UNIVERSITÀ DEGLI STUDI DI CATANIA
DIPARTIMENTO DI INGEGNERIA ELETTRICA
ELETTRONICA E INFORMATICA

International PhD in
Electronic, Automation and Control of Complex Systems

NAVIGATION SYSTEMS FOR AUTONOMOUS ROBOTS BASED ON OPEN SOURCE GIS TECHNOLOGIES

Michele Mangiameli

ADVISOR
Prof. Eng. Giovanni Muscato

COORDINATOR
Prof. Eng. Luigi Fortuna

XXV cycle

Acknowledgments

At the end of my Ph.D. studies I would like to thank all people who made this work possible.

First of all, I would like to express my deepest gratitude to my advisor, Prof. Giovanni Muscato, for his patience, motivation, enthusiasm, and immense knowledge.

I am also very grateful to Prof. Giuseppe Mussumeci for his continuous support to my Ph.D. research, and to Prof. Luigi Fortuna, for his generous advice and encouragement over these three years.

A sincere thank goes to the members of the Laboratory of Robotics and Laboratory of Geomatics, Department of Civil and Environmental Engineering, especially to the Eng. Donato Melita, Eng. Luciano Cantelli, Eng. Filippo Bonaccorso and the surveyor Carmelo Lombardo for providing me the instruments and facilities to accomplish my work.

Last but not the least, I dedicate this thesis to my parents, Aurora and Salvatore, and to my girlfriend Annalisa, who has always been close to me.

Index

Introduction

Man has always dreamed of building artificial beings, which take over tedious or dangerous tasks, with abilities of entertainment and subject to human commands. In everyday language, these artificial beings are called “robots”. Robots are characterized by:

- the capability to act in the environment using a mechanical locomotion system and to interact with the objects present in the environment using a handling system;
- a perceptive capacity to measure parameters relating to their internal state (i.e., wheel speeds, torques, charge of batteries, acceleration and trim) using proprioceptive sensors, and to measure external parameters (i.e., temperature of the environment, humidity and distance from obstacles) using exteroceptive sensors;

- the capability to establish an intelligent link between perceptions and actions, using a control system that works taking into account the mechanical constraints of the robot with respect to those inside the environment (i.e., the location of obstacles).

There are two fundamental aspects of mobile robotics: the estimation of the robot position in the operating environment and the robotic mapping to acquire spatial models of the physical environment.

These challenging aspects can be taken on using different approaches, depending on the application context in which the robot will work.

In some applications it is sufficient to determine the relative displacement with respect to a previous position; in other applications it is useful to determine the absolute position in the environment with respect to a global reference system.

This aspect is very serious when the operating environment of the robot is unknown. In this case, the mobile robot should detect information related to its position and to the topology of the environment where it operates.

The mapping problem is generally considered as one of the most important challenges in the pursuit of building truly autonomous mobile robots, because it requires the integration of information gathered by the robots sensors into a given representation. So the two central aspects in mapping are the representation of the environment and the interpretation of sensor data.

To acquire a map and to estimate its position, a robot should be equipped with sensors that enable it to perceive the outside world. Common sensors usable for this task include cameras, range finders using sonar, laser, and infrared technologies, radar, tactile sensors, compasses, laser scanner and GPS for outdoor applications.

The navigation of a robot in an environment for reaching a goal requires the solution of three tasks: mapping, localization, and path planning.

The scope of this PhD thesis is the management of the navigation for autonomous mobile robots in outdoor environments using geographic information systems.

This technology can be seen as an extension of classical topography but uses advanced functionality for the management of any type of information as a reference spatial and temporal in software environment.

The first author of modern computerized Geographic Information Systems (GIS), also known as the "father of GIS", is Roger F. Tomlinson that, during his tenure with the federal government in the 1960s, planned and directed the development of the Canada

Geographic Information System, the first computerized GIS in the World.
Subsequently, several definitions for GIS technology were published. Some examples are: "GIS is system of databases, hardware, software and organization that manages, processes and integrates the information on a spatial or geographic platform" [*Barrett-Rumor,* 1993], or "A Geographical Information System is a group of procedures that provide data input, storage and retrieval, mapping and spatial analysis for both spatial and attribute data to support the decision-making activities of the organisation" [*Grimshaw,* 1995].
GIS technology finds application in different human activities where it is necessary to place information in a geographical context.
Many of these activities regards the monitoring and management of natural environments, the recording and planning of human-made environments, the understanding of social structures, transport and navigation.
The GIS environment has a layered architecture where the raster layer represents the cartographic base georeferenced with topographic algorithms. On the raster base, different vector layers are overlapped as sets of geometric primitives (points, lines, areas, surfaces and volumes) for the representation of real-world phenomena.

For this reason, the core of this PhD thesis is the development of a navigation system for autonomous robots based on the GIS technology using cartography and maps geo-referenced with the rigorous approach of geomatics to analyze the satellite positioning data detected by the robot and to manage its navigation accurately.

In particular the thesis exploited desktop GIS platforms and developed webGIS platforms using free and open source software for optimizing and customizing these platforms. For managing the navigation of the robot and the spatial data, an external spatial DBMS (DataBase Management System) was also developed with free and open source technologies.

The thesis consists of four main parts corresponding to the following stages of the research:

- *Chapter 1* introduces the State-of-Art of mobile robotics. It introduces the kinematic model, the sensors installed on board the robots and the mathematical approaches used for navigation and localization of mobile robots;
- *Chapter 2* presents a GIS environment, analyzing in detail the topographic approach and the cartography used. It describes the tools of desktop GIS and webGIS platforms and their functionalities;

- *Chapter 3* deals with the spatial database created for the study, focusing on the free and open source DBMS software used for its construction;
- *Chapter 4* describes the design of an architecture for managing the navigation of robots in the GIS environment; the desktop GIS platform employed and the webGIS platform developed to manage the navigation of robots in the GIS environment; the analysis of the GPS signal to represent the position of the robots in the GIS environment; the employment of GIS tools to determine the optimal paths for the mobile robots and to test if the path obtained in the GIS environment is correct; the topographic surveys performed using a GPS. The procedures in the desktop GIS environment to manage the UAV navigation will be also reported, together with the webGIS platform in which it is possible to assign the path to the robots.
- *Conclusions* finally discuss all results obtained.

Chapter 1

Robotic mapping and exploration

In mobile robotics two fundamental aspects should be analyzed: the first one concerns the knowledge of the robot position within an environment, when the map of this environment is known beforehand; the second aspect regards the simultaneous estimation of the robot position and environment map.

Whenever a mobile robot is expected to navigate in unknown environment, it is very important to equip the robot whit a robust and reliable localization algorithm.

Considering the first aspect, the cartography or geo-referenced maps is provided as input to the localization algorithm.

As for the second aspect, the map is estimated in real time using SLAM (Simultaneous Localization And Mapping) techniques.
In general, learning maps whit single-robot systems require the solution of three tasks: mapping, localization and path planning [*Makarenko et al.,* 2002]. A diagram showing these three tasks, as well as the combined problems in the overlapping areas, is reported in Figure 1.

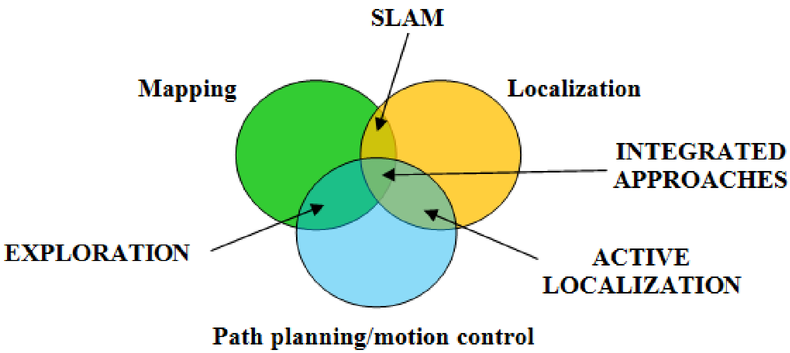


Figure 1 – Scheme of the mapping, localization and path planning tasks

As shown in Figure 1, SLAM is the problem of building a map and localizing the robot within this map simultaneously. The mapping and localizing tasks cannot be solved independently.

Active localization seeks to guide the robot to locations within the map to improve the pose estimate. In contrast, the exploration approaches assume pose information and focus on guiding the robot efficiently through the environment in order to build a map [*Stachniss,* 2009].

The center area of the diagram, that is the intersection of the three tasks, represent the integrated approaches, so the solving of mapping, localization, and path planning simultaneously. This is the so called SPLAM (Simultaneous Planning, Localization And Mapping) problem.

The problem of locating, mapping and SLAM, can be seen as the problem of estimating the state of a discrete system. To solve this problem, different techniques can be used to perform measurements on the system, which, however, do not consider the noisy nature of the measurements itself. Since noise is typically described with statistical approaches, then the problem of localization, mapping, and slam can be solved by stochastic methods.

A typical stochastic method is the Bayes filter. Three mathematical tools used in this method are the Particle filter [*Gordon et al.,* 1993], the Kalman filter [*Kalman,* 1960], and the Extended Kalman filter [*Leonard and Durrant-Whyte,* 1992; *Williams et al.,* 2002].

Particle filter implementations can be found in the area of robot localization, in which the robot position has to be recovered from sensor data [*Engelson and McDermott,* 1992; *Borenstein et al.,* 1996; *Thrun et al.,* 2000].

The Kalman filter has been employed in a wide range of applications, including the control of a dynamic system (state estimation) and to predict the future of dynamic systems that are difficult or even impossible for people to control [*Cox and Wilfong,* 1990; *Kiriy and Buehler,* 2002; *Baltzakis and Trahanias,* 2002].

The Extended Kalman filter [*Smith and Cheeseman,* 1986] is used in most present-day researches on SLAM, even if the high computational complexity prohibits it from operating in real time and makes it unfeasible in large environments [*Lee et al.,* 2007].

The SLAM technology can also map the environment in a more accurate way using a team of mobile robots that automatically recognize the occurrence of map overlapping by matching their current frame with the maps built by other robots [*Léon et al.,* 2008; *Lee and Lee,* 2009]. Moreover, some applications are available where the SLAM is combined to the GPS in order to increase the robustness, scalability and accuracy of localization [*Carlson,* 2010].

Another important technique is 3D laser scanner, particularly used for simultaneous localization and mapping problem, which concerns the solving of 3D maps and robot poses with six degrees of freedom. Examples of 3D SLAMS can be found in humanoid robotics, where the robots are characterized by a great autonomy and the ability to create their own world map on the fly [*Stasse et al.,* 2006].

1.1. Kinematic models for mobile robots

The kinematic model of a mobile robot depends mainly on the architecture of robot locomotion in order to enable easy advancement in the workplace.

The robotic motion is dealt by three different fields of study: locomotion, dynamics and kinematics.

1. Locomotion is the process by which an autonomous robot or vehicle moves. In order to produce motion, forces must be applied to the vehicle;
2. Dynamics is the study of motion in which forces are modeled, including the energies and speeds associated with these motions;
3. Kinematics is study of the mathematics of motion, without considering the forces that affect the motion. It deals with the geometric relationships governing the system and the relationship between control parameters and the behavior of a system in the state space.

Therefore the environment in which the robot will operate can determine a first distinction in:

- Ground robots (Figure 2);
- Flying robots (Figure 3);
- Underwater robots (Figure 4);
- Climbing robots (figure 5).

It is worth noting the structure of robots is inspired by the shape of animals, like spiders of fishes.

Figure 2 – Three examples of ground robots

Figure 3 – Three examples of flying robots

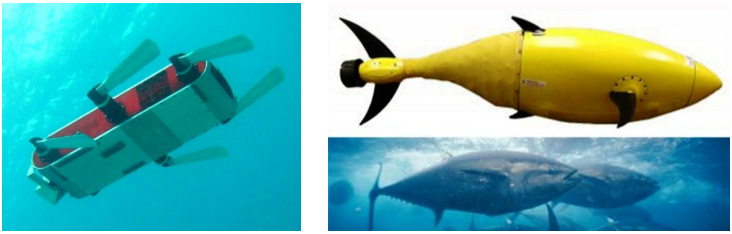


Figure 4 – Three examples of underwater robots

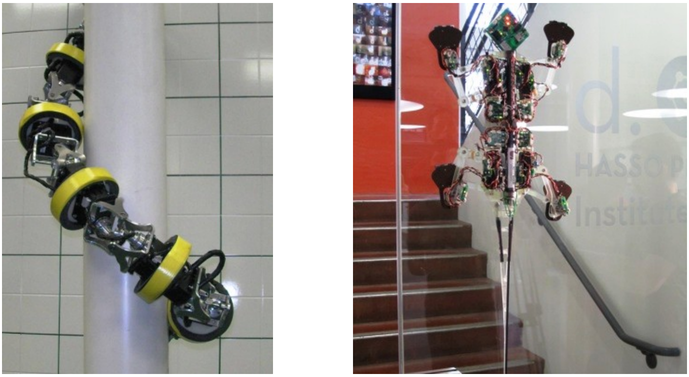


Figure 5 – Two examples of climbing robots

In order to define a kinematic model of a mobile robot, the ground robots and, in particular, Wheeled Mobile Robots (WMRs), are considered. These robots are capable of locomotion on a surface solely through the actuation of wheel assemblies mounted on the robot and in contact with the surface.

Depending on their degree of mobility, WMRs can be divided in robots with locally restricted mobility (non-holonomic) and robots with full mobility (holonomic).

From the mechanical point of view and from the constructive typology, a great importance for the mobility is represented by wheels.

As reported in Figure 6, different types of wheels can be used as system of locomotion for WMRs.

To avoid slip in WMRs, wheels must move instantaneously along some circle of radius so that the center of that circle is located on the zero motion line. This center point, lying anywhere along the zero motion line (Figure 7), is called Instantaneous Center of Rotation (ICR).

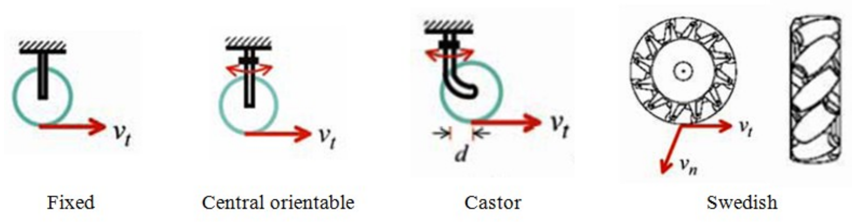


Figure 6 – Different types of wheels for ground robots; v_n is the velocity and v_t is the lateral and transverse velocity

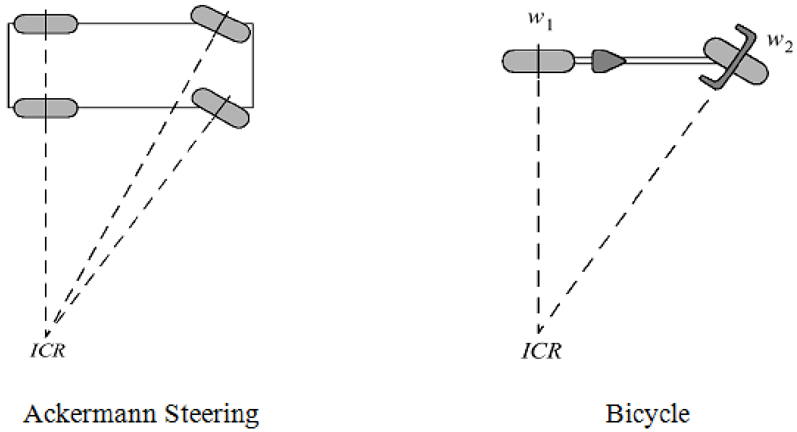


Figure 7 – Instantaneous Center of Rotation for mobile robot maneuverability

The kinematic model is required to robots for planning the eligible paths or trajectories, for devel-oping algorithms of control, for simulations, etc..

The model provides all the directions of motion instantly eligible and correlates inputs in speed with the derivatives of the variables configuration.

$$\frac{dq}{dt} = G(q)v \tag{1}$$

Starting from this kinematic model, other reference models have been developed, i.e. the ideal unicycle model and the unicycle model for differential driving.

Ideal unicycle

The ideal unicycle model consists in a single wheel able to move and change orientation in the plane. If *X* is the vector of the position in the plane and $[x\ y\ \theta]^T$ is its orientation, then the model can be schematized as in Figure 8.

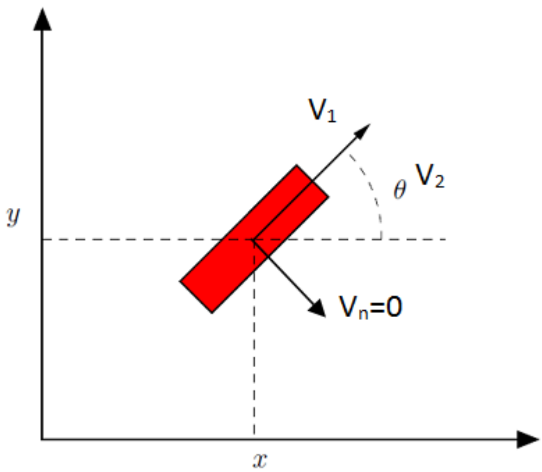


Figure 8 – Schematic representation of the ideal unicycle

In this model only a constraint of pure rolling is present, which can be expressed by the following mathematical formulation:

$$[\sin\theta \quad \cos\theta \quad 0]\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = 0 \tag{2}$$

This constraint imposes that the robot can move in the plane in the single direction defined by θ, but with the plan fully accessible. In fact, given any two configurations q_0 and q_1, the robot can always reach q_1 starting from q_0 through a series of rotations around the axis and translations along θ.

$$\begin{cases}\dot{x} = \cos(\theta)v_1\\ \dot{y} = \sin(\theta)v_2\\ \dot{\theta} = v_2\end{cases} \qquad \dot{q} = G(q)\begin{pmatrix}v_1\\ v_2\end{pmatrix} \qquad G(q) = \begin{pmatrix}\cos(\theta)0\\ \sin(\theta)0\\ 0 \quad 1\end{pmatrix} \tag{3}$$

Given the complete reachability in the plane and the simplicity in construction, the unicycle model is one of the more popular models in kinematics.

There are various achievements for this model. In the most common configuration, the differential driving is characterized by two fixed wheels plus a castor.

Unicycle model for differential driving

The unicycle model for differential driving (Figure 9) is based essentially on two drive wheels aligned each other, and one or more independent wheels (usually castor).

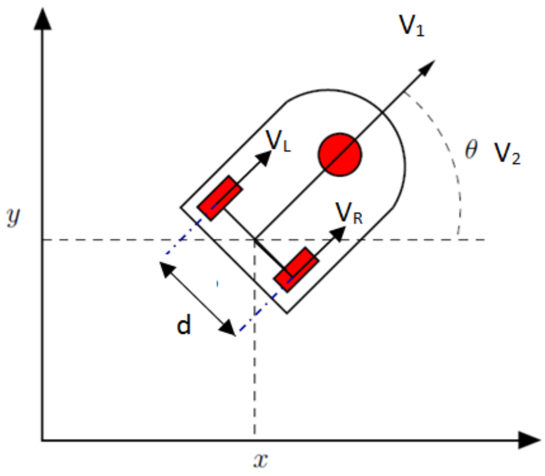


Figure 9 – Schematic representation of the unicycle for differential driving

The direction of translation, identified by θ, is perpendicular to the axis connecting the centers of the two wheels, and represents the orientation of the robot in the plane.

In this perspective, the non-holonomic constraints represented by the two wheels are identical and can identify the constraint for the unicycle ideal described in Equation 2.

This model differs from the ideal one for the fundamental kinematic equations, since the translational and rotational velocity are not identified as first. The robot is controlled via the motors fixed on two wheels, for which the kinematic equations become:

$$\dot{q} = G(q)\begin{pmatrix} v_R \\ v_L \end{pmatrix} \quad G(q) = \begin{pmatrix} \frac{\cos(\theta)}{2} & \frac{\cos(\theta)}{2} \\ \frac{sen(\theta)}{2} & \frac{sen(\theta)}{2} \\ \frac{1}{2d} & -\frac{1}{2d} \end{pmatrix} \quad \begin{cases} v_R = v_1 + dv_2 \\ v_L = v_1 - dv_2 \end{cases} \tag{4}$$

Once the kinematic model of the robot is known and it is equipped with degrees sensors to detect the displacement of the wheels, it is possible to obtain the pose of the robot in the environment.

1.2. Mobile robot sensors

A good and complete set of sensors is a fundamental for a mobile robot. The mobile robot must know its movements using information provided by instrumentation capable of acquiring data regarding its kinematics and dynamic. Furthermore, it should be able to obtain information on the environment where it works.

The sensors installed onboard the robot must detect information related to the metric structure and topological environment, as well as to sense the presence of obstacles, walls or objects in the environment, in order to let the robot build a map of the environment and localize inside.

The sensors can be classified according to different criteria. At the functional level, they are usually classified as proprioceptors or eteroceptors .

The proprioceptors are sensors measuring the internal variables of the robot, such as positions and joint velocity, state of batteries of the electrical system, temperature of motors, etc.

The eteroceptors are sensors that measure the variables external to the robot, as the distance from obstacles, the absolute position of the robot in space, the force applied at the extremity of in the arm by the environment, etc.

In general the sensors used in mobile robotics can be divided into two main categories:

1. *Sensors of relative displacement,* representing the class of sensors that allow to determine the relative displacement of the robot in a given time interval;
2. *Proximity sensors,* being the class of sensors that measures the metric objects in the environment, such as the distance that separates them from the robot.

All sensors are characterized by parameters that define their performance and allow comparing them with each other. These parameters are:

- *Resolution* defines and measures the smallest deviation of the measured quantity that a sensor is able to detect.
- *Repeatability* defines and quantifies the capability to a sensor to measure the same magnitude with measurements made at successive times.
- *Precision and accuracy*, related to random and systematic errors. While the precision measures the quality of measurements, the accuracy expresses the absence of systematic errors in the measurement.

- The *transfer function* describes quantitatively the relationship between the physical signal input and the electrical signal output from the sensor, which represents the measure.
- *Sensitivity* is the ratio between the physical signal input and the electrical output signal
- *Temperature coefficient* measures the dependence of the sensitivity with respect to the operating temperature.
- *Dynamic range* defines the width of the interval of values of the input signal that can be linearly converted into an electrical signal by the sensor. Signals outside of this interval can be converted into an electrical signal only with strong linearity or low accuracy/precision
- *Hysteresis* measures the amplitude of the error given as response by input signals changing value cyclically
- *Nonlinearity* measures the distance from the condition of linearity, that is, from that represented by a linear transfer function
- *Noise* due to random fluctuations or electronic interference. The noise is usually distributed on a wide spectrum of frequencies and many sources noise produce a sound called "white noise", where the power spectral density is the same for each frequency.

The noise is often characterized by providing the spectral density of the effective value of noise.

- *Bandwidth*. All sensors have a finite response time to an instantaneous change the quantities to be measured. In addition, many sensors have decay time, amount of time necessary to return to the original value after a step change in the quantities to be measured. The inverse of these two-stroke provides a rough indication of the upper and lower bound of the cutoff frequency. The bandwidth of the sensor is the frequency interval between these two bounds.

Sensors of relative displacement

Sensors of relative displacement allow knowing the relative displacement of the robot, that are the small variations of orientation and position within the environment.

These sensors perform only kinematic measures and dynamics detectable on board of the robot. Usually the information at the lowest level that it is possible to obtain directly for mobile robots equipped with wheels, is the rotation carried by the wheels themselves. This information is analyzed with the odometry in order to obtain the actual move.

The rotation of the wheels is recorded by devices called encoders, which are mounted on wheels and are able to detect small angle shifts. The most common encoders used in mobile robotics are the optical encoders.

Encoders allow measuring the rotational speed of an axis and, for extension, of a wheel. The encoders can be grouped in absolute encoders and incremental encoders.

Absolute encoders (Figure 10) are able to detect the absolute position of the axis in motion. This means that they can know the position even on the start. For this reason, they are particularly suitable for applications that require high precision and in general where it is required to have absolute positions (i.e. where the position of robotic arm should be known to move it around).

Figure 10 – The absolute encoder Kubler-Sendix 5868 Profibus

Incremental encoders (also called "relative") are instead able to calculate the variation of displacement without knowing anything about the initial position. This type of encoder is very simple to implement and interface with a control circuit, and is particularly indicated for the calculation of the velocity.

Another classification of encoders can be made according to the technology with which they are made. The following are some examples:

- Potentiometric;
- Magnetic;
- Inductive;
- Capacitive;
- Optical.

Potentiometric models exploit the capability of a potentiometer to emit an electrical signal proportional to the position that takes on its rotor. These encoders are only absolute, while the magnetic, inductive, capacitive or optical models can be of both types, absolute and incremental.

There are principally three different types of odometers that differ on the basis of the position of encoders:

- Encoders installed on drive wheels that are on the ICR. Odometer is simple but not very accurate because of possible skidding;
- Encoders installed on additional freewheels. They are affected by load and systematic errors;
- Encoders installed on two added free wheels. For this architecture it is necessary to ensure grip to the ground.

The odometry is subjected to two different types of errors: systematic errors and random errors.

Systematic errors [*Borenstein et al.,* 1996] cause inaccuracy of encoders, different wheel diameter, mean diameter of the wheels different from the nominal value, incorrect distance between the wheels, misalignment of the wheels and large sampling time.

Random errors cause wheels lip, unevenness of the ground and objects on the ground.

In order to reduce these errors, castor wheels should not be used to avoid possible skidding especially when most of the weight of the robot is applied on them. It is better to employ thin and rigid wheels to ensure a small and accurate point of contact with the ground and install wheels or trolleys for the odometry and not for traction.

The angular sensors to the Hall Effect belong to the class of sensors of relative displacement. These sensors provide the absolute measurement of the angle, starting from a "zero" conventional, or angular increase (relative size) of the joints of a kinematic chain or of the motors of the wheels of a mobile robot.

Proximity sensors

During its navigation, the mobile robot needs to interact with the environment. In addition to the information provided by the sensors of the relative displacement, the robot must be able to measure, for example the distance and the angle of obstacles with respect to its position. This information is usually provided by the sensors, as a set of points representing the distance of the obstacle along rays of angle different.

These sensors are divided into active and passive sensors.

The active sensors emit energy in the environment and detect the quantity of reflected energy in order to obtain the distance of the objects (Figure 11).

Active sensors can be classified according to the energy used. Usually the most frequent sources of energy are:

- Electromagnetic source, used by the radar sensors;
- Sound source, used by the sonar sensors;
- Light source, used by laser scanner or infrared sensor.

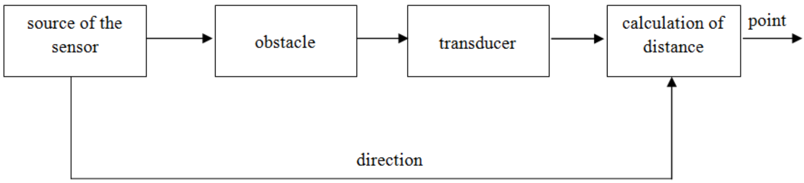


Figure 11 – A rough scheme of the active sensors

Radar sensors have the characteristic of transmitting a radio wave and receive its reflection. It is one of the most vital components of mobile robot, mostly dedicated to the obstacle detection, localization and mapping in extensive outdoor environments [*Rouveure et al.,* 2009].

Sonar are active proximity switches using an emitter of ultrasonic energy as source. The single sonar device is able to obtain information relating to the only direction it is oriented to.

The use of ultrasounds has two fundamental limitations. The first is that it is not possible to have a good directionality in the range of perception (usually with a difference of 30°). The second is that, given the limited velocity of sound in the air, it is not possible to update the information frequently.

Laser scanner is characterized by a laser diode as source of energy. The radius that the device emits is oriented towards the direction to be analyzed and a sensor (typically a photodiode) measures the phase difference or the echo time. The use of light energy, and particularly the laser, allows updating frequencies much higher than those obtained with acoustic energy; moreover an excellent directionality can be obtained thanks to the properties in inherent characteristics of the laser.

Unlike sonar, laser scanners allow to have a wider range of observations, due to the pair laser-diode mounted on a rotating mirror. If the mirror has two degrees of freedom, three-dimensional information concerning the working environment can be also retrieved.

Infrared sensor is a proximity sensor which works on the concept of sonar.

It consists of a transmitter (Tx) installed on the robot that emits a beam of infrared light. If the beam intercepts an obstacle, is reflected toward the robot, and is captured by a sensor receiver (Rx). To make sure that the sensor is not influenced by ambient light or other signals, it modulates the transmitted infrared beam, with a square wave. Only by tuning the receiver to the same frequency of the transmitter, you can receive the signal transmitted.

Passive sensors use energy that is already present in the environment. Devices as video sensors and thermal cameras belong to this category of contact sensors.

A mobile robot that must operate in outdoor environments needs sensors of position and absolute orientation.

These sensors can be considered sensors of absolute distance, or better, sensors of position, as they allow measuring the position of the mobile robot on which they are mounted with respect to a conventional absolute reference system. Using these sensors, it is possible to precisely track the position of the mobile robot from geo-referenced cartographic supports to the same reference system of the sensor, or manage and plan the navigation of the mobile robot on geo-referenced cartographic supports and pass it as input to the sensors of absolute position of the robot.

Typical sensors are absolute positioning GPS, which give the coordinates of the mobile robot according to the reference latitude and longitude universally accepted. Some of these sensors may also provide orientation with respect to the preferential direction (usually the magnetic North).

1.3. Navigation and localization

The navigation of a mobile robot is based on two fundamental aspects: the dead-reckoning and map-based positioning.

The first aspect is related to determine the pose of the robot in its operational environment, and the second aspect is related to the construction of the environment in which the robot is operating.

Techniques belonging to the dead-reckoning are odometry techniques, inertial navigation, the active beacon navigation, the navigational landmark and the Kalman filter. Conversely map-building and map-matching belong to the map-based positioning.

Odometry

The odometry is a method of sensor fusion used to obtain information about the pose of the robot, from measurements of incremental displacement in the time provided by the sensors.

The odometry belongs to localization methodologies for dead-reckoning, with relative measurements where the mobile robot localization is estimated through a wheel motion evaluation.

In order to know the displacement performed by the robot in the time interval considered, the following information should be available:

- the kinematics of the robot;
- rotation carried by the wheels over the time interval.

$$\Delta T_i = (t_{i-1} \,.\,.\,.\, t_i) \qquad (5)$$

This displacement can be considered starting from its two translational [Δx_i Δy_i] and rotational $\Delta\theta_i$ components. Then we can define the vector of relative displacement range ΔT_i as:

$$\Delta U_i=[\Delta x_i \Delta y_i \Delta\theta_i]^T \tag{6}$$

The entire odometric process is developed through three main phases:

1) Transduction of rotation of the wheels into electrical signals using the encoder;
2) Processing of the electrical signals to extrapolate numerically the relative or absolute rotation performed by each wheel;
3) Calculation of the displacement ΔU known the kinematics of the robot.

To obtain the displacement ΔU_i of the robot in ΔT_i range, the discretized version of Equation 4 can be used:

$$\Delta q_i = G(q)\begin{pmatrix} \Delta U_{Ri} \\ \Delta U_{Li} \end{pmatrix} \tag{7}$$

The movements of each wheel can be obtained from the measurement obtained by the encoder:

$$\Delta U_i = c_m N_i \tag{8}$$

where N_i is the number of ticks that the encoder has measured from last reading and c_m is the conversion factor between distance and tick encoder. This factor can be computed through the relation:

$$c_m = \pi \frac{D_n}{nC_e} \tag{9}$$

with D_n the nominal diameter of the wheel, C_e the resolution of the encoder (number of teeths on the wheel) and n the reduction ratio between the encoder and wheel.

Inertial Navigation Systems (INS)

The inertia navigation systems belong to the localization methodologies for dead-reckoning with relative measurements, where the mobile robot localization is estimated through its motion state evaluation (velocities and accelerations).

An inertial navigation system (Figure 12) is a standalone device that determines the trajectory of a mobile means knowing its accelerations and angular velocity.

Figure 12 – The inertial platform MTi

The inertial navigation determines the position and the attitude of a mobile vehicle through a system of inertial sensors, accelerometers, gyroscopes and magnetometers. Known accelerations and angular velocity of the body, the successive integration allows determining position, velocity and attitude of the moving vehicle.

The inertial navigation can be applied in different fields of study:

- *Automotive,* for tests of handling, maneuverability, load analysis and structural optimization, crash test, reconstruction of accidents;
- *Transport,* for monitoring driving conditions and road transport, for optimizing the control system of trains, where the system conserves the function independently by the presence of tunnels or vegetation conditions;

- *Avionics/Aerospace* for navigation systems, attitude control of satellites, autopilot systems for Unmanned Aerial Vehicles (UAVs);
- *Military and remote control,* for supporting the navigation of underwater vehicles, intelligent robotic systems, inertial guidance of torpedoes, missiles, Remotely Operating Vehicles (ROVs), flight systems remotely controlled;
- *Marine applications,* for control systems, for stabilizing and monitoring for boats, active systems for increasing comfort on board boats and oceanographic buoys, for the remote monitoring of wave motion (wave meters);
- *Biomedical,* for the surgery precision;
- *Logistics,* for monitoring the path of the straddle-carriers in commercial ports, or in general, trade flows, for the implementation of systems traceability and archiving routes of products.

An inertial navigation system consists essentially of two parts: an Inertial Measurement Unit (IMU), which is the section that contains the system of inertial sensors; and a navigation computer that is the system with integration of algorithms and processing of data acquired by IMU.

The inertial navigation system presents many advantages, the most important of which are:

- Jamming immunity. It does not receive or transmit radiation and does not require external antennas detectable by radar, allowing it to use for military applications;
- Autonomy of inertial navigation;
- Suitability for driving, controlling and integrating the navigation of the vehicles on which it is installed;
- Low power consumption.

The inertial navigation system has different error sources affecting the performances:

- Presence of a noise at the output of NSI, generated by the NSI electronic components and which overlaps the signal;
- Problems related to bias, that is not anything out of the sensors when the inputs are zero;
- Errors on the scaling factor (constant transduction), often caused by manufacturing tolerances;
- Non-linearity, caused by the sensors;
- Need of a coordinate transformation from the system "body";
- Request of reconstructing the change of trim in time respect to the initial. Moreover the reference system of the "body" is not inertial, so it is necessary to eliminate the contribution of the

gravity and centrifugal acceleration of the Earth and the Coriolis acceleration.

In most applications, an integrated INS/GPS system (Figure 13) is generally used to increase the performance of the INS thanks to the GPS and reduce the overall cost. By combining GPS and INS, deficiencies in both systems can be overcome. The idea is to have regular absolute position fixes, using GPS, and to track position in interim using INS. In this situation the GPS provides short term accuracy, while the INS provides long term stability, complementing each other and producing a sustainable navigational position. The outputs of both systems are compared and suitably filtered, and corrections are made to either or both systems in consequence.

Inertial platform MTi with integrated GPS

GPS Antenna L1/GLONASS for MTi platforms

Figure 13 – Examples of integrated INS/GPS systems

A widely quoted filter for this task is constituted by the Kalman filter, which combines two estimates and provides a weighted mean, using factors chosen to yield the most probable estimate. By adding INS capability to a GPS navigation system, considerable improvements have been observed. Some experiments of particular interest regard a GPS/Dead-reckoning system tested in different big urban areas [*Shair et al.,* 2008]. When tall buildings reduce satellite availability, the stand-alone GPS use three satellites, assuming constant height from previous readings. This produces large inconsistencies where hills are encountered as well. However, the INS-based GPS/DR system overcomes this problem by only taking GPS fixes when full accuracy is attainable (i.e. four satellites in view). Meanwhile, the DR keeps a very good estimate for the position.

The Active Beacon Localization System

The Active Beacon Localization System belongs to the localization methodologies with absolute measurements. This system computes absolute location by measuring the direction of incidence (or the distance to) three or more active beacons. Transmitter locations must be known in inertial frame evaluation (velocities and accelerations).

There are two different types of active beacon systems: Trilateration and Triangulation.

The Trilateration system consists in determining vehicle's pose based on distance measurements to known beacon sources. In the usual configuration, tree or more transmitters are mounted at known locations in the environment and one receiver on board the robot. Usually the sensors used for the calculation of the distance from tree or more points are the GPS and ultrasounds [*Peca,* 2009].

The triangulation system (Figure 14) consists in determining the vehicle's pose (x_0,y_0,θ) on the base of the evaluation of the angles λ_1, λ_2 and λ_3 between the robot longitudinal axis and the direction with which 3 beacons installed in the environment at known positions are detected.

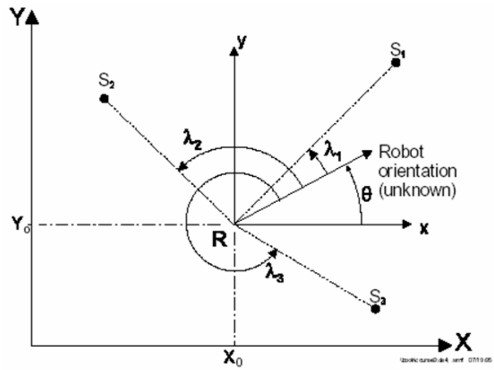
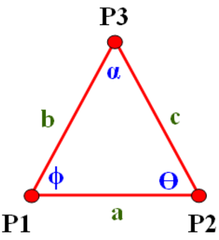


Figure 14 – Triangulation system to determine the vehicle pose

In *2D Triangulation*, the distance from an object can be calculated by simple geometric formulas:



GIVEN

- The lenght of side of a triangle (***a***);
- Two angle of the triangle (***φ***, ***θ***)

IT IS POSSIBLE TO DETERMINE

- The lenght of side of the remaining sides(***b***, ***c***);
- The remaining angle (***α***)

$$b = a\frac{\sin\theta}{\sin(\phi+\theta)}$$

Landmark navigation

The landmark navigation belongs to the absolute localization methodologies for mobile robots. Landmarks are located in known environment places, or they are detected in the environment. Landmarks are distinct features that a robot can recognize from its sensory input. Landmarks can be geometric shapes (e.g., rectangles, lines, circles), and may include additional information (e.g., in the form of bar-codes). In general, landmarks have a fixed and known position, relative to which a robot can localize itself.
Landmarks are carefully chosen to be easy to identify; for example, there must be sufficient contrast to the background. Before a robot

can use landmarks for navigation, the characteristics of the landmarks must be known and stored in the robot's memory. The main task in localization is then to recognize the landmarks reliably and to calculate the robot's position.

In order to simplify the problem of landmark acquisition it is often assumed that the current robot position and orientation are known approximately, so that the robot only needs to look for landmarks in a limited area. For this reason good odometry accuracy is a prerequisite for successful landmark detection.

The general procedure for performing landmark-based positioning is shown in Figure 15, where sensors are used to sense the environment and then extract distinct structures that serve as landmarks for navigation in the future.

There are two types of landmarks: "artificial" and "natural." It is worth noting that “natural” land-marks work better in highly structured environments such as corridors, manufacturing floors, or hospitals. Indeed, “natural” landmarks work better when they are actually man-made (as in the case of highly structured environments). Therefore natural landmarks are those objects or features that are already in the environment and have function different form robot navigation; artificial land-marks are specially designed objects or markers that need to be placed in the environment with the sole purpose of enabling robot navigation.

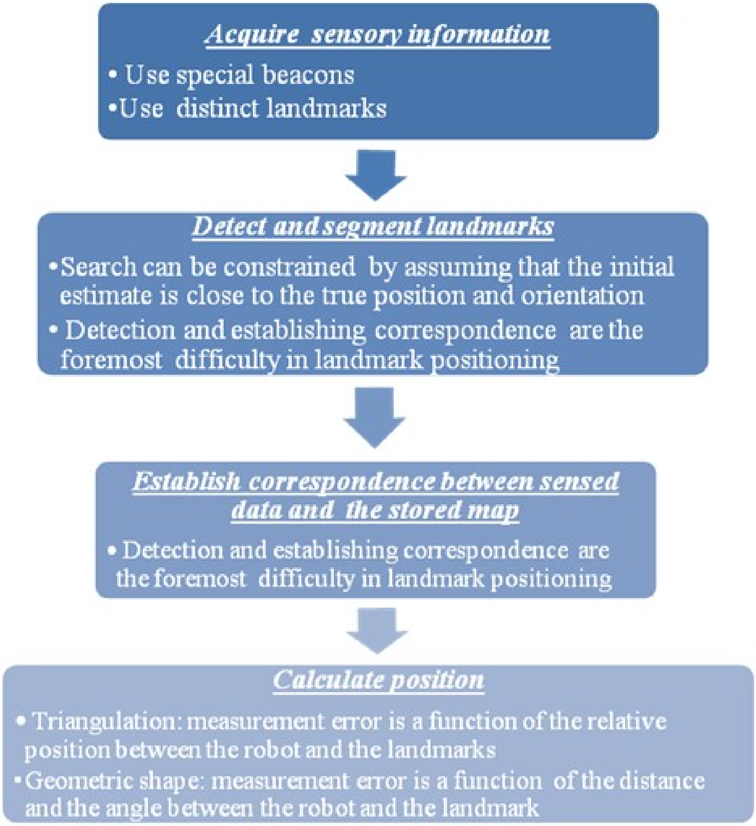


Figure 15 – General procedure for landmark-based positioning

The main problem in natural landmark navigation is to detect and match characteristic features from sensory inputs. The sensor of choice for this task is computer vision. Most computer vision-based natural landmarks are long vertical edges, such as doors and wall junctions, and ceiling lights.

When range sensors are used for natural landmark navigation, distinct signatures, such as those of a corner or an edge, or of long straight walls, are good feature candidates. The selection of features is important since it determines the complexity in feature description, detection, and matching. One system that uses natural landmarks has been developed in Canada and aims at developing a sophis-ticated robot system called the "Autonomous Robot for a Known Environment" (ARK), [*Jenkin et al.,* 1993].

Detection is much easier with artificial landmarks [*Atiya and Hager,* 1993], which are designed for optimal contrast. In addition, the exact size and shape of artificial landmarks are known in advance. Size and shape can yield a wealth of geometric information when transformed under the perspective projection. Researchers have used different kinds of patterns or marks, where the geometry of the method and the associated techniques for position estimation vary in consequence [*Talluri and Aggarwal,* 1993]. Many artificial landmark positioning systems are based on computer vision [*Livatino et al.,* 2009].

1.4. Localization

In mobile robotics, the localization problem consists in identifying the position of the robot and its orientation (in then you get call pose) within the work environment. Indicating with x_t the pose at time t, with u_t the readings provided of the odometry and with z_t those provided by the proximity sensors, the problem can be formalized as the evaluation of the following function density of probability:

$$p\,(x_t \mid u_{1:t}, z_{1:t}) \tag{10}$$

In literature there are several techniques to solve the problem of localization [*Dellaert et al.,* 1998, *Fox et al.,* 2000, *Grisetti et al.,* 2002, *Austin and Jensfelt,* 2000]. In the planar case the complexity of the proposed techniques increases linearly with the size of the environment.

The problem of localization can be correlated to other sub problems as position tracking, global localization and the kidnapping problem.

The position tracking and global localization problems include two assumptions:

- the position at the previous instant of the robot is known;
- at each instant the distribution of poses is monomodal, that is only one maximum is present.

These assumptions have two effects: (i) the need of knowing the position in the previous step limits the search space of possible future positions and establish an unique configuration of departure, and (ii) the distribution of poses should be represented in a canonical form, for example by the mean and the variance of a Gaussian.

Knowing the robot position at the previous instant determines also that an error in a step propagates in successive estimates, and then once the robot loses the location is extremely unlikely to fix it.

In the global localization problem the position of robot can be estimated even under assumptions of global uncertainty, thanks to the absolute position within the environment.

The kidnapping problem occurs when the robot moves within the environment with no shifts registered by the sensors, for example as a result of a manual movement by a human operator. It is commonly used to test the robot ability to recover from catastrophic failures of localization.

Position tracking

Consider that at time $t-1$, the pose of the robot is known with an error of e_{t-1}. Then Equation (10) can be reformulated as:

$$p\,(x_t \mid x_{t-1}, u_{1:t}, z_{1:t}) \tag{11}$$

Knowing the position at time *t-1* for the calculation of the position at time *t* determines that, in case of a big error in the estimate of $x_{t-1},$ it propagates also to x_t. So, once the robot has lost its position, it is extremely unlikely to restore it.

Global localization

The problem of global localization can be solved with different approaches available in literature to estimate the probability function density (PDF) of Equation (10). A typical stochastic method is the Bayes filter, with the mathematical tools, including the Particle filter [*Gordon et al.,* 1993], the Kalman filter [*Kalman,* 1960], and the Extended Kalman filter [*Leonard and Durrant-Whyte,* 1992; *Williams et al.,* 2002].

The filtering problem can be expressed as the estimate of the state x of a discrete dynamical system.

Given the available measurements, the Bayesian filter estimates the evolution of the system state with the following PDF:

$$\mathrm{p}(x_{0:t}|z_{1:t}) \tag{12}$$

Usually it is interesting to calculate the marginal density of the current state, also called distribution filtered:

$$p(x_t|z_{1:t}) \qquad (13)$$

Nevertheless, in most practical applications, the observed process is Markovian, i.e. the current state incorporates all past observations. In probabilistic terms this means that the current observation is stochastically independent of past observations given the current state [*Doucet et al.,* 2001]:

$$p\ (z_t \mid x_t,\ z_{1:\ t-1}) = p\ (z_t \mid x_t) \qquad (14)$$

Among all mathematical tools that can be used for the Bayesian filter, the most famous and widely used is known as Kalman filter. This filter takes its name from R.E. Kalman that in 1960 published a famous article on recursive filtering of discrete systems linear [*Kalman,* 1960]. An interesting introduction to the Kalman filter can be found in *Welch and Bishop* [2001].

The Kalman filter consists of a set of mathematical equations that implement the probability density function of the Bayesian filter with the intake of Markov through the two stages of prediction and update.

The filter is considered good when minimizes the error covariance estimated but the system must be linear and with Gaussian noise. Since the system is usually not linear, in order to use the Kalman filter, the system should be linearized. In this case the filter is called Extended Kalman Filter (EKF). EKF is used in the SLAM even if the computational complexity grows in proportion to the square of the feature number. Some algorithms have been proposed to reduce the computational complexity of EKF-SLAM [*Yong-Ju Lee et al.,* 2007]. EKF is used for the mobile robot localization, when it reduces to a problem of filtering, in which the map is composed in $\{m_1, . . . , m_K\}$ position landmarks, the state vector is the pose of the robot, and the observation vector consists of the landmark positions observed by the robot in its reference system.

The solution to the problem of localization can be derived in a simple way if the landmark can be identified unambiguously (data association problem). Otherwise the problem cannot be resolved and tracking techniques with multiple hypotheses that exceed the limitations of the monomodality of the Kalman filter should be used.

Another way to use the Kalman filter is the forward observation model $p(x \mid z)$ that returns a set of possible positions compatible with current observations. For example in [*Gutmann et al.,* 2001, *Cox,* 1991], the pose of the robot is evaluated through the data matched in proximity to the geometrical environment map. In particular, *Iocchi and Nardi* [1999] use a matching method to efficiently estimate the position in the parameter space of the Hough given the observations. The major limitation of these techniques lies in the need to have a unique landmark or a non-ambiguous observation model. Unfortunately, this it is not possible in the environment with strong symmetries.

There are several approaches to solve the problem of multimodal location. A first approach uses a Kalman filter Multi-Hypothesis [*Chen and Liu,* 2000], which consists of a Bayesian filter where the probability distribution is represented as a set of Gaussians. Each Gaussian is updated through a Kalman filter, while a external Bayesian framework (for example a particle filter or a kalman filter) is used to estimate the weight of the mixture. In *Austin and Jensfelt* [2000] a series of Kalman filters are drawn from a discrete Bayesian filter, which creates and deletes the individual hypotheses represented by Kalman filters depending on the their evolution.

One of the first methods of global localization is the Markov localization [*Fox et al.,* 1998]. This method can be formulated exactly as a Bayesian filter, in which the position of the robot in the plane is represented through a three-dimensional grid, and consequently also the map. Like a Bayesian filter standard, the

Markov localization algorithm works in two steps, prediction and observation.

Markov localization is one of the simplest and most robust methods, but it has high computing costs and it is unstable in dynamic environments.

Another approach for the mobile robot localization uses particle filters. A particle filter is a nonparametric implementation of the Bayes filter and it is frequently used to estimate the state of a dynamic system. In a particle filter, the distribution is computed recursively using Monte Carlo simulations [*Arumampalam et al.,* 2001]. The basic idea is to represent a posteriori the PDF using a series of samples with a weight associated and to calculate the estimate through the samples and weights. Obviously this model represents an approximation of the effective density, but increasing the number of samples the Monte Carlo representation tends to the actual density and the particle filter tends to the excellent Bayesian filter. This particle filter is called SIR (Sampling, Importance weighting, Resampling), and can be summarized with the following steps:

- *Sampling:* Create the next generation of particles on the basis of the previous set of samples. This step is also called sampling or drawing from the proposal distribution;

- *Importance weighting:* Compute an importance weight for each sample in the set at time *t*;
- *Resampling:* Draw N samples from the set at the *t* time, hence the likelihood to draw a particle is proportional to its weight.

Particle filters estimate the state of dynamic systems from sensor information. However, in a lot of real time applications, sensor information arrives at a significantly higher rate than the update rate of the filter. The prevalent approach to dealing with such situations is to update the particle filter as often as possible and to discard sensor information that cannot be processed in time.

Some examples of real-time particle filters have been also proposed, which make use of sensor information even when the filter update rate is below the update rate of the sensors. This is achieved by representing posteriors as mixtures of sample sets, where each mixture component integrates one observation arriving during a filter update. The weights of the mixture components are set so as to minimize the approximation error introduced by the mixture representation. Thereby, the approach focuses computational resources on valuable sensor information. Experiments using data collected with a mobile robot show that the approach yields strong improvements over other approaches [*Kwok et al.,* 2004].

Particle filters are frequently used to solve the simultaneous localization and mapping problem [*Martinez-Cantin et al.,* 2006].

1.5. Map-based positioning

Map-based positioning, also known as "map matching," is a technique where the mobile robot uses a variety of information coming from the sensors installed on board to build a map of the environment where it is operating. This map is compared with a model pre-loaded in the navigation system and if there is a match is calculated by his pose (Figure16).

This approach has several advantages [*Borenstein et al.,* 1996]:

- This method uses the naturally occurring structure of typical indoor environments to derive position information without modifying the environment;
- Map-based positioning can be used to generate an updated map of the environment;
- Map-based positioning allows a robot to learn a new environment and to improve positioning
- Accuracy through exploration.

Disadvantages of map-based positioning are dependent of the specific requirements for satisfactory navigation, in particularly:

- There is enough stationary, easily distinguishable features that can be used for matching;
- The sensor map is enough accurate (depending on the tasks) to be useful;
- A significant amount of sensing and processing power is available.

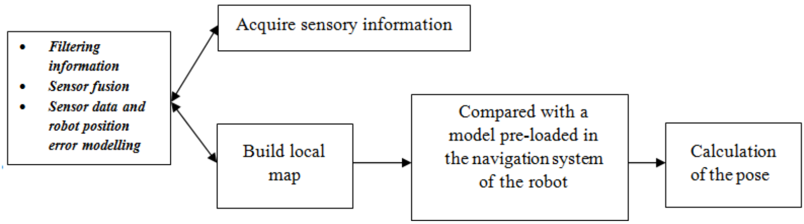


Figure 16 – General procedure for map-based positioning

The first step of map-based positioning, called Map Building, can be traced back to the problem of SLAM in which characteristics of the environment are extracted by the data collected from the aboard the robot.

The second step, called Map Matching, establishes the correspondence between the map generated by the robot with the map-building and map stored in its navigation system.
The algorithms for the map matching can be classified as either icon-based or feature-based.
Schaffer et al.[1992] summarized these two approaches: *"Iconic-based pose estimation pairs sensory data points with features from the map, based on minimum distance. The robot pose is solved for that minimizes the distance error between the range points and their corresponding map features. The robot pose is solved [such as to] minimize the distance error between the range points and their corresponding map features. Based on the new pose, the correspondences are recomputed and the process repeats until the change in aggregate distance error between points and line segments falls below a threshold. This algorithm differs from the feature-based method in that it matches every range data point to the map rather than corresponding the range data into a small set of features to be matched to the map. The feature-based estimator, in general, is faster than the iconic estimator and does not require a good initial heading estimate. The iconic estimator can use fewer points than the feature-based estimator, can handle less-than-ideal environments, and is more accurate. Both estimators are robust to some error in the map."*

The potential fields approach was introduced to the field of navigation by *Khatib* [1979] around 1979, and it is based on the metaphor that the goal should attract the mobile robot towards it, and that the obstacles should repel the agent from them. Combining these two forces upon the robot produces a net force that (in theory) moves the robot towards the goal and away from obstacles simultaneously (Figure 17).

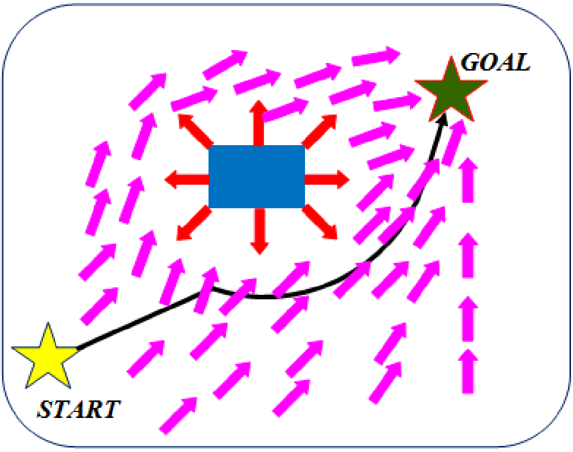


Figure 17 – Example of a potential field approach. Purple arrows represent the long range attractive effect of the goal. Red arrows represent the short range repulsive effect of an obstacle. The black line indicates the path taken by the agent in response to the combination of these two effects

Then for the potential field approach, a potential for the goal and the obstacles should be defined:

$$\begin{aligned} U_{Goal}(q) &= \alpha * dist(q, Goal)^{2} \\ U_{Obstacle}(q) &= \beta * dist(q, Obstacle)^{-1} \end{aligned} \tag{15}$$

The total potential is the sum of the potential of the goal plus the sum of the individual potential obstacles:

$$U(q) = U_{Goal}(q) + \sum U_{Obstacle}(q) \tag{16}$$

The artificial force produced on the mobile robot is:

$$F = -\nabla U(q) = -\begin{pmatrix} \frac{\vartheta U}{\vartheta x} \\ \frac{\vartheta U}{\vartheta y} \end{pmatrix} \tag{17}$$

Chapter 2

The GIS environment

Geographic Information Systems (GIS) are information technologies that have transformed the way to manage all geographic data necessary for all human activities.

This chapter will introduce GIS technology, paying particular attention to the characteristics that identify GIS as extension of the classical topography, and the platforms that allow using, developing and customizing the functionalities of this technology.

All human activities need to place information in a geographical context. Many of these activities concern with the recording and planning of human-made environments, with the monitoring and managing of natural environments, with monitoring and managing transport and navigation systems, and with understanding social structures.

The GIS technology can be considered an engineering work, well defined by an architecture (Figure 18) that is designed, built and tested in all its components.

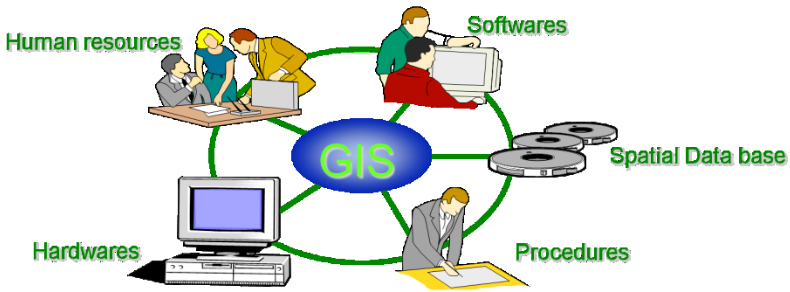


Figure 18 – Schematic representation of the complete architecture of a GIS

The *hardware components* within the architecture of a GIS (Figure 19) allow the analysis, visualization and sharing of the spatial data. Examples of this category are computers, servers, display devices such as monitors, printing devices, satellite positioning sensors such as GPS, computer networks, etc. These devices suffer from the technological development then hardware of a GIS is evolving.

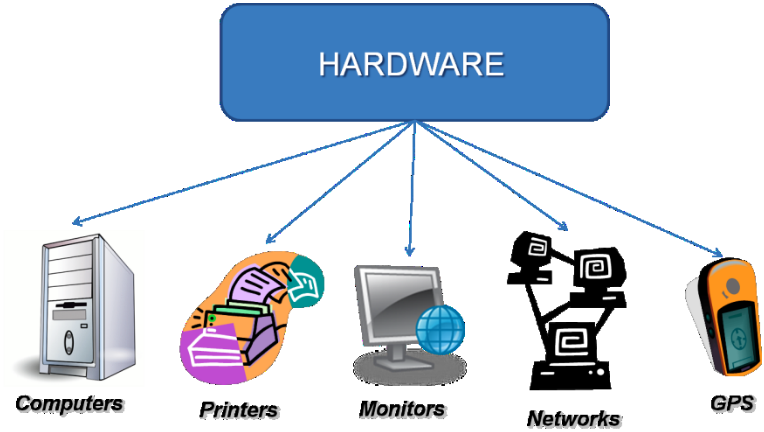


Figure 19 – Hardware components of GIS architecture

The *software* used in GIS technology allows managing all information concerning all GIS components. The software components can be divided into three categories (Figure 20): base software, GIS software and software for territorial applications.

The base software manages the individual hardware components that compose the GIS architecture, i.e. operating systems, networking software, software for printers, scanners, topographic instruments.

In recent years, it is spreading more and more the use of free and open source software. This software is identified by the acronym FOSS (Free and Open Source Software) and is licensed under GPL (General Public License). This license grants the recipients of a computer program the rights of the free software definition and uses copyleft to ensure the freedoms are preserved, even when the work is changed or added to.

This technology is diffused between the GIS software, so much that a community called OSGEO [http://www.osgeo.org/] has been created for supporting the collaborative development of open source geospatial software, and promote its widespread use.

Successively other worldwide satellite communities of developers are born. In Italy, GFOSS (Geospatial Free and Open Source Software) [http://www.gfoss.it] brings in a large community, people from all Italian regions, active in the development and testing of free and open source GIS software in the commercial sector, university and research centers and public administration. Italy has a community of developers and users of free software geographic particularly important since it is the most numerous and active in the world.

The software for territorial applications deals with data enriched by spatial information and software for the development of GIS applications. The software platforms that use the functionalities typical of the GIS are divided in Desktop GIS and WebGIS. This latter permits the distribution on the web of all data and tools of the GIS:

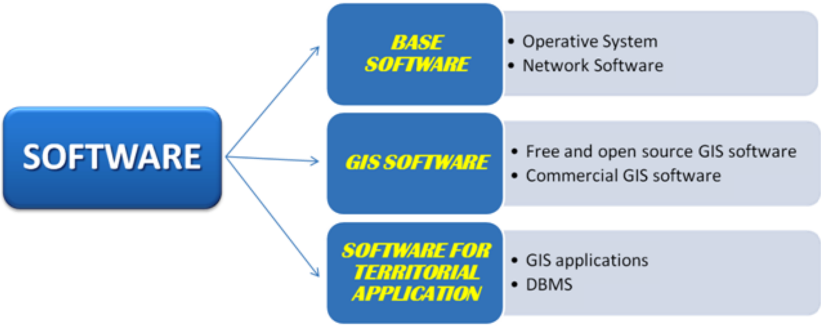


Figure 20 – Software of a GIS architecture

The data managed and used within the GIS architecture (Figure 21) are enriched by spatial characteristics then georeferenced in the territory. These data can be divided into two main categories belonging to two distinct types of numerical cartography, differenced for the specific characteristics and oriented to specific purposes of use. These two main categories are raster and vector data.

The first category represents the cartographic support used in GIS environment, and includes satellite images, orthophotos, etc.

The second category comprises geometric primitives (points, lines, areas) that allow computerizing, interconnecting and representing every element of the territory as towns, road infrastructures, network services, natural phenomena, etc.. To this aim, a database of support is used for the storage and management of the spatial data.

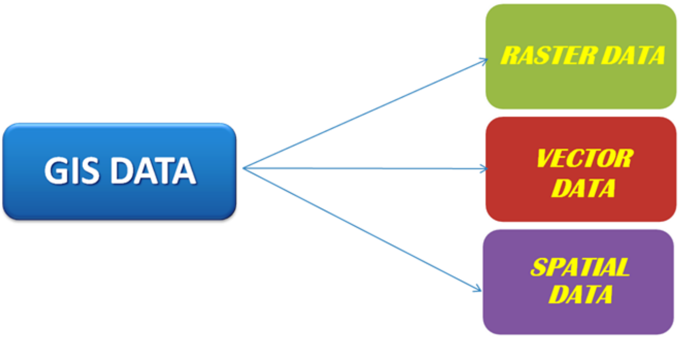


Figure 21 – The different types of GIS data

All types of data within the GIS platform are handled, displayed and shown with a structure of overlapping layers (Figure 22), in which each layer can be used and represented separately:

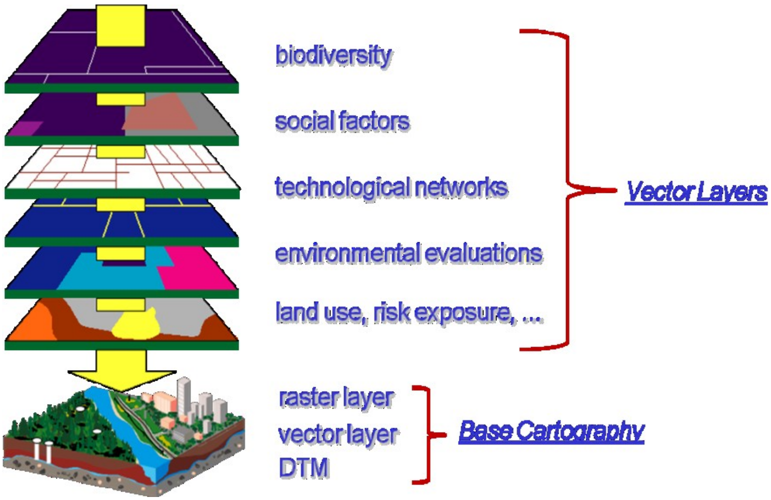


Figure 22 – Data within the GIS platform organized in separate layers

The coordination between the different human resources involved in the design, development and implementation of a GIS application is a key component of GIS architecture.

Usually people involved in all the steps leading to the creation and use of GIS (Figure 23) are: GIS technicians, programmers, managers, users, which contribute with their own specific knowledge to the development of the application.

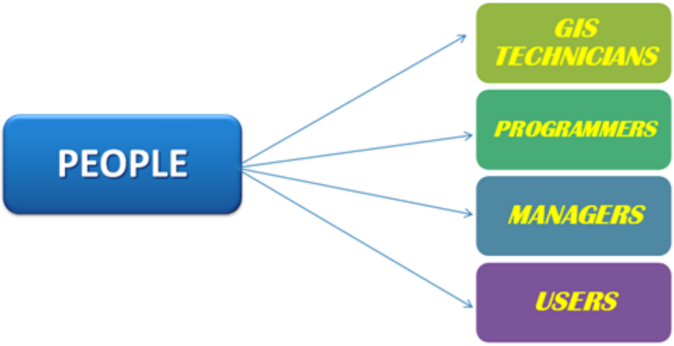


Figure 23 – People involved in the creation and use of GIS

For example, in the generic development of a GIS application, the manager determines the objectives to be attained, the GIS expert chooses the GIS data, functionality and software to use, the computer expert selects the hardware and implements on the computer the informatics procedures, and finally the GIS platform with the developed application is provided to the user.

Figure 24 shows the main steps for planning a GIS where, if the quality check fails, the objectives must be redefined. Otherwise it represents the goals

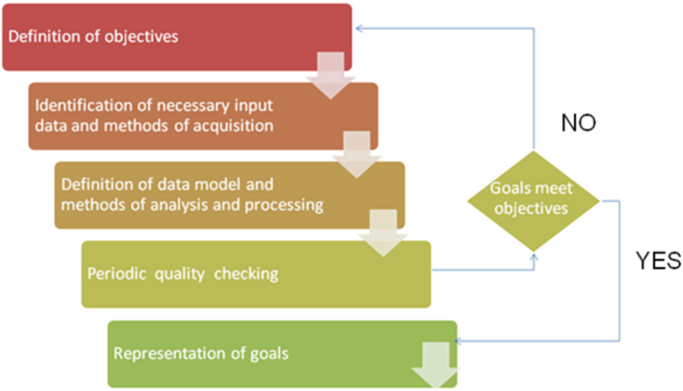


Figure 24 – Scheme of a generic development of a GIS application

GIS application is an engineering work which follows some rules for the architecture and implementation. These rules are: guidelines, specific, standard, procedures (Figure 25). So you have to refer to the regulations, in the application procedures, to the standard realization and detailed technical specifications.

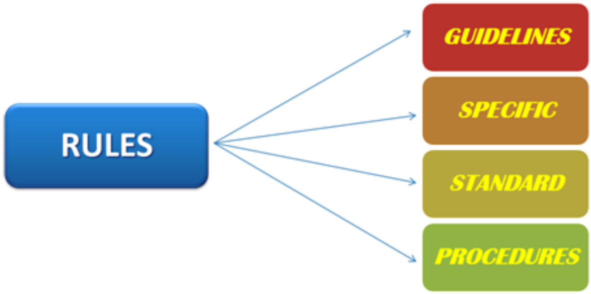


Figure 25 – Rules for GIS application

GIS technologies allow the acquisition, analysis, representation and exportation of the georeferenced spatial data with rigorous topographic approaches (Figure 26).

From these assumptions, it is evident that the GIS environment is much more advanced compared to a CAD (Computer Aided Design), due to some differences:

- CAD can manage and edit the numeric base cartography, but only from the "graphic" point of view;
- In CAD it not possible to bind data to the elements;
- Some typical DB functions are lacking: queries, selections, crossovers, etc.;
- Some type of spatial analysis cannot be performed.

Figure 26 – Procedures provided by GIS

The GIS software is much more than a CAD due to its capability to model phenomena (e.g. propagation of toxic clouds), to define optimal path; to perform spatial analysis (overlay, buffering, union…), to create thematic maps.

In summary, the GIS technology is an integration of numerous software environments: Database Relational Management Systems (DBMS), Computer Aided Design (CAD) and graphics software; Packages for Statistical Analysis and Reporting, as shown in Figure 27.

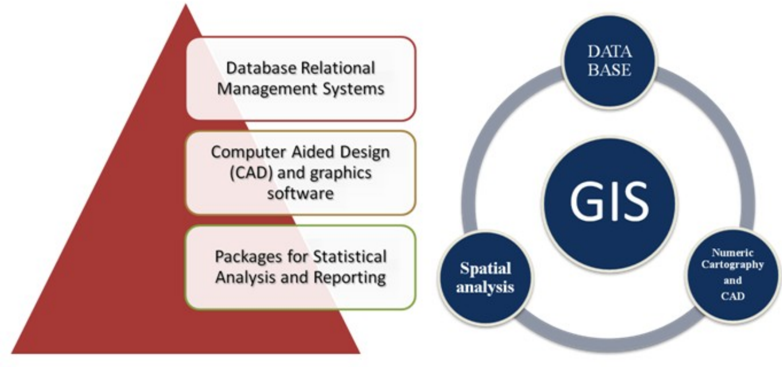


Figure 27 – Different software environments integrated in the GIS technology

For these characteristics, the GIS environment is well suited for the analysis of many local resources as technological networks, land use, transport infrastructure, environmental assessments, social factors, so it can be considered a valuable tool for decision support (Figure 28).

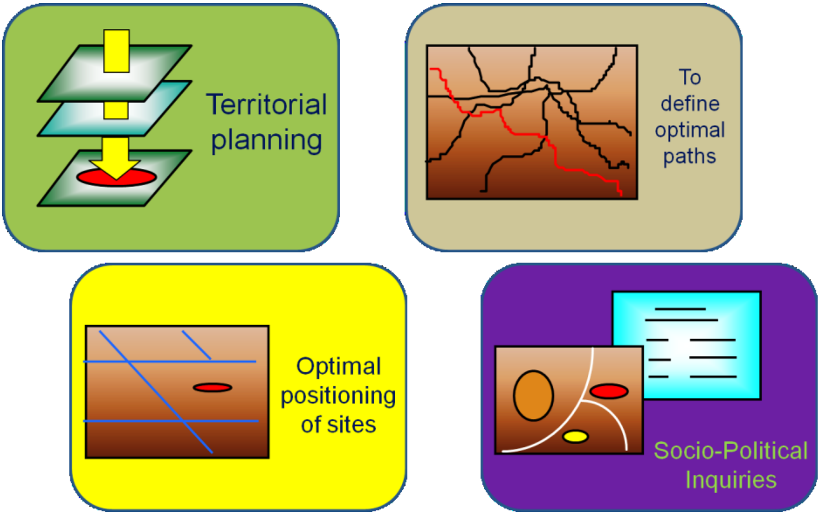


Figure 28 – Some examples of the use of GIS as a tool for decision support: planning the territory, determining optimal paths, defining optimal positioning of sites, and socio-political inquiries

As the range of information can be placed in a geographical context is almost infinite, there are in principle few limits to the variety of possible applications of GIS. This is reflected in a continuing growth in GIS usage across many disciplines [*Stefanini*, 2006]:

Topography and land survey: typically used for monitoring landslides, geo-referencing of a topographic survey, used for intervisibility analysis and as the basis for an analysis of terrain features such as drainage basins [*Giordano et al.,* 2003], [*Famoso et al.,* 2012], or for the evaluation of seismic risk [*Maugeri et al.,* 2009,2010], [*Mussumeci et al.,* 2004]. Information systems that store general-purpose topographic data combined with administrative and land ownership boundaries are often referred to as Land Information Systems (LIS). There is some debate about the distinction between GIS and LIS. *Dale* [1991] suggests that LIS combines institutional and human resources with the technology characterized by GIS, and that the term LIS is in any case in more general usage than GIS in most parts of the world, except the USA and the UK.

Geological surveys: The full potential of a GIS for geological data handling can be realized only with the introduction of flexible 3D tools for modeling, interpretation and visualization. The GIS environment and WebGIS are used to represent a path of deepening geological and geomorphological issues [*Barbieri et al.,* 2008], or for the creation of thematic maps as hazard maps relating to phenomena of collapse of rock walls [*Del Maschio et al.,* 2004].

Soil surveys: In this discipline, the GIS support may be referred to as a soil information system and can be expected to provide facilities for digitizing soil sample points and integrated soil boundaries from maps, for storing the data in a database, for interpolating and re-classifying soil attributes, and for overlaying and plotting these data with other types of data relating for example to terrain models and climatic zones.

Biosphere surveys: GIS technology is used for environmental monitoring focused on the biosphere. In particular for the analysis of the effects of polluting factors on the environment are created, in GIS environmental, ecological database that are integrated with information derived from remote-sensing surveys (especially satellites such us Landsat and Spot), and with information derived from ground surveys of flora and fauna and physical and geochemical environmental parameters.

Navigation: In this sector, GIS is used for many purposes, providing a georeferenced cartographic support of different information from vehicles whose position is known from GPS sensor installed on board. It is also used to plan routes, determine optimal routes to be provided in vehicles or to define flight plans for the navigation of aircraft [*Mangiameli et al.,* 2012]. The GIS environment is an excellent technology to support navigation of mobile robots in outdoor environment (e.g. urban environment) as it uses a georeferenced cartographic support such as satellite images. It also allow building topological maps that contain semantic information such as the width or type street, necessary to determine a path so that the robot can reach its target [*Yu-Cheol Lee et al.,* 2011].

In the following paragraphs, the topographic approach typically used in GIS technology will be treated together with the desktop GIS and WebGIS platforms.

2.1 Reference systems and cartography

The topographic approach is a fundamental component of the GIS environment, thanks to its capability to simultaneously manage data in different projection systems, as Gauss Boaga or UTM.
A coordinate system is a reference system used to represent the locations of geographic features within a common framework. Generally GIS allows dealing with two types of coordinate systems: projected coordinate systems and geographic coordinate systems. Moreover most GIS environments provide additional coordinate systems that can be defined by user (Figure 29). Even if coordinate

systems are complicated topic in GIS, they form the basis for how a GIS can store, analyze, and display sets of spatial data that come from different sources.

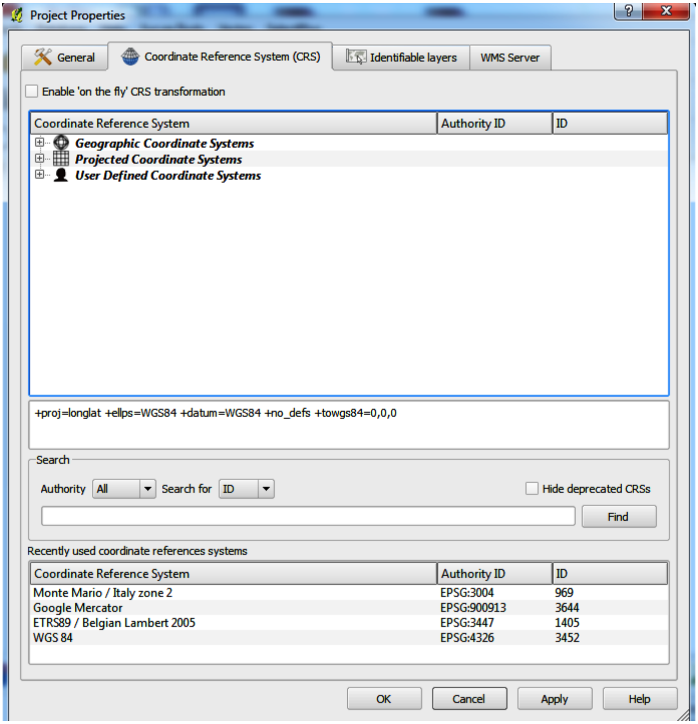


Figure 29 – Screenshot of the graphical interface within the desktop platform GIS QGis, for setting the coordinate system

Projection systems refer to the process of representing the earth's round surface as flat and are always based on a geographic coordinate system. There are many different types of projections, and each one distorts the representation of the earth in some way. When selecting a projection, it is important to be aware of what distortions are in place since that certain types of distortions are more acceptable for some purposes than others. Examples of projections are the Lambert Conformal Conic and the Transverse Mercator.

Geographic coordinate systems use a three-dimensional spherical surface to represent the Earth and define locations on it, which are referenced by longitude and latitude values.

The science that deals with the problem of representing the Earth's surface on a flat plane is the cartography. An important branch of cartography, the geodesy studies the Earth shape and how it is related to its surface's features.

Because the Earth is not completely round, it is not possible to base a coordinate system on a perfect sphere. The main reference surfaces established to approximate the shape of the Earth are the Geoid and the ellipsoid.

The surface of the Geoid (Figure 30) is obtained by extending the mean sea level below the land surface [*Galetto and Spalla*, 1998]. Its main characteristics are:

- The real shape of the Earth is represented like an ideal solid used as a reference surface;
- Surface passing through the mean sea level;
- Equipotential respect to the gravitational force field;
- Orthogonal to the vertical direction at every point.

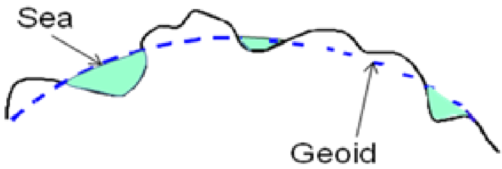


Figure 30 – Representation of the geoid surface obtained extending the mean sea level below the land surface

The equation of the geoid can be written as

$$W(X,Y,Z) = V(X,Y,Z) + U(X,Y) = \text{constant} \tag{18}$$

where V(X,Y,Z) is the attractive Newtonian potential and U(X,Y) is the potential energy of the centrifugal force. Since the distribution of masses inside the Earth is not uniform, it is not possible to calculate the potential V and therefore to use the geoid for the planimetry. On the contrary, the geoid can be used to determine the altitude. In fact, being the geoid equipotential with respect to the gravitational force field, the altitude of a point on the Earth's surface can be determined in relation to mean sea level using the normal direction to the geoid.

The Ellipsoid is a solid generated by an ellipse around its minor axis (Figure 31). In reverse of the geoid, the ellipsoid cannot be used for the construction of the altimetry, but for the planimetry.

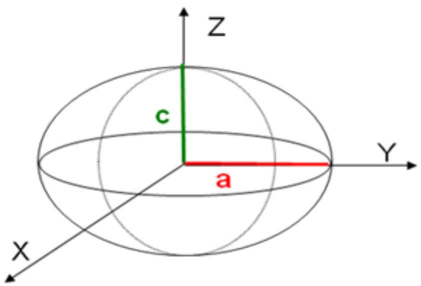


Figure 31 – Representation of the ellipsoid, where c is the minor axis and a is the major axis

The main characteristics of the ellipsoid are:

- Simple mathematical equation to be described (i.e. equation of the ellipse);
- Representation of the real shape of the Earth;
- Correspondence between the points of the terrestrial surface with those of the ellipsoidal surface;
- Vertical direction not normal to the ellipsoidal surface.

Another important concept of the topographic approach is the datum. The datum is a set of reference points on the Earth's surface against which position measurements are made and an associated model of the Earth’s shape (reference ellipsoid) to define a geodetic coordinate system. Horizontal datums are used for describing a point on the Earth's surface, in latitude and longitude or another coordinate system. Vertical datums measure elevations or depths.

Popular datums are NAD 1983 (North American Datum; based on the GRS 1980 spheroid) and WGS 1984 (World Geodetic System; based on the WGS 1984 spheroid).

The WGS84 datum, which is almost identical to the NAD83 datum used in North America, is the only world referencing system in place today. WGS84 is the default standard datum for coordinates stored in the commercial GPS units.

Since 90's, GPS has advanced not only for static or cinematic topographic surveys, but also for mobile mapping and for mobile navigation systems as the mobile robots. This lead to an optimization of the WGS84 system that consists of a new more rapid, more productive and more economic structure. This structure represents a Global Navigation Satellite Systems (GNSS) that is made up of different networks of permanent stations constituted by single receivers permanently operant and distributed on the territory.

The GNSS is based on three main satellite technologies: GPS, Glonass and Galileo. Each of them consists mainly of three segments: pace segment, control segment and user segment. These segments are almost similar in the three satellite technologies, which are all together make up the GNSS.

The real-time interaction among permanent networks provides a correction service that is promptly provided to the user, without the need of postprocessing the data. This kind of architecture is called NRTK (Network Real Time Kinematic).

As for Italy, on 10 November 2011 a Ministerial Decree was published which adopted a new geodetic reference system called ETRF2000 (2008.0). The previous reference systems, still in use in Italy, are the Roma40 (better known as Gauss-Boaga and often called MonteMario), the ED50 (European Datum 1950, often called only

UTM) and ETRF89 (WGS84). This new geodetic reference system was materialized in a set of permanent stations called RDN (Rete Dinamica Nazionale).

2.2 Numerical cartography

The support of one or more basic cartographies is essential for GIS technology. Given the use of computer system, this support should be necessarily of numeric type, and in raster or vector format.

The raster cartographic support is constituted by a "cell array"; the elementary cell is called pixels (picture element) and is characterized by a specific attribute, expressed by a binary code number (number of bits) to which a specific color is associated.

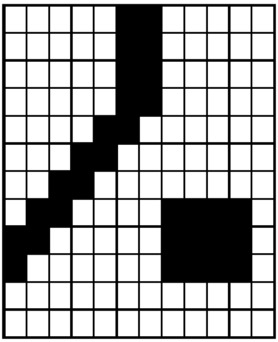
Each pixel is then uniquely identified by the relative position of the line number (no. row) and the column number (no. column) in the reference system image, and a value associated therewith.

The values that can be associated with each pixel are reported in Table 1. Examples of how the number of bits affects the resulting raster images are shown in Figures 32 and 33.

The most common formats for raster image are uncompressed (TIFF, BMP, PCX), compressed (GIF, JPG), and formats specific for mapping and GIS (ERDAS, GeoTIFF, BSQ, BIL, BIP).

PIXEL VALUE	DESCRIPTION	NUMBER OF BITS
0-1	white or black images	1bit 2^1=2 values
0-255	monochrome images (grayscale)	8 bit 2^8=256 values
0-255	Colors images	8 bit 2^8=256 values
(0-255)×3	3 scales to 8 bits for each R, G, B colors	24bit 2^{24}=16.777.216 values

Table 1 – Possible pixel values for raster images



1	1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	0	0	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	0	0	1
0	0	1	1	1	1	1	0	0	0	0	1
0	1	1	1	1	1	1	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1

Figure 32 – Raster image with pixels values of 1 bit

a) b)

Figure 33 – a) An example of 8-bit grayscale raster image; b) A 24-bit color raster image

The applications of numerical cartography in raster formats are:

- Orthophotos, i.e. raster digital cards obtained directly from the aircraft frames or satellite images by applying the necessary corrections, projections and scaling (considering the altitude);
- Scanning of existing paper charts for preservation in time and make them available on the screen. For this application large format scanners (usually rolls) are needed;
- Remote connection with web map services.

GIS environment inside the raster data can be used as a cartographic support since it allows being georeferenced.

Georeferencing a file (or any geographic data) means assigning a datum and a cartographic system through a file associated (world file, Figure 34) that allows to switch from internal system (X = no. column, Y = no. row) to the cartographic system (East, North) with an affine transformation (consisting of six parameters):

$$\begin{aligned} E &= aX + bY + c \\ N &= dX + eY + f \end{aligned} \tag{16}$$

42.300 pixel size in the direction of E
0.000 rotation for the rows
0.000 rotation for the columns
-42.800 Pixel size in the direction of N
2262860.000 coordinate of the pixel E (1.1)
4839940.000 coordinated N of pixels (1.1)

Figure 34 – Example of a "word file" for the georeferencing

The "world file" has an extension that depends on what kind of image file it is associated with. Typically, it is the first and last letters of the image extension, with a “w” at the end. For example, for the file "sicilia.tif", the world file associated will be "sicilia.tfw".

The georeferencing procedure of a raster support is made with Ground Control Points (GCPs) recognizable on the raster on which the coordinates of the reference system (known or specially recorded, i.e. GPS points) are superimposed.

Once assigned the GCPs, different algorithms (i.e. Helmert, Molodenskij, etc.) implemented within the GIS platform for processing datum can be applied to georeference the raster support (Figure 35).

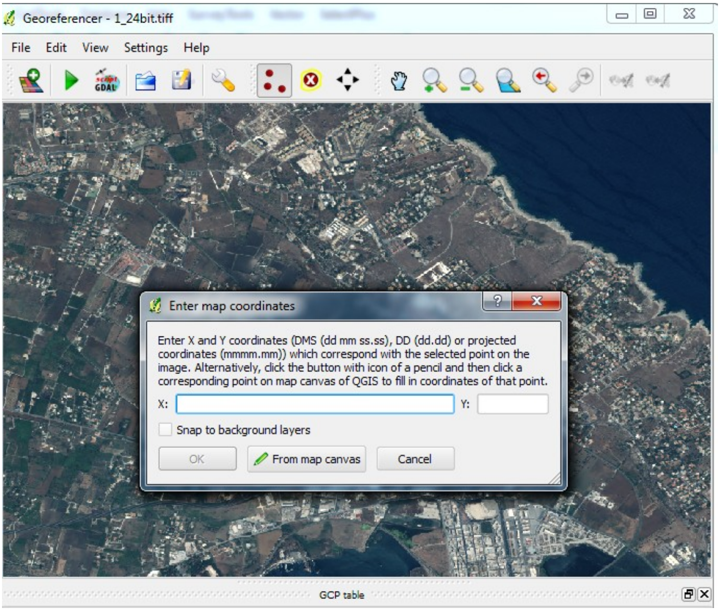


Figure 35 – Screenshots of the georeferencing tool within QGIS

The raster supports typically used in a GIS are the regional technical maps, orthophotos, satellite images and digital terrain models.

The regional technical maps (Figure 36) are obtained by photogrammetric and aerial photography; the nominal scales are typically 1:10,000, 1:5,000, and 1:2,000, 1:500 (only for urban centers. The altitude is reported as units of planimetric details, units of isolated points and contours.

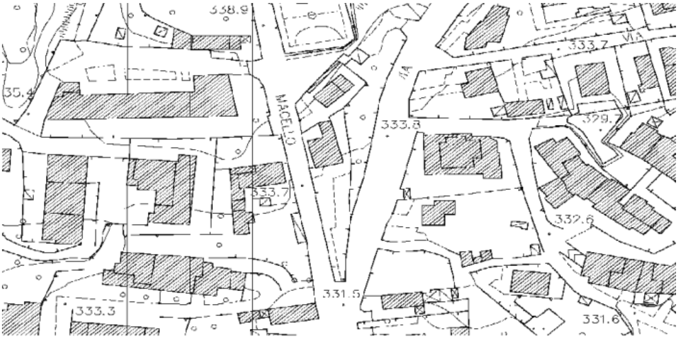


Figure 36 – Excerpt of a regional technical map at scale 1:2000 in UTM WGS84 33N

The orthophoto (Figure 37) is an image obtained from aircraft frames subjected to straightening (from photographic image perspective to orthogonal projection) and scale correction in function of altitude. The nominal scale generally used is of 1:10,000 and 1:2,000 for the major urban centers.

The satellite images (Figure 38) are aerial photographs taken by satellites, usually remote sensing satellites or similar.

Figure 37 – Excerpt of a 2007-2008 orthophotos in UTM WGS84 33N obtained through the WMS service of the Geoportal of the Sicilian region

Figure 38 – Excerpt of a 2006-2007 Quickbird satellite image in UTM WGS84 33N obtained through the WMS service of the Geoportal of the Sicilian region

The Digital terrain models are the purely geometric representations of the ground surface without any indication of the particular topography. They are divided in:

- Digital Terrain Models (DTMs) where the ground surface is modeled with a regular grid mesh (Figure 39);
- Digital Elevation Models (DEMs), which are DTMs with the addition of buildings and artifacts (Figure 40);
- Triangulated Irregular Networks (TINs) where the ground surface is modeled by a triangular mesh (Figure 41).

From the analytical point of view, the planimetric coordinates and the relative altitude are associated to each point of the area.

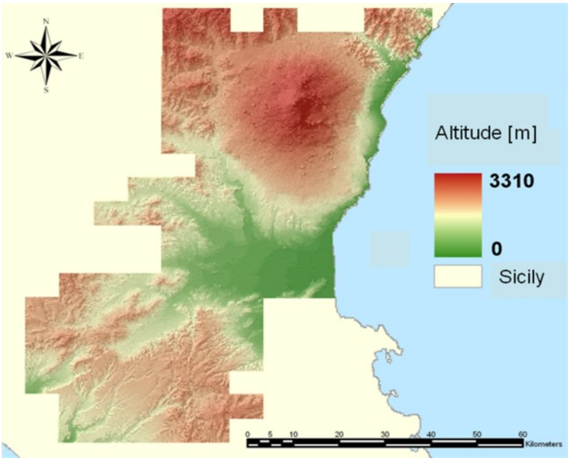


Figure 39 – DTM with a resolution of 40 meters of the Province of Catania

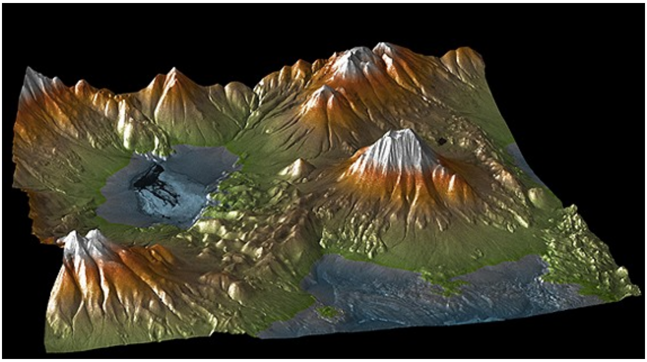


Figure 40 – An example of DEM

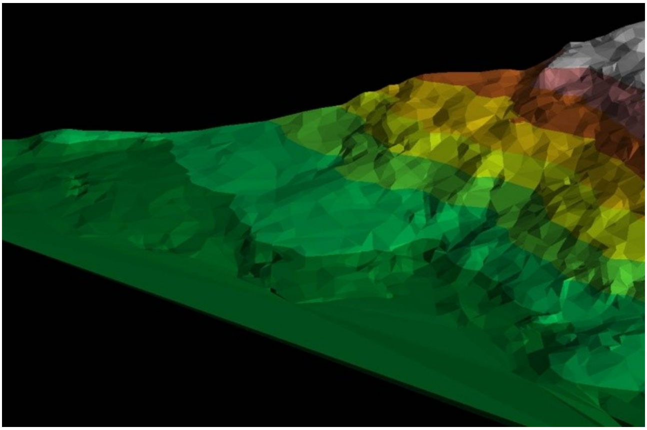


Figure 41 – An example of TIN

The DTMs can be obtained from ground surveys carried out with topographic instruments and/or GPS, from existing maps by digitizing contour lines and spot elevations, from aircraft frames stereoscopic oriented, from measurements by laser-scanning plane or helicopter, by remote sensing.

A DTM of large mesh size is available for the whole world to the public domain with a mesh of 30"of approximately 900 m. DTMs of higher resolution must be purchased or self-produced.

Vector data in GIS environment allow representing and computerizing elements of the territory (infrastructures, buildings, river basins, etc.) using geometric primitives (points, lines, polylines) analytically characterized by the 2D or 3D coordinates of their vertices (Figure 42). The computerizing of these elements is possible thanks to a database with which they are associated to.

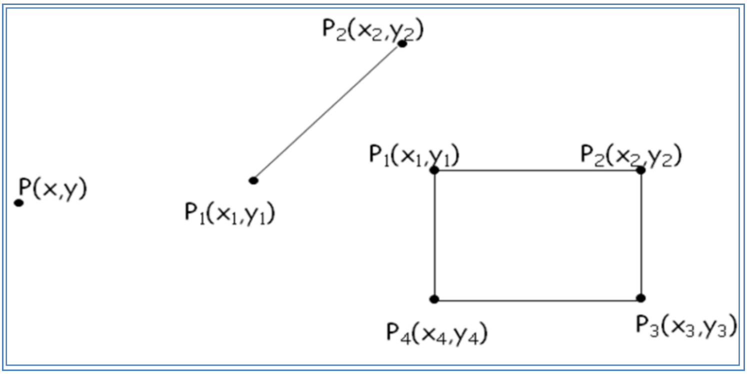


Figure 42 – Geometric primitives used for the representation and digitization of vector data

Vector data for GIS are identified and characterized by the Shapefile format. This format has been developed and regulated by ESRI (Environmental Systems Research Institute) that is the leading worldwide supplier of GIS software and geodatabase management applications. ESRI released the Shapefile as open source format to enhance interoperability ESRI and other GIS platforms. The structure of a vector shapefile is based on three mandatory files having the same name but different extensions. In particular:

- ***".shp"*** is the suffix for the main file;
- ***".shx"*** is the suffix for the positional index of the geometry feature allowing seeking forwards and backwards quickly;
- ***".dbf"*** is the suffix for the database table.

In the common language, the shapefile data vector is related specifically to files with the ".shp" extension, even if it is incomplete for distribution, needing the other two supporting files.

In addition to the three mandatory files, other eight optional files can be stored primarily for data index to improve performance. Each individual file should conform to the MS DOS 8.3 filename convention (8 characters for filename prefix, period and 3 character for the extension) in order to be compatible with past applications that handle shapefiles, many recent software applications accept files with longer names. For this reason, all files should be located in the same folder. These optional files include:

- ***".prj"*** is the suffix for the projection format where the coordinate system and the data relating to the projection used by the shapefile, are described using well-known text format;
- ***".sbx"*** and "***.sbn"*** is the suffix for the spatial indices used by spatial databases to optimize spatial queries;
- ***".ain"*** and "***.aih"*** is an attribute index of the active fields in a table or an attribute spatial table associated to vector themes;
- ***".mxs"*** is a geocoding index for read-write shapefiles in open document format for Office applications;
- ***".ixs" is*** a geocoding index for read-write shapefiles;
- ***".fbn"*** and ***".fbx"*** is a read-only spatial index of the shapefile features;
- ***".atx"*** is an attribute index for the .dbf file in the form of shapefile.*columnname.atx* (ArcGIS 8 and later);
- ***".shp.xml"*** is the suffix for the geospatial metadata in XML format, such as ISO 19115 (international metadata standard for geographic information) or other schemas;
- ***".cpg"*** is the suffix that specify the code page only for .dbf to identify the encoding characters to be used. Code page is another

term for character encoding (for example UTF-8, US-ASCII, EBCDIC, etc.). It consists of a table of values that describes the character set for a particular language.

In a GIS environment it is easy to establish a relation among objects in a layer and a specific database, as shown in Figures 43 and 44.

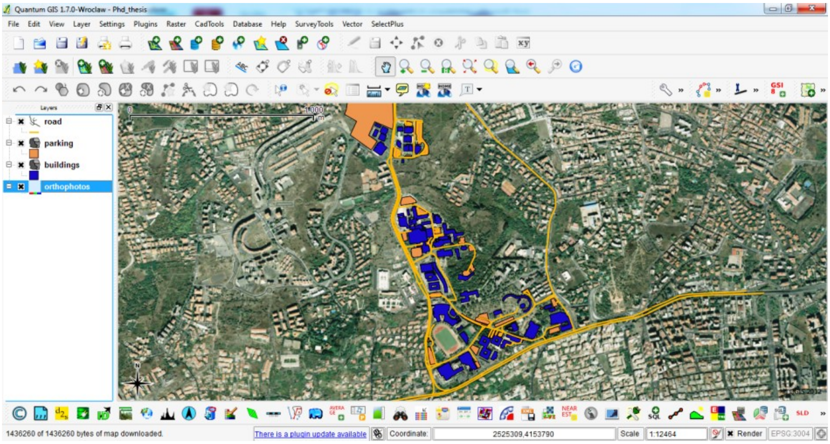


Figure 43 – Vector themes digitized in a desktop GIS and overlapped to an orthophoto with Gauss-Boaga coordinates

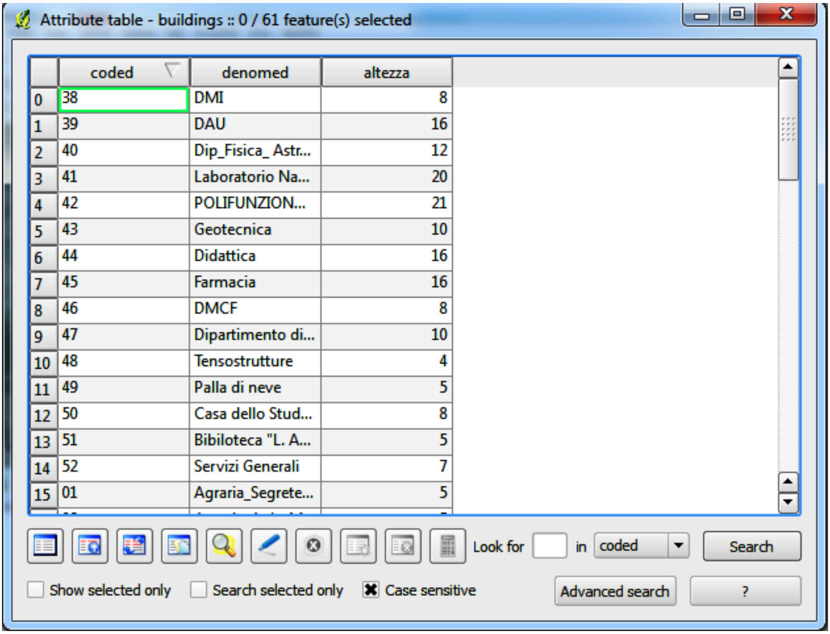


Figure 44 – Table of attributes associated to "buildings" generated in a desktop GIS environment

2.3 Desktop GIS platforms

The desktop GIS technology consists of hardware and software components providing the typical features of GIS. In particular, the desktop configuration allows GIS using the functionality and processing spatial data in the location where the platform is installed.

The desktop GIS software can be distributed with commercial license or under public licenses, as for with free and open source softwares. These latter technologies have become very common in recent years, allowing the user downloading the entire source code of the free desktop GIS to customize it in base of specific requirements.

The most common commercial GIS is ArcGIS developed by ESRI (Figure 45) that by the mid-90s spread the first desktop GIS platform integrated.

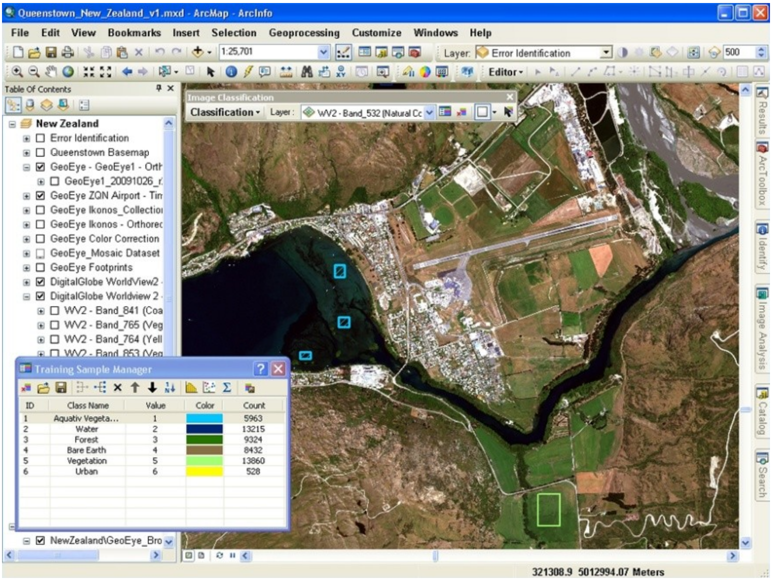


Figure 45 – Screenshot of ArcGIS, the most common commercial GIS developed by ESRI

The real breakthrough for desktop GIS happens with the distribution of free and open source cross-platform, thanks to a global community, OSGeo, which is a not-for-profit organization whose mission is to support the collaborative development of open source geospatial software, and promote its widespread use.

The most common free and open source desktop GIS are: GRASS (http://grass.fbk.eu/index.php), QGIS (www.qgis.org), gvSIG (http://www.gvsig.org/web), etc.

The graphic interface of a desktop GIS generally comprises some standard components (Figure 46): a legend area, in which the thematic layers loaded are listed; an ample map area, where the thematic layers are displayed; a small reference map; a toolbar that provides typical tools for the map navigation, generic plugins for the GIS environment and other instruments to manage specific data types.

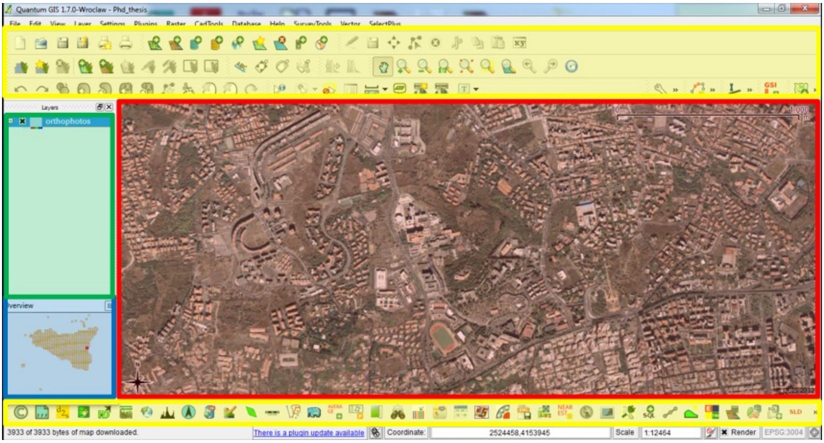


Figure 46 – Standard components available in desktop GIS: legend area (green), map area (red), reference map (blue), and toolbar (yellow). At the bottom of the page, the reference system with the dynamic coordinates and the scale are also displayed

The GIS environment offers some functionalities that can be applied to the different data types, raster or vector. In particular, each layer can be managed in an independent way modifying the graphic features, i.e. transparency, colors, etc. and the topographic properties (Figure 47). The vector thematisms can be digitized and modified using the editing functionality that allows creating geometries (point, line or polygon), and setting the data structure associated to the thematism. Otherwise, the editing functionality permits to modify the existing geometry and update the related database.

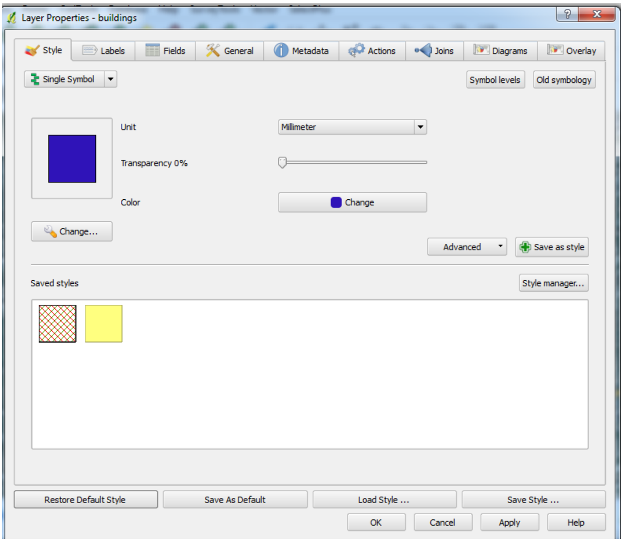


Figure 47 – Screenshot of QGIS showing the available tools to modify graphic and topographic properties

The online resources available to manage GIS data include Web Map Services (WMS) for serving maps, Web Feature Services (WFS) for serving vector features, and Web Coverage Services (WCS) for serving raster data.

The GIS platform can be linked to a spatial database, which can be local or on a remote server, for exploit the advanced functionalities provided by the spatial DBMS. Each table associated to the vector thematism furnishes a tool for query building that allows performing evolved spatial queries on the vector thematism using mathematical or logical operators (Figure 48).

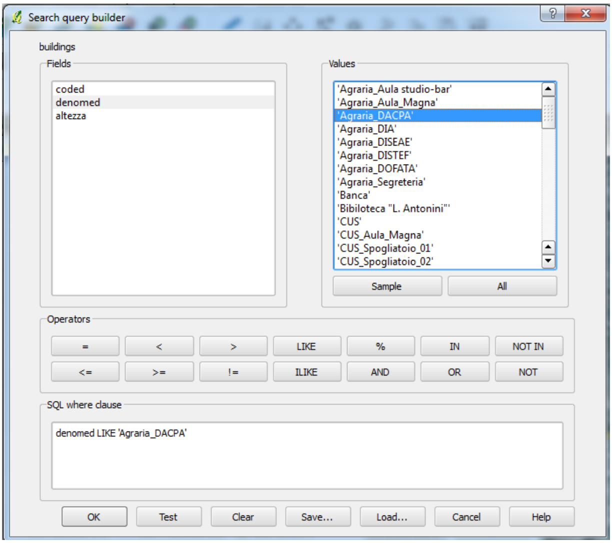


Figure 48 – QGIS query building, where "fields" are the attributes of the vector thematism and "values" are the data stored within the attributes. The bottom of the window also provides the mathematical and logical operators and the editor for composing the spatial query

The main functionalities available for raster data are the “Calculator” (to perform calculations on basis of existing raster pixel values), “Grid” (to create raster from scattered data using interpolation methods), “Clipper” (to automatically mosaic a set of images being in the same coordinate system and having a matching number of bands), and “DEM terrain models” (to analyze and visualize DEMs).

The main functionalities for vector data are available within the geoprocessing tool: “Convex hull” (to create minimum convex hulls for an input layer, or based on an ID field), “Buffer” (to create buffers around features based on distance, or distance field), “Intersect” (to overlay layers such that output contains areas where both layers intersect), “Union” (to overlay layers such that output contains intersecting and non-intersecting areas), “Symmetrical difference” (to overlay layers such that output contains those areas of the input and difference layers that do not intersect, “Clip” (to overlay layers such that output contains areas that intersect the clip layer), “Difference” (to overlay layers such that output contains areas not intersecting the clip layer), “Dissolve” (to merge features based on input field).

Other additional functionalities are available to convert the GIS data from the raster to the vector format and vice versa, to the georeferencing, and to set advanced printing options for the maps.

2.4 WebGIS platforms

WebGIS applications enable the distribution of geo-spatial data in internet and intranet networks, exploiting the analysis derived from GIS software and by means of classical features of web-based publish geographical information on the World Wide Web.

Like classical web architecture, a webGIS system is based on standard client-server functionality. The client is any web browser, such as Mozilla Firefox, while the server side consists of a Web-server (e.g. Apache) and a software webGIS (e.g. Mapserver) that is dedicated to provide the viewing/query's interpretation of georeferenced data.

The process of operation of a webGIS (Figure 49) can be schematized in the following way:

- the user, through a web interface, send a request as defined in the area of interest (geographical extension of the display area) and the content required (list of layers of interest) using a generic browser;
- based on the data received, the webGIS researches in the own archive the required information (files, such as shape, TIFF, ...; making a database connection, or through the access to other server mapping, such as OGC Web Services) by removing the portion of the territory specified;

- one or more images are generated and sent to the client.

The interrogation of data is performed in a similar way: in this case, the client sends to the server a pair of coordinates (in the case of a punctual interrogation) or a set of coordinates to define a region, and the server queries the attributes of cartographic objects present in the area of interest, giving a report with the extracted values.

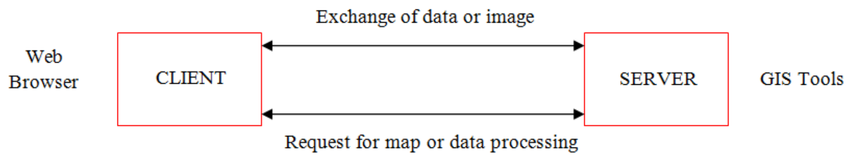


Figure 49 – Structure of the webGIS

As for the Desktop GIS, commercial, and free and open source development environments are available for webGIS applications.

The most common commercial WebGIS software include ESRI ArcIMS (ArcView Internet Map Server; Figure 50), Intergraph Geomedia WebMap Professional, AutoDesk MapGuide, SmallWorld Internet Application Server and MapInfo MapXtreme.

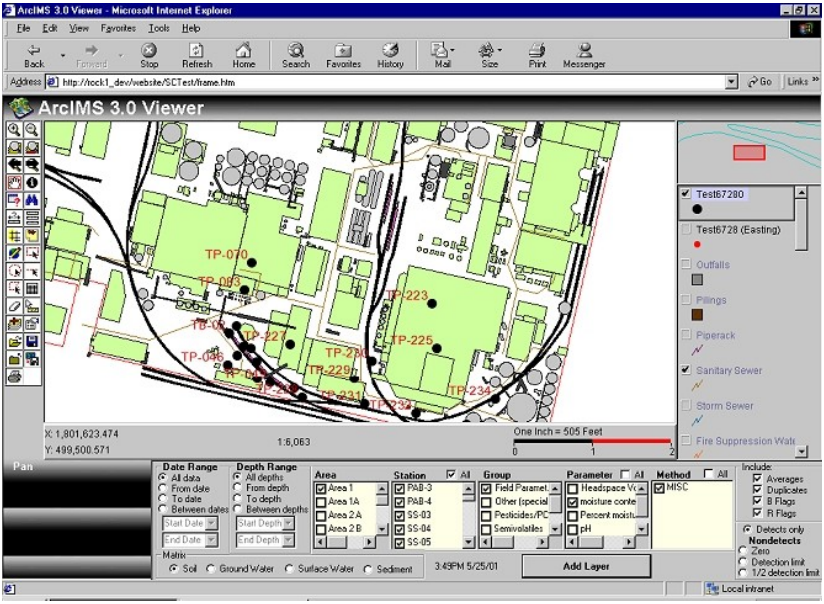


Figure 50 – The development environment ArcIMS for webGIS applications

The most popular free and open source webGIS softwares are Map Server (mapserver.org/), GeoServer (geoserver.org/) and OpenLayers (openlayers.org/). The typical graphic interface of a webGIS contains the similar components of those included in the Desktop GIS, i.e. a legend area, an ample map area, a small reference map and a toolbar (Figure 51).

In the webGIS platforms raster and vector data can be loaded. Raster data comprise orthophotos, satellite images, DTMs, DEMs and TINs, while vector data include shapefiles and ".dxf".

The functionalities available within a webGIS allows navigating the map (i.e. pan, zoom in, zoom out, zoom to the maximum extent, measurement tools, etc.), modifying the topographic parameters, connecting to and querying the external spatial database, and setting the print options for the map.

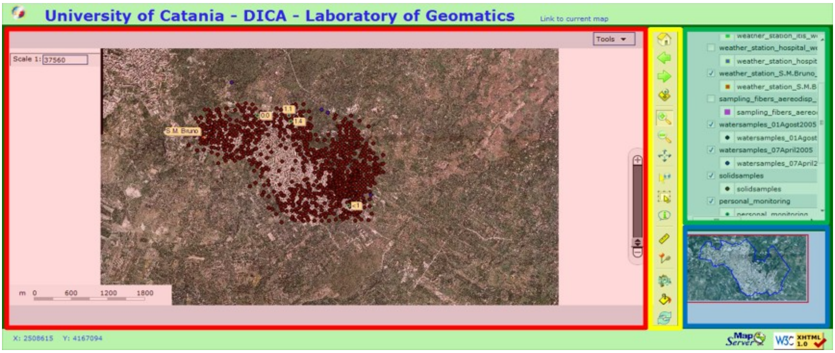


Figure 51 – Standard components available in webGIS: legend area (green), map area (red), reference map (blue), and toolbar (yellow)

Chapter 3

Spatial databases

In the GIS technology, the management of data and information associated therewith is particularly important for the functionality of the GIS application developed.

Although supported by basis tools for managing geographical information, most GIS platforms often resort in external DataBase Management System (DBMS) to leverage the advanced characteristic of the software data manager. Therefore there is a conceptual difference between database and DBMS, being this latter software tool devoted to the management and maintenance of a database.

GIS softwares are really database engines capable of creating and manipulating geographic data in combination with non-geographic attributes and of obtaining representations and mixed associations between the two data types. For this reason, the synonym GeoDBMS (Geographic DataBase Management System) is used for a type designation.

3.1 Introduction to databases

Geographic Information Systems are based on the interaction between a geometrical (map) and a tabular (table) interface used to manage and analyze spatial data. When working with GIS, it is important to define a database design and structure that allow the user to be consistent while moving from the storage phase to the analysis phase of data.

A good database-design allows finding data, and speeds up data processing. Updating and modelling data play a very important role in GIS and therefore have to be taken into consideration while designing a database. Deciding the relations between data and the names of the dataset are also crucial steps in the database design phase. Names have to be understandable and unique for every single dataset.

To analyze in detail the design of a data structure for GIS applications it is important to understand the difference between data and information: data is a set of characters that must be processed and interpreted, while the information is the evolution of data after being interpreted (Figure 52).

An information system is a system for the collection and organization of data, while an informatics system is a hardware and software architecture useful to automate an information system.

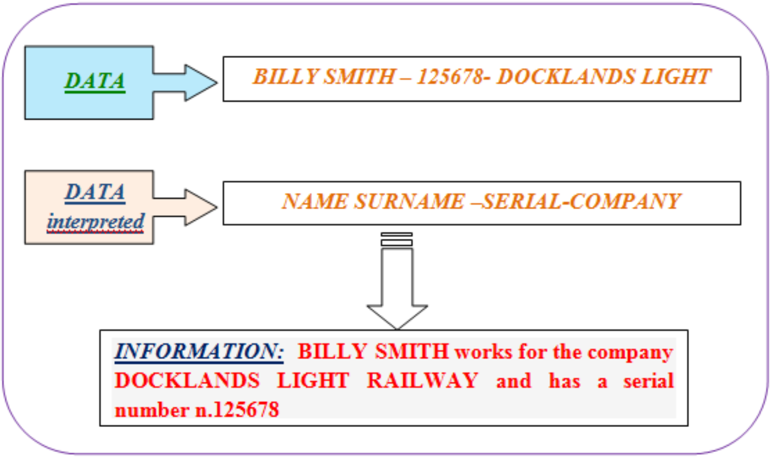


Figure 52 – Example of data end information

The set of data, information, information system and informatic system constitutes a permanent set of information organized according to a structure defined by a data model that represents the real situation to automate called database (Figure 53).

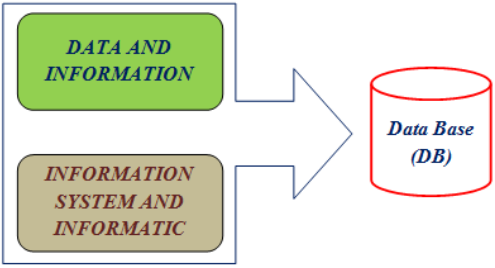


Figure 53 – Example of data end information

The characteristics of a database are the size, persistence and reliability.

The size is the characteristic of a database that contains a large amount of data without posing a problem for its management:

The persistence is the characteristic of a database of storing data for a time of life that is not limited by the execution time of the software that manages them. Without persistence, the data is only stored in RAM so lost when the computer turns off.

The reliability is the characteristic of the database to be able to keep its contents or to allow its reconstruction in the event of a software or hardware malfunction with operations of saving and restoring (backup and restore).

The simplest form of a database consists of a file containing many lines, each one containing the information that must be stored in the database. This type of data storage has considerable limitations: elevated search times, problems of redundancy in storing information, problems in the maintenance of information, etc.. The advantage is that it relies on a very simple structure and it is relatively easy to handle. This type of database is generally used when the amount of information to be stored is very small.

Hierarchical databases are an evolution of this first simple type of database, using multiple files to store information and introducing a hierarchical relationship of parent-child (tree) between them. An example of this database is the IMS (Information Management System).

This type of database has the disadvantage of being inflexible and of presenting difficulties in renovation, expansion and reduction. For this reasons, it is no longer used.

As expansion of the hierarchical database has been developed in the early 70's, with the name of "reticular" or DBTG CODASYL (Conference on Data Systems Languages, Data Base Task Group). This type of database makes it possible the achievement of a node from multiple paths, but presents high costs in management when the number of relationships increases, so much so that nowadays this type of database is no longer used. Being born to manage data with relatively small dimensions (sequences of 0's and 1), another disadvantage of relational databases is the lack of ability to manage BLOBs (Binary Large Objects), which are complex data types, such as images and documents.

This limitation has been resolved with the database objects (object-oriented) that store only references, while the physical memorization of BLOBs is outside the database. The information is stored in objects that have hierarchical relationships. These types of databases are not very popular and generally used only in CAD environments and engineering.

The design of a database can be considered essentially a process of analysis of the real context to be represented, and an abstraction of a model that describes, using the most significant information (Figure 54).

In order to design a database, three main steps of data-model instances should be followed: conceptual model, logic model and physical model.

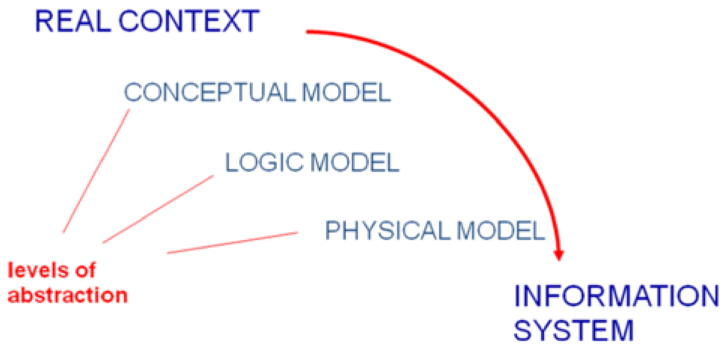


Figure 54 – The data modelling process used to design a database

The models that perform these three steps are called semantic, the best known of which is the Entity-Relationship (ER) model. In the ER model, the concrete or abstract objects of the reality of interest are classified as an entity (table) and each entity has specific properties called attributes (columns).

The conceptual model introduces a graphical representation of reality using three phases: requirements gathering, requirements analysis and ER diagram.

During the requirements gathering, the problems that the system has to solve and entities are identified, which are concrete or abstract objects relevant to the information system.

The requirements analysis organize and define the best requirements collected following appropriate rules, such as avoiding the use of convoluted sentences, standardizing the structure of sentences, identifying and unifying the objects using the same synonym (i.e. place / location / site, teacher / professor / teacher, study / office), avoiding too general or specific specifications.

In the third phase of the conceptual design, the collected and analyzed requirements are converted from the natural language to a graphical representation using precise constructs. This graphical representation is called E-R diagram.

ER diagrams employ special symbols to represent three fundamental and different types of information: boxes are used for entities, diamonds stand for relationships, and small circles symbolize attributes. In addition, other information and kinds of attributes can be represented within E-R diagrams, i.e. simple and compound attributes, relationships and attribute cardinalities, generalizations, subsets, identifiers (Figure 55).

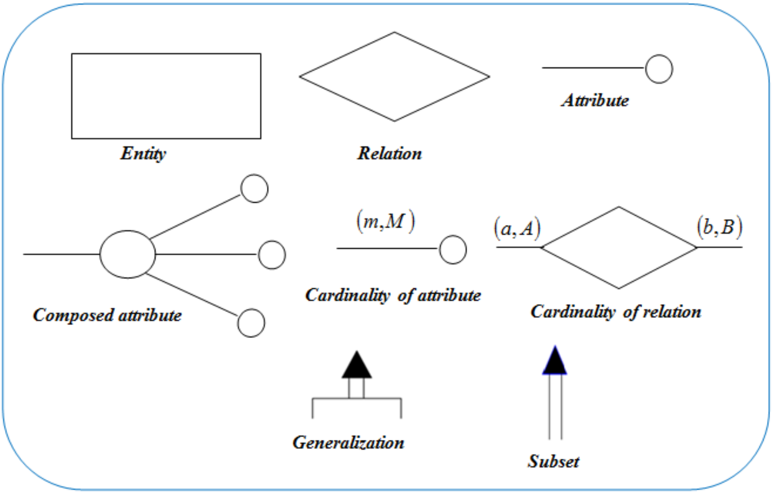


Figure 55 – Graphical representation of the constructs used in the E-R diagrams

An entity may be defined as a set of objects about which information has to be gathered (Figure 56). Physically within the data base are represented by tables.

Figure 56 – Example of graphical representation of an entity

The *relation* is defined as the logical link between entities about which you have interest to collect information (Figure 57).

Figure 57 – Example of the graphical representation of relation between two entities: *The person is related to the city with his residence*

The attributes are the information that must be collected for the entities and relationships. Within the database, they are the columns of the tables (Figure 58). Also the attributes can be composed when described by different information (Figure 59), and characterized by a cardinality to indicate optionality or multivalued attributes (Figure 60). If no cardinality of constraint is specified, the attribute is unique.

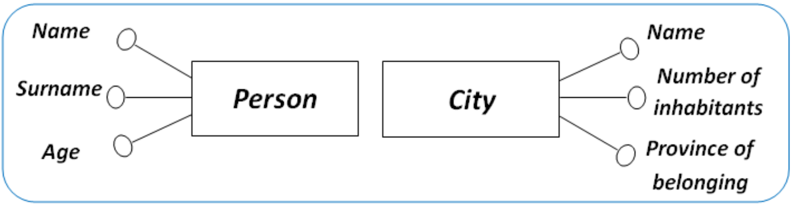


Figure 58 – Graphical representation of the attributes of two entities: *The attributes for Person are name, surname and age. The attributes for City are name, number of inhabitants and province of belonging*

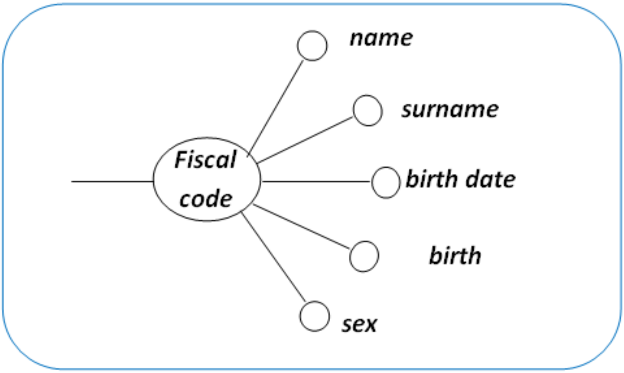


Figure 59 – Graphical representation of a composed attribute: *The fiscal code consists of name, address, date of birth, place of birth and sex*

The cardinality of the relation specifies the minimum and maximum number of occurrences of the entities in relationships. So, if *R* is the relationship between the entity *1* and the entity *2*, then *a* and A represent respectively the minimum and maximum number of times in which the entity *1* participates in the relation *R*, while *b* and B are respectively the minimum and maximum number of times in which the entity 2 participates in the relation R (Figure 61).

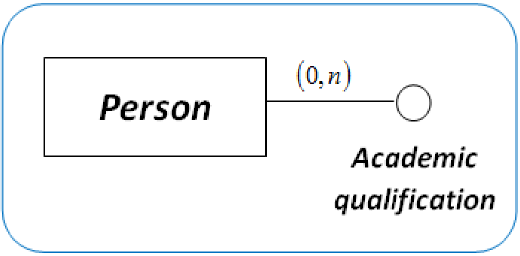


Figure 60 – Example of graphical representation of the cardinality of attribute: *Person can have zero or more academic qualifications*

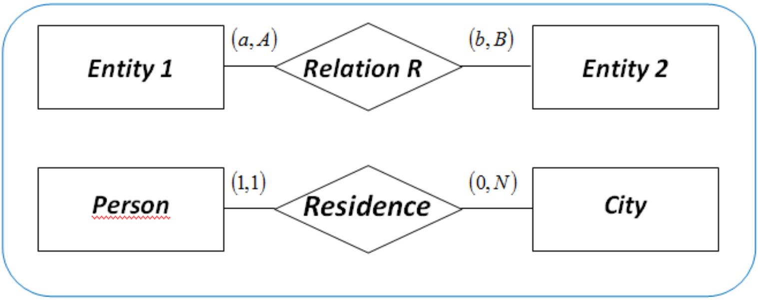


Figure 61 – Examples of graphical representation of the cardinality of relations

In the ER diagram, the generalization represents the logical link between an entity E, called "father", and one or more entities E_1, E_2,…, E_n, called "daughters" (Figure 62).

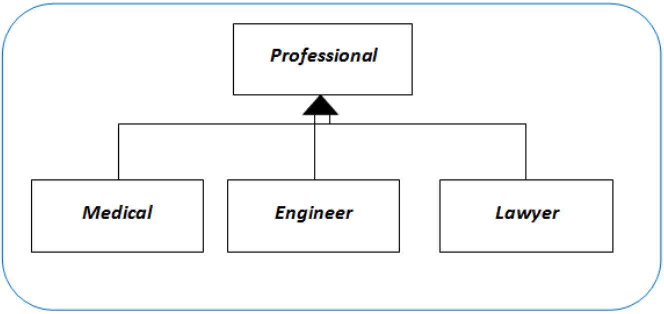


Figure 62 – Graphical representation of generalization: *Professional is a generalization of Medical, Engineer and Lawyer*

The subset is a special case of generalization where the father entity has only one child that is subset of the father.

The identifier is defined as a particular type of attribute that uniquely identifies an entity. If *E* is an entity, (A_1 ..... A_n) is a set of attributes of the entity, E_1 ...... E_m is a set of entities directly related to E and I = {A_1, ..., A_n, E_1, ..., E_m} is an identifier for E, then the following definitions can be defined (Figure 63):

- Internal identifier or primary key if m=0;
- External identifier or foreign key if n=0;
- Mixed identifier if m > 0 and n > 0;
- Simple identifier if m + n = 1;
- Compound identifier if m + n > 1.

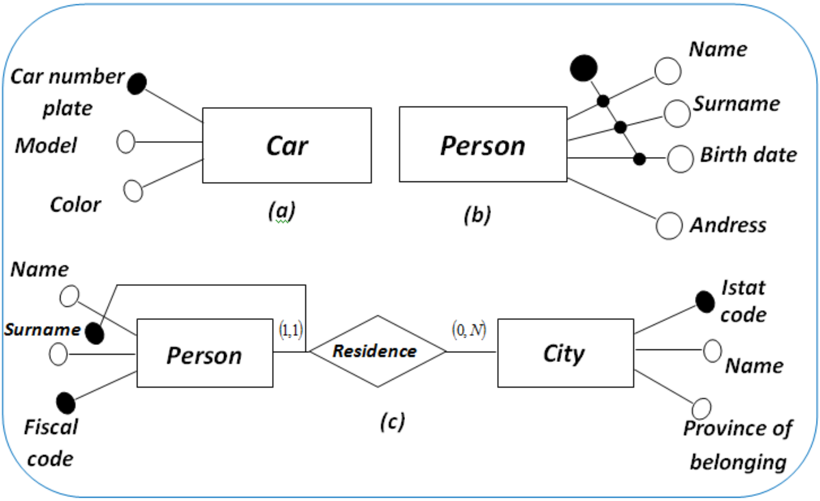


Figure 63 – (a) Example of internal identifier or primary key, (b) Example of compound identifier and (c) Example of external identifier or foreign key

The logical modeling is a technique of organizing and accessing data to be used with a database manager.

In contrast to the conceptual modeling, the logical one strictly depends on the logical model of data representation used by the DBMS, hierarchical, network or relational (this latter used to manage data a GIS environment).

The logical modeling is developed in two phases: restructuring of the E-R diagram and translation towards the logical model.

In the restructuring of the E-R diagram, the ER schema is optimized and simplified through an analysis and removal of redundancies, analysis of the generalization and substitution with other constructs, partition or merging of entities and associations, choice of primary identifiers. After these operations, the translation to logical model can be performed, where the entities of the conceptual model become tables, the relationships between entities are represented using foreign keys (i.e. a set of attributes that correspond to those that make up the primary key in another table), and a reference between the lines of the two tables is established.

Finally, the data structure is complete with physical modeling, where the final logical model is implemented physically creating tables.

As introduced before, another the management and maintenance of a database is performed by a DBMS, the main characteristics of which are privacy, efficiency, effectiveness and sharing.

The privacy is allows the user to indentify uniquely a database and to access it determining a control policy and the possible actions that can be performed on the data.

The efficiency of a DBMS represents the capacity to carry out the operations required using a set of resources (hardware and time) that are acceptable to users.

The effectiveness of a DBMS expresses the capacity to make productive activities performed by users on it.

Finally the sharing is the characteristic to access the database by other applications and users in an appropriate way set by the manager of the database. This characteristic leads to the reduction of redundancy and data inconsistency.

A DBMS provides different languages for interacting with the database. The level of abstraction of these languages strongly depends on the model of data to which it refers, but a common distinction classifies languages based on the functions performed:

- DDL (Data Definition Language) to define the structure (logical, external, internal);
- DML (Data Manipulation Language) to query and change the database;
- DCL (Data Control Language) includes commands of various types, i.e. for access control;
- SQL combines instructions of all three types (therefore called the SQL DDL, DML of SQL and SQL DCL).

The architecture of a DBMS is composed of three levels: external, logical and physical (Figure 64).

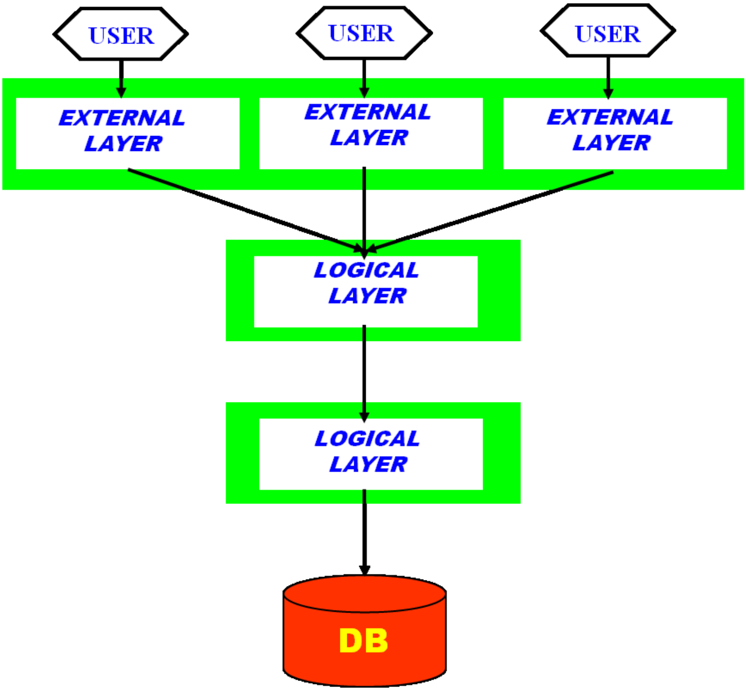


Figure 64 – The three layers of the DBMS architecture

The users of a DBMS can be divided in users of applications with no specific knowledge, users not programmers using the DBMS in an interactive way, programmers of JDBC applications, database designers for conceptual/logic modeling and SQL development (DDL), database administrators with specific knowledge of DBMS and DBMS programmers. Through the external level, the user can operate on the database using SQL or JDBC (Java Database Connectivity) commands.

The logical layer is the overall structure of the database, where all information are represented in an organized manner via an abstract model.

The physical layer is formed by structures of mass memory used to access and store the data in a quick and efficient way. Being the management of the database held by the administrator, the physical layer is transparent to the user, who is engaged only in higher-level operations.

3.2 Development of the spatial DB

An important aspect of GIS technology is the representation of data in a computer system through a representative and flexible model to fit the real phenomena.

Three types of information are essentially managed in GIS: geometric, spatial relationships (topological relations, directional relations and proximity relations), and informative.

Geographic information can manage objects, including size and location, in cartography and hence in the territory. Information concerning spatial relationships may be classified in several ways.

A first distinction is between relationships that are independent of orientation, such topological relations, and those that depend on it, such directional relations: the first mainly concern connection, adjacency, inclusion, segregation, etc. of geometric entities, while the second relate the different geometric entities using a dependence on the orientation (i.e. in front, the other side of, on, under, right, left).

Another distinction regards the "proximity", which describes the distance between geographical entities both in terms metric quantities (distance measurement) and in qualitative terms using expressions such as near, far, in the vicinity of, etc. (Figure 65).

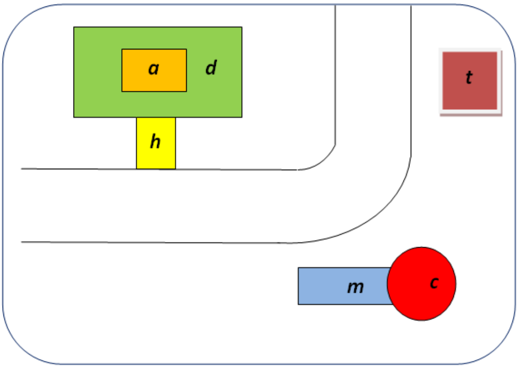


Figure 65 – Spatial relationships between geometric entities. Topological relations: *m* is disjoint from *d*, *h* is connected to *d*, *a* is in internal to *d*. Directional relations: *d* is on top of *h*, *t* is to the right of *d*. Proximity relations: *h* is far from *c*, *d* is near to *h*. It is worth noting that spatial relations can be combined among them

The flexible data structure used to manage the real phenomena described by GIS technology is the relational database. All environments allow the management of a GIS database and therefore of spatial data, but as the GIS software are born not to provide DBMS functions, then the functions related to the management of the data are limited.

The direct management of data in GIS software can lead to anomalies within the database as data redundancy and inconsistency, problems of competition for access to data by multiple simultaneous users, loss of data integrity, security problems, and problems of efficiency from the point of view of search times of data and update data. This determines the impossibility of developing a relational database. For this reason, in a complete and efficient design of a GIS, external Relational Database Management Systems (RDBMS) are used for the management of spatial database containing the data used in the GIS application.

There are several commercial and open source/free software platforms for the RDBMS. They often consist of a standard module for the management of the common database and an extension that is installed separately for the management of the data space.

An example of commercial platform is ORACLE with the spatial extension ORACLE-SPATIAL, while an example of free and open source software is PostgreSQL built on pgAdmin III platform with the spatial extension PostGIS (Figure 66).

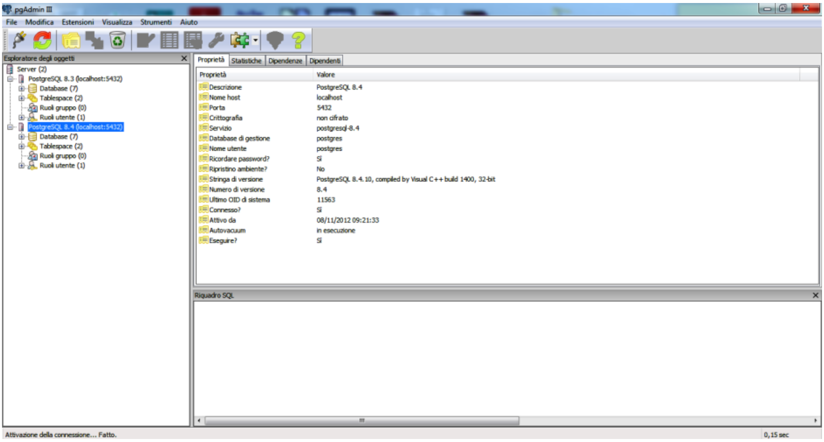


Figure 66 – Graphical interface of the DBMS PostgreSQL

In addition, good hardware architecture allows to have the best choice of using a DBMS outside the GIS platform. In this case the server is in the same machine where the DBMS and the spatial database are located, while all GIS applications (Desktop and Web) they can access are present in the external PC. With this architecture, all the characteristics of DBMS and database are exploited, and the speed for updating the database modification and query is improving, thanks to a totally dedicated machine (Figure 67).

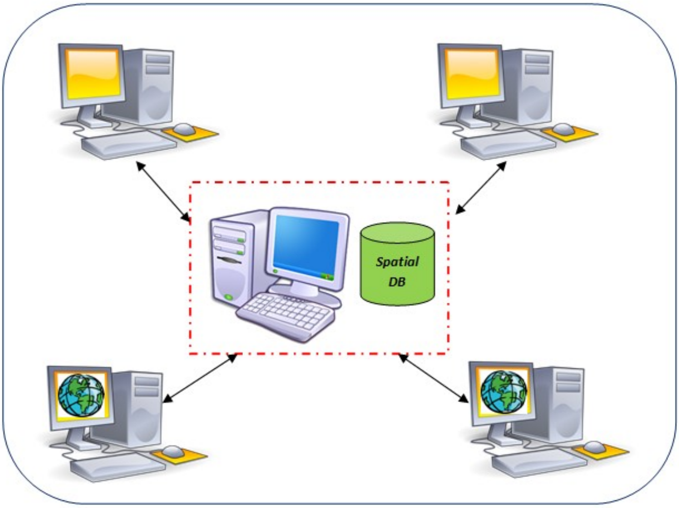
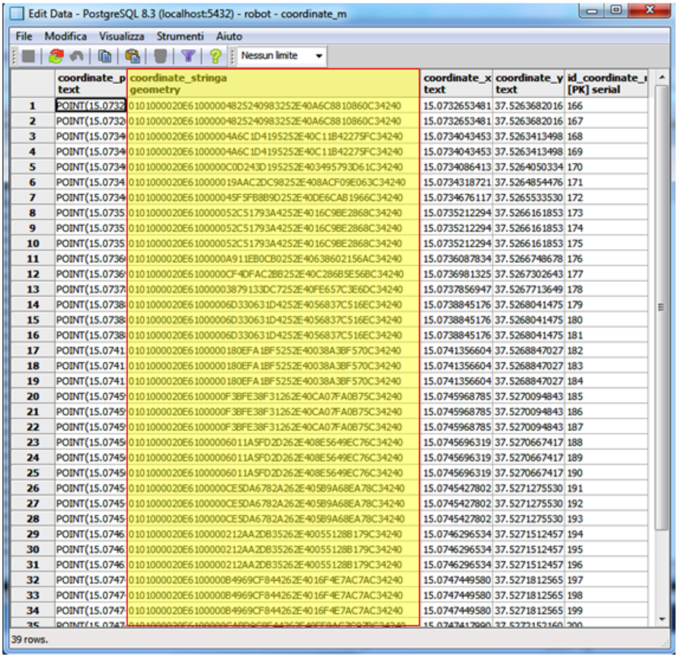


Figure 67 – The GIS users connect themselves to a spatial database located in a dedicated server

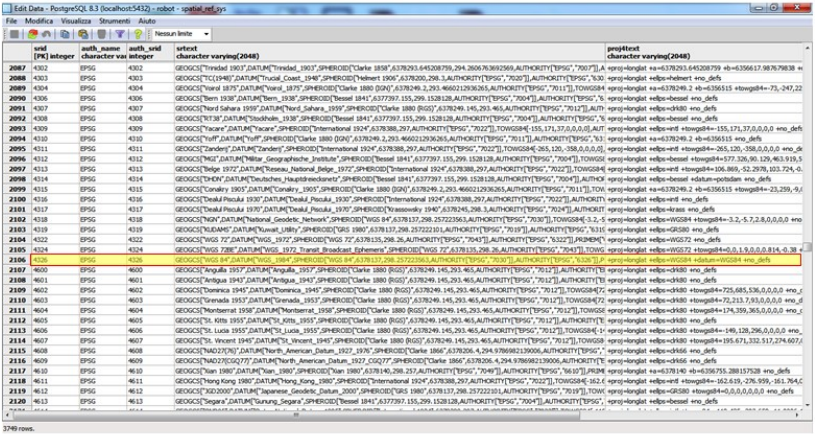
The DBMS used in this thesis to manage the spatial database of the GIS applications supporting the navigation of autonomous robots is the free and open source platform PostgreSQL with the PostGIS spatial extension. The DBMS manages the spatial information within tables in the spatial database using a column containing the geometric or spatial data. Each row of this column contains an alphanumeric string that represents the spatial reference for the entities from the data stored in the table (Figure 68).



	coordinate_p text	coordinate_stringa geometry	coordinate_x text	coordinate_y text	id_coordinate_ [PK] serial
1	POINT(15.0732	0101000020E610000048252409832 52E40A6C8810860C34240	15.0732653481	37.5263682016	166
2	POINT(15.0732	0101000020E6100000482524098325 2E40A6C8810860C34240	15.0732653481	37.5263682016	167
3	POINT(15.0734	0101000020E61000004A6C1D4195252E40C11B42275FC34240	15.0734043453	37.5263413498	168
4	POINT(15.0734	0101000020E61000004A6C1D4195252E40C11B42275FC34240	15.0734043453	37.5263413498	169
5	POINT(15.0734	0101000020E6100000C0D243D195252E403495793D61C34240	15.0734086413	37.5264050334	170
6	POINT(15.0734	0101000020E610000019AAC2DC98252E408ACF09E063C34240	15.0734318721	37.5264854476	171
7	POINT(15.0734	0101000020E610000045F5FB8B9D252E40DE6CAB1966C34240	15.0734676117	37.5265533530	172
8	POINT(15.0735	0101000020E610000052C51793A4252E4016C9BE2868C34240	15.0735212294	37.5266161853	173
9	POINT(15.0735	0101000020E610000052C51793A4252E4016C9BE2868C34240	15.0735212294	37.5266161853	174
10	POINT(15.0735	0101000020E610000052C51793A4252E4016C9BE2868C34240	15.0735212294	37.5266161853	175
11	POINT(15.0736	0101000020E6100000A911EB0CB0252E40638602156AC34240	15.0736087834	37.5266748678	176
12	POINT(15.0736	0101000020E6100000CF4DFAC2BB252E40C286B5E56BC34240	15.0736981325	37.5267302643	177
13	POINT(15.0737	0101000020E61000003879133DC7252E40FE657C3E6DC34240	15.0737856947	37.5267713649	178
14	POINT(15.0738	0101000020E61000006D330631D4252E4056837C516EC34240	15.0738845176	37.5268041475	179
15	POINT(15.0738	0101000020E61000006D330631D4252E4056837C516EC34240	15.0738845176	37.5268041475	180
16	POINT(15.0738	0101000020E61000006D330631D4252E4056837C516EC34240	15.0738845176	37.5268041475	181
17	POINT(15.0741	0101000020E6100000180EFA1BF5252E40038A3BF570C34240	15.0741356604	37.5268847027	182
18	POINT(15.0741	0101000020E6100000180EFA1BF5252E40038A3BF570C34240	15.0741356604	37.5268847027	183
19	POINT(15.0741	0101000020E6100000180EFA1BF5252E40038A3BF570C34240	15.0741356604	37.5268847027	184
20	POINT(15.0745	0101000020E6100000F3BFE38F31262E40CA07FA0B75C34240	15.0745968785	37.5270094843	185
21	POINT(15.0745	0101000020E6100000F3BFE38F31262E40CA07FA0B75C34240	15.0745968785	37.5270094843	186
22	POINT(15.0745	0101000020E6100000F3BFE38F31262E40CA07FA0B75C34240	15.0745968785	37.5270094843	187
23	POINT(15.0745	0101000020E61000006011A5FD2D262E408E5649EC76C34240	15.0745696319	37.5270667417	188
24	POINT(15.0745	0101000020E61000006011A5FD2D262E408E5649EC76C34240	15.0745696319	37.5270667417	189
25	POINT(15.0745	0101000020E61000006011A5FD2D262E408E5649EC76C34240	15.0745696319	37.5270667417	190
26	POINT(15.0745	0101000020E6100000CE5DA6782A262E405B9A68EA78C34240	15.0745427802	37.5271275530	191
27	POINT(15.0745	0101000020E6100000CE5DA6782A262E405B9A68EA78C34240	15.0745427802	37.5271275530	192
28	POINT(15.0745	0101000020E6100000CE5DA6782A262E405B9A68EA78C34240	15.0745427802	37.5271275530	193
29	POINT(15.0746	0101000020E6100000212AA2DB35262E400551288179C34240	15.0746296534	37.5271512457	194
30	POINT(15.0746	0101000020E6100000212AA2DB35262E400551288179C34240	15.0746296534	37.5271512457	195
31	POINT(15.0746	0101000020E6100000212AA2DB35262E400551288179C34240	15.0746296534	37.5271512457	196
32	POINT(15.0747	0101000020E6100000B4969CF844262E4016F4E7AC7AC34240	15.0747449580	37.5271812565	197
33	POINT(15.0747	0101000020E6100000B4969CF844262E4016F4E7AC7AC34240	15.0747449580	37.5271812565	198
34	POINT(15.0747	0101000020E6100000B4969CF844262E4016F4E7AC7AC34240	15.0747449580	37.5271812565	199
35	[illegible]	[illegible]	[illegible]	[illegible]	200

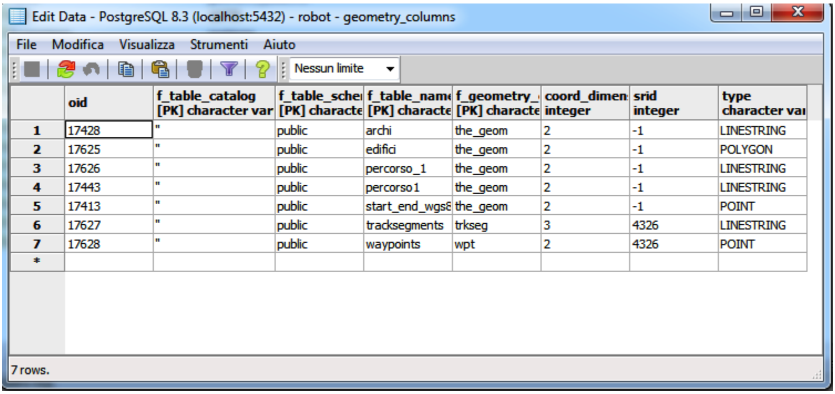
Figure 68 – An example of spatial table. The geometric column is highlighted

The alphanumeric string in the column geometry is generated using the characteristics of the various spatial reference systems and the information about geometric fields created in the database and stored respectively in two tables, spatial_ref_sys and geometry_columns, that are automatically installed with the spatial extension PostGIS (Figures 69 and 70).



	srid [PK] integer	auth_name character var	auth_srid integer
2087	4302	EPSG	4302
2088	4303	EPSG	4303
2089	4304	EPSG	4304
2090	4306	EPSG	4306
2091	4307	EPSG	4307
2092	4308	EPSG	4308
2093	4309	EPSG	4309
2094	4310	EPSG	4310
2095	4311	EPSG	4311
2096	4312	EPSG	4312
2097	4313	EPSG	4313
2098	4314	EPSG	4314
2099	4315	EPSG	4315
2100	4316	EPSG	4316
2101	4317	EPSG	4317
2102	4318	EPSG	4318
2103	4319	EPSG	4319
2104	4322	EPSG	4322
2105	4324	EPSG	4324
2106	4326	EPSG	4326
2107	4600	EPSG	4600
2108	4601	EPSG	4601
2109	4602	EPSG	4602
2110	4603	EPSG	4603
2111	4604	EPSG	4604
2112	4605	EPSG	4605
2113	4606	EPSG	4606
2114	4607	EPSG	4607
2115	4608	EPSG	4608
2116	4609	EPSG	4609
2117	4610	EPSG	4610
2118	4611	EPSG	4611
2119	4612	EPSG	4612
2120	4613	EPSG	4613

Figure 69 – Screenshot of the table *spatial_ref_sys* highlighting the line on the WGS84 reference system identified by the EPSG code 4326



	oid	f_table_catalog [PK] character var	f_table_schei [PK] characte	f_table_namı [PK] characte	f_geometry_ [PK] characte	coord_dimen: integer	srid integer	type character vai
1	17428	"	public	archi	the_geom	2	-1	LINESTRING
2	17625	"	public	edifici	the_geom	2	-1	POLYGON
3	17626	"	public	percorso_1	the_geom	2	-1	LINESTRING
4	17443	"	public	percorso1	the_geom	2	-1	LINESTRING
5	17413	"	public	start_end_wgs8	the_geom	2	-1	POINT
6	17627	"	public	tracksegments	trkseg	3	4326	LINESTRING
7	17628	"	public	waypoints	wpt	2	4326	POINT
*								

Figure 70 – Screen shot of the table *geometry_columns* that contains information about geometric fields created in the database

3.3 Connection with the GIS platforms

All GIS environments provide instruments that allow connection with an external spatial database. Using these tools it is thus possible to import data structures created in the GIS environment within the external database. Conversely, the spatial database created, modified or optimized by using external DBMS can be exported into GIS environment.

Specifically the spatial and semantic information of the GIS environment is contained in the thematic vector, and created and managed with the editing functionality, without the possibility to create a relational database. The shapefiles and the associated data can be imported to the external environment GIS from the spatial database. This procedure is performed for all themes of the GIS application, which allows the development of a relational data structure using the DBMS platform outside the GIS (Figure 71).

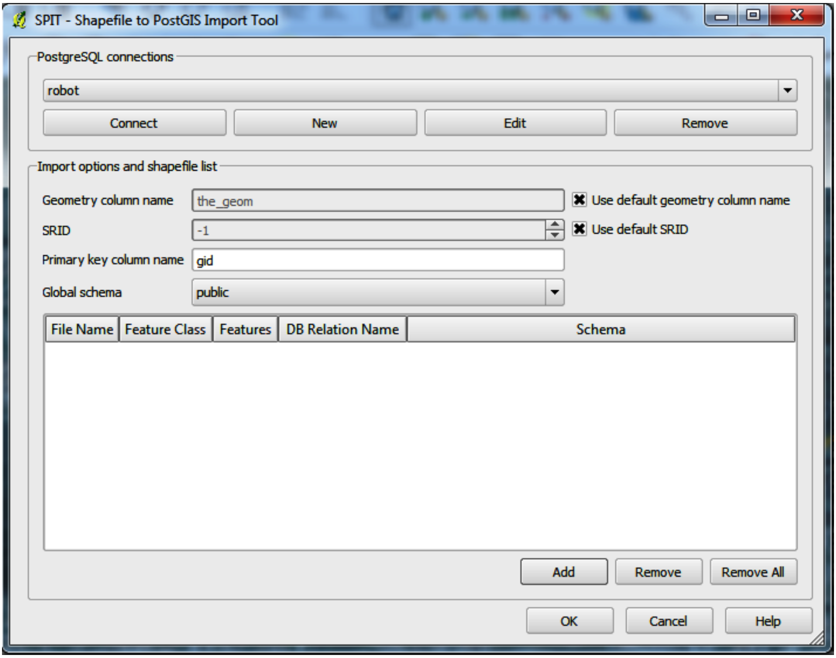


Figure 71 – Screenshot of the QGIS Spit tool to import vector themes in the PostGIS database external to the desktop GIS. It consists of instruments to connect with the remote database (PostgreSQL connections) and import options to select the spatial database and table that will host the geometric vectors themes

On the contrary, a data structure developed in a DBMS platform external to the GIS can be exported in a GIS environment using desktop tools that allow setting the connection parameters (Figura72) of the remote spatial database.

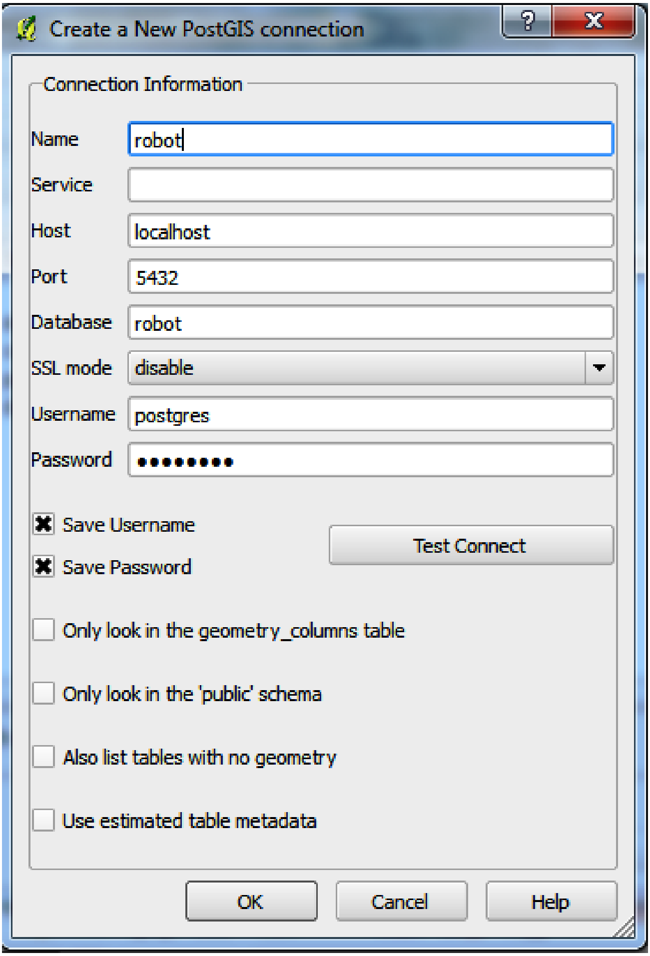


Figure 72 – Screenshot of the interface for the connection with the remote database PostGIS

Once the connection is established, the spatial tables of the database geometric information become available since they are coded in the GIS environment (Figure73).

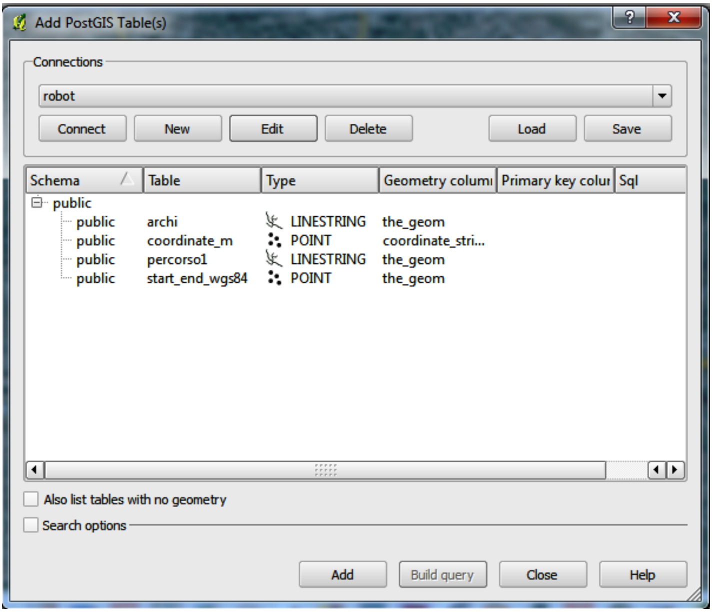


Figure 73 – Screen shot of the interface to add PostGIS layers in QGIS

Obviously the webGIS platforms can connect to the spatial database remotely. Connection systems depend on the platform on which the webGIS application is developed.

For example, a webGIS developed with OpenLayers uses the connection to GeoServer in order to import and represent spatial data available in PostGIS (Figure 74).

Figure 74 – Screenshot of GeoServer, where it is highlighted the PostGIS layer used in the webGIS developed with the Openlayers technology

3.4 Spatial queries

The query spatial can be defined as specific types of database queries, managed, supported and executed on spatial databases. In particular, spatial queries differ from regular queries on database in using geometric and georeferenced data (points, lines, polygons) and in considering the relationships that exist between these geometries.

Spatial queries can be performed in the GIS environment using the query builder, which operates on the data structure of the single vector themes and on the data contained in the columns (attributes) of the table spatial associated. The query builder structure and functioning are similar for all GIS environments.

In order to explain its way of working, let consider the query builder in QGIS (Figure 75), which can be accessed directly by clicking with the right mouse button on the layer and choosing a vector feature, or by advanced search attribute of the table associated with it.

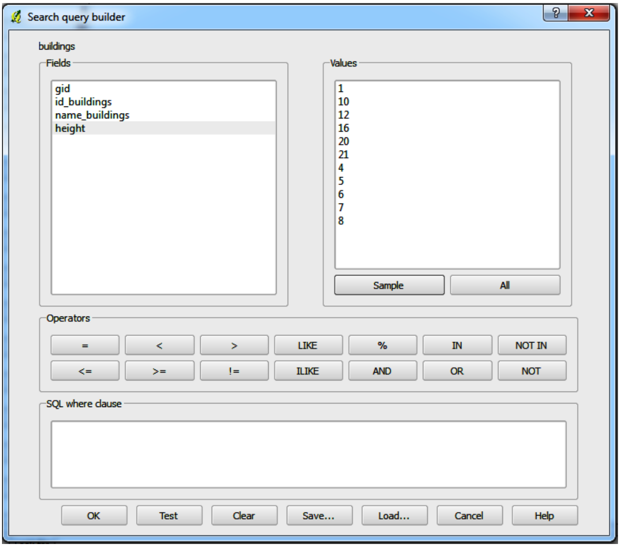


Figure 75 – Query builder in QGIS applied to buildings layer

The Fields, Values and Operators sections of the query builder consent to construct the SQL-like.
The Fields list contains all features of the attribute table of the vector layer considered. Moreover, once double clicking a feature, an attribute can be added through the *SQL where clause* field. Generally various fields, values and operators can be used to construct the query. Otherwise the query can be directly typed into the SQL box editor.
The Values list contains the values of an attribute. To list all possible values or some values of an attribute, the user should select the attribute in the Fields list and then click respectively the *All button* or the *Sample button*. To add a value to the *SQL where clause* field, the name in the Values list should be double clicked.
The Operators section contains all usable operators to compose a spatial query: mathematical (= , > , <, etc.), string comparison (LIKE) or logical (AND , OR , etc.) operators. They use Boolean algebra which operates essentially with only the truth values 0 and 1. Hence the combination of logical operators can develop any Boolean function. To add an operator to the SQL where clause field, the appropriate button should be clicked.
There are different symbols to represent the logical operators based on the field in which they find application, e.g. in mathematics it is

often used the + symbol for OR, and × or * for AND, because in some respects these operators work in a similar way to the sum and multiplication. The negation NOT is often represented by a line drawn on the argument of denial that is the expression that must be negated. In informatics, the symbols | or | | are used for OR, & or && for AND, and ~ or ! for NOT.

If A and B are variables that can take only 0 and 1 as values, then the operators AND, OR and NOT follow the following Boolean algebra:

A	B	A *AND* B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A *OR* B
0	0	0
0	1	1
1	0	1
1	1	1

A	*NOT* A
0	1
1	0

When logical operators are used for the construction of spatial queries, the inputs A and B are queries that can take the value 1 if the query is verified, 0 otherwise.

For example, suppose you want to search all buildings that meet simultaneously the condition of *id_buildings = 02* and *height = 20*, then the spatial query has the following syntax:

id_buildings = 02 AND height = 20

If only one of the clauses in the input is checked, the output of the operator *AND* is zero, hence no geometric entity is found (Figure 76).

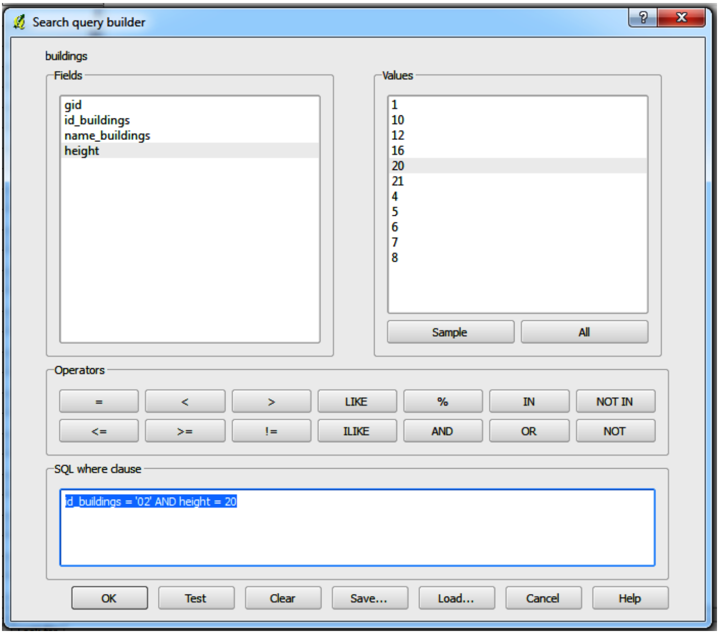


Figure 76 – Spatial query for searching all individual geometric entities present within the buildings vector layer, which satisfy simultaneously *id_buildings = 02 and height =20*

The query editor allows testing the query before being executed, deleting the text in the SQL where clause text field, saving and loading SQL queries using the corresponding buttons located at the bottom of the mask graphics.

From the GIS point of view, the result of spatial query corresponds to a selection in the table attributes on the vector layer applied to the query. Therefore all functionalities typical of the GIS selection can be applied, for example the saving as shapefile.

Even in webGIS platforms, a query editor is available with different functionalities depending on the web platform used. Nevertheless, it is not provided as a standard tool and elaboration software is required to be added to the webGIS platform (Figure 77).

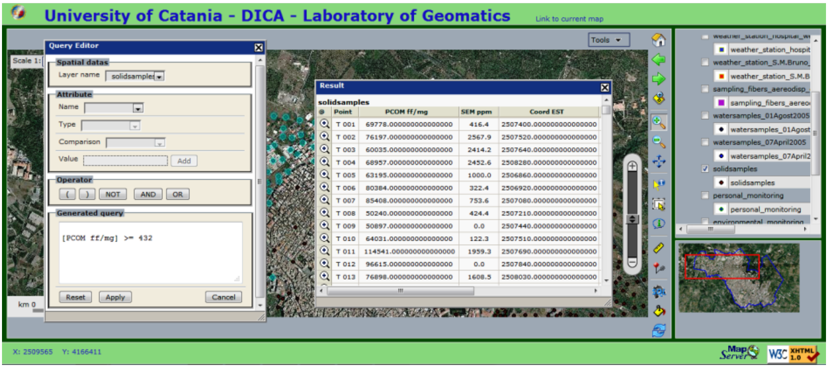


Figure 77 – Query editor available in the MapServer webGIS application

Spatial queries can be also performed using the spatial database external to the GIS platform. If a spatial database is exported and used in desktop GIS platform, then the query builder will operate on it or the query here may be built into the DBMS software that manages the spatial database, using a dedicated editor box. The clauses for the construction of spatial queries are written using the SQL syntax and typical functionalities of the extension spatial of the DBMS used. For example, the PostGIS spatial extension of PostgreSQL acknowledges the following geometric entities: Point, LineString, Polygon, Multipoint, MultiLineString, MultiPolygon, Geometry-Collection (Figure 78).

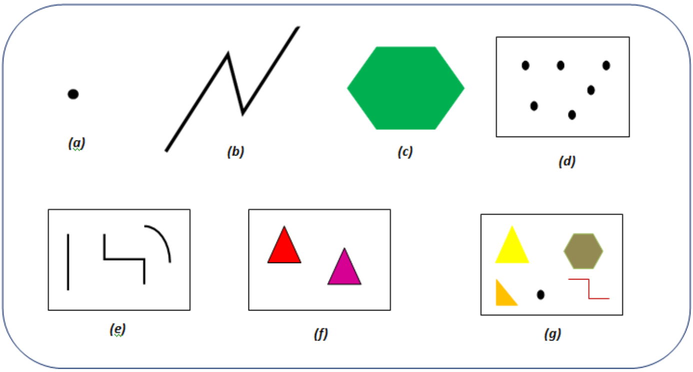


Figure 78 – Geometric entities used within the DBMS PostgreSQL with the PostGIS extension: (a) Point, (b) LineString, (c) Polygon, (d) Multipoint, (e) MultiLineString, (f) MultiPolygon, (g) Geometry-Collection

The functions available in PostGIS are:

- *Basic functionality* for the removal and adding of columns and the geometric assignment of a reference system for spatial data;
- *Functions of relationships* between geometries for calculatin the distance between two geometries, and the verification of possible overlap, intersection, inclusion, etc., between distinct geometric shapes;
- *Calculation functions* on geometries for calculating area, perimeter, centroid, etc., of a given geometry;
- *Function of "information" on the geometry* for knowing the type of geometry present in a given field, the EPSG code and the reference system used, the number of contained points, etc.;
- *Calculation functions of measures* for calculating area, perimeter, length, spheroid, etc., in relation to the reference system and the type of plane (two-dimensional or three-dimensional);
- *Editing functions of the geometry* for manipulating the geometry, i.e. translation or operations of simplification using the Douglas-Peuker algorithm.

An example of spatial query executed in PostgreSQL is reported in Figure 79. A table is created an some operations are performed on it. Then the table space is exported in the GIS environment (Figure 80).

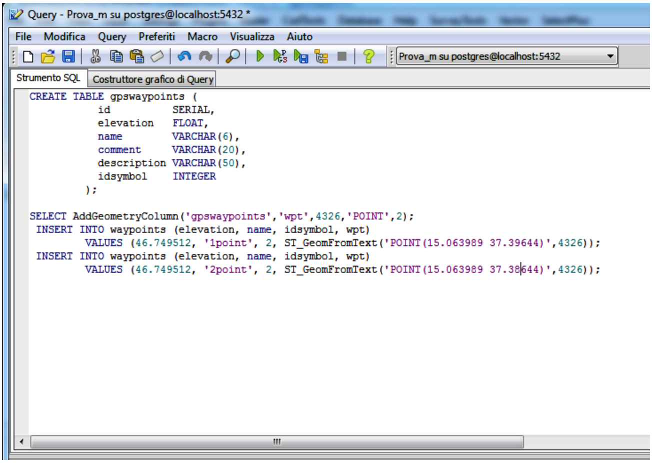


Figure 79 – Spatial query performed in postgreSQL. A table called gpswaypoints is created and a geometric column called wpt with the WGS84 datum is added to it. Finally some records are inserted in the table

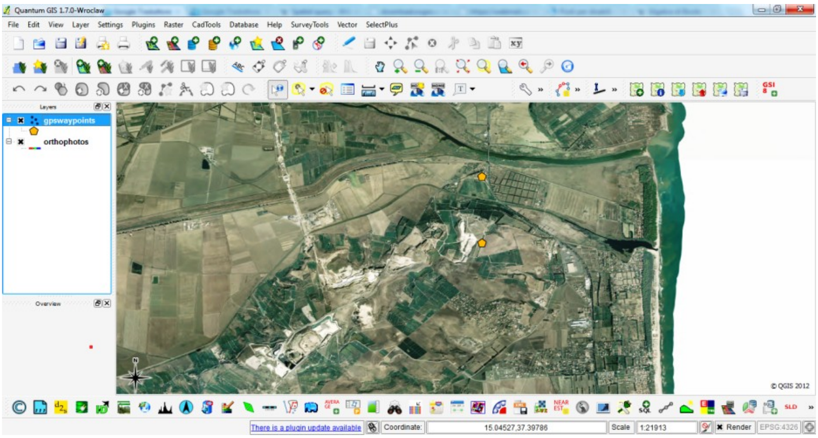


Figure 80 – Geometric entities of the PostgreSQL *gpswaypoints* exported to the QGIS desktop platform

Chapter 4

GIS as support to mobile robot navigation

Nowadays more and more applications require a mobile robot able to operate autonomously in an outdoor environment. The accuracy and precision of mapping systems for the working environment is particularly important for the management, control and efficiency of the activity of the mobile robot. For these reasons, systems mapping must be consistent and efficient, and represent the surroundings in real-time allowing the robot, equipped with a system of autonomous movement, the exploration of unknown environments.

This chapter will discuss the implementation of a system for the management of the navigation of robots based on free and open source GIS technologies. The system furnishes different functionalities, including the storage, processing and representation of all possible information from media geo-maps, satellite positioning systems, innovative systems for remote sensing, modeling and representation of the environment.

Opposed to other approaches available in literature, the system developed in this thesis as support for the navigation of mobile robots, allow using the numerous functionalities typical of GIS technologies to plan and define optimal paths depending of the actual characteristics of the robot and the external environment where it operates.

4.1 Introduction

In recent years the use of autonomous robots has been a valid tool for exploring hostile environments affected by natural disasters such as landslide areas, underground cavities, high pollution zones, volcanic eruptions, buildings at risk of collapse. [*Biondi et al.,* 2004]. The objective of autonomous robots is to minimize the risk for men who work in hazardous environments [*Jun-ichi Meguro et al.,* 2006; *Astuti et al.,* 2009; *Muscato et al.,* 2012].

The navigation of autonomous robots is usually handled through on-board inertial systems, in outdoor applications using techniques such as GNSS satellite positioning for framing space (definition and adoption of an absolute or relative reference) and planning of real-time navigation and remote control systems or, in the case of unknown environments and not visible, making use of integrated navigation systems on board based on the allocation of waypoints (at least beginning-end of the journey of exploration) and the recognition and consequent avoidance of obstacles encountered along the path.
For recognition of obstacles makes use of laser sensors, while bypassing them may be driven by algorithms such Potential Fields.
A fundamental aspect for the management of navigation for autonomous robots that navigate in outdoor environments is to use a precise cartographic support on which control the position, also the variety of media geo-mapping available today and its different reference systems, makes it very useful if you do not need to resort to the use of GIS environments, for the correct georeferencing and interpretation of assigned routes and paths actually traveled.
The GIS makes it possible to manage the navigation of the robot on correctly georeferenced spatial data bases with the topography and rigorous approaches to acquire the same reference system any additional information from the sensors installed on the robot.

The GIS technology can be very useful also for the management of navigation in closed environments, for which it is possible to imagine the construction of a spatial information system to serve as information support base, similarly to what, in the external environment, is done with the use of maps, orthophotos, satellite images, digital elevation models.
In any case, the typical functionalities of GIS permit to define and monitor the optimal paths depending on the operating environment and the mechanical characteristics of the robot.
The GIS technologies chosen in this work are all free and open sources allowing customizing the application to manage the autonomous GIS navigation of robots. In this way, the application is developed specifically for the purpose of robot navigation and is extremely advantageous from the economic point of view, not requiring specific licenses for software.
The architecture of the GIS application consists of a desktop GIS platform installed and used at fixed locations and a webGIS platform to access Internet and the DBMS that manages a single spatial database used by the two applications.

4.2 Architecture for the robot navigation in GIS environment

The GIS platform architecture is based on the spatial database, which communicates with both the GIS (Desktop and Web) and the robot (Figure 81). This architecture is born from the need to guarantee a unique software tool for the management of the spatial data, allowing to exploit all instruments provided by the spatial database and the characteristics available within the DMBS. With such architecture, different kind of access policies can be integrated. For example the administrator can access and modify the whole data, while the uses of the Desktop or WebGIS can only visualize and perform basic operations on data. Another advantage provided by the architecture designed is that the database is located in an “ad hoc” remote server so that the exclusive management of the hardware and software resources is guaranteed.

Within the PostGreSQL / PostGIS DBMS, which remains a work environment external to the GIS software, a geographical database is built and managed from a geometric table that hosts and characterizes the geographical positioning data recorded in the field. This requires, first, a setting of the spatial characteristics of the data in such a way as to make them automatic and immediate placement on the basis of cartographic reference.

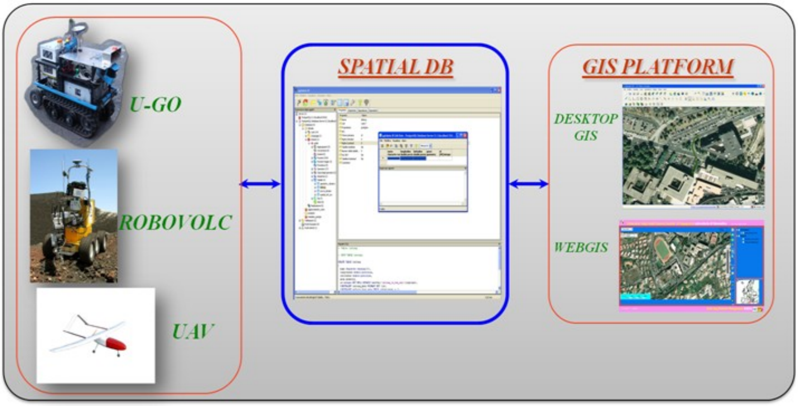


Figure 81 – The GIS architecture supporting the navigation of autonomous robots, showing the centrality of the spatial database, which communicates with both the GIS environments and the robot

The geographical database is connected with the Desktop platform through the Add PostGIS layer described in Paragraph 3.2., which provides a mask for setting the connection parameters with the remote database. The connection with the Web platform is managed through specific internal functionalities of the structure.

The architecture developed allows visualizing the information related to the robot position in the GIS environment thanks to the conversion of the GPS output in geometric columns. Moreover, thanks to the central position of the geographic database, it is possible to exploit functionalities typical of the GIS environment for the robot navigation.

4.3 Desktop GIS and WebGIS platforms

The desktop GIS platforms of the thesis are QGIS 1.7.0-Wroclaw and Grass 6.4.

QGIS (Figure 82) was used as a mapping system to monitor the position of the robot thanks to a simple and intuitive user interface. As topographic features of the GIS project, the datum WGS84 EPSG 4326 was set for the reference system (Figure 83), since the management of the robot involves the use of spatial data from the GPS. As cartographic raster supports, DEMs or DTMs can be used. Otherwise the WMS service within the GIS can provide orthophotos or satellite images, allowing the platform to connect itself with the remote server for obtaining the cartographic raster supports. Several servers provide this functionality, two of which are SITR (http://www.sitr.regione.sicilia.it) and the national cartographic portal (http://www.pcn.minambiente.it/GN/).

As vector supports, the WFS service furnishes vector files already digitized and available on line. Otherwise other solutions include the digitization of vector data through the editing tool provided by QGIS or the loading of vector thematisms from the spatial database.

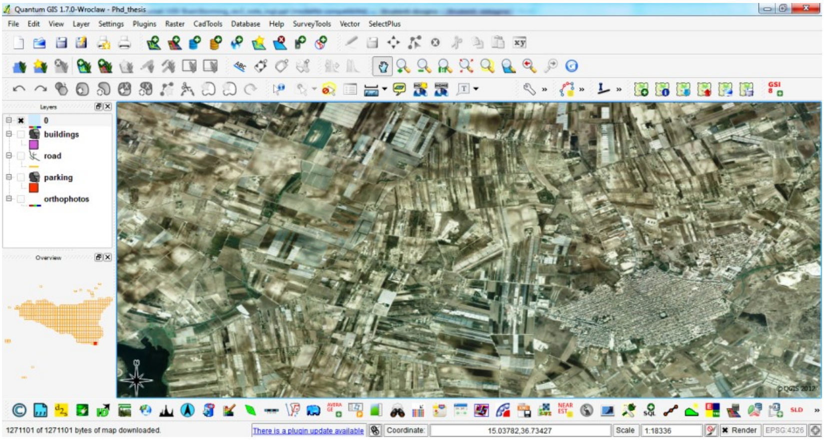


Figure 82 – The QGIS interface

GRASS was used because it provides advanced features that can be called within QGIS. Grass environment (Figure 84) is characterized by a graphic interface that is not very intuitive and user friendly, requiring time and effort to be learnt and remembered. GRASS uses both a windows interface as well as command line syntax for ease of operations. It contains over 350 programs and tools to render maps and images; manipulate raster, vector, and sites data; process multi spectral image data; and create, manage, and store spatial data.

Examples of functions for raster data regard the hydrologic modeling, the terrain analysis and the 3D raster statistics, while tools available for vector data are the topology manager, the network analysis and feature extractions.

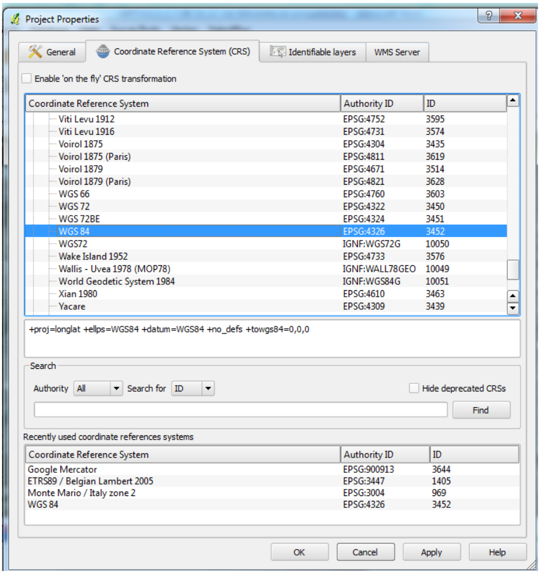


Figure 83 – QGIS mask for setting the topographic parameters

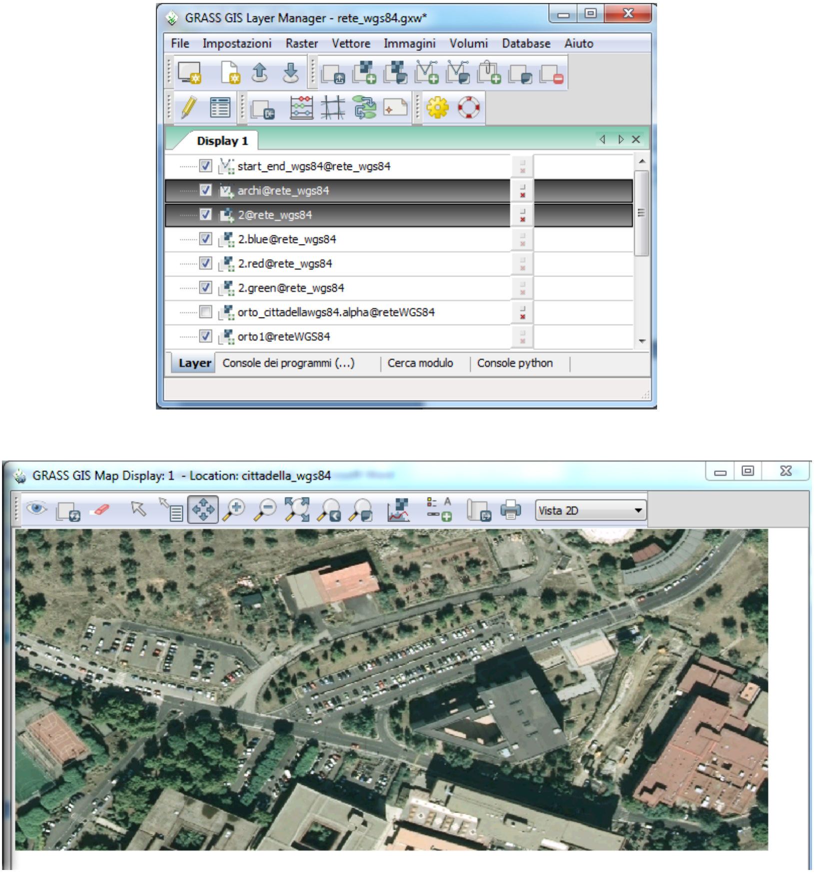


Figure 84 – The Grass interface

The Desktop GIS application can be used only on the machines where it is installed. This limit constitutes and obstacle for its use and for sharing information related to the robot navigation.

So for this thesis a webGIS application has been developed accessible through internet connection. The webGIS allows using the full Desktop GIS functionalities with the support of an external DBMS connected with the web platform.

The webGIS application was developed using the free and open source software MapServer with the PMapper applicative and Apache HTTP Server 2.0.48.

The platform has been customized and optimized by modifying the graphical interface and entering additional plugins as a query builder, software for the direct link with spatial DB, and the ability to populate the spatial DB remotely (Figure 85).

Figure 85 – WebGIS developed in Mapserver with the PMapper applicative

The webGIS application developed and customized in mapserver is based on CGI (Common Gateway Interface) architecture, which is composed of three basic components (Figure86):

- The CGI program;
- The map file;
- The template file.

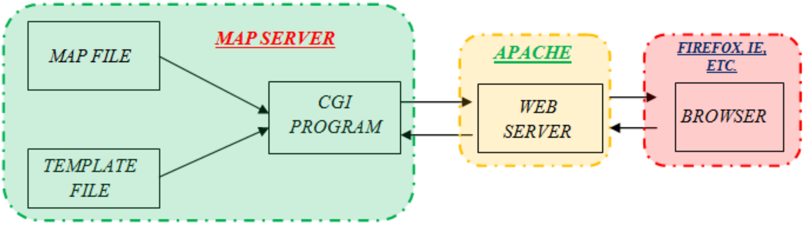


Figure 86 – Components of the webGIS application developed with MapServer

A CGI (Common Gateway Interface) is a program that is activated by a web server (Apache in our case) to perform the processing of cartographic parameters selected by the user with the mouse or using the typical HTML forms and that is the parameters received at input from the mapfile and from the template file. The results of the processing are then presented to the user always within HTML pages, so for the web application is defined GIS engine.

In particular, the CGI program executable by MapServer is called mapserv.exe for Windows and mapserv for UNIX. The mapfile is an ASCII file in which all cartographic parameters are set, including the amplitude of the map, the parameters related to the raster and vector themes. This file, which can be compiled with a common text editor and saved with the extension .map, consists of "objects" structured according to a hierarchical scheme at various levels (Figure 87).

```
# Start of map file
#
MAP
EXTENT  1988372 1400000 6411627 5400000

UNITS meters
#EXTENT -15 30 40 70
#UNITS dd
SIZE 600 500
SHAPEPATH "../../../pmapper_demodata"
SYMBOLSET "../common/symbols/symbols-pmapper.sym"
FONTSET "../common/fonts/msfontset.txt"
RESOLUTION 96
IMAGETYPE png
INTERLACE OFF
#CONFIG "PROJ_LIB" "C:/proj/nad/"
PROJECTION
  # ETRS-LAEA
  #"init=epsg:3035"
  "+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=m +no_defs  no_defs"
END


#
# Image formates for GD
#
OUTPUTFORMAT
  NAME "png"
  DRIVER "GD/PNG"
  MIMETYPE "image/png"
  IMAGEMODE RGB
  FORMATOPTION  INTERLACE=OFF
  TRANSPARENT OFF
  EXTENSION "png"
END
```

Figure 87 – Initial part of the mapfile used in the webGIS application developed in MapServer

In MapServer, the template file is the look and feel of the web page containing the display of thematic raster (mapping support) and vector, and tools, i.e. the tree layer, the reference map, scale bar, the toolbar. However, template files are also used for presenting the results of queries or for signalling errors or constraints imposed on the application, i.e. the operation of the application only in particular scale range or the absence of the Internet connection. In fact the template file is any graphical interface between the webGIS application and the user. It is constituted by a HTML page, provided by JavaScript or others, containing CGI parameters (Figure 88).

```
<!-- WEST -->
<div id="west">
</div>


<!-- EAST -->
<div id="east" class="TOC">
    <!-- LEGEND/TOC -->
    <?php echo writeTocTabs($toctabs) ?>
    <div id="tocContainer">
      <form id="layerform" method="get" action="">
        <div id="toc"       class="TOC" style="<?php echo ($_SESSION['userAgent'] == "mozilla" ? "height:100%" : "height:auto") ?>;"></div>
        <div id="toclegend" class="TOC" style="<?php echo ($_SESSION['userAgent'] == "mozilla" ? "height:100%" : "height:auto;overflow:hidden") ?>; display:none;">DIEEI</div>
      </form>
    </div>
</div>


<!-- MAPZONE -->
<div id="mapZone" class="baselayout">

    <div id="map" class="baselayout">
        <!-- MAIN MAP -->
        <div id="mapimgLayer">
                <img id="mapImg" class="mapImg" src="images/pixel.gif"  style="overflow:hidden;" alt="" />
        </div>
        <div id="zoombox" class="zoombox"></div>
        <div id="measureLayer" class="measureLayer"></div>
        <div id="measureLayerTmp" class="measureLayer"></div>
        <div id="helpMessage"></div>
        <div id="iqueryLayer"></div>
        <div id="loading"><img id="loadingimg" src="images/loading.gif" alt="loading" /></div>
```

Figure 88 – Initial part of the template file used in the webGIS application developed in MapServer

The graphic interface of a webGIS is composed by different parts (Figure 89): a legend area, in which the thematic layers loaded are listed and where operations on the layer (as queries, zoom and transparency on the layer) can be performed clicking with the right mouse button; an ample map area, where the thematic layers are displayed; a small reference map; a toolbar that provides typical tools for the map navigation, generic plugins for the GIS environment, i.e. selection, functionalities on the vector themes, queries with the opportunity to save the results in different formats (Figure 90), or linear or areas measurements (Figure 91), etc.

Figure 89 – WebGIS components: legend area (green), map area (red), reference map (blue) and toolbar (yellow)

Figure 90 – Tool for line or area measurements

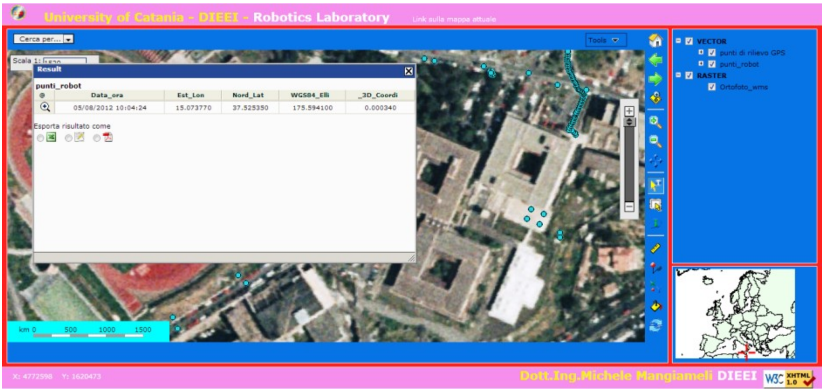


Figure 91 – Tool queries to vector themes with the opportunity to save the results in different formats

In particular, an improvement to the zoom tool compared to the standard one was made: the zoom window has been changed from white to transparent allowing seeing the area under the effect of the zoom window (Figure 92).

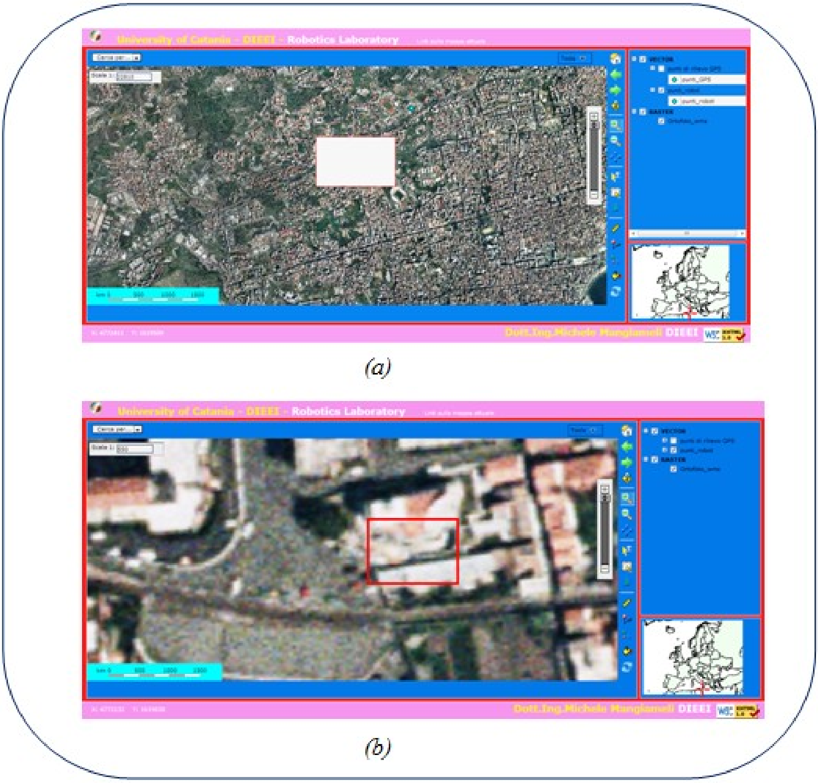


Figure 92 – Optimized zoom window. (a) The white zoon window of the standard platform where the zoomed area cannot be seen; (b) the modified zoom window where the transparence allows visualizing the zoomed area

In the top dropdown menu a printing tool and a plugin developed to dynamically populate the spatial database have been included (Figure 93). A zoom slider, a scale bar were also inserted, with a tool to manually enter the desired scale and a quick search of themes present in legend positioned in the upper left.

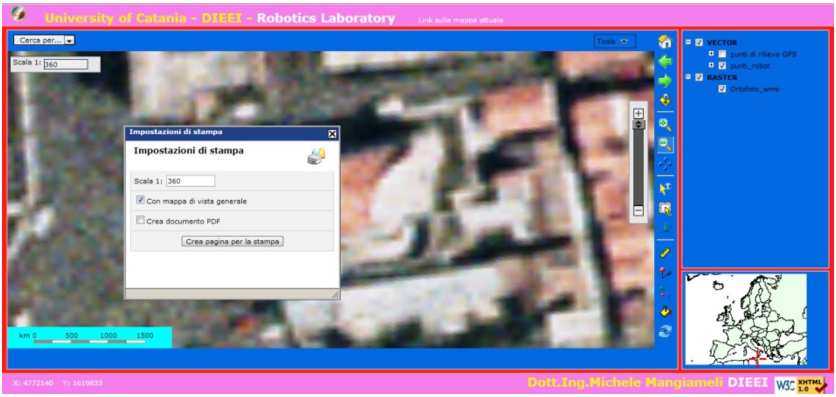


Figure 93 – Printing tool where the print scale can be set and a PDF print can be generated

Using the UML (Unified Modeling Language), can be represented in the application domain of the platforms used, the structure of the architecture and software versions used (Figure 94).

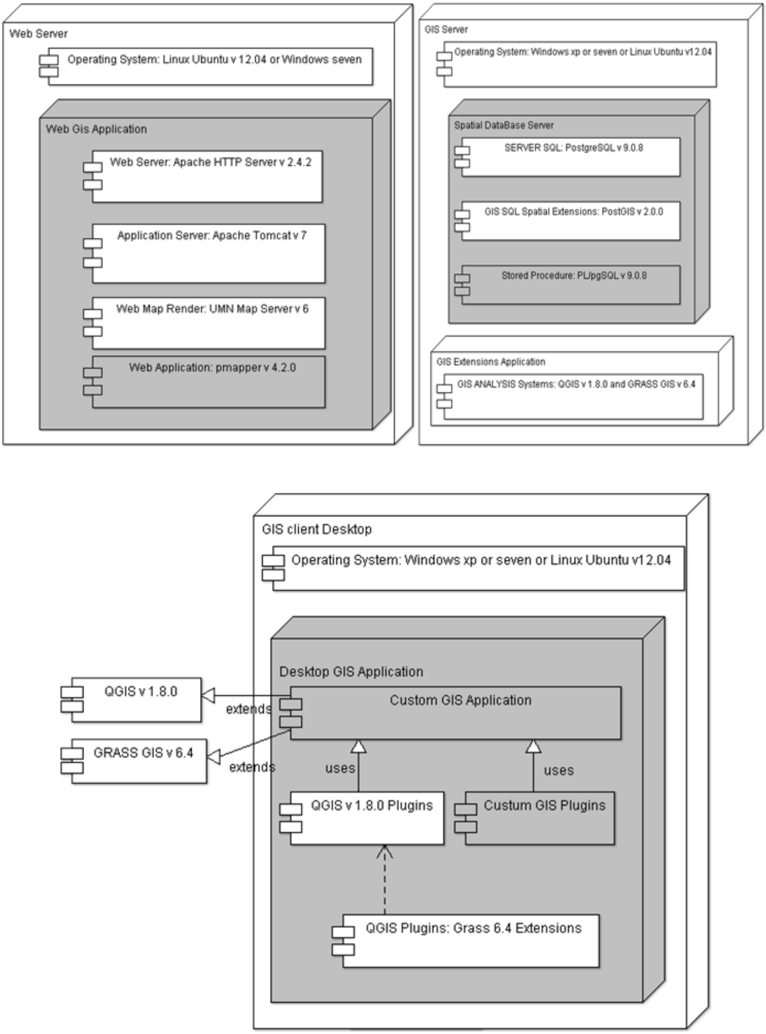


Figure 94 – UML deployment diagram related to GIS applications

4.4 Analysis of the GPS robot signal in the GIS platforms

The geolocalization of a mobile robot in outdoor environments is entrusted to a GPS receiver installed on board the robots.
Obviously the position of the robot movement is not determined in an absolute manner, because in the navigation the error on the position may be even of the order of any ten meters, but uses the technique of positioning NRTK (Network Real Time Kinematic). With this technique errors (ionospheric, tropospheric, of the orbits, watches, multipath) can be determined relative to individual permanent stations that constitute the infrastructure GNSS and on the basis of these the differential corrections of its position are provided to the user in real time [*Melita et al.,* 2012]. Once obtained the robot position with centimeter accuracy, it is necessary to represent it in a georeferenced cartographic support and provide advanced tools to monitor and manage the navigation of the robot. For this purpose we used the GIS technology in which the cartographic support is provided by raster data and the position of the robot can be represented as a vector.
To represent in the GIS environment the robot position as a vector file, the GPS output, that is an NMEA string, is considered and analyzed. The NMEA (National Marine Electronics Association) is a standard data communication mainly used in boating and in the

communication of GPS satellite data. This standard communication protocol consists of alphanumeric strings disposed by rows, where the beginning of each sentence is delimited by a "$", the end of a sequence CR LF and the individual fields are separated by commas for a maximum length of each string is 80 characters.

The prefix, that is the first part of the string, is used to specify what type is the talker, for example, autopilot, GPS device, etc.

For example, in the case of using a GPS device, the prefix is GP, a GLONASS device has a GL prefix, while a GNSS receiver uses the prefix GN, followed by different types of sentences that give information about the visible satellites, minimum information for navigation etc., identified by three letters (e.g.: RMC, RMB, GGA, GSA, etc..) (Figure 95). In the below fields other information are given, such as time of acquisition, the latitude, longitude, etc.

```
$GPGGA,090422.00,3731.5136426,N,01504.4406822,E,4,07,1.1,85.534,M,,,60.43,0003*2A
$GPGGA,090422.00,3731.5136426,N,01504.4406822,E,4,07,1.1,85.534,M,,,61.47,0003*2F
$GPGGA,090422.00,3731.5136426,N,01504.4406822,E,4,07,1.1,85.534,M,,,62.40,0003*2B
$GPGGA,090422.00,3731.5136426,N,01504.4406822,E,4,07,1.1,85.534,M,,,63.52,0003*29
$GPGGA,090422.00,3731.5136426,N,01504.4406822,E,4,07,1.1,85.534,M,,,64.73,0003*2D
$GPGGA,090422.00,3731.5136426,N,01504.4406822,E,4,07,1.1,85.534,M,,,65.42,0003*2E
$GPGGA,090422.00,3731.5136426,N,01504.4406822,E,4,07,1.1,85.534,M,,,66.53,0003*2D
```

Figure 95 – Example of NMEA string

Since the latitude and longitude, as well as any other information shown in NMEA string, are alphanumeric data without a spatial characteristic, it has developed a software procedure to convert the information, relative to the position of the robot, from purely numerical parameters to geographical coordinates.
The procedure involves the use of code developed in PHP language and spatial queries built using SQL language, for taking information from the NMEA string, attach them to a datum and insert into the spatial DB managed by the DBMS PostgreSQL with PostGIS spatial extension (Figure 96).

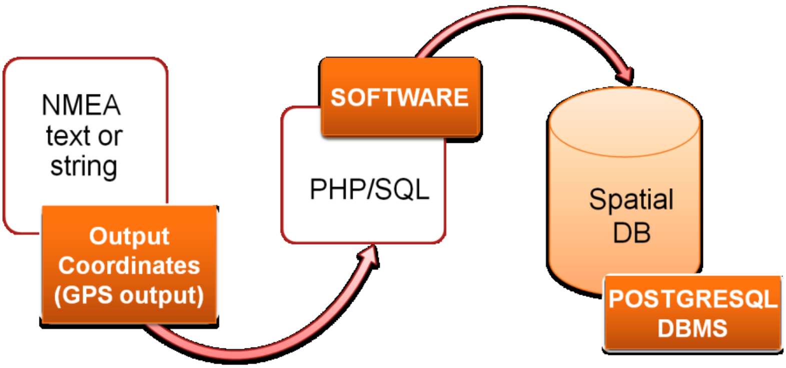
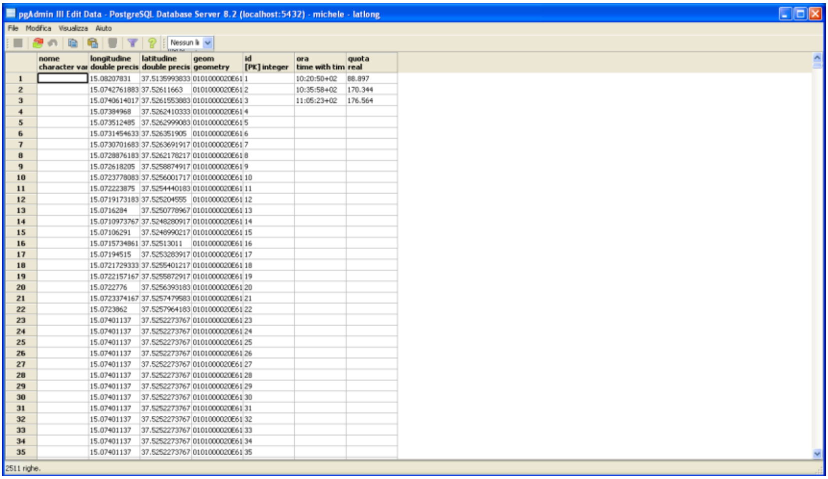


Figure 96 – Software procedure to manage the position of the robot within the spatial database

To explain in detail the software process two phases of operation can be distinguished.

In the first step, a table space within the PostgreSQL DBMS is created that will host the coordinates relative to the position of the robot plus all the information considered useful taken from the NMEA sentence.

In the second phase a PHP code is used that locates in the NMEA string provided as a text file, the columns for latitude and longitude, and using spatial queries executed inside the php code, the topographic datum is associated and subsequently are automatically included in columns latitude and longitude of the spatial table created (Figure 97).



	nome character var	longitudine double precis	latitudine double precis	geom geometry	id [PK] integer	ora time with tim	quota real
1		15.08207831	37.5135993833	0101000020E61	1	10:20:50+02	88.897
2		15.0742761883	37.52611663	0101000020E61	2	10:35:58+02	170.344
3		15.0740614017	37.5261553883	0101000020E61	3	11:05:23+02	176.564
4		15.07384968	37.5262410333	0101000020E61	4		
5		15.073512485	37.5262999083	0101000020E61	5		
6		15.0731454633	37.526351905	0101000020E61	6		
7		15.0730701683	37.5263691917	0101000020E61	7		
8		15.0728876183	37.5262178217	0101000020E61	8		
9		15.072618205	37.5258874917	0101000020E61	9		
10		15.0723778083	37.5256001717	0101000020E61	10		
11		15.072223875	37.5254440183	0101000020E61	11		
12		15.0719173183	37.525204555	0101000020E61	12		
13		15.0716284	37.5250778967	0101000020E61	13		
14		15.0710973767	37.5248280917	0101000020E61	14		
15		15.07106291	37.5248990217	0101000020E61	15		
16		15.0715734861	37.52513011	0101000020E61	16		
17		15.07194515	37.5253283917	0101000020E61	17		
18		15.0721729333	37.5255401217	0101000020E61	18		
19		15.0722157167	37.5255872917	0101000020E61	19		
20		15.0722776	37.5256393183	0101000020E61	20		
21		15.0723374167	37.5257479583	0101000020E61	21		
22		15.0723862	37.5257964183	0101000020E61	22		
23		15.07401137	37.5252273767	0101000020E61	23		
24		15.07401137	37.5252273767	0101000020E61	24		
25		15.07401137	37.5252273767	0101000020E61	25		
26		15.07401137	37.5252273767	0101000020E61	26		
27		15.07401137	37.5252273767	0101000020E61	27		
28		15.07401137	37.5252273767	0101000020E61	28		
29		15.07401137	37.5252273767	0101000020E61	29		
30		15.07401137	37.5252273767	0101000020E61	30		
31		15.07401137	37.5252273767	0101000020E61	31		
32		15.07401137	37.5252273767	0101000020E61	32		
33		15.07401137	37.5252273767	0101000020E61	33		
34		15.07401137	37.5252273767	0101000020E61	34		
35		15.07401137	37.5252273767	0101000020E61	35		

Figure 97 – Geometric table storing data related to the robot position imported from the string NMEA output from the GPS

Using the same procedure other information can be taken from the NMEA sentence, such as the time of acquisition, the altitude etc. To expedite the process, the first phase of the creation of the spatial table in PostgreSQL can be performed directly inside the PHP code by inserting the query space for the creation of geometrical table. The table geometry can thus be used as vector layer in a GIS (Desktop and Web) showing the position of the robot on a georeferenced cartographic support, ortophoto WGS84 (Figure 98), or satellite images (Figure 99).

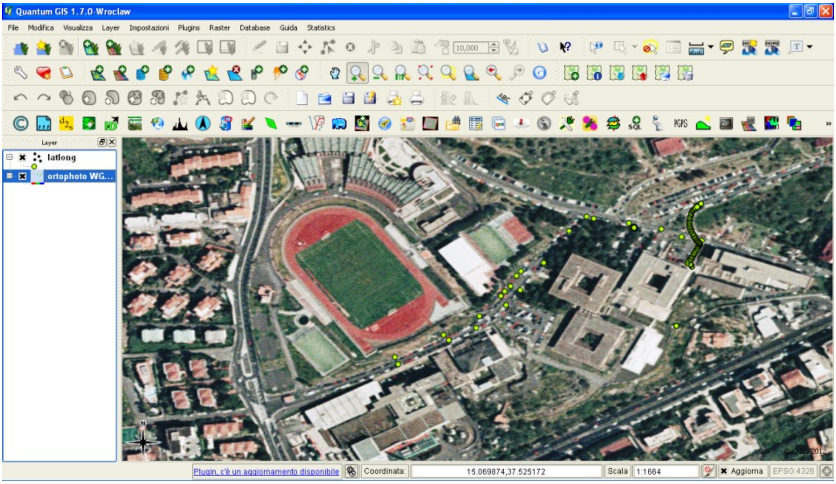


Figure 98 – Geometric table loaded and managed in a GIS environment

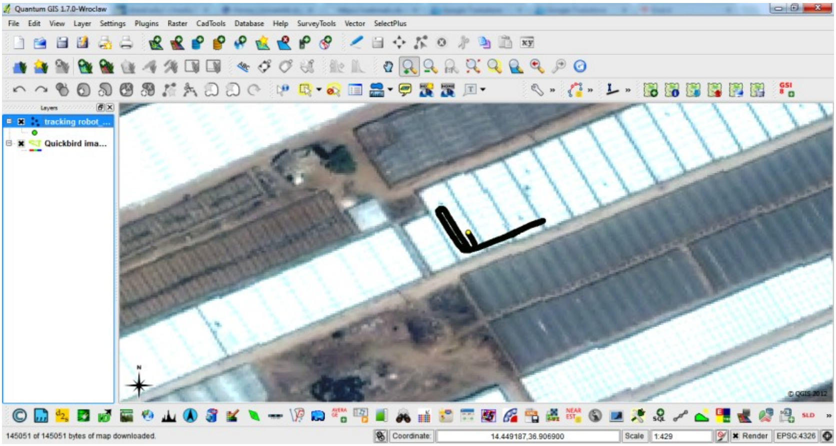


Figure 99 – Tracking robot loaded and managed in a GIS environment

4.5 GIS tools to determine the optimal paths for the robot

The GIS technology lends itself well both for monitoring and controlling the navigation of the mobile robot, but also for planning and managing the operating routes.

This is due to the GIS ability to digitize possible paths for the robot operating on a precise, accurate and updated cartographic support and possibly to relate these routes with additional vector themes that can represent obstacles or geomorphological characteristics of the area that limit the travel of the mobile robot.

In this thesis, software procedures in Grass-GIS, PostGIS and QGIS environments were developed to converting optimal paths (defined in a GIS environment) in waypoints successively assigned to the robot for the autonomous navigation. The conversion consists of extracting the latitude and longitude coordinates from the shape vectors of the optimal paths and of storing them in a spatial database. The coordinates are then saved in a text file and made available to the robot.

In detail the conversion procedure to identify the optimal paths for the autonomous navigation of robot in GIS environment consists of three main steps:

1. Digitization in the desktop GIS platform of possible starting and ending points for the robot together with the paths that can connect them;
2. Use of the “network analysis” tool to determine the optimal path within the GIS Grass environment;
3. Export of the optimal path to the spatial database, and conversion of the path to waypoints successively stored in a text file made available to the robot.

Before the process of digitization in the desktop GIS platform, the cartographic support should be chosen georeferenced with the typical topographic approaches in order to gain a greater accuracy. In particular, for our study, we use a colored WGS84 orthophotos made available as WMS service by the “Sistema Informativo Territoriale Regionale” (SITR) of Sicily, Italy.

Over this cartographic support, we digitized possible starting and ending points in QGIS together with all potential paths that can connect them. The starting and ending points are digitized as vector themes with punctual geometry (see yellow dots in Figure 100) and stored in a typical GIS format called “shape”.

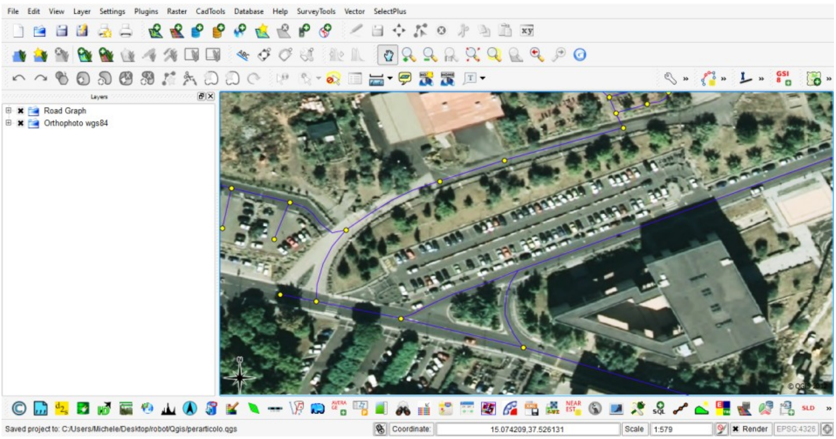


Figure 100 – Screenshot of the desktop GIS platform. On the left, the layer tree containing the legend of vectors and rasters. On the right, the area displaying the cartographic support that is a WGS84 orthophoto and the vector layer that is the road graph

The paths connecting the starting and ending points are digitized as vector themes with linear geometry (see blue lines in Figure 100) and stored in the shape format.

In literature, there are approaches for the construction of road and graph topology maps in a GIS to manage the navigation of mobile robots based on the analysis of maps such as Google Earth.

In this approach the road graph is digitized in GIS environment after designing a relational structure arc-node following the three phases of planning for a DB: conceptual modeling, logical and physical. And the ER diagram that leads to the physical design of the spatial DB is represented in Figure 101.

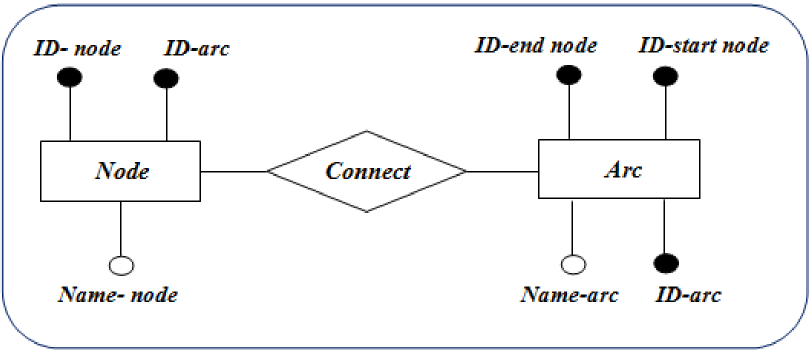


Figure 101 – ER diagram of the topological structure arc-node

To each point and arc, we associated additional information useful for the identification and choice of the optimal path, i.e. the name of the point and arc or the ID of the node and arc. Hence the road graph consists of points (or nodes) representing the possible starting and ending points of the robot navigation and lines (or arcs) constituting potential paths.

The optimal path has been determined within the GIS Grass environment, using the functionality of "network analysis" tool that exploits the road graph constructed as described before. The algorithm that determines the optimal path takes as input the vector file the nodes and vector file of the arches that will be joined and the id-node for the possible starting and ending points (Figure 102).

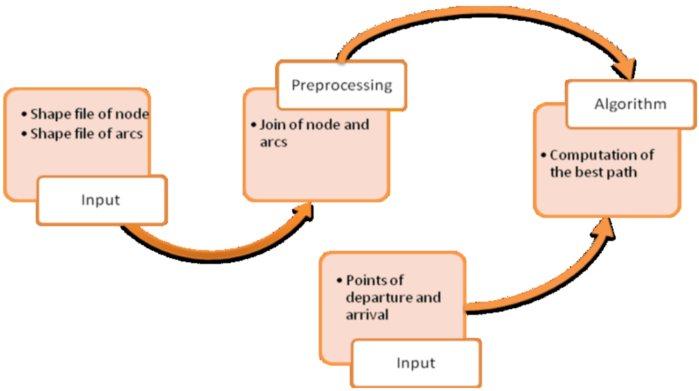


Figure 102 – Input parameters of the algorithm to determine the optimal paths in the GRASS Desktop GIS

The optimality of a path is evaluated on the basis of the shortest distance between the starting and ending points. Otherwise it can be defined as the path having the minimum cost calculated by summing the cost associated to each line which composes the path. The cost of a single line depends on the characteristics of the robot (size, power, locomotion system, equipment…), the characteristics of the operating environment (adherence of the soil, slope, obstacles ...) and objectives of the mission (nature of the investigation, required points, time frames, risk conditions...). To test the algorithm, two possible starting and ending points were chosen (Figure 103).

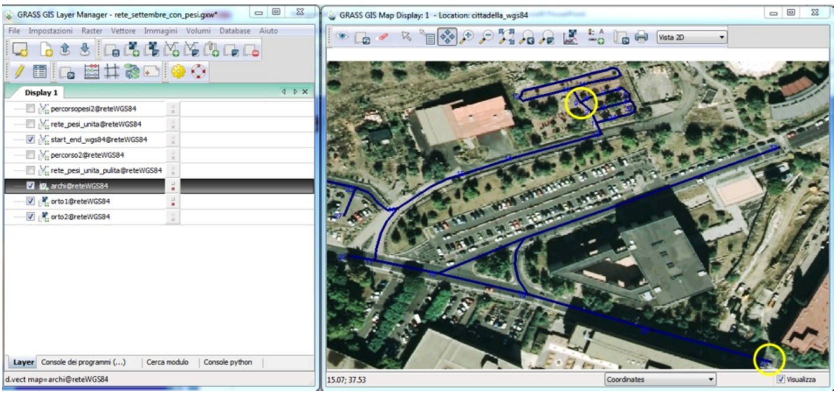


Figure 103 – The yellow rings define two possible starting and ending points provided in input to the algorithm to determine the optimal path

Two tests are performed for determining the optimal paths: with weights assigned to the road graph and without any weights. When no weight is assigned to the road graph, the optimal path is the shortest path between the starting and ending point (Figure104).

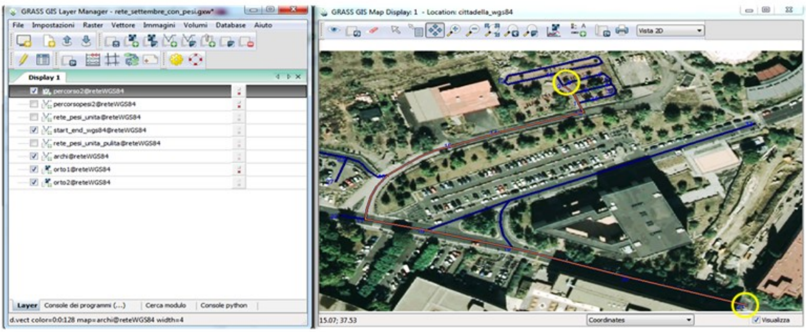


Figure 104 – In red the optimal path determined between two possible starting and ending points without assigning weights to the arcs

When different weights are assigned to the road graph, the optimal path will be determined by discarding the arches with impedance (cost) more (Figure 105). It is worth noting that the optimal path with the weights assigned to the arcs has a greater extent, in fact, the path length without weights is 3.81 km and the path length with weights is 4.26 km (Figure 106).

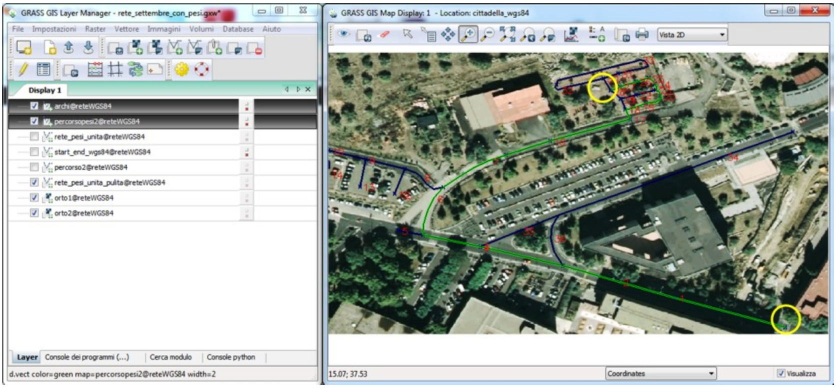
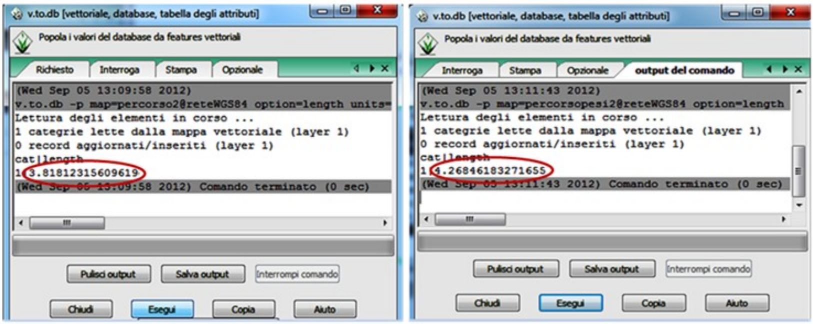


Figure 105 – In green the optimal path determined between two possible starting and ending points with weights assigning to the arcs



(a) *(b)*

Figure 106 – Measurement of optimal paths in GIS environment. (a) Path length without weights and (b) path length with weights

To test if the path obtained in the GIS environment is correct, a topographic survey was performed with the topographic GPS Leica 1250 as rover station in absolute position to determine the position of the receiver with a minimum of 4 satellites. This survey allows acquiring the points in a dynamic way but needs a correction since the accuracy is about 10 meters in kinematic (Figure 107).

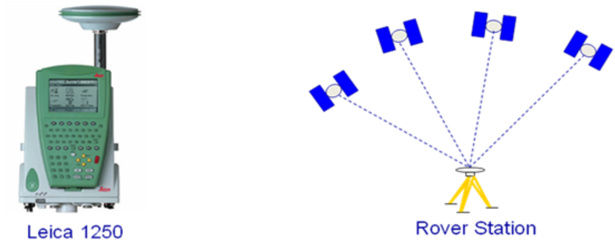


Figure 107 – Rover topographic GPS in absolute positioning

For the correction, another topographic GPS, Trimble 4000, was used as master station with a fixed antenna positioned at the Department of Engineering (Figure 108). This GPS has the same temporal acquisition of the rover GPS, so the rover coordinates are estimated, in postprocessing, with a differential correction of the distances satellite-receiver calculated on the Master Station placed at known coordinates (Figure 109).

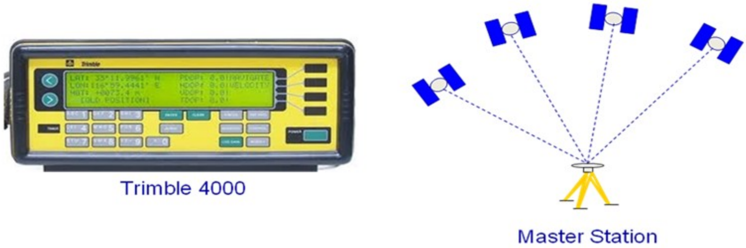


Figure 108 – Master Topographic GPS

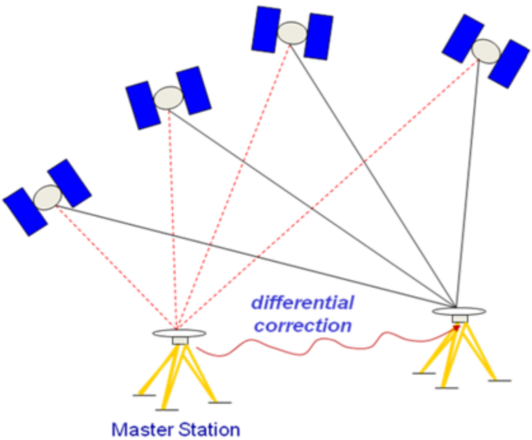


Figure 109 – Differential correction of the distances satellite-receiver calculated on the Master Station placed at known coordinates

The reliability of the algorithm has been validated by comparing the optimal path with the GPS tracks. As evidenced from Figure 110, the GPS track test in blue overlaps in a perfect way the red optimal path obtained in the GIS environment.

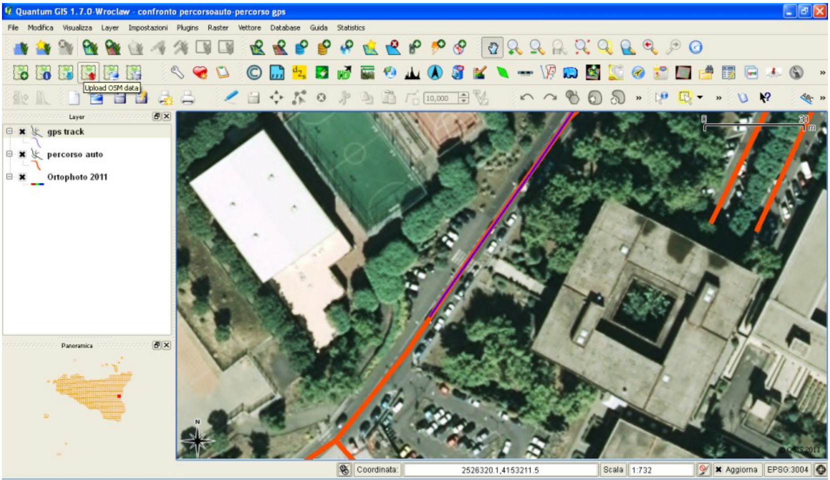
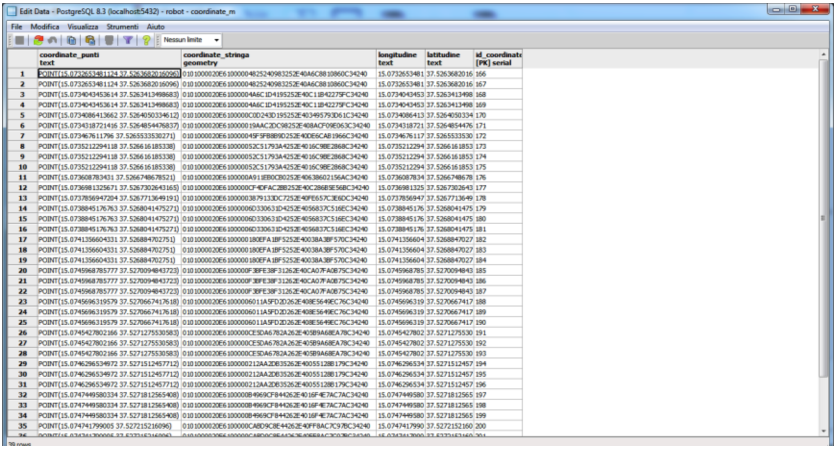


Figure 110 – The GPS track test (blue) overlapping the optimal path obtained in GIS environment (red)

The optimal path is saved as shapefile and exported to the spatial database using SQL spatial queries . The format of the optimal path is an alphanumeric string exported to a row of a table of the spatial database. This string is then converted in a sequence of waypoints, stored in different rows of another table, from which the latitude and longitude coordinates are extracted and stored in two different columns (Figure 111).



Edit Data - PostgreSQL 8.3 (localhost:5432) - robot - coordinate_m

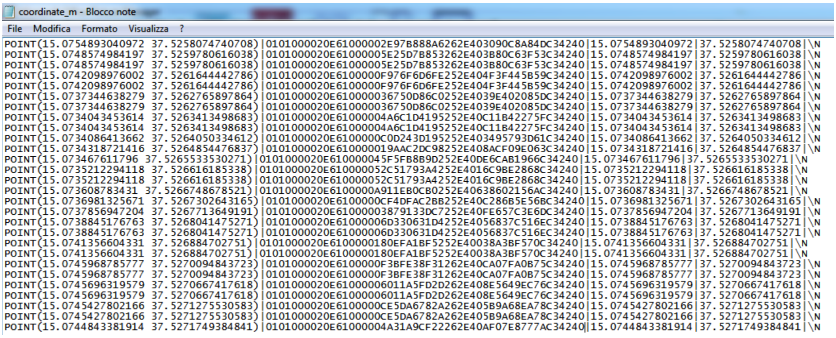
File Modifica Visualizza Strumenti Aiuto

Nessun limite

	coordinate_punti text	coordinate_stringa geometry	longitudine text	latitudine text	id_coordinate [PK] serial
1	POINT(15.0732653481124 37.5263682016096)	0101000020E610000048252409833252E40A6C8810860C34240	15.0732653481	37.5263682016	166
2	POINT(15.0732653481124 37.5263682016096)	0101000020E610000048252409833252E40A6C8810860C34240	15.0732653481	37.5263682016	167
3	POINT(15.073404345361 4 37.5263413498683)	0101000020E61000004A6C1D4195252E40C11B42275FC34240	15.0734043453	37.5263413498	168
4	POINT(15.0734043453614 37.5263413498683)	0101000020E61000004A6C1D4195252E40C11B42275FC34240	15.0734043453	37.5263413498	169
5	POINT(15.0734086413662 37.5264050334612)	0101000020E6100000C0D243D195252E403495793D61C34240	15.0734086413	37.5264050334	170
6	POINT(15.0734318721416 37.5264854476837)	0101000020E610000019AAC2DC98252E408ACF09E063C34240	15.0734318721	37.5264854476	171
7	POINT(15.073467611796 37.5265533530271)	0101000020E610000045F5FB8B9D252E40DE6CAB1966C34240	15.0734676117	37.5265533530	172
8	POINT(15.0735212294118 37.526616185338)	0101000020E610000052C51793A4252E4016C9BE2868C34240	15.0735212294	37.5266161853	173
9	POINT(15.0735212294118 37.526616185338)	0101000020E610000052C51793A4252E4016C9BE2868C34240	15.0735212294	37.5266161853	174
10	POINT(15.0735212294118 37.526616185338)	0101000020E610000052C51793A4252E4016C9BE2868C34240	15.0735212294	37.5266161853	175
11	POINT(15.073608783431 37.5266748678521)	0101000020E6100000A911EB0CB0252E40638602156AC34240	15.0736087834	37.5266748678	176
12	POINT(15.0736981325671 37.5267302643165)	0101000020E6100000CF4DFAC2BB252E40C286B5E56BC34240	15.0736981325	37.5267302643	177
13	POINT(15.0737856947204 37.5267713649191)	0101000020E61000003879133DC7252E40FE657C3E6DC34240	15.0737856947	37.5267713649	178
14	POINT(15.0738845176763 37.5268041475271)	0101000020E61000006D330631D4252E4056837C516EC34240	15.0738845176	37.5268041475	179
15	POINT(15.0738845176763 37.5268041475271)	0101000020E61000006D330631D4252E4056837C516EC34240	15.0738845176	37.5268041475	180
16	POINT(15.0738845176763 37.5268041475271)	0101000020E61000006D330631D4252E4056837C516EC34240	15.0738845176	37.5268041475	181
17	POINT(15.0741356604331 37.526884702751)	0101000020E6100000180EFA1BF5252E40038A3BF570C34240	15.0741356604	37.5268847027	182
18	POINT(15.0741356604331 37.526884702751)	0101000020E6100000180EFA1BF5252E40038A3BF570C34240	15.0741356604	37.5268847027	183
19	POINT(15.0741356604331 37.526884702751)	0101000020E6100000180EFA1BF5252E40038A3BF570C34240	15.0741356604	37.5268847027	184
20	POINT(15.0745968785777 37.5270094843723)	0101000020E6100000F3BFE38F31262E40CA07FA0B75C34240	15.0745968785	37.5270094843	185
21	POINT(15.0745968785777 37.5270094843723)	0101000020E6100000F3BFE38F31262E40CA07FA0B75C34240	15.0745968785	37.5270094843	186
22	POINT(15.0745968785777 37.5270094843723)	0101000020E6100000F3BFE38F31262E40CA07FA0B75C34240	15.0745968785	37.5270094843	187
23	POINT(15.0745696319579 37.5270667417618)	0101000020E61000006011A5FD2D262E408E5649EC76C34240	15.0745696319	37.5270667417	188
24	POINT(15.0745696319579 37.5270667417618)	0101000020E61000006011A5FD2D262E408E5649EC76C34240	15.0745696319	37.5270667417	189
25	POINT(15.0745696319579 37.5270667417618)	0101000020E61000006011A5FD2D262E408E5649EC76C34240	15.0745696319	37.5270667417	190
26	POINT(15.0745427802166 37.5271275530583)	0101000020E6100000CE5DA6782A262E405B9A68EA78C34240	15.0745427802	37.5271275530	191
27	POINT(15.0745427802166 37.5271275530583)	0101000020E6100000CE5DA6782A262E405B9A68EA78C34240	15.0745427802	37.5271275530	192
28	POINT(15.0745427802166 37.5271275530583)	0101000020E6100000CE5DA6782A262E405B9A68EA78C34240	15.0745427802	37.5271275530	193
29	POINT(15.0746296534972 37.5271512457712)	0101000020E6100000212AA2DB35262E400551288179C34240	15.0746296534	37.5271512457	194
30	POINT(15.0746296534972 37.5271512457712)	0101000020E6100000212AA2DB35262E400551288179C34240	15.0746296534	37.5271512457	195
31	POINT(15.0746296534972 37.5271512457712)	0101000020E6100000212AA2DB35262E400551288179C34240	15.0746296534	37.5271512457	196
32	POINT(15.0747449580334 37.5271812565408)	0101000020E6100000B4969CF844262E4016F4E7AC7AC34240	15.0747449580	37.5271812565	197
33	POINT(15.0747449580334 37.5271812565408)	0101000020E6100000B4969CF844262E4016F4E7AC7AC34240	15.0747449580	37.5271812565	198
34	POINT(15.0747449580334 37.5271812565408)	0101000020E6100000B4969CF844262E4016F4E7AC7AC34240	15.0747449580	37.5271812565	199
35	POINT(15.07474179905 37.527215216096)	0101000020E6100000CA8D9C8E44262E40FF8AC7C97BC34240	15.0747417990	37.5272152160	200

Figure 111 – Geometric Table containing the waypoints arranged in two columns obtained by exporting the shape file space in the db of the optimal path determined in GIS environment

The coordinates are finally standardized as a text file (csv, txt...) interpretable and directly usable by the navigation system of the robot (Figure 112).



coordinate_m - Blocco note

File Modifica Formato Visualizza ?

```
POINT(15.0754893040972 37.5258074740708)|0101000020E61000002E97B888A6262E403090C8A84DC34240|15.0754893040972|37.5258074740708|\N
POINT(15.0748574984197 37.5259780616038)|0101000020E61000005E25D7B853262E403B80C63F53C34240|15.0748574984197|37.5259780616038|\N
POINT(15.0748574984197 37.5259780616038)|0101000020E61000005E25D7B853262E403B80C63F53C34240|15.0748574984197|37.5259780616038|\N
POINT(15.0742098976002 37.5261644442786)|0101000020E6100000F976F6D6FE252E404F3F445B59C34240|15.0742098976002|37.5261644442786|\N
POINT(15.0742098976002 37.5261644442786)|0101000020E6100000F976F6D6FE252E404F3F445B59C34240|15.0742098976002|37.5261644442786|\N
POINT(15.0737344638279 37.5262765897864)|0101000020E610000036750D86C0252E4039E402085DC34240|15.0737344638279|37.5262765897864|\N
POINT(15.0737344638279 37.5262765897864)|0101000020E610000036750D86C0252E4039E402085DC34240|15.0737344638279|37.5262765897864|\N
POINT(15.0734043453614 37.5263413498683)|0101000020E61000004A6C1D4195252E40C11B42275FC34240|15.0734043453614|37.5263413498683|\N
POINT(15.0734043453614 37.5263413498683)|0101000020E61000004A6C1D4195252E40C11B42275FC34240|15.0734043453614|37.5263413498683|\N
POINT(15.0734086413662 37.5264050334612)|0101000020E6100000C0D243D195252E403495793D61C34240|15.0734086413662|37.5264050334612|\N
POINT(15.0734318721416 37.5264854476837)|0101000020E610000019AAC2DC98252E408ACF09E063C34240|15.0734318721416|37.5264854476837|\N
POINT(15.073467611796 37.5265533530271)|0101000020E610000045F5FB8B9D252E40DE6CAB1966C34240|15.073467611796|37.5265533530271|\N
POINT(15.0735212294118 37.526616185338)|0101000020E610000052C51793A4252E4016C9BE2868C34240|15.0735212294118|37.526616185338|\N
POINT(15.0735212294118 37.526616185338)|0101000020E610000052C51793A4252E4016C9BE2868C34240|15.0735212294118|37.526616185338|\N
POINT(15.073608783431 37.5266748678521)|0101000020E6100000A911EB0CB0252E40638602156AC34240|15.073608783431|37.5266748678521|\N
POINT(15.0736981325671 37.5267302643165)|0101000020E6100000CF4DFAC2BB252E40C286B5E56BC34240|15.0736981325671|37.5267302643165|\N
POINT(15.0737856947204 37.5267713649191)|0101000020E61000003879133DC7252E40FE657C3E6DC34240|15.0737856947204|37.5267713649191|\N
POINT(15.0738845176763 37.5268041475271)|0101000020E61000006D330631D4252E4056837C516EC34240|15.0738845176763|37.5268041475271|\N
POINT(15.0738845176763 37.5268041475271)|0101000020E61000006D330631D4252E4056837C516EC34240|15.0738845176763|37.5268041475271|\N
POINT(15.0741356604331 37.526884702751)|0101000020E6100000180EFA1BF5252E40038A3BF570C34240|15.0741356604331|37.526884702751|\N
POINT(15.0741356604331 37.526884702751)|0101000020E6100000180EFA1BF5252E40038A3BF570C34240|15.0741356604331|37.526884702751|\N
POINT(15.0745968785777 37.5270094843723)|0101000020E6100000F3BFE38F31262E40CA07FA0B75C34240|15.0745968785777|37.5270094843723|\N
POINT(15.0745968785777 37.5270094843723)|0101000020E6100000F3BFE38F31262E40CA07FA0B75C34240|15.0745968785777|37.5270094843723|\N
POINT(15.0745696319579 37.5270667417618)|0101000020E61000006011A5FD2D262E408E5649EC76C34240|15.0745696319579|37.5270667417618|\N
POINT(15.0745696319579 37.5270667417618)|0101000020E61000006011A5FD2D262E408E5649EC76C34240|15.0745696319579|37.5270667417618|\N
POINT(15.0745427802166 37.5271275530583)|0101000020E6100000CE5DA6782A262E405B9A68EA78C34240|15.0745427802166|37.5271275530583|\N
POINT(15.0745427802166 37.5271275530583)|0101000020E6100000CE5DA6782A262E405B9A68EA78C34240|15.0745427802166|37.5271275530583|\N
POINT(15.0744843381914 37.5271749384841)|0101000020E61000004A31A9CF22262E40AF07E8777AC34240|15.0744843381914|37.5271749384841|\N
```

Figure 112 – Table space in text format obtained from the spatial table in PostgreSQL and directly usable by the navigation system of the robot

Since the robots often operate in environments that are subject to geomorphological variations caused for example of volcanic eruptions, landslides, etc., then this technique of optimal path search can be integrated with information obtained from sensors mounted on board of the robot, i.e. laser scanners, which provide instant information on the geometry of the territory in the neighborhood of the robot that can be georeferenced and used to redetermined the optimal path.

4.6 UAV navigation in the GIS environment

To manage the navigation of UAV robots, not just to trace a path in planimetry, it is essential to manage the path also in terms of altitude in order to identify the optimal route depending on the position and height of possible obstacles present in the operating zone [*Niendorf et al.,* 2011].

To this aim the GIS environment is particularly useful because it allows to have a cartographic support that is accurate and precise for determining the route in planimetry, and to construct maps in which the altitude variation of the obstacles is represented and georeferred.

The main steps to generate raster maps in the GIS environment are:

- Digitization of buildings as vector data shape file;
- Conversion of vector data to raster data using the GIS functionalities;
- Use of the algorithm "spline" to generate a raster that contains the spatial variation of the heights of the buildings.

To test the procedure, a possible area was chosen for UAV navigation presenting buildings as obstacles with different heights. These buildings were digitized in the QGIS environment, associating a corresponding vector layer attribute table that contains an attribute for the height in addition to different information (Figure 113).

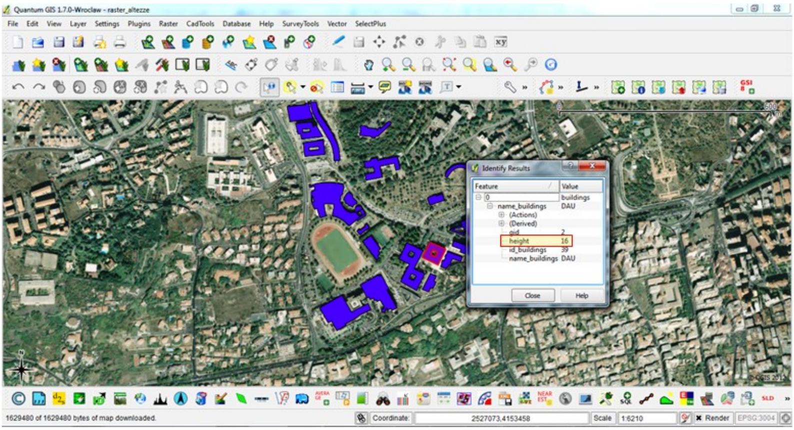


Figure 113 – Buildings in the area chosen for the UAV navigation, digitized in QGIS software and associated to an attribute table that contains different information, including heights

Using the Grass functionalities in QGIS, the vector layer generated is converted into raster data (with the tool "vector to raster") and finally a raster map is generated in which the altitude variation of the buildings is represented, using the algorithm of SPLINE interpolation available between the functionality of surface interpolation related to raster data in GRASS (Figure 114).

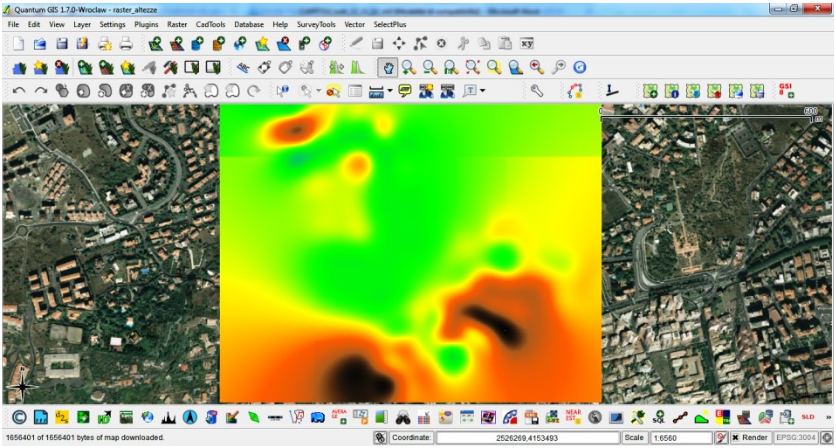


Figure 114 – Map in which the altitude variation of the buildings is represented in the area chosen for UAV navigation

The darkest areas of the raster map in Figure 114 represent buildings with greater heights. Using the properties of the raster layer in GIS environment, the complete legend can be displayed in which the color and the relative height associated to each building are indicated (Figure 115).

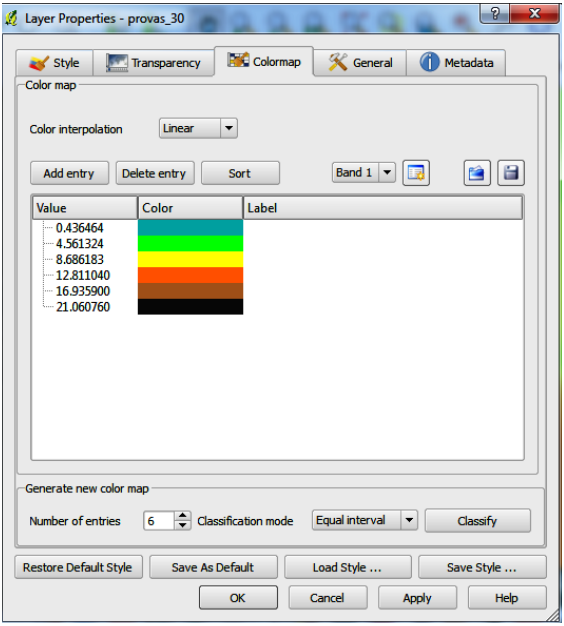


Figure 115 – The color legend in which indicates the color and the related height value associated

Obviously the raster map can be queried in a GIS to obtain information of the pixel queried that is the height value associated with it (Figure 116).

Using this approach, altitude limits can be easily displayed to manage the UAV navigation and to track the planimetric path overlaying the position of obstacles to the raster map of heights (Figure 117).

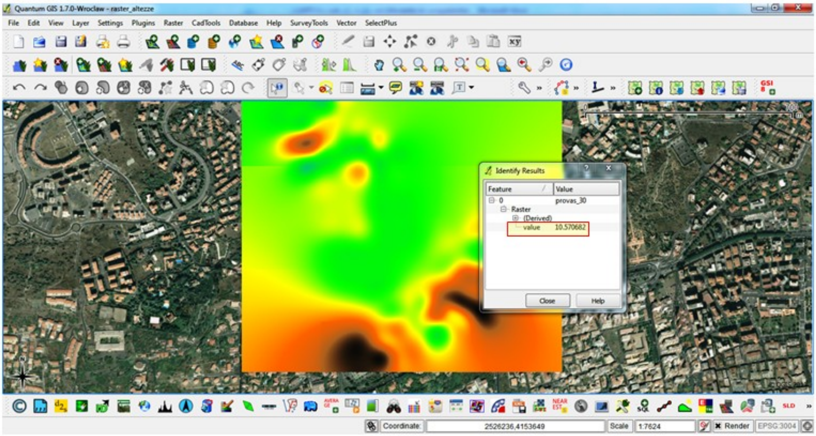


Figure 116 – Query of the map raster resulting in a pixel with a value of about 10,57 meters

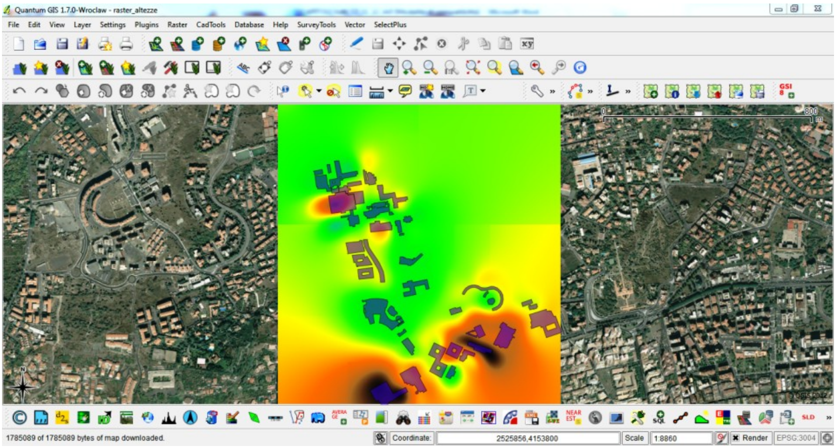


Figure 117 – Vector thematism of buildings overlapped to raster map

4.7 The WebGIS platform to assign the optimal path to the robot

WebGIS platform developed using the free and open source software MapServer allows to visualize, query, and perform other functions typical of GIS technology web side. However some limitations are present: In particular, the use of cartography support loaded by WMS service is dependent on the speed of the internet connection and this results in long response times for the user who needs to perform dynamic operations on raster data support (even if good for static views of layers overlapped to it). Another limitation is that the webGIS platform in MapServer does not allow digitizing and saving the paths as vector themes.

For overcoming these limitations and provide an environment independent webGIS able to manage the navigation of mobile robots, another webGIS has been developed with the free JavaScript library OpenLayers [http://openlayers.org/].

The webGIS platform so designed is characterized by a large map area, latitude longitude of the possible paths digitalized displayed in the left side, a retractable layer tree in the right side, tools at the top bar, and a reference map at the bottom where the map area is represented with respect to a global map (Figure 118).

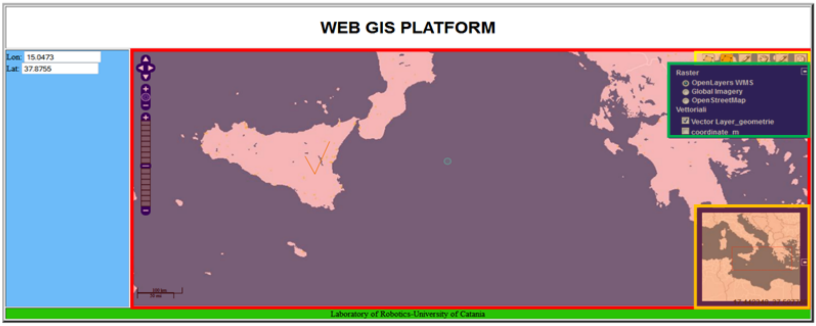


Figure 118 – WebGIS platform developed with the free JavaScript library OpenLayers. The components available are: legend (green), map area (red), reference map (orange), and toolbar (yellow)

As for the cartographic support, Google Maps were not used due to limitations in terms of services and because they are often not very precise and accurate. For this reasons, three cartographic supports were chosen available on remote servers that have excellent characteristics in terms of precision and accuracy. These raster supports are: Generic base image, Satellite global image and Open street map (Figure 119).

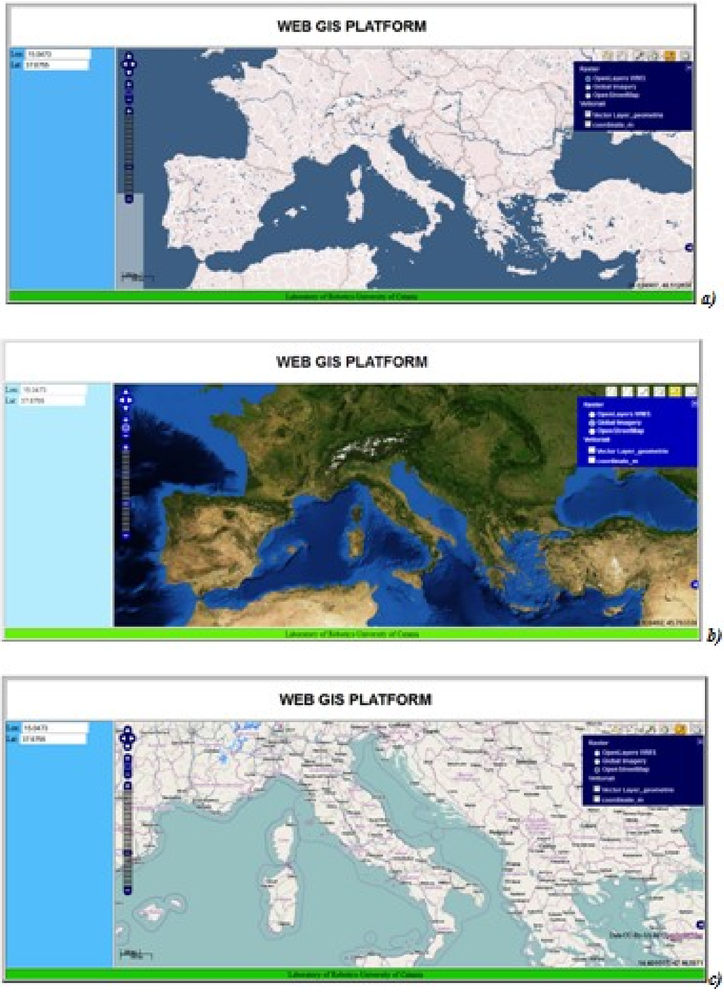


Figure 119 – Cartographic supports to the Open Layers webGIS platform: (a) Generic base image, (b) Satellite global image and (c) Open street map

The generic base image is a global representation of the Earth, the satellite global image is a high resolution global image mosaic of the earth, produced from more than 8200 individual Landsat7 scenes and OpenStreetMap (OSM) is a detailed and accurate cartography created by a global collaborative project to collect geographic data. These supports raster can be chosen directly by selecting in the layers tree.

Since the cartography is provided on a projected coordinate system type encoded with a Google Mercator EPSG: 900013, a coordinate transformation is performed for setting the WGS84 datum encoded with an EPSG: 4326 used by the GPS receiver. By performing this conversion, the position of the mobile robot can be perfectly georeferenced on the cartographic support and conversly a path digitalized on the raster support can be saved as waypoints in latitude and longitude and provided with the navigation system of the mobile robot.This transformation was performed directly in the source code using the commands provided by the JavaScript library OpenLayers (Figure 120).

The zoom functionality on the map are managed through a zoom slider and a navigator to move the map. With respect to the standard configuration provided by the JavaScript library OpenLayers, a zoom to full extension was inserted between the slider and the zoom navigator (Figure 121).

```
function init() {

    map = new OpenLayers.Map ("map", {
        controls:[
        new OpenLayers.Control.Navigation(),
                    new OpenLayers.Control.PanZoomBar(),
                    //new OpenLayers.Control.LayerSwitcher({'ascending':false})
                    new OpenLayers.Control.ScaleLine(),
                    //new OpenLayers.Control.Permalink('permalink'),
                    new OpenLayers.Control.MousePosition(),
                    new OpenLayers.Control.Attribution(),
                    new OpenLayers.Control.OverviewMap(),//referencemap
                    new OpenLayers.Control.KeyboardDefaults()
                ],
        projection: new OpenLayers.Projection("EPSG:900913"),
        displayProjection: new OpenLayers.Projection("EPSG:4326")
        } );
```

Figure 120 – Coordinate conversion performed within the source code

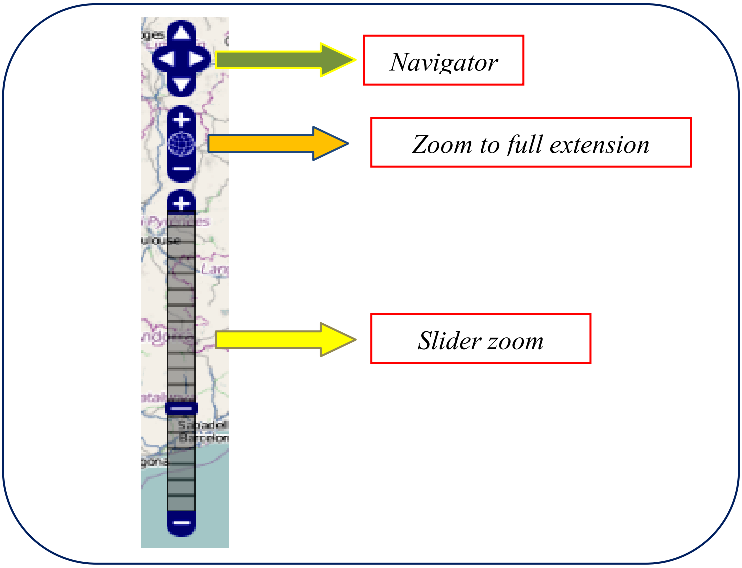


Figure 121 – Zoom and navigation functionalities of the GIS platform

Moreover two bars were inserted on the left and right bottom corners for scaling and for dynamically representing the coordinates when the mouse moves on the map (Figure 122).

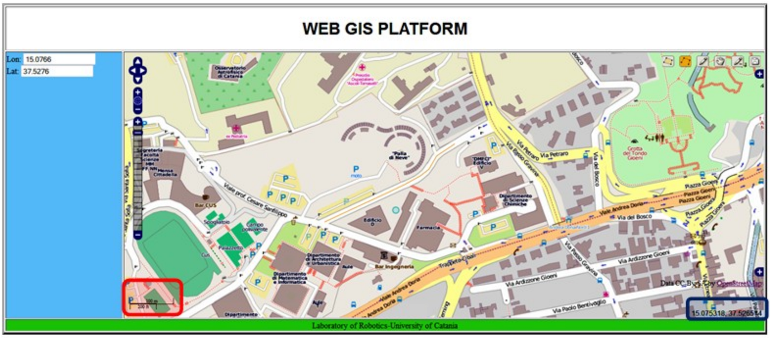


Figure 122 – Scale bar (red) and dynamic coordinates (blue) in the GIS platform

The toolbar located at the right top of the graphic interface contains standard icons associated with functionalities specifically developed to manage the navigation of mobile robots. Using the standard libraries, simple geometric primitives can be digitized on the cartographic support, even if they are simple drawings, not usable as georeferenced spatial data. To overcome this limitation, a JavaScript and PHP application was implemented to digitize geometric primitives (points, lines and polylines) on the raster supports available in the webGIS platform that can be managed as vector layers. In this way they are characterized by spatial information, i.e. the latitude and longitude of each vertex, which can be saved as waypoints in text file used by the navigation system of the robot. With this procedure webGIS platform can be used to assign the path to the robot.

By clicking on the toolbar, polylines, paths or individual points can be so digitized. Other functionalities have been implemented to edit these these geometries, i.e. tool "*pan*" for moving, the "*delete feature*" for removal and "*save waypoints*" for saving them in a text file (Figure 123).

The vector themes created can be turned on or off using the layer tree.

Also the platform has been designed to display other thematic vector managed as shapefile, connect to GeoServer or layer PostGIS.

Inside of the webGIS, additional plugins can be inserted to enrich the platform with typical GIS functionality, optimizing and customizing JavaScript library provided by OpenLayers.

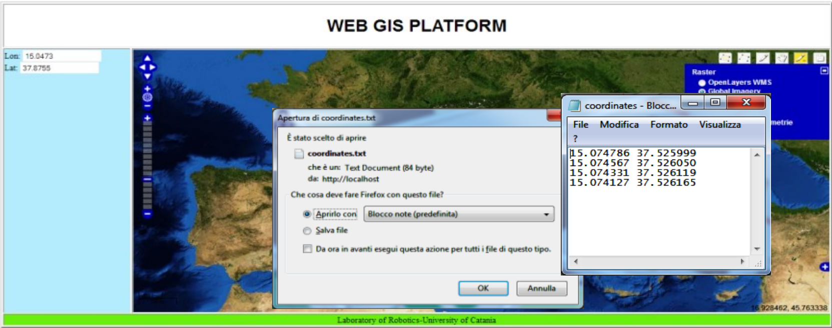


Figure 123 – Tool for saving the waypoints path as txt file

The accuracy of the raster cartographic support and the procedure for the determination of paths in the web interface have been verified uploading the points of the path determined in the web platform (in the open street map) on geo-referenced orthophotos with datum WGS84 in QGIS. The path determined in the webGIS environment overlaps with a great accuracy and precision that on the ortophoto in desktop GIS (Figure 124).

The navigation on the cartographic support is fast so the waiting time for the user are reduced significantly compared to the architecture of WeGis MapServer. Therefore the webGIS platform made is a valid instrument for the management of the navigation of mobile robots.

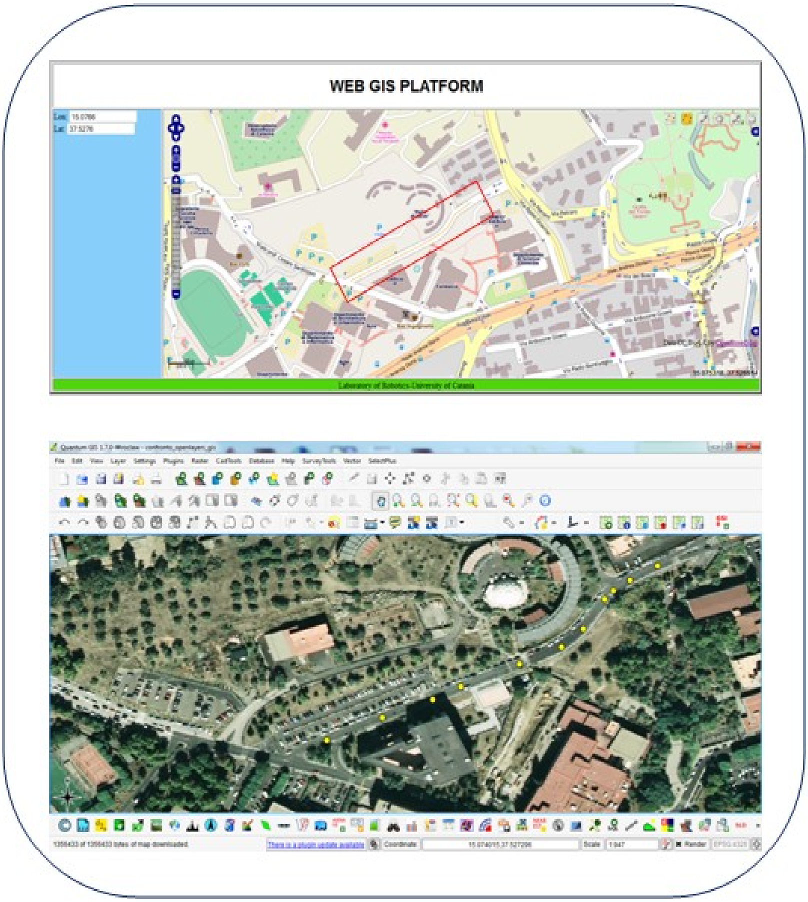


Figure 124 – Path determined in the webGIS platform overlapped as waypoints on an ortophoto with a WGS84 datum in a desktop GIS environment

Conclusions

The use of autonomous robots is a valuable tool for the exploration of hostile environments and sites affected by natural disasters, i.e. unstable areas or structures, underground cavities, polluted sites, buildings at risk of collapse, areas affected by volcanic phenomena. The navigation of the robot is usually monitored and controlled by inertial platforms and in outdoor applications, including satellite positioning.

The variety of geo-mapping supports and of different reference systems available nowadays, makes it very useful, even necessary, to resort to the use of GIS (Geographic Information System) environments for the correct interpretation of georeferencing and assigned routes and trajectories actually covered.

The GIS allows to manage the navigation of the robot on the bases of correctly georeferenced spatial data and to acquire any further information on the same reference system from the sensors installed aboard the robot. The GIS technology can be very useful also for the management of navigation in closed environments, for which it is possible to imagine the construction of a spatial information system to serve as base information support, similarly to what happens in the external environment with the use of cartographic maps, orthophotos, satellite images and digital elevation models.

The aim of this thesis is to provide free and open source GIS technologies supported by a spatial database for the management of the spatial navigation of mobile robots. The proposed architecture is characterized by a free and open source spatial database that communicates both with the GIS platform and the robot.

The spatial database used to manage the spatial data provides different advantages and characteristics which are exploited by the GIS application and the robot. The robot acquires its position through a GPS installed aboard, which output is a NMEA string. This information is stored inside the spatial database using a software procedure expressly developed in PHP and SQL that automatically convert the purely numerical parameters of the NMEA string to geographical coordinates associated to an appropriate geographic datum. Obviously, using this procedure, different kinds of information can be deduced and inserted into the spatial database from the NMEA string (as time, altitude, etc.) or from the sensors aboard the robot (as temperature, structure, images, etc.) .

The two free and open source desktop GIS platforms used in this thesis are QGIS and Grass GIS. QGIS is characterized by an easy and intuitive user interface, and by the ability to map evolved functionalities for the management of geo-referenced maps. This desktop platform can be easily managed and there is a spatial database repository to download plugins that enable it to enrich the platform with advanced tools.

GRASS is characterized by a graphic interface that is not very intuitive and user friendly, but allowing performing advanced analysis in GIS environment such as environmental analysis, hydrological modeling, and network analysis. This latter functionality was used for the management of navigation of mobile robots to determine optimal paths where the optimality of a path can be evaluated as the shortest distance between the starting and ending points or as the path having the minimum cost calculated by summing the cost associated to each arc which composes the path. The cost of a single arc will depend of the characteristics of the robot, the characteristics of the operating environment and the objectives of the mission.

In recent years the use of flying robots, especially of UAVs, has become of huge interest due to their great reliability, to the reduction of costs and to the intrinsic characteristics of these flying platforms often of very contained dimensions. Thanks to all these characteristics, flying robots have become ideal tools for monitoring the territory and hostile environments.

The scientific literature is full of UAV applications in military and civil fields, because this technology permits to monitor large areas subjected to natural disasters, i.e. landslides, floods, polluted areas, forest fires, or to cooperate with terrestrial robots. The use of UAVs provides considerable advantages, such as the reduction of the number of operators at possible risk or the decrease of the intervention time. In these cases, the GIS technology helps to provide accurate and precise cartographic supports for the management of the FVO mission, as well as utile tools for the mission planning.

This thesis provides Desktop and Web GIS platforms containing cartographic supports georeferenced with the accurate approaches of topography. Moreover a procedure for the management of the UAV autonomous navigation has been developed, in terms of both planimetry and altimetry by generating a georeferenced raster map that represents the variation in altitude of the obstacles. In this way, flight plans for the UAVs can be defined in real-time in relation to the possible obstacles present in the area.

As the desktop GIS applications can be used only on the machines where they are installed, two webGIS platforms were developed to provide GIS applications to the user using the Internet network.

The first platform has been developed using the software MapServer. Thanks to the fact it is open source, it was customized using cartographic supports from remote servers, and connected to a PostGIS spatial database. The test platform has highlighted the limitations in performance caused by long waiting time for the user in the use of data from remote servers.

To overcome these limitations another platform webGIS has been designed and implemented with the free JavaScript library OpenLayers. This platform is characterized by high performance of mapping, as the navigation on the cartographic support by the user is rapid with very reduced waiting times in addition to being accurate and precise, but in its standard configuration it has limitations for the management of vector themes, which are simple drawings. To add informative contents, a JavaScript-PHP application was implemented for digitizing the geometric primitives on the raster supports available in the webGIS platform and managing the spatial information in terms of latitude and longitude for each vertex of the primitives digitized. They can be saved as waypoints in text file and used by the navigation system of the robot. So the webGIS is a valid tool to manage the navigation of mobile robots, with the possibility to assign individual tracks or waypoints on a georeferenced cartographic support to the robot using a web connection.

In conclusion, the applications developed using GIS technology is a valid tool for the management of autonomous navigation of robots. Compared to the common mapping platforms, GIS allow using more accurate and precise cartographic supports since they are georeferenced with algorithms and approaches typical of topography, to structure the data in a relational manner in order to manage the different types related to the navigation data of the robot, from sensors installed on board and on the environment in which the robot operates. Moreover GIS allow using different types of data simultaneously. A great difference compared to the common platforms mapping is the ability to use typical functionalities of GIS technology, as spatial analysis and geo-processing functionalities. In addition, the possibility of using free and open source software for developing GIS application provides economic advantages since they can be freely downloadable from the web and open source allowing the application to be customized according to the scope of use.

Bibliography

Arulampalam S., Maskell S., Gordon N. J., Clapp T. (2001). A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking, IEEE Transactions of Signal Processing, Vol. 50, No.2, pp.174-188, 2001, ISSN 1053-587X.

Astuti G., Giudice G., Longo D., Melita C. D., Muscato G., Orlando A. (2009) "An overview of the "Volcan Project": an UAS for exploration of volcanic environments". Journal of Intelligent and Robotic systems, Springer, Vol.54, 471-494, doi: 10.1007/s10846-008-9275-9.

Austin, D.J., and Jensfelt, P. (2000). Using multiple Gaussian hypotheses to represent probability distributions for mobile robot localization. Proc. of the Int. Conference on Robotics and Automation ICRA-00, San Francisco, CA, USA, pp.1036-1041.

Baltzakis H. and Trahanias P. (2002). Hybrid mobile robot localization using switching state-space models. Proceedings of the 2002 IEEE International Conference on Robotics and Automation. Washington D.C., USA, pp.366-373.

Barbieri M., Coratza P., Piacentini D. (2008). Mem. Descr. Carta Geol. d'It.LXXVIII, pp.7-12, fig. 5 - tab. 1.

Biondi G., Condorelli A., Maugeri M., Mussumeci G. (2004). Earthquake-triggered landslide hazard in the Catania area - 4th International Conference on Computer Simulation in Risk Analysis and Hazard Mitigation RISK ANALYSIS 2004 (Wessex Institute of Technology), pp. 115-130, 27-29 Settembre 2004, Rhodes (Greece).

Borenstein J., B. Everett, and L. Feng (1996). Navigating Mobile Robots: Systems and Techniques. A. K. Peter.

Borenstein J., B. Everett, and L. Feng (1996). Book Systems and Methods for Mobile Robot Positioning. Chapter 8 MAP-BASED POSITIONING.

Borenstein, I., Liqiang Fag(1996). Measurement and correction of systematic odometry errors in mobile robots, IEEE Transaction on Robotics and Automation, vol.12, Issue:6, pp.869-880, Dec.1996.

Carlson J. (2010). Mapping Large, Urban Environments with GPS-aided Slam. PhD dissertation.

Chen R. and Liu J. S. (2000). Mixture kalman filters. Journal of the Royal Statistical Society B 62(3), pp.493–508,.

Cox I.J. and G. Wilfong (1990), Autonomous Robot Vehicles, Springer-Verlag.

Cox I.J., NEC Res. Inst., Princeton NJ. (1991). Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. Robotics and Automation, IEEE Transactions on. Volume: 7 , 2, pp.193- 204. Product Type: Journals & Magazines.

Del Maschio L., Pizziolo M., Gozza G., Piacentini D.(2004). Una metodologia integrata in ambiente gis per l'analisi dei fenomeni di crollo: il caso studio di Monte delle Formiche(BO). In "Il Geologo dell'Emilia-Romagna" - Anno IV . N. 19, pp.43 – 51.

Doucet A., Cambridge University, Cambridge, United Kingdom; De Freitas N., University of California, Berkeley, CA, USA; Gordon N., Defence Evaluation and Research Agency,

Malvern Worcs, UK (Eds.)(2001). Sequential Monte Carlo Methods in Practice. Springer, Series: Information Science and Statistics, XIV, 581 p.168 illus.

Engelson S. and D. McDermott (1992). Error correction in mobile robot map learning. In ICRA-92.

Fox D., Burgard W., and Thrun S. (1998). Active Markov localization for mobile robots. Robotics and Autonomous Systems, 25, pp. 195-207.

Galetto R., Spalla A. (1998). Lezioni di topografia. CUSL (Pavia).

Giordano F., Mattei G., Peluso F., Troisi S. (2003). Atti 7a Conferenza Nazionale ASITA "L'Informazione Territoriale e la dimensione tempo" (2003).

Gordon, N.J.; Salmond, D. J. and Smith, A.F.M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEEE Proceedings F on Radar and Signal Processing 140 (2): pp.107–113. doi:10.1049/ip-f-2.1993.0015.

Gutmann J.-S., Weigel T., Nebel B.(2001). A fast, accurate and robust method for self-localization in polygonal environments using laser range finders. Springer Advanced Robotics (2001) 14: pp. 651-667.

Iocchi, L. and Nardi, D. (1999). Hough trasformbased localization for mobile robots. In Mastorakis, N., editor, Advances in Intelligent Systems and Computer Science, pp.359–364. World Scientific Engineering Society.

Meguro J., Ishikawa K., Hashizume T., Takiguchi J., Hatayama M. (2006). Disaster Information Integration into Geographic Information System using Rescue Robots. Intelligent Robots and

Systems, 2006 IEEE/RSJ International Conference, pp.3514 – 3520.

Kalman R.E. (1960). A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering, 82(Series D): pp.35–45.

Khatib, O.; Commande dynamique dans l'espace operationnel des robots manipulateurs en presence d'obstacles, PhD thesis, L'Ecole Nationale Superieure de l'Aeronautique et de l'Espace, 1979.

Kiriy E. and M. Buehler (2002). Three-state Extended Kalman Filter for Mobile Robot Localization.

Kwok L.F. and Longley D.(2004). Security Modelling for Risk Anal ysis. Proc. 18th IFIP World Computer Congress. IFIP 2004, 22-27 August 2004, Toulouse, France, p.2945.

Lee H.S. and K.M. Lee (2009). Multi-robot SLAM using ceiling vision. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pages 912–917, St. Louis, USA, October 2009.

Lee, Y.T. Kwon, and J. Song (2007). SLAM of a Mobile Robot using Thinning-based Topological Information," International Journal of Control Automation and Systems, vol. 5, no. 5, pp.577.

León A., R. Barea, L.M. Bergasa, E. López, M. Ocaña and D. Schleicher (2008). Multi-robot SLAM and Map Merging, IX Workshop of Physical Agents, pp.171-176.

Leonard J. J. and H. F. Durrant-Whyte (1992). Directed Sonar Sensing for Mobile Robot Navigation, Kluwer Academic Publishers, Boston, MA, 1992.

Livatino S., Muscato G., Privitera F., Stereo Viewing and Virtual Reality Technologies in Mobile Robot Teleguide, IEEE Transactions on Robotics, Vol.25,No.6, December 2009 (DOI: 10.1109/TRO.2009.2028765).

Makarenko A., Williams S., Bourgault F., and Durrant-Whyte H.F. (2002). An Experiment in Integrated Exploration. In Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), pp.534–539, Lausanne, Switzerland, October 2002.

Martinez R.-Cantin, De Freitas N., and J.A(2006). Castellanos. Marginal-SLAM: A Convergent Particle Method for Simultaneous Robot Localization and Mapping.

Maugeri M., Motta E., Mussumeci G., Raciti E. (2010). GIS Techniques in the Evaluation of Pipeline Network Seismic Hazard. In: M. KONECNY, S. ZLANATOVA, T.L. BANDROVA EDS.. Geographic Information and Cartography for Risk and Crisis Management. Towards better solutions. pp. 143-154, BERLIN, HEIDELBERG:Springer, ISBN: 978-3-642-03441-1, doi: 10.10007/978-3-642-03442-8

Maugeri M., Mussumeci G, Panarello G., Raciti E. (2009). Evaluation of seismic hazard of water networks by GIS techniques. In: Joint Symposium ICA-JBGIS "Cartography and Geoinformatics for Early Warning and Emergency Managements: Towards Better Solutions - January, 19-22, 2009, Prague, Czech Republic - Proceedings. Prague, January, 19-22, 2009, pp. 404-411, BRNO:© Masaryk University, ISBN: 978-80-210-4796-9.

Melita C.D., Muscato G., Poncelet M., "A Simulation Environment for an Augmented Global Navigation Satellite System Assisted Autonomous Robotic Lawn-Mower", Accepted to Journal of Intelligent and Robotic Systems, Springer, 2012, Published Online: DOI 10.1007/s10846-012-9770-x .

Muscato G., Bonaccorso F., Cantelli L., Longo D. and Melita C.D.. "Volcanic environments: Robots for Exploration and Measurement", IEEE Robotics & Automation Magazine, Vol.19, No.1, March 2012, pp.40-49, DOI: 10.1109/MRA.2011.2181683.

Mussumeci G, Condorelli A, Maugeri M, Biondi G (2004). Earthquake-triggered landslide hazard in the Catania area . In: Risk Analysis IV. WIT TRANSACTIONS ON ECOLOGY AND THE ENVIRONMENT, vol. 77, pp. 115-130, WIT Press, ISBN: 1-85312-736-1, ISSN: 1743-3541, doi: 10.2495/RISK040111.

Niendorf M., Bednary M. and Reichelz R. (2011). 3D Interpolation-Based Path Planning for a Fixed-Wing Unmanned Aircraft. In proceeding of: RED-UAS 2011.

Peca M.(2009). Ultrasonic Localization of Mobile Robot Using Active Beacons and Code Correlation. Research and Education in Robotics - EUROBOT 2009. Communications in Computer and Information Science Volume 82, 2010, pp.116-130.

Rouveure R., Faure P., Monod M.O.(2009). Radar imager for perception and mapping in outdoor environment (2009). Springer-Verlag.

Shair, S., Chandler, J.H., Gonzalez-Villela, V.J., Parkin, R.M., Jackson, M.R.(2008). The Use of Aerial Images and GPS for Mobile Robot Waypoint Navigation. Mechatronics, IEEE/ASME Transactions on, Volume: 13, Issue: 6, doi: 10.1109/TMECH.2008.2005197, pp. 692- 699. .

Smith R. C. and P. Cheeseman (1986). On the representation and estimation of spatial uncertainty. Int. J. Robotics Research, 5(4).

Stachniss C. (2009). Robotic Mapping and Exploration, Series: Springer Tracts in Advanced Robotics, Vol. 55, XVIII, 198 p.89 illus.

Stasse O., A.J. Davison, R. Sellaouti and K. Yokoi (2006). Real-time 3D SLAM for humanoid robot considering pattern generator information. In IEEE/RSJ International Conference of Intelligent Robots and Systems.

Stefanini L.(2006). I sistemi informativi geografici (GIS) e le nuove applicazioni dell'informazione georeferenziata. In Argomenti, 16, pp.65 – 93.

Talluri R., Aggarwal J.K.(1993). Image Map Correspondence for Mobile Robot Self-Location Using Computer Graphics. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 15 no. 6, pp.597-601.

Thrun S., D. Fox, W. Burgard, and F. Dellaert (2000). Robust monte carlo localization for mobile robots. Artificial Intelligence, 128 (1-2).

Williams S. B., H. F. Durrant-Whyte, and T. Baily (2002). Map management for efficient simultaneous localization and mapping (SLAM), Autonomous Robots, vol. 12, no. 3, pp.267-286, May 2002.