresponding coordinates  $x$  and  $y$  of every height  $Z(x, y)$  as its indexes. The spectral density of this collection of heights can be described by  $S_Z(u, v)$  as Equation 14 shows

$$S_Z(u, v) = \frac{\sigma^2}{(u^2 + v^2)^{H+1}} \quad (14)$$

In Equation 14,  $\sigma$  is a proportional constant representing the amplitude coefficient of a given terrain. In other words,  $\sigma$  represents the behavior of the heights  $Z(x, y)$  within the natural terrain to be generated. Based on the characteristic line of the Fourier transform,  $\sigma$  can be obtained from the values of the height  $Z(x, y)$ . If we call  $F_Z(u, v)$  the Fourier transform of  $Z(x, y)$ , the power spectral density  $S_Z(u, v)$  becomes Equation 15,

$$S_Z(u, v) = \frac{1}{N\Delta} |F_Z(u, v)|^2 \quad (15)$$

where  $N \times N$  is the number of nodes into which the terrain was discretized,  $\Delta$  is the grid's spacing between nodes which is in accordance to  $S_X(f) = \frac{|X(f)|}{\Delta f}$  which is an equivalent definition of the spectral density given by Equation 9. From Equation 14 and Equation 15, the  $F_Z(u, v)$  of  $Z(x, y)$  can be found as presented by Equation 16,

$$|F_Z(u, v)| = \frac{\sigma \sqrt{(N\Delta)}}{(u^2 + v^2)^{\frac{H+1}{2}}} \quad (16)$$

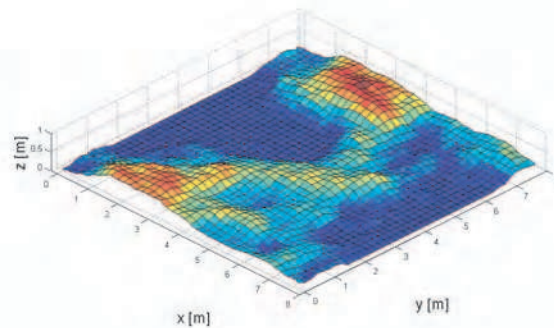


Figure 6. A natural terrain generated with the implemented fBM model.

The parameters required by the implemented model to generate a simulated natural terrain are  $\sigma$ ,  $H$ ,  $N$  and  $\Delta$ , where  $N$  and  $\Delta$  are kept constant in order to maintain the same constrained area on every terrain generation. Therefore, the parameters that will determine the roughness and amplitude (height) of the simulated natural terrain will be determined by  $\sigma$  and  $H$ . Using this characterizing parameters, it's possible to compute  $|F_Z(u, v)|$  as shown in Equation 16. Now that the Fourier transform is known based on the characterizing parameters, it is possible to calculate  $Z(x, y)$  by finding its inverse Discrete Fourier transform. It should be noticed that  $Z(x, y)$  will have white noise added due to this computational process. In order to avoid the issues this added noise may introduce in the calculation, random values can be utilized. The inverse Fourier transform of  $F_Z(u, v)$  can be determined as shown in Equation 17,

$$Z(\hat{x}, \hat{y}) = \frac{1}{(N\Delta)^2} \sum_{\hat{u}=0}^{N-1} \sum_{\hat{v}=0}^{N-1} F_Z(\hat{u}, \hat{v}) e^{2i\pi(\hat{u}\hat{x} + \hat{v}\hat{y})/N} \quad (17)$$

where the variables that have a "hat" such as  $\hat{x}$  are in the interval spanning from 0 to  $N - 1$  and the sampling cycle  $\Delta$  can be considered as Equation 18:

$$u = \left( \frac{\hat{u}}{N\Delta} \right), x = \hat{x}\Delta \quad (18)$$

Then since  $Z(x, y)$  has real numbers, the conjugate of  $F_Z(\hat{u}, \hat{v})$  can be found as described by Equation 19,

$$F_Z(\hat{u}, \hat{v}) = F_Z^*(N - \hat{u}, N - \hat{v}) \quad (19)$$

Now if one combines Equation 16 and Equation 18 in order to avoid the indefinision caused by  $u = v = 0$ , we can obtain the result given by Equation 20:

$$|F_Z(u, v)| = \frac{\sigma(N\Delta)^{H+\frac{3}{2}}}{(\hat{u}^2 + \hat{v}^2)^{\frac{H+1}{2}}} \quad (20)$$

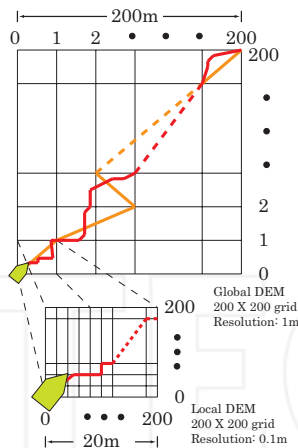


Fig. 7. Description of the path planning strategy based on the Global and Local DEMs

which can be introduced in Equation 17 to obtain the desired natural terrain model. Fig. 6 shows an example of a natural terrain that can be obtained through the implementation of this procedure.

## 7. Path Planning and Execution

Several techniques have been designed and utilized in the robotics field to generate paths for a robot to move from its initial position to the desired destination. In this section, the authors describe the two different techniques that have been implemented: a “global path planner” and a “local path planner”. The first method determines a path based on the information gathered by an orbiter and represented in the GDEM. The local path planner, in the other hand, is based on the topographic information of the local environment gathered by the laser range sensor. As shown by Fig. 7, the global path planner will serve as an “guideline” the mobile robot would follow. As the vehicle navigates the environment, its knowledge of the terrain increases and the originally determined global path is reconfigured.

### 7.1 Global path planner

In this section, a procedure that calculates a candidate path based on the information provided by the GDEM is explained. The path planner’s objective function is to find a candidate path that can be composed of three criteria indices: terrain roughness, path length and inclination. This path planner takes into account the features of the terrain to be traversed, however, it does not include an index that weights the visibility from the actual position of the mobile robot Ishigami et al. (2006).

The path planner used in this research assumes a perfect, namely without uncertainties, knowledge of the terrain map and the DEM obtained from it. The terrain map is represented by a DEM, which is defined as a series of elevations at a grid’s node  $n_i$  in  $(x_i, y_i, z_i)$ , where  $i$  is the index of the node. This DEM-based terrain map can be measured by an orbiter flying around the Moon or celestial body of interest.



The terrain roughness index aims to define the traversability over an uneven terrain. The planner avoids defining the path over a too rough zone that the vehicle would not be able to overcome. The terrain roughness index  $B_i$  is given as a standard deviation of the terrain elevation over a projection region (described in Fig. 5) of the robot  $R_i$  Iagnemma & Dubowsky (2004) as shown by Equation 21 :

$$B_i = \sqrt{\frac{1}{n} \sum_{R_i} (z(R_i) - \bar{z}(R_i))^2} \quad (21)$$

where,  $n$  represents the number of the node inside the region  $R_i$  and  $\bar{z}(R_i)$  denotes an average elevation inside the same region. The rougher the terrain is, the larger  $B_i$  becomes. The projection region  $R_i$  includes the set of terrain elevation points inside the region surrounded by the wheels of the mobile robot.

This index aims to define the shortest path from the initial position of the rover to a desired destination. The path length index  $L_i$  between adjacent nodes is calculated by Equation 22:

$$L_i = |n_i - n_j| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (22)$$

If the nodes  $n_i$  are not adjacent to the nodes  $n_j$ ,  $L_i$  gets a large value.

When the robot is climbing up a hill or traversing the slope of a crater, the risk of slippage or tipping over increases. The index of the terrain inclination aims to mitigate such risks. The terrain inclination angles are divided into two axis related to the robot body coordinates. An inclination angle around  $x$ -axis of the robot coordinates is denoted by  $\theta_x$ , while one in the  $y$ -axis is  $\theta_y$ . The indices  $\Theta_{xi}$  and  $\Theta_{yi}$ , associated with each terrain inclination are respectively determined by the average inclination at the region  $R_i$  as presented by Equation 23:

$$\Theta_{xi} = \bar{\theta}_x(R_i) \quad (23)$$

$$\Theta_{yi} = \bar{\theta}_y(R_i) \quad (24)$$

The above criteria indices are weighted in order to define the  $C(p)$  economical function to minimize, that generates a candidate path  $p$ .

$$C(p) = \sum_{i=p} (W_B N_B B_i + W_L N_L L_i + W_{\theta_x} N_{\theta_x} \Theta_{xi} + W_{\theta_y} N_{\theta_y} \Theta_{yi}) \quad (25)$$

where,  $W_B$ ,  $W_L$ ,  $W_{\theta_x}$  and  $W_{\theta_y}$  are the weighting factors to give specific priorities between the terrain roughness, path length, and terrain inclinations. Note that  $W_{\theta_x}$  or  $W_{\theta_y}$  respectively take large enough values when the index  $\Theta_{xi}$  or  $\Theta_{yi}$  exceed threshold angles  $\theta_{x_{max}}$  and  $\theta_{y_{max}}$ .  $N_B$ ,  $N_L$ ,  $N_{\theta_x}$  and  $N_{\theta_y}$  are constants to normalize each corresponding indices and eliminate the dimensions. The path  $p$  consists of a series of neighboring nodes,  $p = \{n_{start}, \dots, n_i, \dots, n_{goal}\}$ . A small index means that the robot is less affected by the criteria. For example the smaller  $W_B$  is, the less the planner will try to avoid rocks and boulders.

Therefore the smallest the sum of the weighted indices is, the optimal as less hazardous the path is, and supposing that the objective function is a hypothetical distance function, the path planning problem is considered as a shortest path search. And considering that the minimum objective function derives the "shortest" path  $p_s$ , the following equation can be defined by Equation 26:

$$\min C(p) = C(p_s) \quad (26)$$

where  $p_s$  is derived by the commonly used Dijkstra algorithm LaValle (2006).

## 7.2 Local path planner

Once a global destination has been determined and the global path calculated, the procedure to generate a local path follows. After having introduced the methods to characterize the terrain and determine the occluded areas in Sections 4- 5, it is possible to calculate the destination at every LDEM to where the vehicle should go to. This position is called the “next sensing position”.

### 7.2.1 Next sensing position

The next sensing position in this paper is defined as the position within the map space free  $\chi_{free}$ , at a given LDEM, from where it is estimated that the LIDAR sensory system will cover a larger non-occluded area on the next scanning procedure.

Consider Fig. 8 as a conceptual example. Assume that the robot knows the coordinates of the global goal and that this goal is behind an obstacle Fig. 8(a). Once the information from the LIDAR sensory system has been acquired and the LDEM has been created, the visibility from the initial position of the vehicle can be determined by the algorithm presented in Subsection 5.2. This algorithm will classify the nodes occupied by the viewable edges of the obstacle as non-occluded and the area behind it in the other hand, will be determined to be occluded. We can then complete the characterization of the LDEM and determine its  $\chi_{free}$ . In Fig. 8(b) a node representing a new location is evaluated and its  $\chi_{free}$  compared to other visited locations such as the one described in Fig. 8(c). The estimated location for the next sensing position will sense the same viewable edges of the obstacle detected previously, but due to its new orientation towards the global goal and its position in the  $\chi_{free}$ , it will obtain the largest non-occluded area from the sampled positions as presented in Fig. 8(d). This estimation procedure is based on the evaluation of various parameters of a node contained in  $\chi_{free}$ . These parameters are: the visibility index  $Vis$  presented in Subsection 5.2, the Euclidean distance  $L_{n_i}$  separating the inspected node and the global goal as given by Equation 27,

$$L_{n_i} = \sqrt{(x_g - x_{n_i})^2 + (y_g - y_{n_i})^2 + (z_g - z_{n_i})^2} \quad (27)$$

where  $(x_g, y_g, z_g)$  denote the coordinates of the global destination and  $(x_{n_i}, y_{n_i}, z_{n_i})$  represent the coordinates of the currently inspected node. The third parameter is the value  $obs$  a node is given to if it is a node element of  $\chi_{obs}$ . The evaluation function combining these parameters is presented in Equation 28;

$$E_{n_i} = W_{occ} \cdot Vis_{n_i} + W_L \cdot L_{n_i} + obs_{n_i} \quad (28)$$

where  $E_{n_i}$  is the value of the evaluation function at node  $n_i$  and  $W_{occ}$  and  $W_L$  represent the weighting factors that give specific priorities between the size of the occluded area in the next sensing position and distance from the inspected node  $n_i$  to the global goal. The parameter  $obs_{n_i}$  works as a penalty index, since it takes large values if the inspected node  $n_i$  is contained in  $\chi_{obs}$ . It should be noted that if a higher value is set to a given weighting factor, a higher priority will be given to that parameter. For example, the higher the weighting factor  $W_{occ}$  is, the more importance to the amount of non-occluded area in the next sensing procedure will be. In the same way, the higher the weighting factor  $W_L$  is, the more importance to a position closer to the global goal will be given. In our implementation, the node  $n_i$  with the evaluation function having the highest value from the sample taken will be selected as the next sensing position. This next sensing position is consider as optimal with respect to the priorities given to the weighting factors.

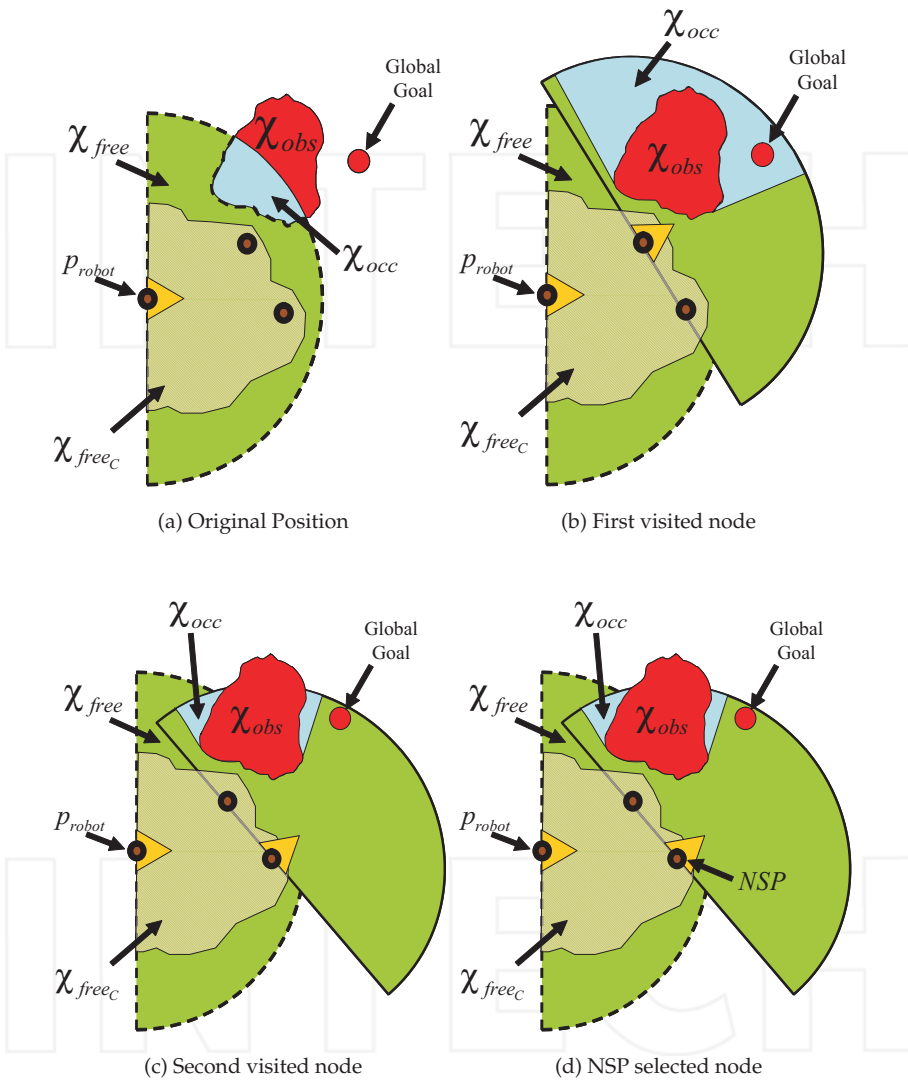


Fig. 8. Conceptual representation of an estimated next sensing position (NSP) for the mobile robot.

**Algorithm 2** Next Sensing Position in a given LDEM**Input:** Nodes contained in a LDEM's  $\chi_{free}$ **Input:** Number of seed nodes for the Hill Climbing algorithm,  $numSeeds$ **Output:** Next sensing position within a LDEM,  $nextSenPos$ 

```

1: for  $i = 0$  to  $numSeeds$  do
2:   Select a random node  $rndmNode$  from  $\chi_{free}$ 
3:   Set  $currentNode \leftarrow rndmNode$ 
4:   while TRUE do
5:     Calculate  $Nghbrs = neighbors(currentNode)$ 
6:     Set  $nxtEval = -\infty$  and  $nxtNode = NULL$ 
7:     for  $\forall n_i \in Nghbrs$  do
8:       Calculate distance to goal  $L_{n_i}$  from  $n_i$ 
9:       Determine the visibility index  $Vis_{n_i}$ 
10:      if  $n_i \in \chi_{obs}$  then
11:        Set  $obs_{n_i} = \text{arbitrary low value}$ 
12:      else
13:        Set  $obs_{n_i} = \text{arbitrary high value}$ 
14:      end if
15:      Calculate  $E_{n_i}$ 
16:      if  $E_{n_i} > nxtEval$  then
17:         $nxtNode \leftarrow n_i$ 
18:         $nxtEval \leftarrow E_{n_i}$ 
19:         $tmpEval \leftarrow nxtEval$ 
20:      end if
21:    end for
22:    if  $nxtEval \leq E_{n_i}$  then
23:      return  $nxtNode$ 
24:    end if
25:     $currentNode \leftarrow nxtNode$ 
26:  end while
27:  if  $tmpEval > tmpEvalMax$  then
28:     $tmpEvalMax \leftarrow tmpEval$ 
29:     $nextSenPos \leftarrow nxtNode$ 
30:  end if
31: end for
32: return  $nextSenPos$ 

```

Since the number of nodes in the  $\chi_{free}$  can be large, the computation of Equation 28 for every node inside the  $\chi_{free}$  is time-consuming. Therefore, the authors utilized a hill-climbing search within the  $\chi_{free}$ . To avoid the local minima issue that can arise by the use of this algorithm, several random nodes were used as initial nodes. The Algorithm 2 describes in pseudocode the procedure to obtain the next sensing position.

A set of results obtained from a numerical simulation are depicted in Fig. 9. The global goal is represented by a red sphere and the local goal determined by the next sensing position is shown as a purple sphere in Fig. 9. Various local goal candidates that were considered by the algorithm are depicted as blue spheres. Fig. 9 shows the results obtained by the Algo-

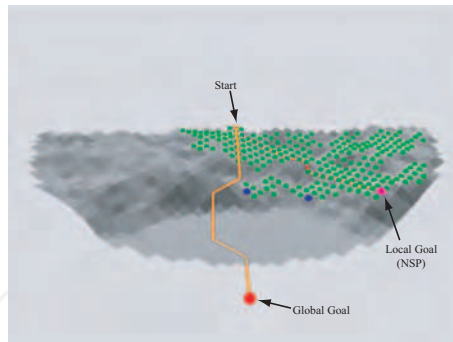


Figure 9. Visual description of the next sensing position algorithm

rithm 2 after 5 iterations. It is possible however, that the same node may be selected multiple times among the 5 iterations as is the case presented in Fig. 9. After the pre-set number of iterations is reached the node with the highest evaluation function among these nodes is determined to be the next sensing position.

Once the next sensing position has been obtained, the destination where the mobile robot would go to is passed on to the implemented local path planner. The local path planner utilized in this research is called the Distance Transform and it was introduced by (Jarvis, 1994). This algorithm finds the optimal collision-free path between the initial and final destination of the vehicle based on a given known stationary environment map. Its output is a two-dimensional array of exact cell decomposition. In our implementation, each cell is equivalent to a node of the LDEM and each cell has a value that represents a collision-free distance cost to reach the goal.

The local goal, or the subglobal goal, is initially given an arbitrary low distance cost, i.e. zero, while other nodes are given much higher values. Notice that the inverse procedure works equally well. The algorithm then visits every node and assigns to each of them a different distance cost based on the propagating distance costs of the surrounding nodes. The optimal path is found by using the steepest descent or ascent track from the actual location of the vehicle in the  $\chi_{free}$ . The generated path has no risk of falling into the local minima problem since every node has been assigned a different distance cost at the propagating phase of the algorithm.

The distance transform is a robust algorithm that can be efficiently implemented either with an a priori knowledge of the  $\chi_{obs}$  or without it. In our proposed scenario, we have a completely known global environment with a low resolution and a partially known local environment with a high resolution.

## 8. Path following strategy

Having defined a path as a set of way points and the mobile robot being able to determine its actual position based on the features the laser range sensor registers in the environment surrounding the rover, it is necessary to adopt a control strategy that assists the rover to comply as close as possible with the path it was commanded to follow. This control strategy aims to

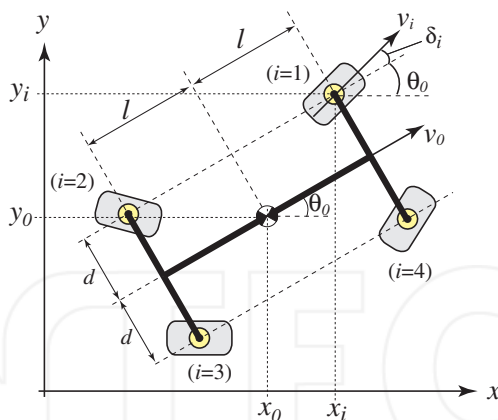


Fig. 10. Kinematic model of a four-wheeled vehicle

correct the steering and driving maneuvers generated by the vehicle while traversing an area in order to reach its destination.

In this section the nonholonomic kinematic constraints that govern the vehicle, and the control method known as “following the carrot” that implements them to minimize the steering and driving errors are presented.

### 8.1 Nonholonomic kinematic model and constraints

In the nonholonomic kinematic model of a slipless vehicle shown in Fig. 10, the following assumptions are considered: the distance between wheels are strictly fixed, the steering axle of each wheel is perpendicular to the terrain surface and the vehicle does not consist of any flexible parts. In Fig. 10, the subscript  $i$  denotes the identification of the wheel,  $(x_0, y_0, \theta_0)$  the position and orientation of the center of gravity of the vehicle,  $(x_i, y_i)$  the position of each wheel,  $v_i$  the linear velocity of each wheel,  $l$  longitudinal distance from the centroid of the vehicle to the front or rear wheel and  $d$  the lateral distance from the centroid of the vehicle to the left or right wheel.

The nonholonomic constraints of the implemented mobile robot are expressed by Equation 29 and Equation 29,

$$\dot{x}_0 \sin \phi_0 - \dot{y}_0 \cos \phi_0 = 0 \quad (29)$$

$$\dot{x}_i \sin \phi_i - \dot{y}_i \cos \phi_i = 0 \quad (30)$$

where,  $\phi_0 = \theta_0$ , and  $\phi_i = \theta_0 + \delta_i$ , having  $\delta_i$  as the wheel steering angle. By looking at the configuration between every wheel and the center of the gravity of the vehicle, the geometric constraints can be given by

$$\left. \begin{aligned} x_1 &= x_0 + l \cos \theta_0 - d \sin \theta_0 \\ x_2 &= x_0 - l \cos \theta_0 - d \sin \theta_0 \\ x_3 &= x_0 - l \cos \theta_0 + d \sin \theta_0 \\ x_4 &= x_0 + l \cos \theta_0 + d \sin \theta_0 \end{aligned} \right\} \quad (31)$$



$$\left. \begin{aligned} y_1 &= y_0 + l \sin \theta_0 + d \cos \theta_0 \\ y_2 &= y_0 - l \sin \theta_0 + d \cos \theta_0 \\ y_3 &= y_0 - l \sin \theta_0 - d \cos \theta_0 \\ y_4 &= y_0 + l \sin \theta_0 - d \cos \theta_0 \end{aligned} \right\} \quad (32)$$

which can be rewritten as  $x_i = x_0 + X_i$  and  $y_i = y_0 + Y_i$ .

## 8.2 Steering and driving maneuvers

Having defined the nonholonomic constraints of the implemented mobile robot, it is possible to determine the steering angle of each wheel  $\delta_i$  and the driving maneuvers of the vehicle. By transforming the nonholonomic constraints given by Equation 30, the steering angle can be obtained as follows,

$$\delta_i = \tan^{-1}(\dot{y}_i / \dot{x}_i) - \theta_0 \quad (33)$$

Then, by substituting Equation 31 and Equation 32 into Equation 33,  $\delta_i$  is derived by the following equation:

$$\delta_i = \tan^{-1} \left( \frac{v_0 \sin \theta_0 - \dot{Y}_i(\dot{\theta}_0)}{v_0 \cos \theta_0 - \dot{X}_i(\dot{\theta}_0)} \right) - \theta_0 \quad (34)$$

where,  $v_0$  represents the linear velocity of vehicle and  $\dot{\theta}_0$  is the turning angular velocity of vehicle. The driving maneuver of the vehicle can be achieved by the manipulating the wheel's angular velocity  $\omega_i$ . The relationship between  $\omega_i$  and wheel's linear velocity  $v_i$  can be written as:

$$\omega_i = v_i \cos \delta_i / r \quad (35)$$

where,  $r$  is wheel radius. Here,  $v_i$  can be expressed by  $\dot{x}_i$  or  $\dot{y}_i$  as:

$$v_i = \dot{x}_i / \cos \phi_i = \dot{y}_i / \sin \phi_i \quad (36)$$

Substituting Equation 31 or Equation 32 into the above equations, the angular velocity of the wheel  $\omega_i$  can be finally derived as shown by Equation 37:

$$\omega_i = \begin{cases} (v_0 \cos \theta_0 + \dot{X}_i(\dot{\theta}_0)) \cos \delta_i / r \cos \phi_i & (\theta_0 \leq \pi/4) \\ (v_0 \sin \theta_0 + \dot{Y}_i(\dot{\theta}_0)) \cos \delta_i / r \sin \phi_i & (\theta_0 \geq \pi/4) \end{cases} \quad (37)$$

## 8.3 Follow the carrot

Once a local path has been generated from the actual location of the vehicle to the optimal position found by the algorithm proposed on Subsection 7.2.1 the robot will execute the path by following its set of configurations. The method used in this research to track the path provided by the path planner is called "follow the carrot" and is illustrated in Fig. 11(a). This approach calculates the desired heading  $\phi_{carrot}$  based on the current position of the mobile robot and the local path. This approach draws a line equivalent to the radius  $r$  of a circle, from the center of the vehicle's coordinate system to the desired path. The carrot point, or goal point, is then defined to be the point on the path that intersects the generated circle and that has an orientation given by  $\phi_{err}$ . The most important parameter of this method is the orientation error, defined to be the angle between the current vehicle heading and the line drawn to the carrot point. A proportional control law can then aim at minimizing the orientation error. The magnitude of a turn is decided by Equation 38

$$\varphi = K_p \phi_{err} \quad (38)$$

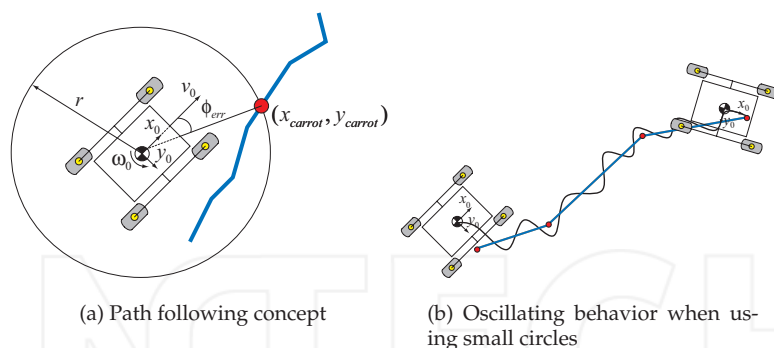


Fig. 11. Conceptual detail of the implemented path following strategy.

where  $K_p$  is the proportional gain and  $\phi_{err}$  is the orientation error defined by

$$\phi_{err} = \phi_{carrot} - \phi_{heading} \quad (39)$$

In Equation 39,  $\phi_{heading} = \delta_i$  and the desired carrot angle  $\phi_{carrot} = \theta_d$ , where  $\theta_d$  represents the desired steering orientation angle of each the wheels of the vehicle. With this in mind Equation 39 can be rewritten as Equation 40;

$$\theta_{err} = \theta_d - \delta_i \quad (40)$$

The mobile robot can move towards the carrot by controlling the turning angular velocity  $\dot{\theta}_d$ . Thus by substituting  $\theta_0 = \theta_d$  and  $v_0 = v_d$  into Equation 37, the desired angular velocity of each wheel  $\omega_{di}$  can be achieved.

One of the most important characteristics of this approach is its simplicity in terms of implementation, alas, it has a couple of major drawbacks (Helmick et al., 2004), (Kelly, 1994). First, the vehicle has a tendency to naturally cut corners. This happens because the vehicle immediately tries to turn towards each new carrot point. Another drawback is that the vehicle could oscillate about the path, particularly in the case of a small "lookahead distance" (smaller circles) or at higher speeds as presented in Fig. 11(b). In our implementation we solve these issues by having a constant linear speed of  $v_d = 11[\text{cm/s}]$  which allows the vehicle to change its heading without losing track of the path.

A large lookahead distance will tend to filter out small features of the path, but results in a smooth motion of the vehicle. A small value results in large heading changes of the vehicle for small path errors (which is extremely inefficient), but results in an overall smaller path following error. A lookahead distance that balances the tradeoff between getting closer to the path and decreasing the magnitude of the heading error  $\phi_{err}$  is then determined. Under nominal conditions, the vehicle path error will always be smaller than the lookahead distance. If this is not the case, then the radius is grown until an intersection does occur.

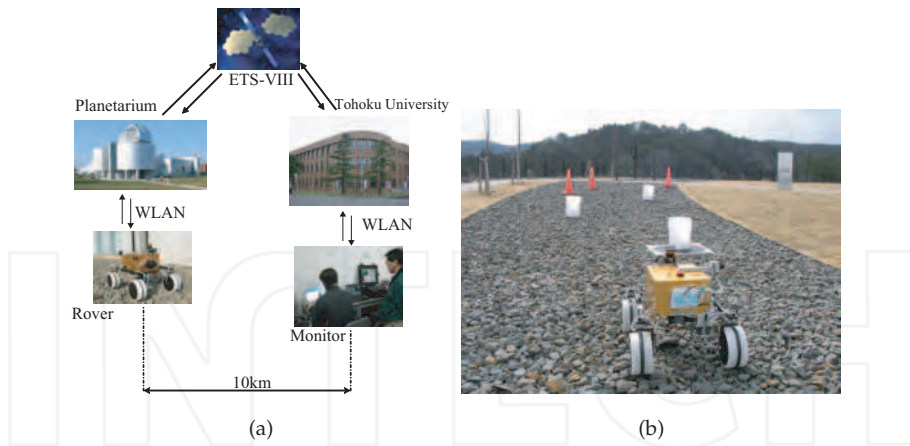


Fig. 12. Setup of the experiments realized at Sendai city's planetarium.

## 9. Experimental Results

In order to validate the procedures presented in the previous sections of this chapter, a set of experiments that recreated the occluding effect on the laser range sensor implemented in our mobile robotic testbed were carried out. This set of experiments were conducted between Tohoku University and the Planetarium of the city, both located in Sendai, Japan but separated by a distance of approximately 10 [km]. The communication between the human monitors at Tohoku University and the mobile robotic platform at the Planetarium relied on the Japanese Engineering Test Satellite VIII (ETS-VIII). The communication setup implemented between both ends is presented in Fig. 12(a). The communication link provided by the ETS-VIII can be set to three possible bandwidth specifications and data package loss rates in order to simulate the communication conditions on a possible lunar mission. The experiments consisted on the navigation of our mobile robotic platform through an environment where various obstacles were located in such a way that the field of view of the laser range sensor would be limited due to the characteristics of the environment as can be seen in Fig. 12(b). This location at the Planetarium has an area of 20[m] × 20[m] and is covered by loose rocks that simulate the possible conditions the vehicle may encounter on the lunar surface. During the navigation, the vehicle acquires the information of its environment and through the procedures presented in this chapter, determined the next sensing position, the local path to that position and finally it executed the generated local path. The process was then repeated until the global destination was achieved.

The comparison between the generated global path, the succession of the generated local paths that reconfigured the global path and the actual path followed by the vehicle (represented by the odometry obtained from its proprioceptive sensors) are described in Fig. 13(a). The orientation profile the vehicle realized through out the navigation experiment is depicted by Fig. 13(b).

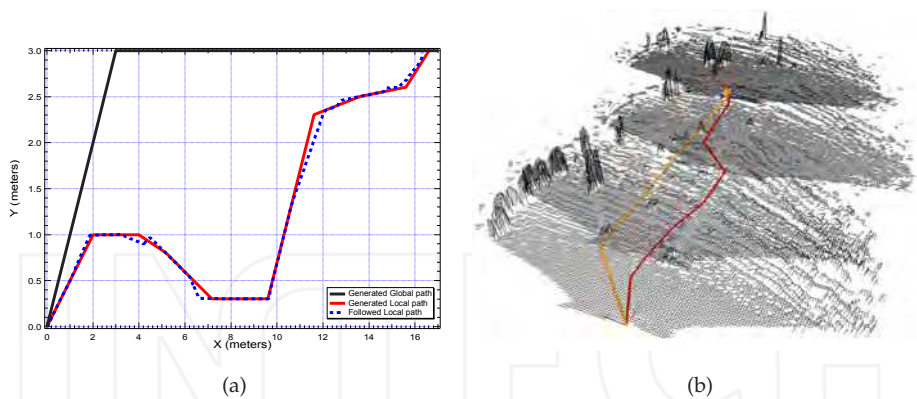


Fig. 13. Experimental results: generated and followed paths of the vehicle.

As it can be seen from Fig. 13(a), the vehicle initially follows the global path but it soon steers towards the left in order to avoid the obstacle on the right and when it reaches its local destination it rotates around its z-axis in order to cover a bigger area in the next mapping procedure. Instead of following the path suggested by the global path, the vehicle continuously moves towards a position that will enable it maximize the number of nodes it can see from the determined next sensing position.

## 10. Concluding Remarks

In this chapter, the combination of two path planning strategies, a global path planner and a local path planner, was discussed for a mobile robot (rover) to explorer in a lunar surface environment with poor illumination and rich obstacles. The global path planner would use the representation of the elevation data called the *Global DEM* (digital elevation map) gathered by a satellite mission orbiting the targeted celestial body to calculate a general path to the desired destination. The local path planner in the other hand, uses the representation of the elevation data called *Local DEM* obtained by its laser range sensor mounted on the rover to compute a path that would reconfigure the global path initially determined.

An analysis of the LDEM, or local map, was made to classify the different areas according to the occlusion effect due to the presence of obstacles. This analysis allowed the quantification of the occluded and non-occluded regions within the area that can be covered by a laser range sensor. The algorithm that quantifies the non-occluded areas that can be perceived by the sensory system from the position where the rover is located within a local map was addressed. This quantification is given in the form of an index called the visibility index which is later incorporated in the local path planner through a method called the next sensing position. The next sensing position concept and its algorithm were explained thoroughly, and numerical simulation results were presented. The local path planner evaluates the terrain and finds an optimal path in terms of distance to the local goal obtained by the next sensing position algorithm.

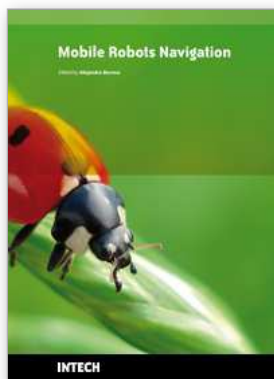
Fractal geometry was introduced to generate a discrete natural terrain representation which can describe topographic features similar to those of the lunar surface. Particularly, a method based on the fractional Brownian motion approach was implemented to validate through numerical simulations the concepts presented in this chapter. Experiments that utilized the Japanese Engineering Test Satellite VIII (ETS-VIII) in order to simulate the conditions of a mobile robotic lunar exploration mission were carried out. In these experiments, the vehicle navigated through an irregular, rocky terrain by perceiving its environment, determining the next sensing position that maximizes the non-occluded region within each local map and executing the local path generated.

## 11. References

- Adams, M. D. (1999). Sensor modelling, design and data processing for autonomous navigation, *World Scientific*.
- Barnea, D. I. & Silverman, H. F. (1972). A class of algorithms for fast digital image registration, *IEEE Transactions on Computers* pp. 179–186.
- Barnsley, M. F., Devaney, R. L., Mandelbrot, B. B., H. O. Peitgen, D. S. & Voss, R. F. (1998). The science of fractal images, *Springer-Verlag*.
- Borenstein, J., Everett, H. R. & Feng, L. (1996). Navigating mobile robots, *A. K. Peters, Wellesley, MA*.
- Danner, T. & Kavraki, L. E. (2000). Randomized planning for short inspection paths, *Proc of the 2000 IEEE Int. Conf. on Robotics and Automation*.
- Dupuis, E., Allard, P., Bakambu, J., Lamarche, T., Zhu, W. & Rekleitis, I. (2005). Towards autonomous long-range navigation, *Proceedings 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.
- Everett, H. R. (1995). Sensor for mobile robots, theory and application, *A. K. Peters*.
- Heckman, N., Lalonde, J., Vandapei, N. & Hebert, M. (2007). Potential negative obstacle detection by occlusion labeling, *Proceedings IEEE International Conference on Intelligent Robotics and Systems*.
- Helmick, D. M., Cheng, Y., Clouse, D. S., Matthies, L. H. & Roumeliotis, S. I. (2004). Path following using visual odometry for a mars rover in high-slip environments, *Proc of the IEEE International Aerospace Conference*.
- Hokuyo (2008). Scanning laser range finder utm-30lx/ln specification, *Hokuyo Automatic Specification Report*. Hokuyo Automatic Co. LTD.
- Iagnemma, K. & Dubowsky, S. (2004). Mobile robots on rough terrain, *Springer Tracts in Advanced Robotics*.
- Ishigami, G., Nagatani, K. & Yoshida, K. (2006). Path following control with slip compensation on loose soil for exploration rover, *Proc of the 2006 IEEE International Conference on Intelligent Robots and Systems*.
- Ishigami, G., Nagatani, K. & Yoshida, K. (2007). Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics, *Proc of the IEEE Int. Conf. on Robotics and Automation*.
- Jarvis, R. A. (1994). On distance transform based collision-free path planning for robot navigation in known, unknown and time-varying environments, *Advanced Mobile Robots*, World Scientific Publishing Co. Pty. Lt, pp. 3–31.
- Kalisz, J. (2004). Review of the methods for time interval measurements with picosecond resolution, *Metrologia* **41**: 17–32.

- Kato, M., Takizawa, Y., Sasaki, S. & Team, S. P. (2007). The selene mission: Present status and science goals, *Proceedings 38th Lunar and Planetary Science Conference* .
- Kelly, A. (1994). A feedforward control approach to the local navigation problem for autonomous vehicles, *Robotics Institute Technical Report, CMU-RI-TR-94-17* .
- Kunii, Y. & Kubota, T. (2006). Human machine cooperative tele-drive by path compensation for long range traversability, *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems* .
- Langer, D., Mettenleiter, M., Hart, F. & Frohlich, C. (2000). Imaging laser scanners for 3-d modeling and surveying applications, *Proc. of the 2000 IEEE International Conference on Robotics and Automation (ICRA 2000)*, San Francisco, CA, U.S.A.
- LaValle, S. (2006). Planning algorithms, *Cambridge University Press* .
- Lucas, B. D. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision, *Proc. of Imaging Understanding Workshop* .
- Mandelbrot, B. B. (1983). The fractal geometry of nature, *W. H. Freeman* .
- Mazo, R. M. (2002). Brownian motion fluctuations, dynamics, and applications, *Oxford University Press* .
- Morales, Y., Takeuchi, E., Carballo, A., Tokunaga, W., Kuniyoshi, H., Aburadani, A., Hiro-sawa, A., Nagasaka, Y., Suzuki, Y. & Tsubouchi, T. (2008). Imaging laser scanners for 3-d modeling and surveying applications, *Proc. of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France*.
- Moravec, H. P. (1979). Visual mapping by a robot rover, *Sixth International Joint Conference on Artificial Intelligence* .
- Pascoal, J., Marques, L. & de Almeida, A. T. (2008). Assessment of laser range finders in risky environments, *Proc. of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* .
- Rekleitis, I., Bedwani, J.-L. & Dupuis, E. (2009). Autonomous planetary exploration using lidar data, *Proceedings of the IEEE International Conference on Robotics and Automation* .
- Sanger, T. D. (1988). Stereo disparity computation using gabor filters, *Journal of Biological Cybernetics* **59**(6): 405–418.
- Sharma, S. & Vishwamittar (2005). Brownian motion problem: Random walk and beyond, *Resonance* **10**(8): 49–66.
- Skrzypcznski, P. (2008). How to recognize and remove qualitative errors in time-of-flight laser range measurements, *Proc. of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France*.
- Watson, K., Murray, B. & Brown, H. (1961). On the possible presence of ice on the moon, *Journal of Geophysical Research* **Vol. 66**: 1598–1600.
- Ye, C. & Borenstein, J. (2002). Characterization of a 2-d laser scanner for mobile robot obstacle negotiation, *Proc. of the 2002 IEEE International Conference on Robotics and Automation (ICRA 2002)*, Washington, D.C., U.S.A.





## **Mobile Robots Navigation**

Edited by Alejandra Barrera

ISBN 978-953-307-076-6

Hard cover, 666 pages

**Publisher** InTech

**Published online** 01, March, 2010

**Published in print edition** March, 2010

Mobile robots navigation includes different interrelated activities: (i) perception, as obtaining and interpreting sensory information; (ii) exploration, as the strategy that guides the robot to select the next direction to go; (iii) mapping, involving the construction of a spatial representation by using the sensory information perceived; (iv) localization, as the strategy to estimate the robot position within the spatial map; (v) path planning, as the strategy to find a path towards a goal location being optimal or not; and (vi) path execution, where motor actions are determined and adapted to environmental changes. The book addresses those activities by integrating results from the research work of several authors all over the world. Research cases are documented in 32 chapters organized within 7 categories next described.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Andres Mora, Keiji Nagatani and Kazuya Yoshida (2010). Path Planning and Execution for Planetary Exploration Rovers Based on 3D Mapping, *Mobile Robots Navigation*, Alejandra Barrera (Ed.), ISBN: 978-953-307-076-6, InTech, Available from: <http://www.intechopen.com/books/mobile-robots-navigation/path-planning-and-execution-for-planetary-exploration-rovers-based-on-3d-mapping>

**INTeCH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821