

Geographic Information Systems

Manuel Campagnolo

Instituto Superior de Agronomia, Universidade de Lisboa

2013-2014

What is GIS?

Definition (Geographic Information System)

A GIS is a computer-based system to aid in the collection, maintenance, storage, analysis, output and distribution of spatial data and information.

In this course, we will focus mainly in collection, maintenance, analysis and output of spatial data.

Main goals of the course:

- 1 Understand some basic principles of Geographic Information Science behind GIS;
- 2 Become familiar with the use of GIS tools (in particular QGIS);
- 3 Prepare yourself to undertake new analyses using GIS beyond this course.

Major topics of the course

- 1 Abstracting the World to Digital Maps
- 2 Working with Data in GIS
- 3 Coordinate Reference Systems
- 4 Vector Analyses
- 5 Raster Analyses
- 6 GIS Modeling
- 7 Collecting Data

References:

- 1 Class slides;
- 2 Paul Longley, Michael Goodchild, David Maguire and David Rhind, Geographic Information Systems and Science, 3rd edition (2011) Wiley. BISA U40-142

Abstracting the World to Digital Maps

Two basic views of spatial phenomena

When working with GIS, a basic question is “What does need to be represented in the GIS?” In particular, do we want to represent just objects in space or do we want to represent the space itself, or do we need both types of representation?

Objects in space For instance, we may want to represent information about cities (e.g. location, name, population, area, . . .), or about roads (e.g. location, type, maximum speed, . . .). Typically **vector data** and **spatial database systems** are used to represent this phenomena. The representation is **entity-oriented** and the view is **feature-based**.

Space itself In alternative, if we want to describe every point in space as, for instance, a land use map or a land surface temperature map, a distinct kind of representation can be more adequate.

A typical choice for the representation of this phenomenon is **raster data**. The representation is **space-oriented** and the view is **position-based**.

Objects in space: Geographic features

Objects in space that are represented in GIS are usually called **features**. Features of the same nature are grouped in **themes**.

Definition (Geographic Feature)

A feature is an *abstraction of real world phenomena*; it is a geographic feature if it is associated with a location relative to the Earth. So a digital representation of the real world may be thought of as a set of features. Source: Open Geospatial Consortium, GML v3.3

Features can be:

- 1 **Actual physical entities or events**, e.g. lakes, rivers, wetlands, elevation contours, roads, forested areas, rare species habitats, soils, earthquakes, vehicle thefts, electricity distribution lines, and groundwater reservoirs;
- 2 **Conceptual entities**, i.e. conceptual features do not physically exist on the landscape, but are imposed by us for various reasons and can be represented in a geographic context. Examples are census tract boundaries, administrative boundaries, parcel boundaries.

John Snow and the Broad Street cholera outbreak (1854)

John Snow was a British physician, born on the 15th of March, 1813. He was one of the first Epidemiologists, and he showed that cholera was water-borne, and debunked the (then-popular) Miasma theory, the theory suggested that diseases were spread through “bad air”. Later, in 1885, after Snow’s death, Robert Koch identified *V. cholerae* as the bacillus causing the disease.



Figure: John Snow, The Father of Modern Epidemiology

Adapted from <http://blogs.plos.org/publichealth/2013/03/11/john-snow-the-first-epidemiologist/>

The Broad Street cholera outbreak and John Snow's investigation

As the summer of 1854 wound down, a major cholera outbreak struck Soho, a neighbourhood in London. From August 31st to September 3rd, 127 people died of Cholera. Within a week, 500 people had died and around one in seven people who developed cholera eventually died from it. This all occurred within 250 yards of the Cambridge Street and Broad Street intersection.

John Snow came in and started his investigation.

He examined the neighbourhood, and talked to everyone he could. He was looking for an underlying theme that linked these people together. He suspected some contamination of the water, but couldn't find any organic matter in it, which you would expect under the Miasma theory.

Adapted from <http://blogs.plos.org/publichealth/2013/03/11/john-snow-the-first-epidemiologist/>

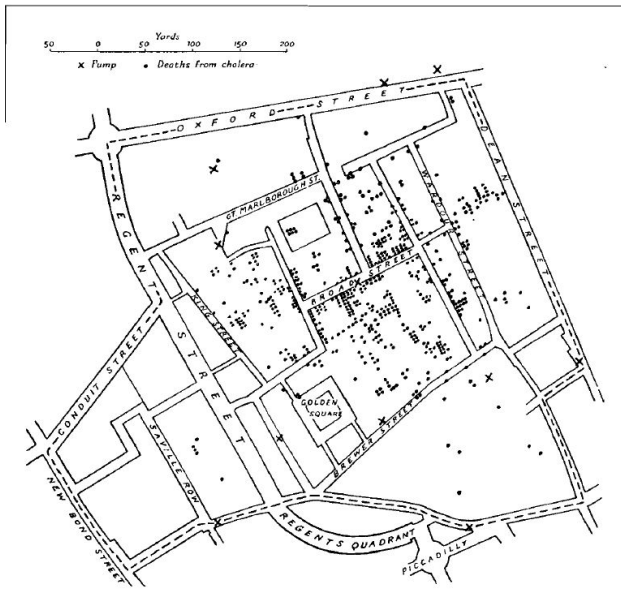
The Broad Street cholera outbreak and John Snow's investigation (cont.)

Almost all the cases of cholera occurred close to the Broad Street Pump. There were only 10 cases that were decidedly nearer to another pump. Of these, 5 preferred the water from the Broad Street Pump (and got their water from the Broad Street Pump) and 3 were children who went to school near the Broad Street Pump. The last two were unrelated, and likely just background levels of cholera in the population. This was pretty convincing, but Snow **mapped it out** to make sure that he was on the right track.

The evidence Snow presented in favour of his findings were too compelling for the local council to ignore, and while there was resistance to this finding, Snow had said enough for the local council to remove the pump handle, halting the spread of the disease. But, as Snow later pointed out, he couldn't be sure that this stopped the disease, and the incidence of the disease might have been declining. But the end result was the same: cholera cases went down.

Adapted from <http://blogs.plos.org/publichealth/2013/03/11/john-snow-the-first-epidemiologist/>

Gilbert's 1958 rendering of John Snow's map



John Snow own account of the investigation

On proceeding to the spot, I found that nearly all the deaths had taken place within a short distance of the [Broad Street] pump. There were only ten deaths in houses situated decidedly nearer to another street-pump. In five of these cases the families of the deceased persons informed me that they always sent to the pump in Broad Street, as they preferred the water to that of the pumps which were nearer. In three other cases, the deceased were children who went to school near the pump in Broad Street [...]

I had an interview with the Board of Guardians of St Jame's parish, on the evening of the 7th inst [7 September], and represented the above circumstances to them. In consequence of what I said, the handle of the pump was removed on the following day.

John Snow, letter to the editor of the Medical Times and Gazette

Broad Street cholera outbreak as a GIS case

The Broad Street cholera outbreak and John Snow's investigation is a typical example of spatial data analysis. The steps of John Snow's approach can be described as:

- 1 **Define the problem to be solved:** finding the source of the outbreak to prevent further infection.
- 2 **Acquiring the relevant spatial (or non spatial) information:** locations of the pumps and locations of the victims of the outbreak.
- 3 **Performing the analysis:** correlating the location of the pumps and of the victims.
- 4 **Describing the results in a comprehensive manner,** such that a decision can be made: the evidence was compelling; as a result, the handle of the pump was removed.

Most of GIS analyses can be framed similarly.

Geographic features in the Broad Street cholera outbreak example

In the Broad Street cholera outbreak example, the geographic features that John Snow considered can be essentially grouped in two themes: the **pumps** and the **addresses** of the victims.

Each feature is an actual event that can be represented by a single spatial location.

At the core of John Snow's analysis, there are only two groups (themes) of geographic features that can be represented as **points** on a map.

The core of the analysis consists on relating the **coordinates** of the points that represent pumps with the coordinates of addresses.

Note: Actually, John Snow took more information into consideration to relate the 10 deaths that occurred closer to a pump other than the infamous Broad Street pump. For more details about John Snow and Cartography, see Tom Koch (2004) The Map as Intent: Variations on the Theme of John Snow, Cartographica, 39, 4.

The complexity of reality versus representation

The world is effectively hugely complex, and a complete representation is impossible. The closer we look at the world, the more details it reveals.

Thus, in practice, any representation must be partial: it has to simplify the reality in some way, e.g. limit the level of detail, ignore change through time, ignore some attributes, or other type of simplification.

Simplification must not be confounded with **information compression**: the latter allows to represent the same amount of information with much less resources. For instance, a map of an homogeneous region can be very effectively compressed without loss of information.

To achieve a reasonable level of efficiency (in the representation of reality, or in data processing), the good representation has to be chosen (e.g. “objects in space” or “space itself”) and an adequate **data format** has to be selected.

Three fundamental means for organizing data for GIS use

GIS allow us to use spatial and non spatial information in a variety of formats.

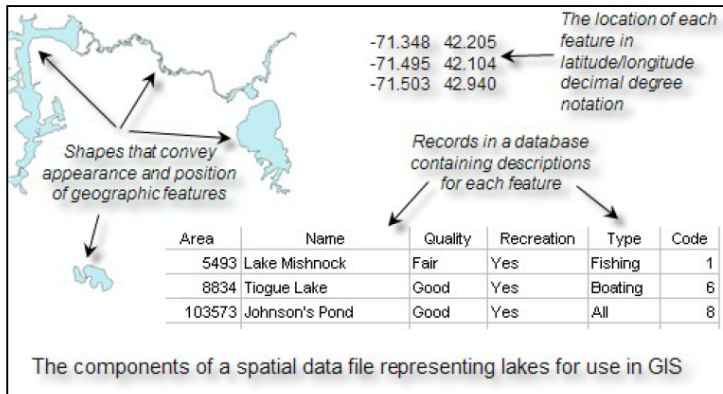
- 1 **Vector Data Formats**, e.g. shape files (open), coverage, ESRI geodatabases, DWG and DXF AutoDesk formats, Web Feature Services (see Open Geospatial Consortium OGC Web Feature Service Interface Standard WFS),
- 2 **Raster Image Formats**, e.g. bitmap (binary), GIF, PNG, TIFF and GeoTIFF (georeferenced), jpeg and jpeg2000 (compressed and potentially georeferenced), GRID (ESRI), IMG (ERDAS), Image Map Services (see OGC),
- 3 Tables (non spatial data format)

If data formats support **georeferencing** then files are called **spatial data files**:

- 1 vector spatial data files;
- 2 raster spatial data files.

Vector spatial data files

- 1 digital representations of geographic features;
 - 2 vector spatial data files have **attributes**;
 - 3 vector spatial data files are **georeferenced**;
 - 4 each vector spatial data file typically represents a common **theme**.
- In the example below, the theme is “lakes”.

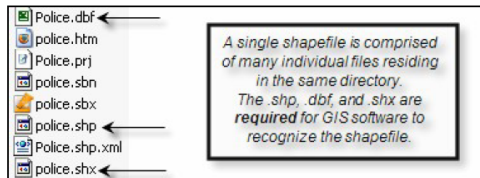


The Shapefile spatial data file format

- 1 A shapefile is a open vector data format.
- 2 In this format, geographic features can be represented in one of three ways: **points**, **polylines**, **polygons**.
- 3 If you create your own shapefile, you will have to determine which representation is best for your work or application. The determination is made based on several factors, including but not limited to:
 - 1 the need to depict features at a specific scale (e.g. cities as points or cities as polygons);
 - 2 the need to depict features as realistically as possible (e.g. the center line of rivers versus the width of the entire river with both banks constituting a polygon);
 - 3 the need to quantify some aspect of geographic features (e.g. the size of a lake would require it be created as a polygon).

The Shapefile spatial data file format (cont.)

- ④ A single shapefile is actually comprised of three files that absolutely must reside together in the same directory on your disk, or else it will not be recognized by the software. These three files will always have the same prefix for their file name, but have different extensions.
 - ① the **.shp** file contains the shapes (e.g. rivers.shp)
 - ② the **.dbf** file contains the tabular database records (e.g. rivers.dbf)
 - ③ the **.shx** file contains an index which connects the table to the shapes (e.g. rivers.shx)
- ⑤ While not required for GIS software to recognize a shapefile, a **.prj** file (e.g. rivers.prj) is very important. This file contains the coordinate system definition for the shapefile.



Source: <http://gisatbrown.typepad.com/gis/files/spatialdatafiles.pdf>

The Shapefile spatial data file format (cont.)

- 6 After you have worked with a shapefile, you may find that additional files appear in the directory along with the three base files. For example, **.sbn** and **.sbx** are common. Each one of these extra files is generated by the GIS software for a specific purpose (e.g. to speed drawing time). Another file is the **.shp.xml** of meta data.

Even though they are not required for the software to recognize the shapefile, it is generally preferable to retain them unless you have a specific reason or need to delete them.

- 7 *If you obtain a shapefile from another organization, person, or from a website, it will often be provided as a compressed .zip file. Once you have unzipped the file check to see that, at a minimum, the three base files are in the directory. Otherwise, the file will be useless.*

Adapted from <http://gisatbrown.typepad.com/gis/files/spatialdatafiles.pdf>

Working with Data in GIS

There are several available options of GIS software in the computer network of ISA:

- 1 QGIS (formerly Quantum GIS), an open source GIS that runs on Windows, OSX (Mac) and Linux. This is the GIS we'll use in lab classes;
- 2 ArcGIS, licensed from ESRI (runs on Windows);
- 3 IDRISI, licenced from Clark Labs.

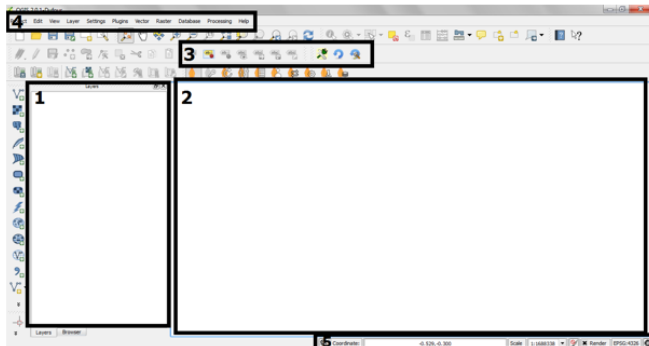
LAB: Create a GIS project in QGIS

The data for this first project is available in the folder

`\\dspace\home\cadeiras\sig\2013-2014\john_snow_data`

- 1 Map network drive `\\dspace\home`
- 2 Navigate to the folder;
- 3 Copy the folder `john_snow_data` to your local PC (suggestion: copy it to the `Documents` folder);
- 4 Start QGIS: QGIS desktop

LAB: QGIS interface



- **1. Map Legend/Layers** – where you add layers and interact with them, such as grouping them, setting symbology, and such.
- **2. Map View** – the main map area where the layers show up and all your visualization is done.
- **3. Tool Bar** – button bars. Like most modern windows applications you can drag your button bars around and move them between layers. Hover over each button and you will get a short description of what it is for.
- **4. Menu Bar** – most of the functionality in the Toolbar are up here as well. Don't forget the all-important *Help*. QGIS has good documentation so use it.
- **5. Status Bar** – shows status, coordinates and scale of the Map View. Sometimes map display updates also called rendering can be slow when using large data files. You can turn off rendering when dealing with large files until you want to see the layers.

Source: http://pm-qgis-inasafe-training.readthedocs.org/en/latest/qgis_interface_sc.html

LAB: Configure QGIS

- 1 Set language to English: at the menu bar follow *Configurações/Opções/Região* activate *Sobrepôr idioma do sistema* and switch to U.S. English, then [OK]; close and re-open QGIS.
- 2 Activate plugins: at the menu bar, open Plugin Manager at Plugins/Manage and Install Plugins Coordinate ; Activate Capture, Dxf2Shp Converter, fTools, GdalTools, Georeferencer GDAL, Interpolation plugin, Processing, Raster Terrain Analysis plugin, Spatial Query Plugin, Topology Checker, Zonal statistics plugin; [Close].
- 3 (optional) Change selection color: at the menu bar, follow Project/Project Properties/General choose Selection color;[OK].

LAB: Create project and add spatial data

- 1 Create project: at the menu bar, follow Project/Save and choose the folder `john_snow_data`; create (empty) project with name `snow`; [OK]
- 2 Add a raster spatial data file to the project: on the far left of the QGIS interface, click on the icon [Add Raster layer] and choose `SnowMap.tif`; [OK].
- 3 (optional) You can proceed similarly to add `OSMap.tif`, which shows a more recent map of the same neighborhood of London;
- 4 Save the updated project with [CTRL-S] or Project/Save.

Maps and scale

Definition (Scale of a map)

The scale of a map is the ratio of a distance on the map to the corresponding distance on the ground. This ratio is called the representative fraction (RF) and is usually expressed as $1/N$ or $1:N$, where N is typically larger than 1.

Given the scale, one can convert a map distance d into field distance D or vice-versa, using the relation $D = N \times d$.

Classification of scales:

- 1 **Large scale**: N smaller than, say, 50000 (e.g. 1:5000 for a walking map of a town);
- 2 **Medium scale**: N between 50000 and 250000 (e.g. a 1:100'000 country map);
- 3 **Small scale**: N larger than 250000 (e.g. World map at a 1:100'000'000 scale).

Note: Because of Earth's curvature, the scale is not constant across a map. This depends on the projection that is used (see later).

LAB: Scale in GIS

- 1 What is the meaning of the scale in the status bar (at the bottom)? Notice that in a GIS one can choose an arbitrary scale;
- 2 However, there is a native resolution that cannot be arbitrarily improved. On the bottom of Snow's map, there is the indication "30 inches to a mile". What is the native scale of Snow's map? Answer using the standard format 1:N. (Note: 1 inch=2.54cm and 1 mile= 1609.344m.)

Answer: the scale, or representative fraction is then

$$RF = \frac{\text{map distance}}{\text{field distance}} = \frac{30 \times 2.54}{160934.4} = 0.0004734848 = \frac{1}{2112}, \text{ i.e.}$$

the scale is 1 : 2112 (N=2112, large scale map).

- 3 Adjust the scale to the value you determined above (use 1:2000 otherwise): is the map clear? Change the resolution to a value 10 times larger (e.g. 1:200): what do you notice?

LAB: Scale in GIS (cont.)

- ④ Measure the actual length of Broad Street on the map using the ruler in the tool bar area; repeat using a different scale of representation; Why do the measurements differ?
- ⑤ How precise is a location on the paper map? *Even if the location is known perfectly*, there is an intrinsic error that derives from map drawing. The norm for map drawing states that the minimum width of a point or line is 0.2mm. How does this translate into positional uncertainty on the map? In particular, what is the resulting location (in meters) uncertainty in Snow's map?

To answer to this question, one just has to convert the drawing precision to a distance on the ground. In this case, the uncertainty of the drawing corresponds to $0.2 \times 2112\text{mm} = 422.4\text{mm}$, i.e approximately 0.4m.

Map scale and positional accuracy

Definition (Positional accuracy)

Degree of compliance with which the coordinates of points determined from a map agree with the coordinates determined by survey or other independent means accepted as accurate (Glossary of Mapping Sciences, 1994).

Positional accuracy uses *sampling* to estimate the discrepancy between map coordinates and their *true* location on the Earth's surface.

For example, the 1947 US National Map Accuracy Standard stipulates that *for horizontal accuracy, not more than 10% of the points tested may be in error by more than 0.846mm (at a map scale) for maps larger than 1:20'000 scale [...]*.

*This standard for map accuracy depends on the map scale.
How does this translate to Snow's map?*

Modern view of positional accuracy

Modern standards for positional accuracy share the following characteristics:

- 1 they rely on **ground units** (absolute measurements), instead of *map units*;
- 2 they consider the **distribution** of positional error from a sample of points, instead of just the quantile of the observations;

Positional accuracy is described for instance as: *At the 90% confidence level, 80% of the tested points are within 1.76m from their true location.*

Since points in GIS have coordinates (GIS coordinates can be stored with many digits), those coordinates can be compared to the corresponding real coordinates (true coordinates).

However, when spatial data files in GIS are **derived from existing maps** (as in John Snow's example) the map accuracy related to the original map scale still apply.

Analog and digital maps: scale and positional accuracy (conclusion)

Analog maps:

- 1 The scale is fixed;
- 2 The positional accuracy of the points or objects is measured on the map, and therefore it depends on the scale

Digital maps, in GIS:

- 1 The scale is variable;
- 2 The positional accuracy is estimated from the actual coordinates of the points or objects represented on the map, in ground units, so it is independent of the scale.

If digital maps are derived from analog maps (by digitalization), the positional accuracy of the digital map cannot be in principle better than the accuracy of the underlying analog map.

LAB: Add a vector spatial data to project

- 1 **Open project:** just click on the file `snow.qgs` in your folder if it is available;
- 2 **Add two vector spatial data files** to the project: on the far left of the QGIS interface, click on the icon [Add Vector Layer] and browse: select `Pumps.shp`; press [CTRL] and select `Cholera_Deaths.shp`; [OK].
- 3 Change the color of the graphical representation of points in the layer `Pumps` double-click the symbol below the layer name to open **layer properties** for that layer; Choose **Style** on the left menu and choose a new symbol and color to represent the points; [OK];
- 4 (optional) Proceed similarly for the layer `Cholera_Deaths`
- 5 (optional) Don't forget to save regularly the updated project with [CTRL-S] or Project/Save.

Tables, attributes, values and domains

Slide

Each geographic feature of a theme (for example, the Broad Street water pump of the layer `Pumps`) is associated to a **row** of the layer's **attribute table**.

An attribute table has to satisfy the following conditions (database table structure, Codd 1970):

- 1 There is only **one value** in each cell at the intersection of a row and column;
- 2 All values in a column are about the same **attribute** of the features;
- 3 Each row is unique;
- 4 There is no significance to the sequence of columns;
- 5 There is no significance to the sequence of rows.

Each attribute has a **domain**, i.e. the values of the cells of the respective column have to belong to that domain. The domain can be for instance a set of integers, or a set of decimal numbers, or a set of dates (with a certain format) or a set of text strings (e.g. up to 20 characters).

LAB: Inspect the attribute table of a layer

The `Cholera_deaths` layer feature represents addresses of the victims of the cholera outbreak; However, the number of victims at that location can be distinct. The information about the number of victims at each location is available in the **attribute table** of `Cholera_deaths`.

- 1 Click on the layer name `Cholera_deaths` to select it (the layer name turns into **blue**); then, click on the **Open Attribute Table** icon on the tool bar (or right-click on the layer name and select “Open Attribute Table”);
- 2 Is each row unique as required by a database table structure? What are the attributes of this table?
- 3 Sort the rows of the table by the attribute `Count` (Click on the name of the attribute); Verify that the map layer doesn't change as expected given the characteristics of the database table structure;
- 4 Select the location with the largest number of victims (15); Use the menu icons of top of the Attribute table: zoom in on that location; and then unselect.

LAB: Inspect the attribute table of a layer (cont.)

The `Pumps` layer feature represents the location of the pump.

- 1 Click on the layer name `Pumps` to select it (the layer name turns into **blue**); then, click on the **Open Attribute Table** icon on the tool bar (or right-click on the layer name and select “Open Attribute Table”); What are the attributes of this table?
- 2 To understand what is the domain of each attribute you need to inspect the **Fields** window in the **Layer properties**; Change color representation of the layer `Pumps`. Double-click to open **Layer properties** and click on tab **Fields**; Could a pump feature be called “Great Marlborough Street”? The answer is “no” since “Great Marlborough Street” has more than 20 characters and the domain of the attribute `name` is the set of character strings with at most 20 characters.

LAB: Use the attribute table of a layer to change the map color scheme and label points

The information about the number of victims at each location is available in the **attribute table** of `Cholera_deaths`. This information can be used to change the map representation, so locations with a higher number of victims are depicted in a stronger color.

- 1 Change color representation of the layer `Cholera_deaths`. Double-click to open **Layer properties** and select **Style**. Click tab **Single Symbol** and select instead **Graduated**. The **Column** has to be “Count” since there is no other attribute available; you can choose the symbol (suggestion: choose a larger size) and the color ramp; you can also choose the number of classes and the mode; [APPLY] or [OK];
- 2 As an alternative, you can label each location with the number of victims; Double-click to open **Layer properties** and select **Labels**; activate **Label this layer with** `Count`; change the color to red and the size to 12.

LAB: Select by attributes

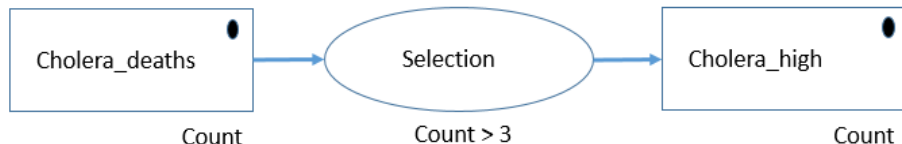
The information about the number of victims available in **attribute table** of `Cholera_deaths` can also be used to select a subset of locations depending on the number of victims.

Consider the problem: *Select the locations where the number of victims was equal or larger than 4, and create a new vector spatial data file with that theme.*

- 1 Click on the layer name `Cholera_deaths` to select it (the layer name turns into **blue**); then click on the **Select features using an expression** ϵ icon on the menu bar;
- 2 On the window **Select by expression** you have a list of functions that you can use to define an expression. At the end you find **Fields and values**: click and then double-click on `Count`, "`Count`" appears in the *Expression* box. Then, using the keyboard complete the expression to get "`Count`" `> 3` (or you can just type the full expression in the Expression box); Click [Select];
- 3 As a result the locations that satisfy the expression are selected on the map and in the attribute table; Check the table.

LAB: Select by attributes (cont.)

- 4 To save the selection as a new shapefile, right-click on the layer name `Cholera_deaths` and select **Save selection as ...**, browse to your working folder and call it `Cholera_high`; [OK]
- 5 Now, you can add the new layer to your project if you want and compare it with the existing `Cholera_deaths` layer; Look at the attribute table of the new layer: what are the differences? (same columns, fewer rows);
- 6 Let's document what we just did with a diagram:



LAB: A little flavour of spatial analysis ...

At the core of John Snow's investigation is the idea of proximity of the victim's address to each one of the pumps. One way of **formalizing the problem** is: (A) look for the **area of influence** of each pump; (B) **identify the pump** which area of influence contains the largest number of victim's addresses.

As a first approximation, the *area of influence* can be defined in geometric terms from the location of the pumps: the area of influence of a pump is the set of all points in space that is closest to that pump than to any other pump. Those sets, which are convex polygons, are called the **Voronoi polygons** (or Thiessen polytopes, or Dirichlet regions) of the given set of points.

- 1 In the menu bar, follow **Vector/Geometry tools/Voronoi Polygons**; As the **input point vector layer** choose `Pumps`; Write 100% in the **Buffer region box**; Choose your working folder and a name for the output polygon shape file; [OK];
- 2 Move the resulting layer to a position below your other vector layers; Add a label to each polygon with the name of the pump.

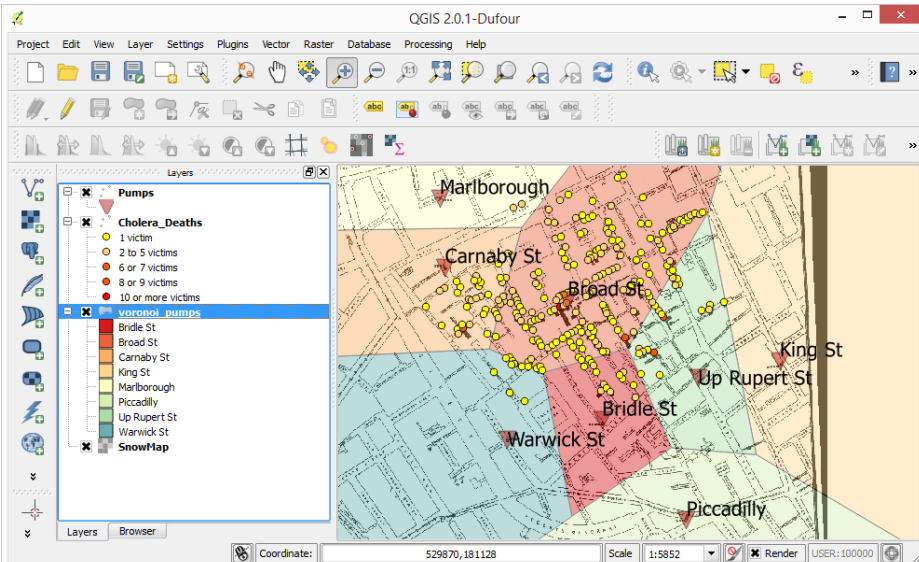
LAB: The Broad Street cholera outbreak as a GIS case (revisit)

Revisit of the steps of the approach and resolution of the study case.

- 1 **Define the problem to be solved:** finding the source of the outbreak to prevent further infection.
- 2 **Acquiring the relevant spatial (or non spatial) information:** locations of the pumps and locations of the victims of the outbreak.
- 3 **Performing the analysis:** correlating the location of the pumps and of the victims. The approach was to define the Voronoi polygons of the pump locations as approximations of the *area of influence* of each pump.
- 4 **Describing the results in a comprehensive manner.** The analysis allowed us to produce a map that shows clearly that there are more occurrences of Cholera deaths in the *area of influence* of the Broad Street pump than in the area

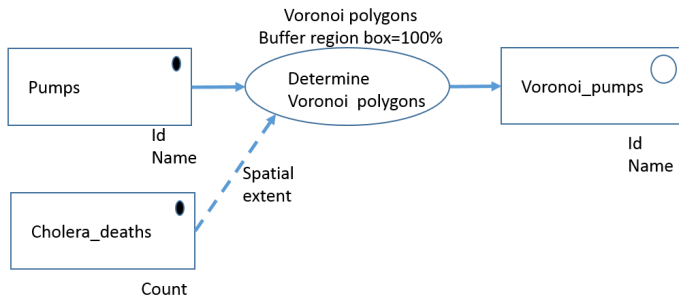
LAB: Presenting the results as a map

The digital map that is produced strongly suggests that the Broad St pump is the most likely cause of the outbreak.



LAB: Documenting the spatial analysis

Although the analysis also includes the construction of the map with colors, labels, and transparencies, the simple diagram below describes the spatial operation performed in GIS. There are two inputs, one for each set of features, and a new vector spatial data is created. While the inputs' geometry type is **point**, the new geometry type is **polygons**. The function is a well-defined geometric function and only has one **parameter**: the extent of the bordering polygons. This extent is chosen so all the points in `Cholera_deaths` are included in some polygon.



Coordinate Reference Systems

- ➊ The Earth's Reference Frame
- ➋ Coordinate Reference Systems
 - ➊ Approximating the Earth's shape with an ellipsoid
 - ➊ Positions, distances and angles
 - ➋ Elevation
 - ➋ Flattening the Earth
 - ➊ Map projections
 - ➋ Coordinate Transformations
 - ➌ Measurements on maps and associated errors
- ➌ Georegistration and orthorectification

Earth's shape

Definition (Geoid)

The equipotential surface of the Earth's gravity field which best fits, in a least squares sense, global mean sea level.

The geoid's shape is approximated by different mathematical models. The current standard model has 360 coefficients and is named **EGM96** (Earth's Gravitational Model). There is not *one* EGM96 geoid: several parameters are necessary to uniquely identify it (like tide model and reference ellipsoid).

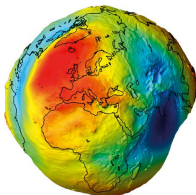


Figure: Geoid representation: the highest point is over the Atlantic Ocean; the lowest point is over the Indian Ocean.

Terrestrial Reference Systems and their realizations

Definition (Terrestrial Reference System, TRS)

It is a spatial reference system co-rotating with the Earth in its diurnal motion in space. In such a system, positions of points attached to the solid surface of the Earth have coordinates which undergo only **small variations with time**, due to geophysical effects (tectonic or tidal deformations).

Definition (Terrestrial Reference Frame ,TRF)

It is defined as the realization of a TRS, through the realization of its origin, orientation axes and scale, and their time evolution. **Seven parameters are needed to fix a TRF at a given epoch**, to which are added their time derivatives to define the TRF time evolution.

Note: The **International Earth Rotation and Reference Systems Service** (IERS) defines and maintains the TRS and its TRF realizations. It also establishes the International Reference Meridian, the International Reference Pole,

Geodetic datum

Definition (Geodetic datum)

A geodetic datum (plural datums, not data) is a reference from which measurements are made. In surveying and geodesy, a datum is a set of reference points on the Earth's surface against which position measurements are made, and (often) an associated model of the shape of the earth (reference ellipsoid) to define a geographic coordinate system.

Two types of datum:

- 1 **Horizontal** datums are used for describing a point on the earth's surface, in latitude and longitude or another coordinate system.
- 2 **Vertical** datums measure elevations or depths.

Source: https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Datum_%28geodesy%29.html

The World Geodetic System (WGS84)

WGS84 is an Earth-centered, Earth-fixed **terrestrial reference system** and **geodetic datum**. WGS84 is based on a consistent set of constants and model parameters that describe the Earth's size, shape, and gravity and geomagnetic fields.

WGS 84 is the standard U.S. DoD definition of a global reference system for geospatial information and is the reference system for the **Global Positioning System** (GPS). It is compatible with the International Terrestrial Reference System (ITRS).

- 1 Origin: Earth's center of mass;
- 2 Axes: **Z** is the direction of the IERS Reference Pole; **X** is the direction of the intersection of the IERS Reference Meridian and the plane passing through the origin and normal to the Z-axis; **Y** completes a right-handed, Earth-Centered Earth-Fixed orthogonal coordinate system.

The IERS reference meridian

The IERS reference meridian doesn't coincide with the Greenwich Meridian: currently it is 5.3 arc seconds ($\approx 102\text{m}$) east of the Greenwich Royal Observatory mark.



Figure: A GPS reading at the Greenwich Royal Observatory prime meridian mark. The GPS longitude for this measurement is 0.00152° W , i.e. 5.472 arc seconds.

Earth representation: ellipsoid

Definition (Spheroid)

An oblate spheroid is an ellipsoid of rotation, formed by the rotation of an ellipse about its shorter axis. The spheroid has then two semi-axes, the **semi-major axis** a and the **semi-minor axis** b . The **flattening** of the spheroid measures the reduction in the minor axis relative to the major axis and is defined as $f = (a - b)/b$.

The Earth's shape is well approximated by a slightly flattened spheroid, where $f \approx 1/300$. For some measurements, the spherical model $f = 0$ can be a good enough approximation of the geoid.

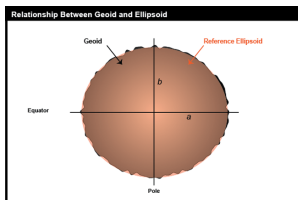


Figure: Earth's surface, geoid and approximation by a spheroid.

Global versus local approximation of the Earth's surface by an ellipsoid

Much effort was expended over the last centuries by many countries to find the best ellipsoid that approximate the shape of the Earth over those countries. This led to a large diversity of ellipsoids, with distinct semi-axis and centers, often not centered on the Earth's center of mass. Globalization (namely military or transportation requirements) led to international standards, and to the adoption of global models of the Earth's surface.

Therefore, we can distinguish two types of ellipsoidal models of the Earth's surface:

- 1 **Global** Earth ellipsoids, which are centered at the estimated Earth's center of mass, and give a good approximation of the whole surface;
- 2 **Local** Earth ellipsoids, which are usually not centered at the estimated Earth's center of mass, and give a good approximation of a given region of the world surface, at the cost of significant inaccuracies in other areas.

Global versus local approximation of the Earth's surface by an ellipsoid (cont.)

Although global ellipsoids are currently the natural choice to represent the Earth's surface, local ellipsoids are still important since a large quantity of available geographic data is georeferenced with respect to older standards.

GIS software has the capability of converting coordinates from one reference system to another.

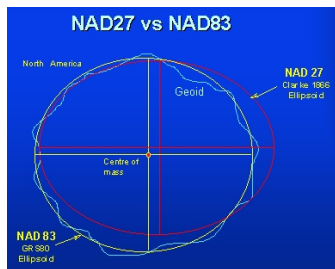


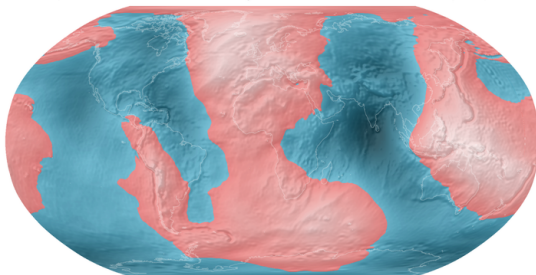
Figure: The reference systems NAD27 (local) and NAD83 (global) are important in North American mapping.

How accurate can a global Earth ellipsoid be?

The most widely accepted ellipsoid is from the **World Geodesic System of 1984**, aka WGS84, which differs from the standard geoid by at most 107m.

Deviation of the Geoid from the idealized figure of the Earth

(difference between the EGM96 geoid and the WGS84 reference ellipsoid)



Red areas are above the idealized ellipsoid; blue areas are below.



Figure: The line that separates pink and blue corresponds to the locations on Earth where the EGM96 geoid and the WGS84 ellipsoid intersect.

Some global and local Earth ellipsoid models

The reference ellipsoid models listed below have had utility in geodetic work and many are still in use. Some were adopted at moments in time by the international community like the International Ellipsoid (first proposed by Hayford in 1910 and adopted in 1924), and the more recent the Geodetic Reference System or the almost identical WGS84 ellipsoid.

Ellipsoid model	Semi-major axis a	$1/f$	
GRS80	6 378 137.0 m	298.257222	global
WGS84	6 378 137.0 m	298.257223	global
Bessel (1841)	6 377 397.15 m	299.152812	Europe, Japan
Hayford (1924)	6 378 388 m	297	Europe, USA
Krassovsky (1940)	6 378 245 m	298.3	USSR, Russia Romania
...			
Sphere	6 371 000 m		

Coordinate Reference System (CRS)

A **Coordinate Reference System** (also called a *spatial reference system*) contains two different elements - the datum and the coordinate system:

- 1 The **datum** defines how the CRS is related to the Earth (position of the origin, the scale and the orientation of coordinate axis) e.g. WGS84, ETRS89;
- 2 The **coordinate system** describes how the coordinates are to be assigned to points on the surface by means of mathematical rules. The coordinates can be cartesian coordinates, ellipsoidal coordinates or coordinates of a map projection.

The need for standardization

In order to identify the multitude of Earth models, the GIS community rely on Spatial Reference System Identifiers (SRID). For instance, <http://spatialreference.org/> lists Coordinate Reference Systems (CRS) in a number of formats:

- 1 EPSG (www.epsg.org) codes are defined by the Geomatics Committee of the International Association of Oil and Gas Producers (OGP), formerly European Petroleum Survey Group. The EPSG Geodetic Parameter Dataset is a structured repository of data required to (A) identify the CRS; (B) define transformations and conversions that allow coordinates to be changed from one CRS to another CRS.
- 2 ESRI;
- 3 IAU2000 (International Astronomical Union);
- 4 other CRS references.

The need for standardization (cont.)

The **Open Geospatial Consortium (OGC)**

<http://www.opengeospatial.org/> is an international voluntary consensus standards organization. In the OGC, more than 400 commercial, governmental, nonprofit and research organizations worldwide (including NASA, EADS, ESRI, ORACLE, GOOGLE and many universities) collaborate in a consensus process encouraging development and implementation of open standards for geospatial content and services, GIS data processing and data sharing.

The **Open Source Geospatial Foundation (OSGeo)**

<http://www.osgeo.org/> was created to support the collaborative development of open source geospatial software, and promote its widespread use. OSGeo collaborates with the OGC. OSGeo develops desktop applications like **QGIS**, GRASS and Marble; it manages the **Geospatial Data Abstraction Library** (GDAL: raster formats/OGR: vector formats). The OSGeo coordinate system engine is built on **PROJ.4** and **OGC Well Known Text** (WKT) coordinate system descriptions.

The need for standardization (cont.)

Comparison of two alternative descriptions of a particular CRS – the WGS84 (SRID 4326):

❶ Description in OGC WKT (“well known text”):

```
GEOGCS["WGS 84",  
    DATUM["WGS_1984",  
        SPHEROID["WGS 84",6378137,298.257223563,  
            AUTHORITY["EPSG","7030"]],  
        AUTHORITY["EPSG","6326"]],  
    PRIMEM["Greenwich",0,  
        AUTHORITY["EPSG","8901"]],  
    UNIT["degree",0.01745329251994328,  
        AUTHORITY["EPSG","9122"]],  
    AUTHORITY["EPSG","4326"]]
```

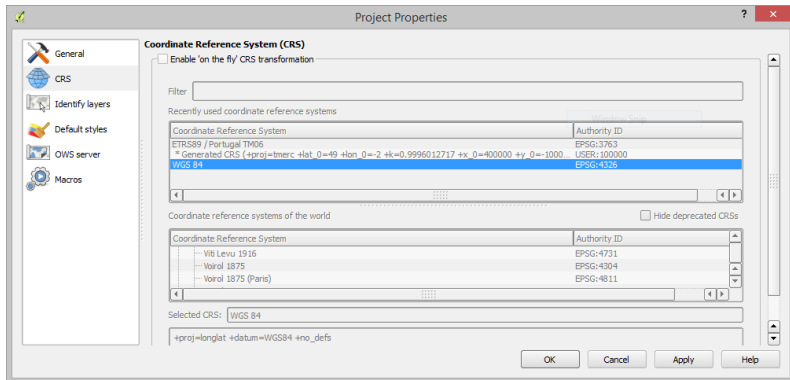
❷ Compact PROJ.4 description:

```
Proj4js.defs["EPSG:4326"] = "+proj=longlat  
+ellps=WGS84 +datum=WGS84 +no_defs"
```

LAB: introduction to Coordinate Reference Systems in QGIS

What is the default CRS in QGIS?

- 1 Start QGIS: QGIS desktop
- 2 Look at the current CRS (bottom right of the interface);
- 3 Click on the CRS status button (bottom far right) or on **Project/Properties/CRS** to check that the default CRS is WGS84.



LAB: Geodetic coordinates; “On the fly” conversions

Data for the LAB exercises:

- 1 Capitals of the World – Coordinate reference system: ETRS89
Source: <http://epp.eurostat.ec.europa.eu>,
EuroGeographics for the administrative boundaries
File: compressed data in `cntr_capt_etr89.7z`
- 2 Global administrative areas – Coordinate reference system: WGS84
Source: <http://www.gadm.org/>
File: compressed data in
`global_adm_countries_sl_wgs84.7z`
- 3 SRTM - Shuttle Radar Topography Mission, 3-arc second, tile N38W008 – Coordinate reference system: WGS84
Source: <http://earthexplorer.usgs.gov>
File: compressed data in `SRTM_N38W008.7z`
- 4 Geoid ondulation with respect to WGS84
File: `geoid_ondulation.img`

LAB: Geodetic coordinates

Goal: understand “on the fly” coordinate transformations.

The data for this first project is available in the folder

`\\dspace\home\cadeiras\sig\2013-2014\world`

- 1 File Explorer: Map network drive `\\dspace\home`
- 2 Navigate to the folder;
- 3 Copy the folder `world_data` to your local PC (suggestion: copy it to the `Documents` folder);
- 4 File Explorer: Unzip `cntr_capt_etr89.7z` and `global_adm_countries_sl_wgs84.7z`;
- 5 QGIS: on the legend/layer area add a **browser window** in (**view/panels**);
- 6 Use **browser window** to add working folder to favorites; navigate and right-click on the working folder to **add favorite**.

LAB: “on the fly” (OTF) coordinate transformation

- ➊ In the **browser window/favorites**, double click on `cntr_capt_etrs89.shp` to add the layer to the QGIS map;
- ➋ Check that the current project CRS is EPSG:4258 (ETRS89);
- ➌ Remove the `CNTR_CAPT_PT_2010` layer from the map;
- ➍ In **Project/Properties/CRS** check **Enable 'on the fly' CRS transformation** and select **WGS84 – EPSG:4326** (can use filter, and then double-click on WGS84);
- ➎ Again, in favorites, double click on `cntr_capt_etrs89.shp` to add the layer `CNTR_CAPT_PT_2010` to the QGIS map;
- ➏ Check that the current project CRS is now EPSG:4326 (WGS 84);
- ➐ Check in **layer properties** that the CRS associated to the layer is still EPSG:4258 - ETRS89

The project CRS can either be defined *a priori* or can be inherited from the layer CRS. Check the menu **Settings/Options/CRS**.

LAB: “on the fly” (OTF) coordinate transformation

In the simple previous example, the use of **on the fly** CRS transformation was illustrated.

In short, there are two distinct CRS settings in QGIS:

- 1 The **project CRS**, which is defined in **Project/Properties/CRS** ;
- 2 Each **layer CRS**, which is displayed in the layer properties/CRS.

When the **on the fly** CRS transformation is enabled, then QGIS converts any new layer coordinates into the Project CRS, ensuring that all layers are displayed in a common CRS. QGIS handles about 3000 distinct Coordinate Reference Systems.

- 1 In the **browser window/favorites**, double click on `global_adm_countries_sl_wgs84.shp` to add the layer to the QGIS map;
- 2 Check that the two layers have distinct CRS; however, even if you disable OTF in **Project/Properties/CRS** the two datasets still match since the two CRS (WGS84 and ETRS89) are almost identical.

Lets's carry on...

We saw that WGS84 is a well accepted global reference framework, but we haven't discussed yet:

- 1 Why global reference frameworks other than WGS84 (e.g. ETRS89) are used;
- 2 How locations on the Earth are described within a reference framework, using coordinates.

Why global datums other than WGS84 are important?

To calculate the WGS84 (G1150) datum the United States Department of Defense, owner of the GPS system, used a subset of the ground reference stations of the International Earth Rotation and Reference System Service (IERS) responsible for the International Terrestrial Reference System (ITRS). This reference system imposes a **no net rotation condition** for horizontal motions which means that the WGS84 datum is not tied to any specific tectonic plate.

In practice this means that the coordinates of any point located in the surface of the planet Earth will slowly change over time because **tectonic plates slowly drift over time** and the reference frame is not tied to any of them.

Adapted from <http://support.novatel.com>.

The European reference system

To avoid the ITRS slow temporal drift the European Commission mandated the creation of a reference frame consistent with the Euroasiatic plate motion named **European Terrestrial Reference System 1989 (ETRS89)**, which uses the Earth-centered GRS80 ellipsoid.

When the ETRS89 datum was first defined in 1989 it was coincident to the ITRS. However, because the European continent displaces at **2.5cm per year** when seen from the ITRS datum the positional difference between those two datums has progressively grown to the point that today, more than 20 years after the creation of the ETRS89 datum **the coordinate difference becomes obvious** even in mid accuracy GIS applications.

It is a common practice for the European reference network operators to configure the coordinates their ground reference stations expressed in ETRS89 datum instead of WGS84, unfortunately very rarely is clearly stated what exact datum is in being used.

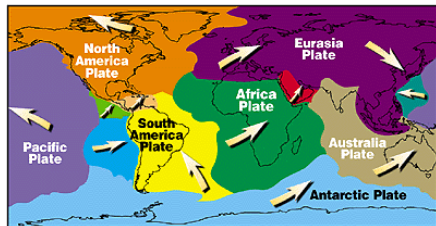
Adapted from <http://support.novatel.com>.

The datum for the North American Plate

The analogue of the ETRS89 for North America is the North American Datum (NAD83) which is tied to the North American Plate and uses the Earth-centered GRS80 ellipsoid. It deviates 1–2cm per year from WGS84.

As an example, we mentioned 3 Earth-centered datums:

- 1 WGS84 (uses the WGS84 ellipsoid), not tied to any tectonic plate;
- 2 ETRS89 (uses the GRS80 ellipsoid), tied to the Euroasiatic Plate;
- 3 NAD83 (uses the GRS80 ellipsoid), tied to the North American Plate.



For many purposes, different Earth-centered datums are interchangeable

NATO/PfP UNCLASSIFIED

ANNEX D to
STANAG No. 2211
(Edition 6)

DATUM TRANSFORMATIONS

INTRODUCTION

1. World Geodetic System (WGS) 84 is the U.S. Department of Defense definition of a global reference system for geospatial information. The number of local geodetic datums (including those based on astronomic definitions alone) used throughout the

[. . .]

IMPLEMENTATIONS OF WGS 84

3. NATO operations do not require the use of datum transformations between datums that conform to the International Earth Rotation Service (IERS) convention 1996 for an Earth-centered, Earth-fixed coordinate reference system. The following datums are considered to be implementations of WGS 84 within the NAOR:

- a. World Geodetic System 1984 (e.g., realization G873)
- b. International Terrestrial Reference System (ITRS)
- c. European Terrestrial Reference System 1989 (ETRS89), also referred to as EUREF.

Coordinates referenced to any of these systems shall be considered WGS 84 coordinates for NATO operations.

Figure: For NATO, geodetic coordinates from WGS84, ETRS89 or NAD83 are considered sufficiently similar.

Coordinates on the ellipsoid

Coordinates on the ellipsoid can be:

- 1 Cartesian coordinates (X, Y, Z), as in the WGS84 system;
- 2 Geodetic coordinates: latitude, longitude, and elevation.

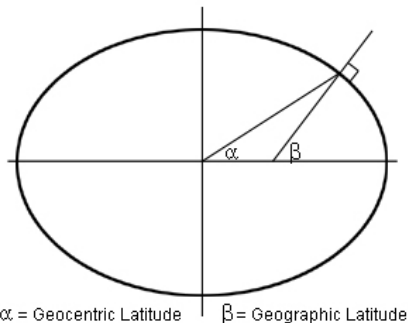
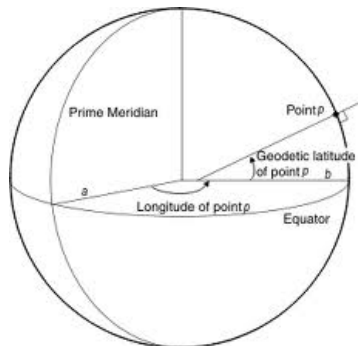


Figure: Geometry of the ellipsoid. There are two definitions of latitude: the **geocentric** latitude and the **geodetic** (or **geographic**) latitude. The usual coordinate reference systems use the **geodetic** latitude.

Geodetic coordinates on the ellipsoid

The **geodetic coordinate system** has 3 coordinates: latitude, longitude and elevation.

(Geodetic) latitude Angle ϕ between the **equatorial plane** and the line that passes through the point and is **normal** to the surface of the ellipsoid;

Longitude Angle λ (West or East) from a **reference meridian** to the meridian that passes through the point.

(Geodetic) Elevation The distance (positive or negative) to the **ellipsoid**.

Recall that latitude ranges from -90° (S) to 90° (N), and longitude ranges from -180° (W) to 180° (E).

Distances on the ellipsoid

Conventional measures of distance were derived from the Earth's natural distances and from the measurements of latitude and longitude:

- 1 The **meter** was initially proposed in 1791 by the French Academy of Sciences as the length equal to one ten-millionth of the distance between the North Pole and the Equator;
- 2 The **nautical mile** is the distance on the surface that corresponds to one minute of latitude along a meridian.

However, distances on the surface are not proportional to variations in longitude, since the distance on the surface that corresponds to one unit (e.g. degree) of longitude depends on the latitude.

Angle units

Latitude and longitude are **angles**, which can be expressed as

- Degrees, minutes and decimal seconds (e.g. lat= $38^{\circ} 50' 7.296''$), the DMS notation. There are 360° in the circle, each degree is divided in 60 minutes, and each minute is divided in 60 seconds;
- Decimal degrees (e.g. lat= 38.83536°);
- In radians: the radian is the standard unit of angular measure. There are 2π radians in the circle, so $1\text{rad} = \frac{360}{2\pi} = \frac{180}{\pi} \approx 57.29577^{\circ}$

Conversions:

- DMS to decimal degrees:
 $38^{\circ}50'7.296'' = 38 + 50/60 + 7.296/3600 = 38.83536$ decimal degrees; Or we can convert DMS to seconds and then to degrees:
 $38^{\circ}50' 7.296'' = 38 \times 3600 + 50 \times 60 + 7.296 = 139807.296''$ and $139807.296''/3600 = 38.83536^{\circ}$.
- Decimal degrees to radians: we multiply by $\frac{\pi}{180}$. For instance, $38.83536^{\circ} = \frac{\pi}{180} 38.83536 = 0.6778049\text{rad}$;
- Radians to decimal degrees: we multiply by $\frac{180}{\pi}$. For instance, $0.8\text{rad} = \frac{180}{\pi} 0.8 = 45.83662^{\circ}$.

Distances as angles or lengths over the surface

The **distance** between two points can be expressed as a **length over the surface** or as an **angular measurement**. As discussed earlier, angular measurements can be made in **radians**, in **degrees**, but also in **arc minutes** or **arc seconds**.

On a circle of radius R , an angle θ (in radians) corresponds to a length $R \times \theta$.

Example: Global Multi-resolution Terrain Elevation Data 2010 (GMTED2010) are available in 30-, 15- and 7.5-arc second resolutions. This translates to the following spatial resolution on the ground over the Equator ($R=6371\text{km}$):

- 1 30-arc seconds are $30/3600 = 0.008333^\circ = 0.00014544\text{rad}$.
Therefore, the corresponding length at the Equator is
 $6371 \times 0.00014544 = 0.9266244\text{km} \approx 926\text{m}$;
- 2 Similarly, 15-arc seconds corresponds to $\approx 463\text{m}$ and 7.5-arc seconds corresponds to $\approx 231\text{m}$ on the ground at the Equator.

Distances on the Earth ellipsoid versus distances on the sphere

For the following geodetic latitudes, the table gives the distances on the WGS84 ellipsoid for 1° of variation in latitude or longitude.

latitude (ϕ)	0°	15°	30°	45°	60°	75°	90°
Δ_{lat}^1 (km)	110.574	110.649	110.852	111.132	111.412	111.618	111.694
Δ_{long}^1 (km)	111.320	107.551	96.486	78.847	55.800	28.902	0

Exercise. Consider the sphere of radius 6371km as a simplified model of the Earth to answer the following questions.

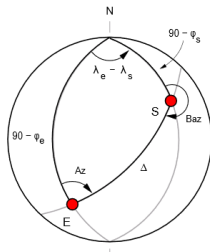
- 1 Determine the length of a meridian (in km).
- 2 Determine the distance (in km) on the sphere for a variation of 1° in latitude (an approximation to Δ_{lat}^1)
- 3 Determine the length in km of the 45° latitude parallel;
- 4 Compute the distance on the sphere for a variation of 1° of longitude when the latitude is 45°. Compare with the value 78.847km in the table above.

All over the globe, the error from spherical approximation is at most 0.5% of the real distance.

Distances and azimuths on the sphere: mathematical expressions

Computing distances and azimuths on the ellipsoid is complicated. Since the sphere is a good approximation of the ellipsoid, we review how those measurements can be done on the sphere of radius R (for the Earth, $R=6371\text{km}$). Given two points with coordinates $E = (\phi_e, \lambda_e)$ and $S = (\phi_s, \lambda_s)$, where $\text{lat}=\phi$ and $\text{long}=\lambda$:

- The **distance** (km) between E and S over the sphere of radius R is $R \times \Delta$, where $\Delta = \arccos[\sin \phi_e \sin \phi_s + \cos \phi_e \cos \phi_s \cos(\lambda_e - \lambda_s)]$.
- The **azimuth** of the point S at the point E is the angle between EN and ES . It is $Az = \arcsin[\sin(\pi/2 - \phi_s) \sin(\lambda_s - \lambda_e) / \sin \Delta]$.



LAB: extracting coordinates from a point; calculating distances and azimuths on the ellipsoid

- 1 Check the that current project CRS is **WGS84 – EPSG:4326** and that “on the fly” is **NOT** enabled in the **CRS** menu of **Project Properties**; Check also that in the **General** menu in **Project Properties** the **Measure tool (CRS transformation: OFF)** is set to **None/Planimetric**;
- 2 Double-click on the layer `CNTR_CAPT_PT_2010` and add labels using the attribute `NAME_ENGL`;
- 3 In the menu **view/decorations**, add a grid with a 1° interval for X over the map;
- 4 Compute the distance on the map between Lisbon and Madrid with the **measure line** button; you should obtain approximately 5.708° ($=0.0996\text{rad}$);
- 5 Compute the azimuth of the direction Lisbon–Madrid, at Lisbon, using the **measure angle** button and following the grid; you should obtain something like 72.5° .

LAB: coordinates, distances and azimuths on the ellipsoid (cont.)

- 6 Extract the lat/long coordinates of Lisbon and Madrid: use the **identify feature** button and right-click to copy attribute value;
- 7 Those coordinates allow you to compute with the equations of Slide 74 the (A) angular measurement $\Delta = 0.07900877\text{rad}$ and (B) the azimuth= 66.06278° at Lisbon;
- 8 Why measured values quite different from these (correct) values? The reason is that when the **Measure tool** in the **General** menu in **Project Properties** is set to **None/Planimetric**, QGIS computes **Euclidean (planar) distances** instead of distances over the ellipsoid;
- 9 To fix this, (A) enable “on the fly” in the **CRS** menu of **Project Properties**; (B) set in the **General** menu in **Project Properties** the **Measure tool (CRS transformation: ON)** to **WGS84** (since the data's CRS is WGS84);
- 10 Repeat steps 4 and 5: now you should obtain the correct distance (504327m) **over the ellipsoid** and the correct azimuth (66.142°).

Elevation

There are different definitions of elevation:

- 1 **Orthometric height** or **mean sea level (MSL) elevation** H refers to the distance to the geoid (e.g. elevation used in Topography, Surveying, elevation given by SRTM and GDEM);
- 2 **Geodetic (ellipsoidal) height** h is the distance to the reference ellipsoid (e.g. elevation provided by the Global Positioning System **GPS**).

$N = h - H$ is called the **geoid height** or **geoid undulation**.

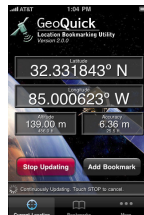
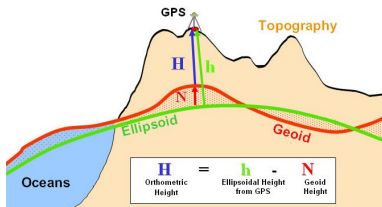


Figure: GPS geodetic elevation measurements have to be shifted by the value of N to be converted into orthometric heights.

Height measurements and the zero surface reference

A sailor aboard an anchored destroyer turns on his Navstar Global Positioning System (GPS) receiver and reads that his present height is minus 25 meters. Knowing the ship hasn't sunk, he questions the accuracy of his equipment. The manual says his receiver has a vertical accuracy of plus or minus 9 meters, so he decides to record the measurements for several hours to see if the elevation will ever be positive. Later, when he compares the readings, he finds that all are negative with a mean value of minus 20 meters.

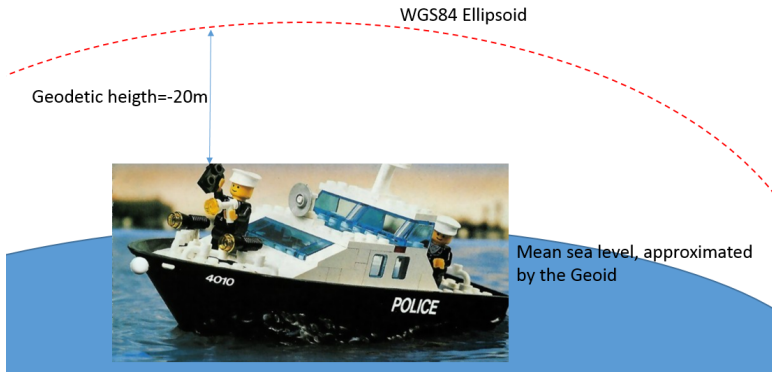
What's the problem here – is the receiver broken? Probably not. Standard GPS receivers and software have built-in flexibility to meet a variety of applications. Different applications need heights or elevations with respect to different **zero surfaces**. Most GPS receivers have built-in software to do these transformations. The **user needs to specify the correct vertical reference system**.

Source: National Imagery and Mapping Agency,

http://www.usna.edu/Users/oceano/pguth/website/so432web/e-text/NIMA_datum/VERTICAL%20DATUMS.htm

Illustration of the previous example

On certain locations on the Earth (e.g. off the coast of Sri Lanka) the geoid lies well **below** the WGS84 ellipsoid. Therefore, the geodetic height at sea level is likely to be **negative**.



Global Digital Elevation Models

In 2000 a space shuttle mission acquired data for the global digital elevation map SRTM (Shuttle Radar Topography Mission) with a global spatial resolution of 3-arc seconds.

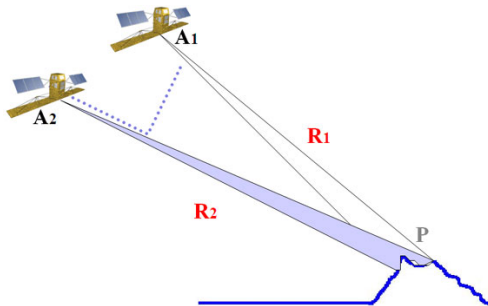
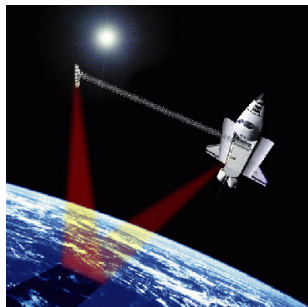


Figure: Illustration of the application of interferometric synthetic aperture radar (InSAR) technology. Interferograms can be used to produce digital elevation maps using the stereoscopic effect caused by slight differences in observation position between two images.

Global Digital Elevation Models (cont.)

The following data are available at

https://lpdaac.usgs.gov/get_data/

(the user needs to be registered).

- 1 3-arc second (global coverage) SRTM data are described in <http://glcf.umiacs.umd.edu/data/srtm/description.shtml>. The original data are from February 2000.
- 2 Aster Global Digital Elevation Map (GDEM version 2, 2011) are described in <http://www.jspacesystems.or.jp/ersdac/GDEM/ver2Validation/GDEM/QuickGuide2.pdf>. The original data were acquired by the sensor ASTER on-board NASA's TERRA satellite. ASTER GDEM's nominal resolution is 30m. Version 2 is derived from data collected between 2000 and 2010.

Global Digital Elevation Maps (cont.)

- 3 30-, 15- and 7.5-arc second Global Multi-resolution Terrain Elevation Data 2010 (GMTED2010), described in http://topotools.cr.usgs.gov/GMTED_viewer/. Seven different aggregation methods are available: minimum elevation, maximum elevation, mean elevation, median elevation, standard deviation of elevation, systematic subsample, and breakline emphasis.

*All those digital elevation models contain **mean sea level (orthometric)** heights.*

LAB: Compare orthometric (MSL) and geodetic elevation

- 1 File explorer: unzip `SRTM_N38W008.7z` in your working directory;
- 2 Add `n38_w008_3arc_v1`, which is a GeoTIFF file of a SRTM3 tile, to your project;
- 3 Confirm in **Layer properties/Metadata** that this raster layer extent is $\approx 1^\circ \times 1^\circ$, gridded in 1200 columns and rows, i.e. 1200^2 cells;
- 4 Confirm that the cellsize is (0.0008333, 0.0008333). Which are the units? Why is this called a 3 arc second dataset?
- 5 All pixels look square on the latlong map displayed on QGIS; Show that actually the approximate cell size on the ground at latitude 38.5° is 92.66m (along N-S) and 72.51m (along E-W)?
- 6 Add to your project the raster `geoid_ondulation.img`. This raster contains the geoid ondulation $N = h - H$ (in meters) for the WGS84 ellipsoid (h) and the EGM96 geoid (H);
- 7 For a flat area on the SRTM3 tile, calculate the approximate value of the orthometric (MSL) elevation. What should be the geodetic elevation value that you could read at that location using a GPS instrument?

LAB: Resolution of question 5 on slide 83

To answer the question we use the spherical model ($R=6371\text{km}$) to represent the Earth's surface.

- 1 First we determine the approximate size of the cell in the N-S direction: this distance is measured over a circle of radius $R=6371\text{km}=6371000\text{m}$, and the angular measurement for the cell is $\theta=3 \text{ arc seconds}=0.0008333^\circ=0.00001454383\text{rad}$. Hence (see Slide 72) the cell size is $0.00001454383 \times 6371000 = 92.66\text{m}$.
- 2 For the E-W direction we have to do a similar calculation but over a convenient parallel.
 - 1 Since the the `n38_w008_3arc_v1` spatail data set contains elevation from points located between the 38° N parallel and the 39° N parallel, we can consider that the average latitude is 38.5°
 - 2 Next, we determine the radius R' of the 38.5° parallel: this is just $R \times \cos(38.5^\circ)$ where
$$\cos\left(\frac{\pi}{180}38.5\text{rad}\right) = \cos(0.6719518\text{rad}) = 0.7826082.$$
Therefore, $R' = 6371 \times 0.7826082 = 4985.997\text{km}$;
 - 3 Finally, we do the same calculation that we did for the N-S direction, but considering this time a circle of radius $R' = 4985.997\text{km}$ and we obtain the cell size $0.00001454383 \times 4985997\text{m} = 72.51\text{m}$.

Quiz

Explain the meaning of the following sentences:

- 1 “The conversion of GPS ellipsoidal height differences to elevation differences will depend on how well geoid height differences can be determined.” (from “Applications of GPS for Surveying and Other Positioning Needs in Departments of Transportation, Cerzniak et al., 1998”)
- 2 “ newer GPS devices output orthometric (geoid) height measurements as a product of 'behind the scenes' calculations based on a combination of formulas, tables, and matrices that use geographic coordinates as inputs. ” (from “Mean Sea Level, GPS, and the Geoid, By Witold Fraczek, Esri Applications Prototype Lab, 2003”).
- 3 “Heights are in meters referenced to the WGS84/EGM96 geoid” (from the documentation of SRTM version 2).

Coordinate Reference Systems: Map Projections

So far we have discussed Coordinate Reference Systems that use

- **3D Cartesian coordinates**: Earth-centered, Earth-fixed orthogonal axis X , Y , Z , where Z points toward the North Pole and X points toward the intersection of the equator and the prime Meridian, as in the standard definition of the WGS84 global reference system.
- **Geodetic coordinates** (also called *geographic coordinates*): latitude and longitude. Example: Latitude/Longitude coordinates for the WGS84 and ETRS89CRS, and GPS coordinates.

For many applications, it is more adequate to work with a map coordinate system, i.e.

- **2D Cartesian coordinates** (also called *projected coordinates*) of a **plane coordinate system**: a 2D orthogonal coordinate system defined with respect to a plane.

To convert geographic coordinates to planar coordinates we need a **map projection**.

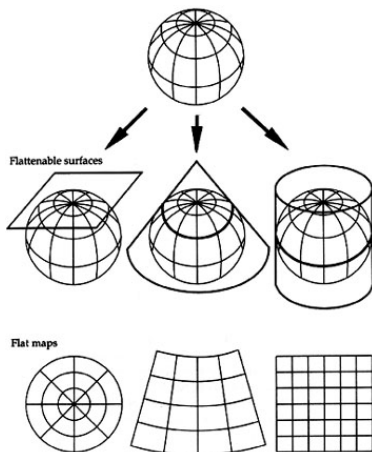
Map projections

Definition (Map projection)

It is a method to produce all or part of a spheroid (ellipsoid of revolution) on a flat surface.

The three basic families of map projections:

- azimuthal,
- conic,
- cylindrical



Classification of map projections

Map projections can be classified mainly according to:

- ① The **form of the surface** used for the projection and the **viewing origin**:
 - ① Flat surface where the positions of the spatial objects are projected: cylinder, cone and disk
 - ② Position of the surface in relation to the ellipsoid: each surface can be tangential or secant to the ellipsoid.
- ② **Spatial properties**, preserved or distorted:
 - ① distance: distances are preserved, i.e. distances on the map projection are proportional to the distances on the ellipsoid;
 - ② direction: directions are preserved;
 - ③ shape: angles are preserved;
 - ④ area: area is preserved;

Rules for choosing a projection

1 Location and extent (size and direction).

- ▶ latitude higher than 70° (polar): **azimuthal normal**;
- ▶ latitude between 20° and 70° :
 - ★ round extent: **azimuthal oblique**;
 - ★ north-south extent: **cylindrical transverse**;
 - ★ east-west extent: **conical normal**;
- ▶ latitude lower than 20° (equatorial): **cylindrical normal**.

For example, for Portugal a cylindrical transverse projection is natural.

2 Purpose of the map.

- ▶ show correct scale between a point (or two) and every point on the map: **equidistant**;
- ▶ two equal shapes on the map cover the same area of the actual Earth: **equal-area**;
- ▶ relative local angles are showed correctly at about every point on the map: **conformal**;
- ▶ other geometric requirements.

The EU case

Europe is a patchwork of several countries with different traditions in terms of their geographic choices. Around year 2000, European countries still used 5 different types of reference ellipsoids and 8 different map projections at country level. In 2003, a broad technical assessment at the EU level recommended:

- 1 the adoption of ETRS89 as the geodetic datum for the geo-referenced coordinates within the EU;
- 2 the extensive use of ellipsoidal coordinates related to the ETRS89 for expressing and storing positions;
- 3 the use of 3 selected types of map projections, sufficient for the needs of the European programs.

It also recognizes that the recommended coordinate reference systems and the existing national and local coordinate reference systems will continue to coexist for many years to come.

The distribution of map projections in Europe in 2000

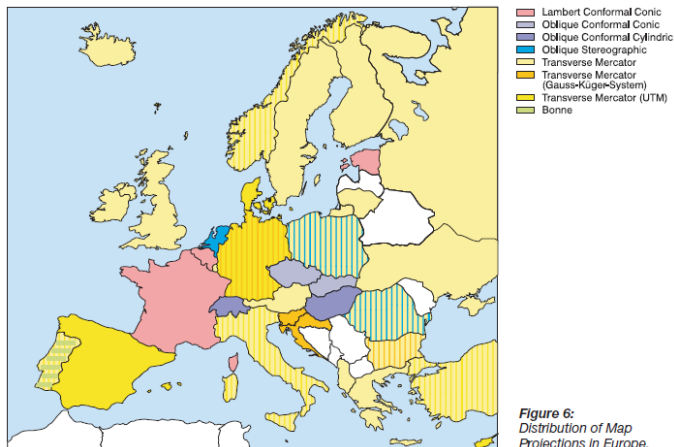


Figure: Map projections in Europe (Annoni *et al.*, 2003). The projection depends on the shape and size of each country.

The EU recommended Map Projection Coordinate Reference Systems

The INSPIRE Directive 2007/2/EC establishes the following map projections for Europe:

- 1 ETRS-TMzn: **ETRS89 Transverse Mercator CRS**, which is **Conformal** and is recommended for large and medium scale mapping (scale larger than 1:500 000); This is a family of CRS, which depend on the zone (zn) where the map projection is centered;
- 2 ETRS-LCC: **ETRS89 Lambert Conic Conformal CRS**, which is also **Conformal** and is recommended for small scale mapping (scale smaller than 1:500 000);
- 3 ETRS-LAEA: **ETRS89 Lambert Azimuthal Equal Area CRS**, which is a **Equal Area** map projection and is recommended for European statistical maps at all scales.

Transverse Mercator

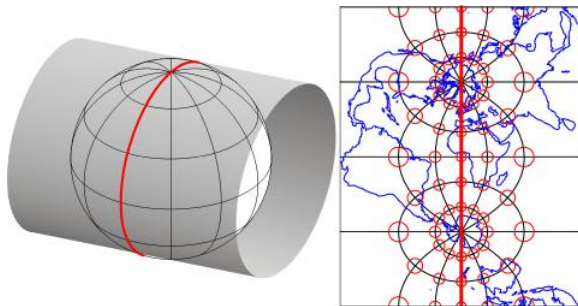


Figure: The Transverse Mercator projection corresponds to a cylinder wrapped along a particular line of longitude. It is a conformal map projection. Moreover, distances are accurate up to a 0.04% error within a 6° span of longitude. It is used by the UTM coordinate system.

Transverse Mercator projection zones

Since distortions grows with the distance to the reference meridian, the transverse Mercator map projection works better over a narrow span of **longitudes**. A typical range is 6° , which is used by the Universal Transverse Mercator coordinate system and by the standard zones of the ETRS-TMzn, which leads to distortions lower than 1m per km.

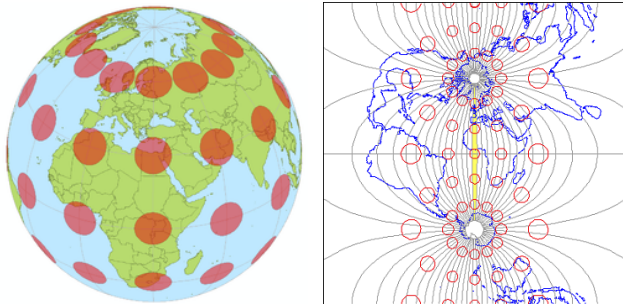


Figure: Distortions from the **Transverse Mercator** map projection along the Greenwich meridian, represented by its Tissot indicatrix. The yellow zone corresponds to a 6° range of longitude.

Lambert Conic Conformal map projection

A Lambert Conic Conformal projection (LCC) is a conic map projection. In essence, the projection seats a cone over the sphere of the Earth and projects conformally onto the cone. The cone is unrolled, and the parallel touching the sphere is assigned unitary scale in the simplest case. This parallel is called the **reference parallel**. There can be two reference parallels.

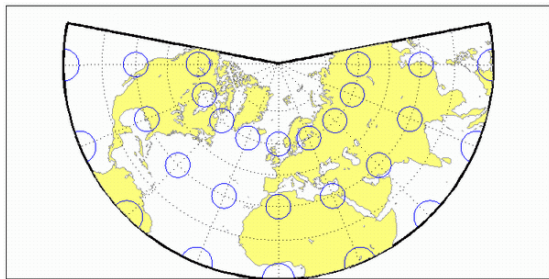


Figure: Distortions from the Lambert Conformal **Conic** map projection, represented by its Tissot indicatrix.

Lambert Azimuthal Equal Area map projection

It has a center point S , which defines the plane of projection. The LAEA map projection maps every point of the surface of Earth along a circle that is centered in S . This technique preserves areas.

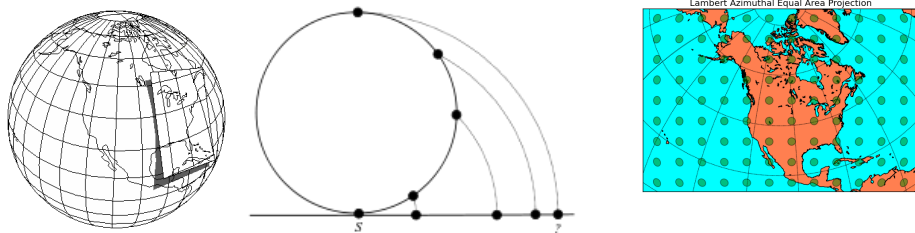


Figure: (a) Illustration of the oblique azimuthal map projection centered on a point over North America. (b) Illustration of the Lambert azimuthal equal area map projection. (c) Distortions from this projection represented by its Tissot indicatrix.

ETRS89-LCC and ETRS89-LAEA map projections

The INSPIRE Directive 2007/2/EC adopts the following parameters for the Lambert Conic Conformal and the Lambert Azimuthal equal Area map projections:

- ETRS89-LCC has two standard parallels (35° N and 65° N). The latitude grid origin is 52° N and the longitude grid origin is 10° E; false northing: 2 800 000m; false easting: 4 000 000m. The scale factor ranges between 0.9656 (at $\approx 51^{\circ}$ N) and 1.0437 (at 71° N).
- ETRS89-LAEA. latitude of origin: 52° N; longitude of origin: 10° E; false northing: 3 210 000m; false easting: 4 321 000m.

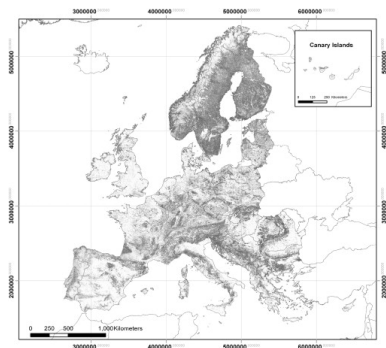


Figure: A map of forested areas for Europe using the ETRS89-LAEA CRS (Pekkarinen *at al.* 2009)

Other map projections

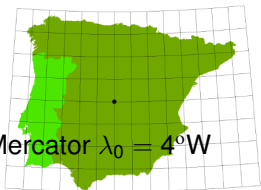
There are many map projections which are more or less adequate depending on the purpose of the map. Some projections achieve the goal of being almost conformal and area preserving all over the surface of Earth. This is achieved at the cost of “tearing” the map.



Figure: The Dymaxion map or Fuller map is a projection of a world map onto the surface of an icosahedron, which can be unfolded and flattened to two dimensions.

Comparison of projections for the Iberian Peninsula

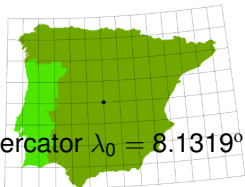
Trans. Mercator $\lambda_0 = 4^\circ\text{W}$



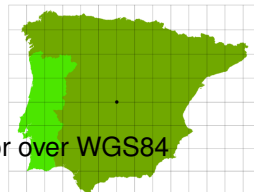
ETRS-LCC



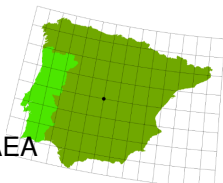
Trans. Mercator $\lambda_0 = 8.1319^\circ\text{W}$



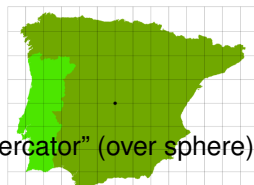
Mercator over WGS84



ETRS-LAEA



“Web Mercator” (over sphere)



LAB: Map projections and planar coordinates in QGIS

Data for the LAB exercises (which are in the folder `world_data` previously used).

- 1 Capitals of the World, coordinate reference system: ETRS89;
Source: <http://epp.eurostat.ec.europa.eu>,
EuroGeographics for the administrative boundaries;
File: compressed data in `cntr_capt_etr89.7z`
- 2 Countries 2010 (1:3'000'000): coordinate reference system: ETRS89;
Source: <http://epp.eurostat.ec.europa.eu>,
EuroGeographics for the administrative boundaries;
File: compressed data in `cntr_3_etr89.7z`;
- 3 Same as above;
Coordinate reference system: ETRS89 projected with the Lambert Conformal Conic map projection... but *the projection file is missing*;
File: compressed data in `etr89_to_lcc.7z`

LAB: Work with map projections in QGIS

- 1 File explorer: unzip `cntr_3_etr89.7z` and `etr89_to_lcc.7z` in your working directory;
- 2 Add the layer `CNTR_RG_03M_2010` and create a QGIS project;
- 3 Verify that the project CRS is ETRS89 (EPSG:4258) and that the coordinates are lat/long;
- 4 In **Project/Properties/CRS** check **Enable 'on the fly' CRS transformation** and select `ETRS89/ETRS-LAEA` (can use filter, and then double-click on `ETRS89/ETRS-LAEA`); [OK]
- 5 Observe that the current CRS is now EPSG:3035 (bottom right of the screen);
- 6 Observe that the map is an equal-area map; What are the type of distortions that you can observe? Compare with the Tissot indicatrix for the Lambert azimuthal map projection;
- 7 Add the layer `CNTR_CAPT_PT_2010` to your project; why does it fit nicely with the previous layer?

LAB: Work with map projections in QGIS (cont.)

- 8 Add the layer `etrs_to_lcc` to the QGIS project;
- 9 When you're prompted to "Specify CRS for the layer" just click [OK];
- 10 Explain why the data don't match;
- 11 Is the map of the World in `etrs_to_lcc` an equal-area projection? Does it preserve the shape of the continents? Check the previous slide that describes the Lambert Conic Conformal projection and the corresponding Tissot indicatrix;
- 12 To make the data match, we need to set the correct projection to `etrs_to_lcc`; this is done by creating a `.prj` file;
- 13 Remove the layer `etrs_to_lcc` from the QGIS project;
- 14 Add it again, but now when you are prompted for a CRS select `ETRS89 / ETRS-LCC`; [OK];
- 15 To save the CRS in a `.prj` file: right-click on the layer and choose **save as...**; notice that `ETRS89/ETRS-LCC` is now the layer's CRS; browse to the working folder and choose a new name.

LAB: calculating distances on map projections

Earlier (see Slide 75) we saw that one can do a **line measurement** with the appropriate tool of QGIS either over a **planar surface** or over an **ellipsoid** (QGIS let you choose from a list of around 30 distinct ellipsoids).

Measurements with the **measuring tool** are tedious, imprecise, and cannot be automatically saved. A better alternative is to use the spatial tool in menu **Vector/Analysis tools/Distance matrix** that outputs a file with the distances between points of interest. However this tool only computes **plane (Euclidean) distances**. Therefore, if the spatial data uses geographic coordinates, then one needs to:

- ➊ Project geographic coordinates into planar coordinates, using some map projection; Note that this possibly can introduce some errors;
- ➋ Compute the Euclidean distance between the projected planar (x and y) coordinates.

LAB: calculating distances on map projections (cont.)

Let's determine the distance between four European towns (Lisbon, Madrid, Berlin and Amsterdam) over the ETRS89-LAEA map projection, which is the map projection of choice to represent the whole EU. We will see that even if the distance between Lisbon and Madrid and between Amsterdam and Berlin are similar, the error induced by the projection is much more pronounced for the pair Lisbon-Madrid.

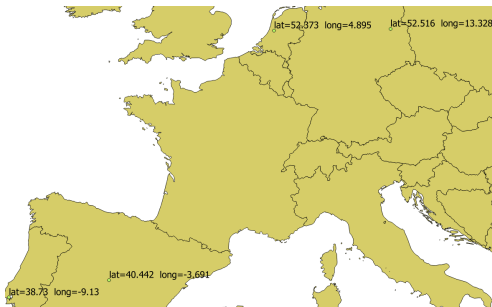


Figure: ETRS89-LAEA map projection with the original geographic coordinates of the four capital cities.

LAB: calculating distances on map projections (cont.)

- 1 Create a QGIS project with the layers `cntr_rg_03m_2010` and `cntr_capt_pt_2010`;
- 2 In **Project/Properties/CRS** check **Enable 'on the fly' CRS transformation** and select `ETRS89/ETRS-LAEA` (can use filter, and then double-click on `ETRS89/ETRS-LAEA, EPSG:3035`); [OK]
- 3 To project the coordinates of `cntr_capt_pt_2010` into `ETRS-LAEA` right-click on the layer and choose **Save As...**; Browse for CRS `ETRS-LAEA`; give the data set a new name `capitals_ETRS-LAEA (.shp)`; [OK] (you can check the option **Add saved file to map** to automatically add the new layer to the QGIS project);
- 4 Zoom over Europe; with the layer `capitals_ETRS-LAEA` active **select** the four capitals (Lisbon, Madrid, Berlin and Amsterdam);
- 5 Right-click on the layer `capitals_ETRS-LAEA` name and with **Save Selection As** create a new shapefile in your working folder called `LAMB_ETRS-LAEA (.shp)`.

LAB: calculating distances on map projections (cont.)

The last vector spatial data data set you created `LAMB_ETRS_LAEA` has planar coordinates; hence planar distances can be calculated between the four capitals.

- ⑥ Open **Distance matrix** tool in menu **Vector/Analysis tools/Distance matrix**; choose as **Input point layer** `LAMB_ETRS_LAEA` and as **Input Unique ID field** `NAME_ENGL`; repeat the same choices for Target point layer and Target unique ID field; Choose the **Linear** output matrix type and save it as `LAMB_distances (.csv)` in your working folder; [OK]; [CLOSE];
- ⑦ With the **Add Delimited Text Layer** button, add to the project the *non spatial* data set `LAMB_distances.csv`; In geometry definition choose **no geometry**; [OK];
- ⑧ Open attribute table of the new layer `LAMB_distances` to find the distances that were calculated;

The distances are calculated in meters, which are the units of the `ETRS_LAEA` map projection.

LAB: evaluating errors due to the map projection

Correct measurements versus measurements over the map projection:

1 Coordinates:

	Lisbon	Madrid
(lat,long)	38.729997,-9.129995	40.44222,-3.690972
(x,y)	2666496.388, 1947391.733	3160928.376, 2032430.657
	Berlin	Amsterdam
(lat,long)	52.516269,13.32757	52.373043, 4.894839
(x,y)	4546805.998, 3272610.385	3973634.911 , 3263695.960

2 Distances

- 1 **Geodetic distances.** Lisbon-Madrid: 504327.63m;
Amsterdam-Berlin: 573282.772m
- 2 **Euclidean distances.** Lisbon-Madrid: 501691.7m;
Amsterdam-Berlin: 573240.4m.

The difference is 0.52% for Lisbon-Madrid and 0.00739% for Amsterdam-Berlin.

3 Azimuths

- 1 **Geodetic azimuth.** Lisbon-Madrid: 66.14274669° at Lisbon;
Amsterdam-Berlin: 85.06588563° at Amsterdam.
- 2 **Cartesian azimuth.** Lisbon-Madrid: 80.241° at Lisbon;
Amsterdam-Berlin: 89.10896° at Amsterdam.

LAB: discussion of the previous results

In the example we used a projection `ETRS_LAEA` that preserves areas but distorts distances and angles (e.g. azimuth).

Why are the distortions larger in Portugal than in Germany? To understand this we can look at the description of the `ETRS_LAEA` map projection that you can find, for instance, in **Project Properties** and that you can compare with the description in Slide 97:

```
+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000  
+y_0=3210000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0  
+units=m +no_defs
```

`+lat_0=52 +lon_0=10` tells you what is the **origin of the projection**, where the **distortion vanishes**. That location is much closer to Berlin (lat=52.515, long=13.327) than Madrid. Since the distortions of the `ETRS_LAEA` map projection grow with the distance to the origin of the projection, then larger errors will occur on the margins of the map.

Decrypting PROJ.4 projections

Description of some common parameters that specify conversions between geographic and projection coordinates. Example: EPSG 3034 - Lambert Conic Conformal projection: `+proj=lcc +lat_1=35 +lat_2=65 +lat_0=52 +lon_0=10 +x_0=4000000 +y_0=2800000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m`

- ➊ `+proj=lcc`: Lambert Conic Conformal projection;
- ➋ `+lat_1=35 +lat_2=65`: latitude of first and second standard parallels;
- ➌ `+lat_0=52 +lon_0=10`: geographic coordinates of the origin of the projection;
- ➍ `+x_0=4000000 +y_0=2800000`: false easting and false northing, i.e. distances (m) from the origin ($x=0, y=0$) of the projection coordinates to the origin of the projection;
- ➎ `+ellps=GRS80`: ellipsoid name
- ➏ `+towgs84=0,0,0,0,0,0,0` : 3 or 7 term datum transform parameters to WGS84
- ➐ `+units=m`: units of projection coordinates

source: <http://trac.osgeo.org/proj/wiki/GenParms>

Decrypting PROJ.4 projections (cont.)

Example: EPSG Projection 22391 - Carthage / Nord Tunisie:

```
+proj=lcc +lat_1=36 +lat_0=36 +lon_0=9.9 +k_0=0.999625544  
+x_0=500000 +y_0=300000 +a=6378249.2 +b=6356515 +units=m
```

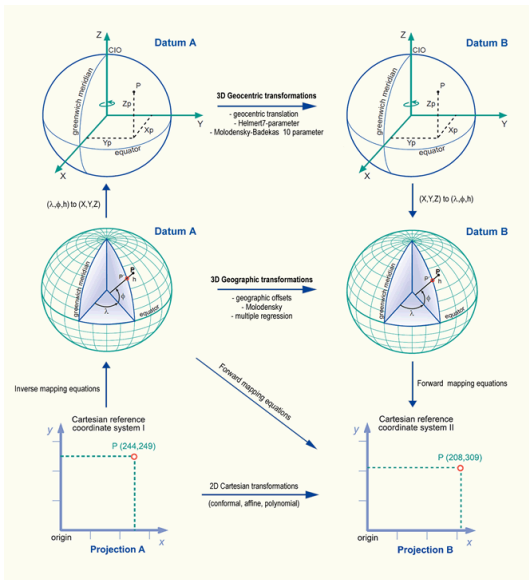
- ➊ `+proj=lcc`: Lambert Conic Conformal projection;
- ➋ `+lat_1=36`: latitude of the standard parallel;
- ➌ `+lat_0=36 +lon_0=9.9`: geographic coordinates of the origin of the projection;
- ➍ `+k_0=0.999625544`: scale parameter over the central meridian;
- ➎ `+x_0=500000 +y_0=300000`: false easting and false northing;
- ➏ `+a=6378249.2 +b=6356515`: ellipsoid semi-axes;
- ➐ `+units=m`: units of projection coordinates.

Overview of coordinate transformations

If there is only **one datum** (datum A), we only need to use the inverse projection and re-project;

If there are **two datums** (one associated to each CRS) then we can either:

- (a) Do a direct lat/long transformation;
- (b) Convert lat/long into 3D Cartesian coordinates and perform a transformation of the (X, Y, Z) coordinates from datum A to datum B.



Why do we need datum transformations?

Horizontal datums differ in the choice of ellipsoid and its point of origin that are used to approximate the surface of the earth:

- **Global datums** are defined by **the origin of the coordinate system**, its **orientation** and the dimensions of the **ellipsoid**;
- **Local horizontal datums** position a selected ellipsoid so as to closely fit the surface of the earth in a local area, and their origin is fixed to a location on the Earth's surface.

Since datums can be quite different, in order to convert one Coordinate Reference System into another, we need to transform the geodetic coordinates of datum A into the geodetic coordinates datum B – this is called **datum transformation**

The datum transformation (also known as **geographic transformation**) is typically done using some approximate method. GIS software in general has several options to do datum transformations.

Types of datum transformations

There are several types of data transformations. The most common are known as:

- ① Transformations of 3D Cartesian coordinates (X, Y, Z):
 - ① **Three parameter method**, just a translation of the center of the ellipsoid;
 - ② **Seven parameter method**, also known as Helmert transformation, Bursa-Wolf method, Coordinate Frame Rotation transformation, Position Vector Transformation; involves translation, rotation and change of scale (see Slide 114)
- ② Transformation of geodetic coordinates (lat,long):
 - ① **Molodensky method**; low number of parameters: low precision for large areas;
 - ② **Grid-based methods** use a dense set of points with known coordinates in both datums. Instead of estimating the parameters of the transformation, coordinates are obtained by spatial interpolation. This is potentially the most accurate method.

Datum transformation: 3D Cartesian coordinates

The transformation is based usually on the following parameters:

- 1 d_x, d_y, d_z : three parameters that define **translations** between the coordinate axes – the three parameter method uses only these parameters (units=meters);
- 2 r_x, r_y, r_z : three parameters that define **rotations** around the coordinate axes (angle units=seconds);
- 3 Δs relative scale factor (units=parts per million, can be positive or negative).

The parameters are typically determined by a **least-square method**. For a given area, the lat/long coordinates are known and lead to the estimation of the best set of parameters.

However, the transformation is not in general error-free. Seven-parameter methods tend to be associated to smaller errors than three parameter methods.

Decrypting PROJ.4: the `+towgs84` parameter

Consider the following datums:

- 1 the Portuguese Datum 73 (EPSG:4274)
`+proj=longlat +ellps=intl` (International 1909 or Hayford ellipsoid)
- 2 the ETRS89 datum (EPSG:4258)
`+proj=longlat +ellps=GRS80`

The datum transformation EPSG:4274 to EPSG:4258 is done with respect to a **reference datum** which is **WGS84**.

It can be defined by 3 or 7 parameters (see Slide 114):

- 1 EPSG:4274 (`+proj=longlat +ellps=intl`)
 - 1 three parameters: `+towgs84=-223,110,37,0,0,0,0;`
 - 2 seven parameters:
`+towgs84=-230.994,102.591,25.199,0.633,-0.239,0.900,1.950`
- 2 EPSG:4258 (`+proj=longlat +ellps=GRS80`)
 - 1 `+towgs84=0,0,0,0,0,0,0` since the datum GRS80 is considered essentially identical to the datum WGS84

Decrypting PROJ.4: putting it all together

Example: A **transformation between projected coordinates**

EPSG:27492 \Rightarrow EPSG:3763
(Portugal HG73) (ETRS 1989 - Portugal TM06)

where the projected CRS are described by

```
EPSG:27492 +proj=tmerc +lat_0=39.666666  
+lon_0=-8.131906111111112 +k=1+x_0=180.598  
+y_0=-86.98999999999999 +ellps=intl +units=m
```

```
EPSG:3763 +proj=tmerc +lat_0=39.66825833333333  
+lon_0=-8.133108333333334 +k=1 +x_0=0 +y_0=0  
+ellps=GRS80 +units=m
```

involves a **datum transformation over geographic coordinates** between

EPSG:4274 \Rightarrow EPSG:4258
+proj=longlat +ellps=intl +proj=longlat +ellps=GRS80

Decrypting PROJ.4: putting it all together (cont.)

Therefore, a complete description of the transformation of Portugal HG73 (EPSG:27492) **projected coordinates** to ETRS 1989 - Portugal TM06 (EPSG:3763) **projected coordinates** can be written using:

- 1 7 parameters, leading to positional errors up to 1m for Portugal:

```
+init=epsg:27492 +towgs84=-230.994,102.591,25.199,  
0.633,-0.239,0.900,1.950
```

```
+to
```

```
+init=epsg:3763 +towgs84=0,0,0,0,0,0,0
```

- 2 3 parameters, leading to positional errors up to 2m for Portugal:

```
+init=epsg:27492 +towgs84=-223,110,37,0,0,0,0
```

```
+to
```

```
+init=epsg:3763 +towgs84=0,0,0,0,0,0,0
```

For more details see http://proj.maptools.org/man_cs2cs.html.

Datum transformations: grid-based methods

Grid transformation is a method of converting coordinates from one datum to another by interpolation within a grid of points.

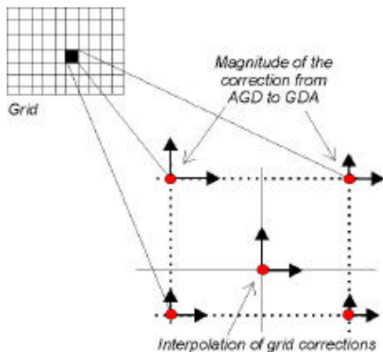


Figure: Illustration of grid transformation. In this example from Australia, the goal is to convert geographic coordinates from the old AGD datum to the GPS compatible GDA datum. The arrows at the grid points represent differences between the two datums for the same location.

Datum transformations: grid-based methods (cont.)

The illustration of Slide 118 shows how the transformation is done: to get the actual datum shift for a coordinate at any location, the **four closest known coordinates** are retrieved from a grid file, and then they are interpolated using **bi-linear interpolation**.

Among other formats, grid-based datum transformations can be done with NTV2 files (National Transformation version 2). A single file can contain various grids with different accuracies. Available grids can be downloaded and integrated into GIS software. Software has to be designed in a way that a coordinate is converted using the most accurate grid available. The default extension for NTV2 grid files is **.GSB** (Grid Shift Binary)

Currently, grid files are available for North America, Australia, Brazil, Canada, France, Germany, New Zealand, Portugal, South Africa, Spain, Switzerland, United Kingdom and Venezuela.

Decrypting PROJ.4: the `+nadgrid` parameter

Consider again the transformation of
Portugal HG73 (EPSG:27492) **projected coordinates**
to
ETRS 1989 - Portugal TM06 (EPSG:3763) **projected coordinates**

Using the NTV2 file `pt73_e89.gsb` the transformation can be described as:

```
+init=epsg:27492 +nadgrids=pt73_e89.gsb  
+to  
+init=epsg:3763
```

This transformation leads to much lower positional errors than the transformations described in Slide 117, since the errors are below 0.28m for Portugal.

Quiz: decrypting PROJ.4 parameters

Explain the meaning of the following PROJ.4 parameters:

- ❶ **EU recommended transverse Mercator projection for Portugal:**
`+ellps=GRS80 +towgs84=0,0,0 +proj=tmerc
+lat_0=39d40'05.73''N+lon_0=08d07'59.19''W;`
- ❷ **EPSG Projection 2176 - ETRS89 / Poland CS2000 zone 5:**
`+proj=tmerc +lat_0=0 +lon_0=15 +k=0.999923
+x_0=5500000 +y_0=0 +ellps=GRS80 +units=m;`
- ❸ **SR-ORG Projection 7564 - China Albers Equal Area Conic**
`+proj=aea +lat_1=27 +lat_2=45 +lat_0=35 +lon_0=105
+x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=km.`

Quiz: decrypting PROJ.4 parameters (cont.)

Compare the following descriptions of the map projection known as *Lisbon Portuguese Grid* (EPSG:20791):

❶ From QGIS:

```
+proj=tmerc +lat_0=39.66666 +lon_0=1 +k=1 +x_0=0  
+y_0=0 +ellps=intl  
+towgs84=-304.046,-60.576,103.64,0,0,0,0  
+pm=lisbon +units=m +no_defs;
```

❷ From <http://spatialreference.org>:

```
+proj=tmerc +lat_0=39.66666 +lon_0=1 +k=1 +x_0=0  
+y_0=0 +ellps=intl +pm=lisbon +units=m +no_defs.
```

Explain why the second description to perform a coordinate transformation that would involve a **datum transformation** would lead to high positional errors.

Quiz: decrypting PROJ.4 parameters (cont.)

In the PROJ.4 description

```
+proj=tmerc +lat_0=39.66666 +lon_0=1 +k=1 +x_0=0  
+y_0=0 +ellps=intl  
+towgs84=-304.046,-60.576,103.64,0,0,0,0 +pm=lisbon  
+units=m +no_defs,
```

- ➊ `+towgs84=-304.046,-60.576,103.64,0,0,0,0` is the sole parameter of interest with regard to the datum transformation;
- ➋ `+pm=lisbon` is a parameter that indicates the *alternate prime meridian*; it would be the International Reference Meridian (Greenwich) by default;
- ➌ `+lon_0=1` indicate that the central meridian of the projection (where the projection distortion vanishes) is 1° East of the prime meridian;
- ➍ `+no_defs`: prevents from using the default file if installed;

LAB: projected coordinates transformation in QGIS

The goals of the exercise are: (A) create point spatial layers in QGIS from a list of coordinates; (B) Use QGIS and available data to evaluate positional errors of a projected coordinate transformation involving a datum transformation.

Data are in folder `datum_transformation_portugal_data`:

- 1 `ETRS89_point_coordinates.csv` is a *comma separated values* file that contains projected coordinates of a set of points in Portugal in the CRS: ETRS 1989 - Portugal TM06 (EPSG:3763); it also contains the DMS lat/long coordinates of the same points in ETRS89 datum (EPSG:4258);
- 2 `HG73_point_coordinates.csv` is also a *comma separated values* file that contains projected coordinates of a set of points in Portugal in the CRS: Portugal HG73 (EPSG:27492);
- 3 `Portugal` is a spatial vector polygon data file (shapefile format) that includes a `.prj` definition.

LAB: projected coordinates transformation in QGIS (cont.)

- 1 Copy folder `datum_transformation_portugal_data` to your PC;
- 2 Open QGIS and add to the empty project `Portugal.shp`: verify that the CRS is ETRS89 (EPSG:4258) with geographic coordinates;
- 3 Set the project CRS to **ETRS89 / Portugal TM06** (EPSG:3763) in the **Project Properties** menu; Enable “on the fly” CRS transformation.
- 4 Click the button **Add delimited text layer** to open the dialog menu **Create a Layer from a delimited Text file** to create a layer from the coordinates in `HG73_point_coordinates.csv`; see Slide 126 for options to use;
- 5 Specify CRS: select EPSG:27492 – Datum 73 / Modified Portuguese Grid (deprecated);
- 6 Observe the `+towgs84` parameter of the CRS: what does it mean?

LAB: projected coordinates transformation in QGIS (cont.)

File Name: Y:/Aulas/SIG MedFor/sig_medfor_2013_2014_isa/portugal-coordinates/HG73_point_coordinates.csv Browse...

Layer name: HG73_point_coordinates Encoding: ISO-8859-1

File format: ☐ CSV (comma separated values) ☒ Custom delimiters ☐ Regular expression delimiter

☐ Comma ☒ Tab ☐ Space ☐ Colon ☒ Semicolon

Other delimiters: Quote: " Escape: "

Record options: Number of header lines to discard: 0 ☒ First record has field names

Field options: ☐ Trim fields ☐ Discard empty fields ☐ Decimal separator is comma

Geometry definition: ☒ Point coordinates ☐ Well known text (WKT) ☐ No geometry (attribute only table)

X field: X-HG73 Y field: Y-HG73 ☐ DMS coordinates

Layer settings: ☐ Use spatial index ☐ Use subset index ☐ Watch file

	Location	X-HG73	Y-HG73
1	PATAIAS	-74583,460	2462,139
2	BARROCAL 2	80933,352	120655,173
3	BATEL	-70042,006	-102637,802
4	CASCALHEIRA	-23357,254	-111933,066
5	CRASTA ALTA	-64387,393	32482,001
6	FANADIA	-81021,530	-33211,393

OK Cancel Help

Figure: Example of QGIS dialog menu **Create a Layer from a delimited Text file.**

LAB: projected coordinates transformation in QGIS (cont.)

- 7 Convert the layer `HG73_point_coordinates` to **ETRS89 / Portugal TM06** (EPSG:3763) projected coordinates and create a new shapefile with that CRS; to do this right-click on `HG73_point_coordinates` and select **save as ...**; select a name and the correct CRS; [OK]; add new shapefile to your project;
- 8 What was the applied datum transformation? Notice that the standard current version of QGIS (Dufour 2.0.1) doesn't prompt for a datum transformation – this is going to be addressed in the forthcoming version (QGIS 2.2.0);
- 9 Click the button **Add delimited text layer** to open the dialog menu **Create a Layer from a delimited Text file** to create a layer from the coordinates in `ETRS89_point_coordinates.csv`; These coordinates were measured over ETRS89 independently of the `HG73_point_coordinates`;

- 10 Compare over **ETRS89 / Portugal TM06** (EPSG:3763) the projected coordinates obtained by the transformation of `HG73_point_coordinates` with the (correct) coordinates in `ETRS89_point_coordinates.csv`; You can zoom in to observe that there are differences; You can use the tool **matrix distance** of Slide 103 to calculate those differences (in meters).

Georegistration

Definition (Georegistration, or georeferencing)

The georegistration process consists of establishing a location of a data set (e.g. an image) within a coordinate reference system.

Our goal can be, for instance, to merge a high resolution imagery with a geographic data set with known coordinate reference system (CRS).

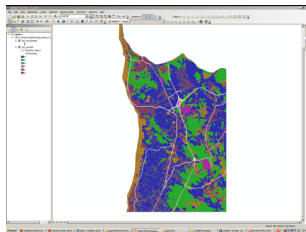


Figure: On the left a land cover map with known coordinates. On the right the image to be mapped onto it.

Registration points (aka control points)

The idea is to use control points to establish the coordinate transformation:

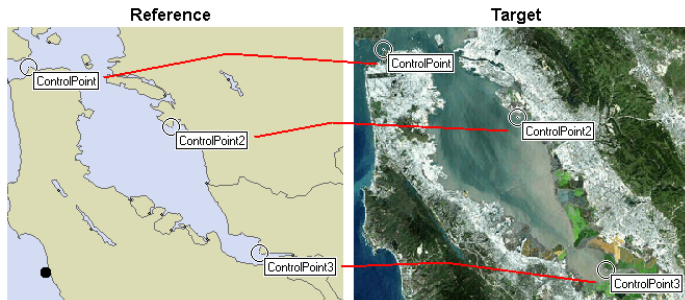


Figure: Control points can be identified on the **reference** (the map with known CRS) and the **target** image to be georegistered. At the end of the process of georegistration the target image will be resampled over the reference CRS using the control points as a guide.

Related concepts

Definition (Georegistration with control points)

The georegistration process consist of mapping one image to another (with known CRS) in a way that “optimally” aligns the two images based on a set of control points (i.e. it is *feature-based*).

Definition (Image registration)

Process of transforming different sets of data into one coordinate system. Data may be multiple photographs, data from different sensors, from different times, or from different viewpoints (can be *feature-based* or *intensity-based*)

Definition (Orthorectification)

The process of correcting the geometry of an image so that it appears as though each pixel were acquired from *directly overhead*. Orthorectification uses elevation data to correct terrain distortion in aerial or satellite imagery.

Georegistration steps

- 1 Identify control points on the target image and on the reference;
- 2 Choose a transformation from the target image coordinates to the CRS of the reference and estimate its parameters;
- 3 Resample the target image using a certain resampling rule and the estimated transformation;
- 4 Estimate the georegistration error.

The georegistration process is typically *iterative*, i.e., if the error is too high, control points must be deleted or added and the transformation needs to be adjusted.

Coordinates of the target image and reference CRS

Even if the image to be georegistered doesn't have a well defined CRS, we can still consider that it has its own coordinates, e.g. row number and column number of each pixel in the image.

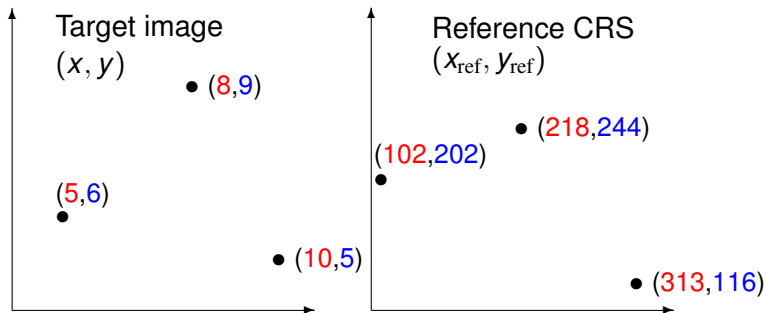


Figure: The figure illustrates 3 control points with their coordinates in the image (e.g. column number, row number) and their coordinates in the reference CRS.

1. Control points

Control points are features on the surface: they can be road intersections, lakes, watercourses, coast line features, among other features with clear boundaries.

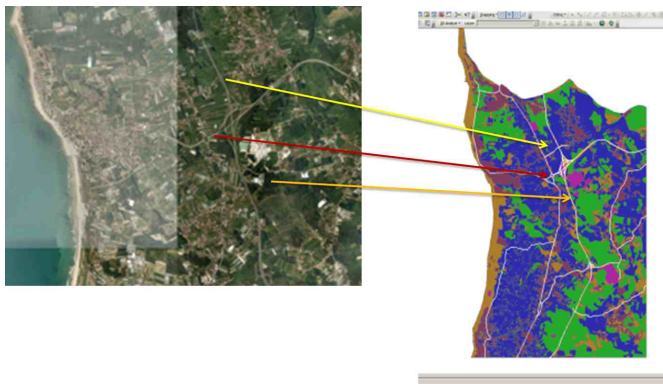


Figure: Here we identify on the target feature 3 points: two are road intersections and the third is a corner of a distinct parcel on the ground. Those 3 points can be identified on the map with known CRS.

Distribution of control points over the target image

Control points must be distributed homogeneously within the image. Some areas like the images boundaries, areas of lowest and highest elevation as well as areas where adjacent images overlap must be well represented.

The transformation will only be accurate over the subset of the target image which contains control points.

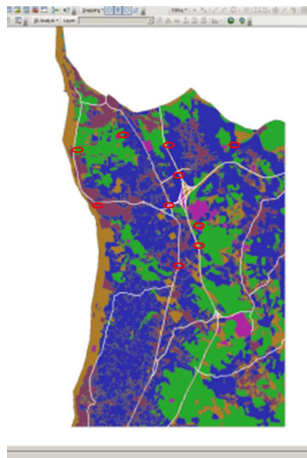


Figure: Control points that will determine the transformation.

2. Coordinate transformation

The transformation of an image is known as *rubber-sheeting* or *rubber-masking*.

- (a) Choose a coordinate transformation function (usually a polynomial);
- (b) Estimate the parameters of the transformation (the polynomial coefficients).

Number of coefficients . A polynomial of degree N has

$(N + 1)(N + 2)/2$ coefficients:

- $p(x, y) = a + bx + cy$ (affine function) has 3 coefficients;
- $p(x, y) = a + bx + cy + dxy + ex^2 + fy^2$ has 6 coefficients;
- if the polynomial has degree 3 it has 10 coefficients.

Type of geometric transformation :

Affine function – translation, rotation, change of scale;

Higher degree polynomial – those and other more complex distortions of the image.

Polynomial transformation of an image

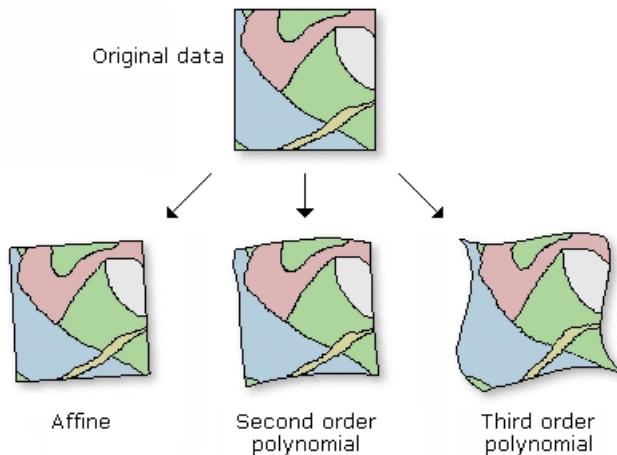


Figure: The type of transformation of the image depends on the degree of the polynomial.

Estimation of the coefficients of the polynomial transformation

Let any control point be defined by

- x_i, y_i are the coordinates on the target image;
- $x_{i,\text{ref}}, y_{i,\text{ref}}$ are the coordinates on the reference CRS.

The transformation is defined by:

$$\begin{cases} x_{\text{transf}} &= p_x(x_{\text{ref}}, y_{\text{ref}}) \\ y_{\text{transf}} &= p_y(x_{\text{ref}}, y_{\text{ref}}) \end{cases}$$

where p_x and p_y are polynomials. p_x returns the new x coordinate on the target image and p_y returns y coordinate on the image target.

The estimation of the coefficients of p_x and p_y is a *mean square* estimation, i.e. it minimizes the **positional error** of the transformation. This depends on the **residuals** for the set of control points. The **residual** of the i -th control point **with respect to the reference CRS** is

$$\sqrt{(p_x^{-1}(x_i, y_i) - x_{i,\text{ref}})^2 + (p_y^{-1}(x_i, y_i) - y_{i,\text{ref}})^2}.$$

where $p_x^{-1}(x_i, y_i)$ and $p_y^{-1}(x_i, y_i)$ are the x and y coordinates of the inverse transformation of the control point (x_i, y_i) onto the CRS.

Example: an affine transformation estimated from just 3 (non colinear) control points

The 3 control points have (5,6), (8,9) and (10,5) coordinates on the target image and (102,202), (218,244) and (313,116) coordinates on the CRS. What is the linear system that defines the transformation?

$$\begin{cases} 5 &= a_x + 102 b_x + 202 c_x \\ 8 &= a_x + 218 b_x + 244 c_x \\ 10 &= a_x + 313 b_x + 116 c_x \\ 6 &= a_y + 102 b_y + 202 c_y \\ 9 &= a_y + 218 b_y + 244 c_y \\ 5 &= a_y + 313 b_y + 116 c_y. \end{cases}$$

This system has one unique solution which is $a_x = 1.8977$, $b_x = 0.0248$, $c_x = 0.0028$, $a_y = -3.2011$, $b_y = 0.0115$, $c_y = 0.0398$

Therefore, the coordinate transformation for the target image is:

$$\begin{cases} x_{\text{transf}} &= 1.8977 + 0.0248 x_{\text{ref}} + 0.0028 y_{\text{ref}} \\ y_{\text{transf}} &= -3.2011 + 0.0115 x_{\text{ref}} + 0.0398 y_{\text{ref}} \end{cases}$$

Transformation with 3 control points (illustration)

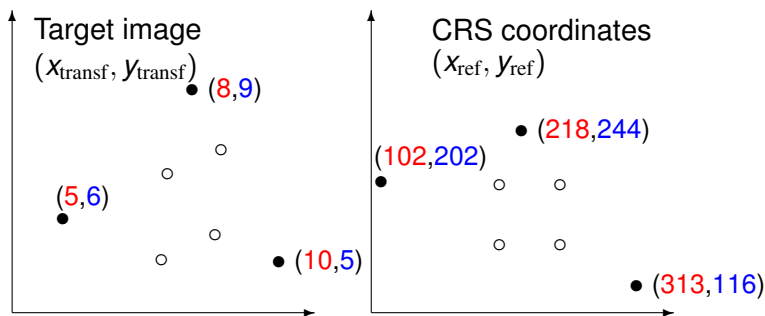


Figure: The transformation maps the “good” coordinates (the CRS coordinates) to the “arbitrary” target image coordinates. Since the transformation is affine the 4 points represented by \circ can be shifted, rotated, and their x and y coordinates can be re-scaled: the original square is transformed into a parallelogram.

$$\begin{cases} x_{\text{transf}} &= 1.8977 + 0.0248 x_{\text{ref}} + 0.0028 y_{\text{ref}} \\ y_{\text{transf}} &= -3.2011 + 0.0115 x_{\text{ref}} + 0.0398 y_{\text{ref}} \end{cases}$$

Controlling the error

The previous approach (number of control points equal to the number of coefficients) is not adequate.

In fact, regardless to the accuracy of the coordinates of the control points (over the target image and over the CRS) the residuals are always zero.

To evaluate the error of the transformation, one needs to have more control points than coefficients: the rule of thumb is to have **2 to 3 times more control points than coefficients of the transformation**.

In that case, the residuals can be used to estimate the error associated to the georegistration.

An affine transformation with more than 3 control points

Suppose we have a 4th control points with coordinates (7,5) e (190,150). What is the new system of equations?

$$\left\{ \begin{array}{lcl} 5 & = & a_x + 102 b_x + 202 c_x \\ 8 & = & a_x + 218 b_x + 244 c_x \\ 10 & = & a_x + 313 b_x + 116 c_x \\ 6 & = & a_y + 102 b_y + 202 c_y \\ 9 & = & a_y + 218 b_y + 244 c_y \\ 5 & = & a_y + 313 b_y + 116 c_y \\ 7 & = & a_x + 190 b_x + 150 c_x \\ 5 & = & a_y + 190 b_y + 150 c_y \end{array} \right.$$

This system (8 equations, 6 variables) doesn't have in general a solution. However, there is a set of optimal $a_x, a_y, b_x, b_y, c_x, c_y$ that **minimize** the residuals (more precisely, the root mean square of the residuals) between the reference coordinates for the control points and the coordinates obtained by georeferencing them.

3. Resampling: re-project the target image into the CRS)

Since the target image is a set of pixels with a certain dimension (for instance, the spatial resolution of a satellite image), one can consider a grid in the CRS where the cells have the same dimensions.

Denote by $(\bar{x}_{\text{ref}}, \bar{y}_{\text{ref}})$ the center of a cell in the CRS and denote by $\bar{x}_{\text{transf}} = p_x(\bar{x}_{\text{ref}}, \bar{y}_{\text{ref}})$ and $\bar{y}_{\text{transf}} = p_y(\bar{x}_{\text{ref}}, \bar{y}_{\text{ref}})$ the transformed coordinates.

$(\bar{x}_{\text{transf}}, \bar{y}_{\text{transf}})$ lies somewhere over the target image, not necessarily over the center of a pixel of the target image.

Therefore, we need a **resampling** criteria to associate to each cell – with center $(\bar{x}_{\text{ref}}, \bar{y}_{\text{ref}})$ – a value from the target image.

nearest neighbor : we associate the target image value of the pixel closest to $(\bar{x}_{\text{transf}}, \bar{y}_{\text{transf}})$;

interpolation rule : we associate a combination of the values of the pixels closest to $(\bar{x}_{\text{transf}}, \bar{y}_{\text{transf}})$.

Illustration of the resampling procedure

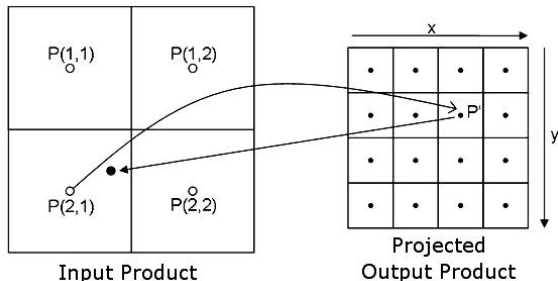


Figure: On the left is the target image and on the right the *gridded* CRS. Point (\bullet) with coordinates $(\bar{x}_{\text{transf}}, \bar{y}_{\text{transf}})$ is the transformation of $P' = (\bar{x}_{\text{ref}}, \bar{y}_{\text{ref}})$, the center of a CRS cell. Since (\bullet) lies on pixel with center $P(2, 1)$, then by the **nearest neighbor** criteria the value of $P(2, 1)$ is assigned to cell P' . By an **interpolation rule** a weighted average of the values of pixels $P(1, 1)$, $P(1, 2)$, $P(2, 1)$ e $P(2, 2)$ is assigned to P' .

The most common interpolations are **bilinear** and **cubic**.

4. Evaluate the error

If the transformation is accurate, then the residuals for the control points should be low.

There are two ways to estimate the error:

- (a) *real error*: when the positional error is computed from a set of control points which was not used to estimate the coefficients of the transformation;
- (b) *apparent error*: when the positional error is computed over the same control points that are used to estimate the coefficients of the transformation.

The positional error **shouldn't be larger than the spatial resolution of the georeferenced target image**. Ideally, it should even be *smaller than half of its spatial resolution*.

LAB: Georeferencer tool in QGIS

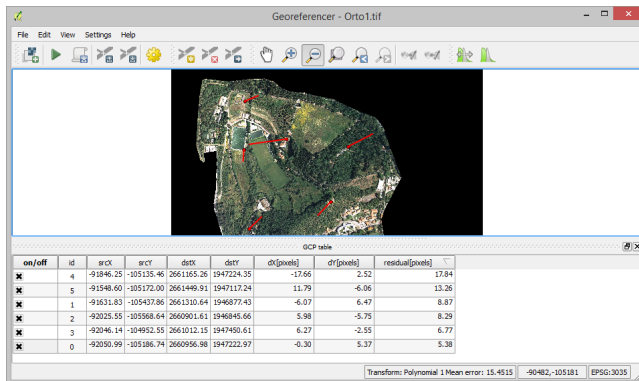


Figure: The coordinates of the ground control points are listed in the **GCP table**. The **src** (source) coordinates are the original coordinates over the target image. The **dst** coordinates are the coordinates over the reference CRS. Each line of the table corresponds to one control point. Each **residual** is listed at the end of the line: control points with high residuals should be double-checked and perhaps removed. The user can choose the **settings** of the transformation; transformation type, resampling method among others.

The target image after georegistration



Figure: In this example, georegistration was done with 10 control points, an affine transformation and a nearest neighbor resampling rule.

Basic rules to follow when georegistering an image

- I The **number of control points** depends on the degree of the transformation polynomial: its should be 2 to 3 times larger than the number of coefficients of the polynomial.
- II Those control points should be **homogeneously distributed** across the target image.
- III Georegistration is an **iterative** process:
 - (a) Identify the coordinates of a set S of n control points;
 - (b) estimate the coefficients of the polynomial;
 - (c) If the *positional error* is acceptable, then resample the target image using the transformation;
 - (d) Otherwise, remove control points with high residuals and/or add new control points and go back to (b).

To evaluate the positional error one has to estimate the residuals over the reference CRS.

LAB: Georeferencing in QGIS

The goals of the exercise are: (A) perform a georeferentiation of two high resolution images over the EPSG:3035 CRS; (B) Produce a mosaic of those two images.

Data are in folder `georegistration_lisbon_data`:

- 1 A Bing image of part of Lisbon.
Coordinate reference system (EPSG:3035): ETRS89-LAEA
file name `ISA_bing600dpi`
- 2 Two very fine resolution (10cm) aerial images of ISA
Coordinate reference system: unknown
file names: `orto1.tif` and `orto2.tif`

The documentation on the QGIS Georeferencing plugin can be found [here](#).

LAB: Georeferencing in QGIS (cont.)

- 1 Copy folder `georegistration_lisbon_data` to your PC and unzip `georegistration_lisbon_data.7z` (0.5GBytes);
- 2 Add `ISA_bing600dpi` to your project; Check its Coordinate Reference System (CRS);
- 3 Check that the spatial resolution (cell size) of that spatial raster data set is around 55cm;
- 4 In **view/toolbars** activate **raster** toolbar;
- 5 Click on the **Georeferencer** button to open the georeferencer window;
- 6 It's best to have both the QGIS desktop window and the Georeferencer window visible;
- 7 Use **File/open raster** (or the icon **open raster**) to open `Orto1.tif` in the Georeferencer window; When prompted for a CRS just click [OK];

LAB: Georeferencing in QGIS (cont.)

- 8 Use **edit/add point** (or the icon) to start adding control points; first identify the control point on the target (Georeferencer windows) and on the reference (QGIS map) – use zoom if necessary – and then digitize the location on the target and, when prompted, select **From map canvas** to digitize the location on the reference map; The point is added to the GCP table;
- 9 Repeat for control points well distributed around the target image;
- 10 When you have enough points (see discussion in Slide 148) you can choose the settings for the transformation in **Settings/Transformation settings** (or the icon); You need to select the correct CRS (under **Target SRS**) which will be EPSG:3035 since you're using coordinates from that CRS to define the transformation; You need to define the name of the output layer;
- 11 Control points can be deleted or moved if necessary with the **edit** menu;

LAB: Georeferencing in QGIS (cont.)

- 12 On the GCP table, click on the **residual** column to sort the points by decreasing order; then you can use the **on/off** column to exclude points with large residuals and re-estimate the errors of the transformation;
- 13 When you have enough points you can click on **Start georeferencing**; Check that the CRS of the output layer is the CRS that you have chosen (EPSG:3035);
- 14 If you have more than 6 points (you should!) compare the results you obtain with a 1st degree polynomial and a 2nd degree polynomial in terms of residuals and in terms of the quality of the fit between the georeferenced target image and the reference image; Even if the residuals are smaller for the 2nd degree polynomial, it is likely that the fit is better for the simpler transformation for the data in hand;
- 15 Proceed similarly for the `Orto2.tif` target image.

LAB: Georeferencing in QGIS (cont.)

- 16 To merge the resulting georeferenced images `Orto1_reg.tif` and `Orto2_reg.tif` you can use the **merge** tool under the menu **raster/Miscellaneous/Merge**;
- 17 Choose as input files `Orto1_reg.tif` and `Orto2_reg.tif` and name the output `mosaic.tif`; Areas where both images overlap will contain the values from the last input.

Image orthorectification

The distortions in a target image caused by the **relief** cannot in general be corrected by georegistration, even with a transformation of high degree (say, 3rd degree polynomial).

When the relief is moderate the target image should be corrected using a digital elevation model. This can be done as long as one knows what is the precise geometry of acquisition of the image (aerial photography, satellite imagery).

Orthorectification is done in two steps. Firstly, the (column, row) coordinates of the target image pixels are converted into 3D (x, y, z) coordinates. Secondly, the (x, y, z) coordinates are projected onto the surface.

Illustration of the orthorectification process

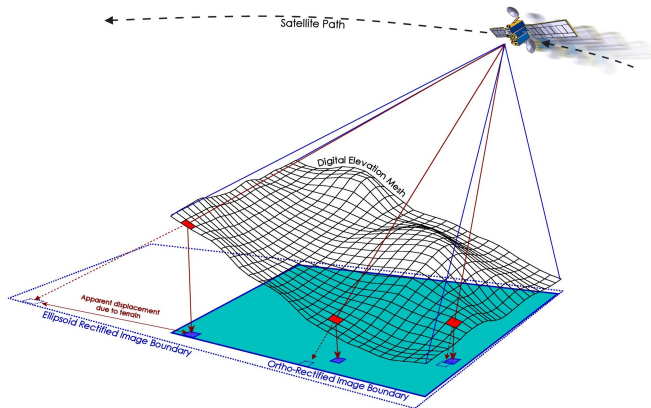


Figure: Illustration of the distortions caused by the relief and geometry of image acquisition. Both contribute to the distortion. The pixels should be projected vertically but they are displaced in the target image. If the elevation of a pixel is h , and the view zenith angle (angle with respect to the zenith) is θ then the horizontal displacement is roughly $h \times \cot(\pi/2 - \theta)$, so it increases with h .

Georegistration and orthorectification

The figures below show images with 1m spatial resolution for an area with elevation with a 60m range. The orange polygons indicate the streets location. Without orthorectification the match is poor, which means that there is a substantial horizontal displacement due to the relief.



Figure: This image is georegistered but not orthorectified.

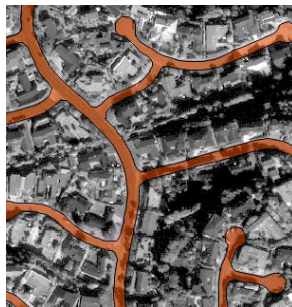


Figure: This image is both georegistered and orthorectified.

Image distortions over large areas: combining techniques

Image over large areas (e.g. coarse resolution satellite imagery) suffer distortions due to Earth's shape. In this case, it is better to first perform a coarse correction of the image using mathematical models that combine the acquisition geometry and the Earth's shape. The resulting image can then be orthorectified and georegistered.

Georegistration alone only works well in small and relatively flat areas.

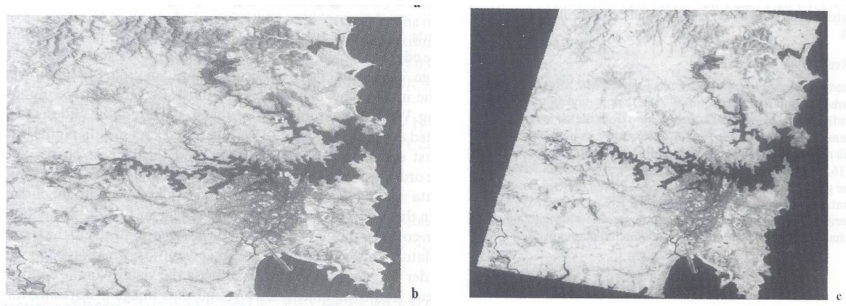


Figure: An raw satellite image on the left and the same image after geometrical corrections on the right.

Quiz

Suppose you want to georegister an image which coordinate reference system is unknown.

- 1 What is the major challenge you face when applying this technique?
- 2 Which role can have a GPS device in this process?
- 3 Suppose that the distortions in the image *cannot* be corrected by shifting, rotating or rescaling it. What consequences this has in the application of the technique? In particular, how does this affect the choice of the transformation and the number of control points to be collected?
- 4 Suppose that the target image is a panchromatic aerial photograph where each pixel takes *integer* values between 0 and 255 (1 bit/pixel). After georegistration, you observe that some pixels have decimal (non integer) values (e.g. some pixel has value 155.6). What does this tell you about the resampling method that was used?

Quiz (cont.)

- 5 Suppose that your target image has 10m spatial resolution. Does it make sense to resample it to a georeferenced grid where each cell is 5m wide? What will happen if you use a nearest neighbor rule?

Creating and maintaining geographic databases

GIS data

To do spatial analysis we need **data**. In general, there is a large quantity of data, which means that data needs to be organized to be effectively and efficiently used.

- 1 a **dataset** is a single collection of information, without any requirement as to form or organization;
- 2 a **database** is a collection of non-redundant data shareable among different applications.

We can use the concepts of database and database management to describe the necessary requirements.

Definition (Database)

A database is an integrated set of data on a particular subject.

Definition (Database Management Systems)

A DBMS is a software application designed to organize the efficient and effective storage and access of data.

Source: Longley *al.* 2011

Relational databases and geographic extensions

All geographic information systems use database management systems (DBMS) technology for data management.

Main types of DBMS:

- 1 **Relational DBMS**: essentially applied to a set of tables;
- 2 **Object DBMS**: object-oriented DBMS;
- 3 **Object-Relational DBMS**: an **extension** of relational DBMS to deal with objects, e.g. spatial objects. These are also known as **geographic DBMS extensions**.

Most GIS systems rely on **Object-Relational DBMS**, which means that they have the capabilities of relational DBMS (e.g. they allow **attribute queries**), but also the capability of using spatial characteristics of the objects in the database (for instance, they support **spatial queries**).

Many relevant DBMS have been extended with spatial capabilities (e.g. the open source DBMS PostgreSQL has been extended to PostGIS) to permit integration with GIS.

Database tables and keys

A **table** is just a two-dimensional array of rows and columns. However, **database tables** have to follow the principles described in Slide 33.

A **primary key** (or key) is an attribute of the table which has distinct values. We will admit that the every table has a key.

	ISO_2DIGIT	ISO_3DIGIT	LONG_NAME	STATUS	POP2007
0	FK	FLK	Falkland Islands (Islas Malvinas)	Non-Self-Governing Territory of the UK	-99999
1	GF	GUF	Department of Guiana	Oversea Department of France	203321
2	GY	GUY	Co-operative Republic of Guyana	UN Member State	769095
3	PN	PCN	Pitcairn, Henderson, Ducie, and Oeno Isl	Non-Self-Governing Territory of the UK	-99999
4	GS	SGS	South Georgia and the South Sandwich Is	UK Territory	-99999
5	SH	SHN	St. Helena	Non-Self-Governing Territory of the UK	7543
6	SR	SUR	Republic of Suriname	UN Member State	470784
7	TT	TTO	Republic of Trinidad and Tobago	UN Member State	1056608

In attribute table of layer `entry08` above, we can suppose that the first column (with values $\{0, 1, 2, \dots\}$) is an attribute called `ID` (hidden). This attribute, or any of the attributes `ISO_2DIGIT`, `ISO_3DIGIT`, `LONG_NAME` are all possible keys, but the attributes `STATUS` or `POP2007` are not since the same value can appear more than once.

Describing database tables: the schema and attribute domains

A database table can be described by **schema** that lists the attributes and identifies the key of the table.

For instance, the schema of the database table in Slide 163 is

```
cntry08 (ID, ISO_2DIGIT, ISO_3DIGIT, LONG_NAME, STATUS, POP2007)
```

The underlined attribute is the designated **key** for the table and therefore identifies uniquely all rows of the table. It is usual to create a key by enumerating the rows hence ensuring that no rows have identical values.

In addition to the schema above, the **domains** of the attributes can be described. As indicated in Slide 33 the domain can be specified by a data type (e.g. integers, “real” numbers, character strings, date formats).

For instance, if the domain of ISO_2DIGIT is the set of 2 character strings, we can write

```
ISO_2DIGIT [string; 2].
```


Feature classes and database tables

GIS can handle two kinds of database tables:

- 1 **spatial database tables**: each row of the table is associated to a geographic feature.
- 2 **non spatial database tables**: tables *per se*, not associated to geographic features.

In Slide 6 we discussed the concept of **geographic feature** which is used to represent **objects in space** in GIS.

We also saw that geographic features are organized in collection of data on a particular **theme** (e.g. water pumps, victims locations, world countries), which are called **layers** of **feature classes**.

There is a **one-to-one correspondence** between geographic features and rows of the layer's attribute table (see illustration in Slide 16)

Relational databases are made up of tables. Each feature class is stored as a table. To save space and improve performance, coordinate values are usually stored separately in a compressed form.

Relational join among database tables

Database tables can be joined together to create new views of the database.

Tables are joined together using a common attribute. Two database tables can be joined if **the key of one table matches some attribute of the other table**.

	Name	OfName	ISO2
174	Tonga	the Kingdom of Tonga	TO
177	Turkey	the Republic of Turkey	TR
175	Trinidad and Tobago	the Republic of Trinidad and Tobago	TT
179	Tuvalu	Tuvalu	TV
184	United Republic of Tanzania	the United Republic of Tanzania	TZ
181	Ukraine	Ukraine	UA
180	Uganda	the Republic of Uganda	UG

In the table above, the key `ISO2` matches the attribute `ISO_2DIGIT` of the table in Slide 163, allowing the tables to be joined.

LAB: Relational joins with QGIS

Suppose you want to create a map that shows the *variation in CO₂ equivalent emissions caused by agricultural activity* over the world between 2009 and 2010. A world map in format shapefile is easy to find. The Food and Agriculture Organization (FAO) produces tables of estimated emissions for all countries. The difficulty is to find the right way to relate those two datasets.

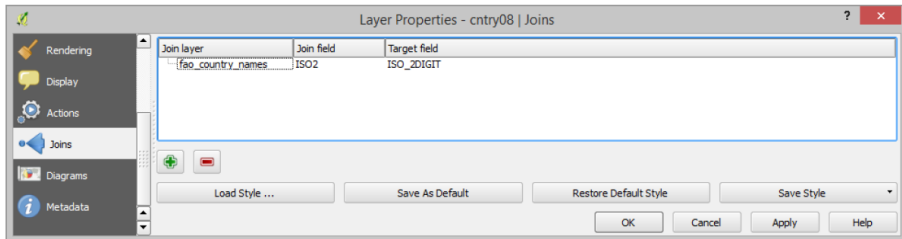
Data are in folder `world_agriculture_CO2_emissions`.

- 1 Map of world countries from 2008 in format shapefile `cntry08.shp`; the CRS is WGS84 (EPSG:4326); each country is identified by its **ISO 2 letter code**
Source: www.baruch.cuny.edu/geoportal/data;
- 2 FAO tables with emissions of CO₂ equivalent emissions (Gigagrams) caused by all sectors of agricultural activity; Each country is identified by its FAO **short name**; format: `.csv`; Source: faostat3.fao.org;
- 3 FAO table `fao_country_names.csv` that relates countries ISO codes and FAO names; Source: www.fao.org/countryprofiles.

LAB: Relational joins with QGIS (cont.)

- 1 Copy folder `world_agriculture_CO2_emissions` to your PC;
- 2 Open QGIS desktop and add `cntry08.shp` to your project;
- 3 Check the attributes of that shpefile;
- 4 Add the table `fao_country_names.csv` to your project and check its attributes;
- 5 Right-click on the **properties** of the `cntry08` layer and choose the option **Joins** to join the tables of `fao_country_names.csv` and `cntry08.shp` using the *key* attributes `ISO2` and `ISO_2DIGIT`; [OK].

LAB: Relational joins with QGIS (cont.)



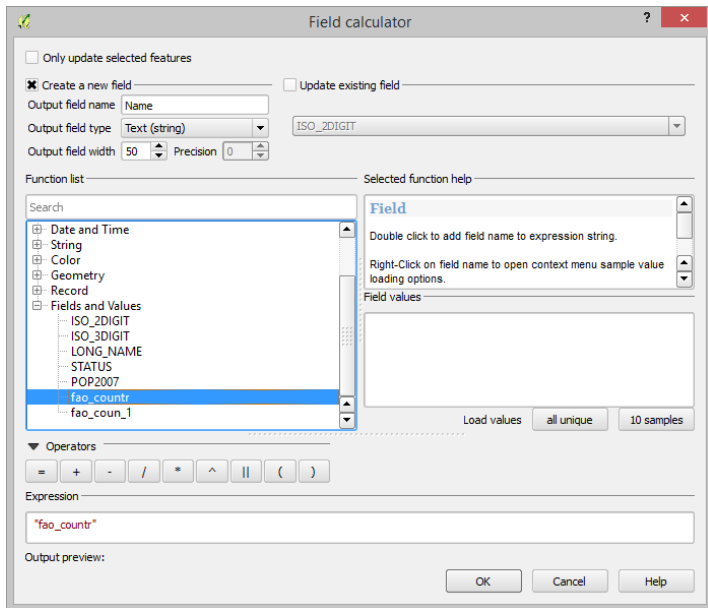
- 6 Open the attribute table of `cntry08` and check that the two new attributes with the short and the official names of the countries were added;
- 7 Save the layer `cntry08` as `cntry08_2.shp` and add it to the project.

LAB: Working with attribute tables ... a little housekeeping

The shapefile format limits the attribute names to 10 characters. As a result, QGIS does some unexpected name conversions when the layer is saved as a shapefile. One can edit the attribute table and create/delete columns to customize it at wish.

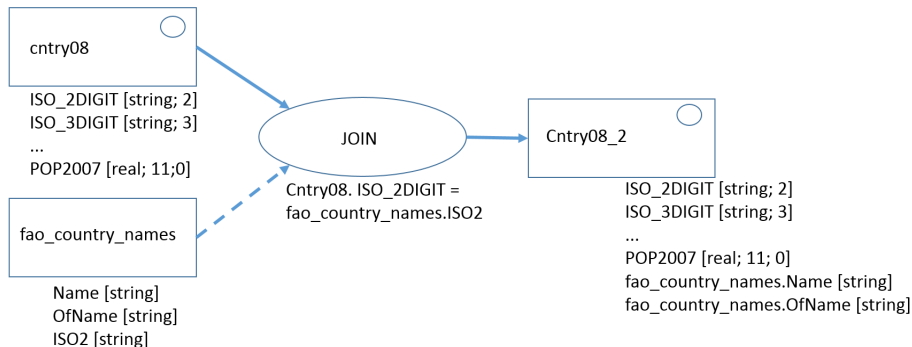
- 1 Check the attributes of `cntry08_2` and verify that the contents of the attribute `Name` from `fao_country_names.csv` is now available although it is called `fao_countr`;
- 2 Toggle the **editing mode on** (the map features become enhanced);
- 3 With the “abacus” button, create a new field called `Name` of **type** Text (string) with **width** 50. Then define its value by the expression `"fao_countr"`;
- 4 Repeat for the attribute `fao_coun_1` which contains countries official names; Call the new attribute `OfName`;
- 5 With the **Delete column** button, you can delete the attributes `fao_countr` and `fao_coun_1` which aren't necessary anymore;
- 6 Toggle the **editing mode off** and save changes.

LAB: adding a new attribute with QGIS



LAB: documenting relational joins

The relational join of the tables from the layers `cntry08` and `fao_country_names` and the creation of the new vector spatial data file `cntry08_2` can be represented by the diagram below. In this diagram, the notation `Cntry08.ISO_2DIGIT` is used to represent the attribute `ISO_2DIGIT` from `Cntry08`.



LAB: Relational joins with QGIS (cont.)

Next, we can join the information about CO₂ equivalent emissions caused by agricultural activity to the world map.

- 1 Right-click on the **properties** of the `cntry08_2` layer and choose the option **Joins** to join tables
`emissions_2009_CO2eq_gigagrams_all_agri_sectors_all`
and
`emissions_2010_CO2eq_gigagrams_all_agri_sectors_all`
In both cases the attribute **key** is `country` and `Name`;
- 2 Save the layer `cntry08_2` as `cntry08_3.shp` and add it to the project;
- 3 Again, QGIS renames the new attributes `emissions_2009` and `emissions_2010` and (this turn, respectively to `emissions_` and `emission_1`).

LAB: Attribute tables in QGIS ... a little more housekeeping

When reading and joining non-spatial data in QGIS, not only the names of the attributes may have to be modified (because of name constraints) but **types** (i.e. the **domains of attributes**) may have to be adjusted.

A typical problem that can occur has to do with *numerical values been read as (text) strings*. This can be fixed using the editor tools already discussed in Slide 170.

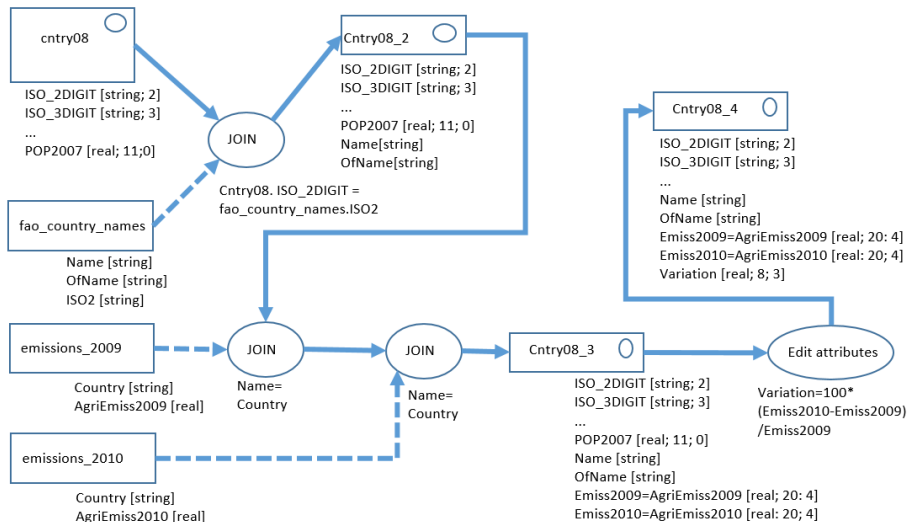
- 1 In the layer `cntry08_3` properties, check the domain of the attributes to check of the attributes that store emission values have a *numeric* type;
- 2 Open the attribute table of `cntry08_3` and toggle the editing mode on;
- 3 With the “abacus” button, create a new field called `Emiss2009` of **type** decimal with **width** 20 and precision (i.e. number of decimal digits) equal to 4; Then define its value by the expression `"emissions_";`

LAB: Relational joins with QGIS (cont.)

Next, we can create a new attribute with the *variation in CO₂ equivalent emissions between 2009 and 2010*. The process is identical, but the **expression** now involves two attributes.

- 4 You can verify that the new attribute is created and the numbers are now left justified;
- 5 If you save the changes and toggle off the editor you can verify in the **Properties/Field** for the layer that the new attribute is of type *double and Real*;
- 6 Repeat the procedure for `Emiss2010`.
- 7 Use again the “abacus” button to create a new attribute `Variation` of **decimal** type, **width 8**, **precision 3**, and set expression to
$$100 * (\text{"Emiss2010"} - \text{"Emiss2009"}) / \text{"Emiss2009"}$$
to compute a new attribute whose value are variations (in %) of emissions over the period 2009–2010; toggle off the editor; [save];
- 8 Now you can draw a map using `Variation` for the legend.

LAB: Putting it all together and documenting with a diagram



LAB: Relational joins with QGIS (cont.)

Notice that QGIS **expressions** can contain a large diversity of functions.

In particular, one could have used the **conversion** function `toreal`.

As a result, the desired `Variation` attribute could have been defined straight from the attributes `emissions_` and `emission_1` as described below.

```
100 * ( toreal( "emissions_" ) -  
toreal( "emission_1" ) ) / toreal( "emission_1"
```

Structuring database tables: Normalization

Definition (Normalization)

The process of removing redundancy from a database.

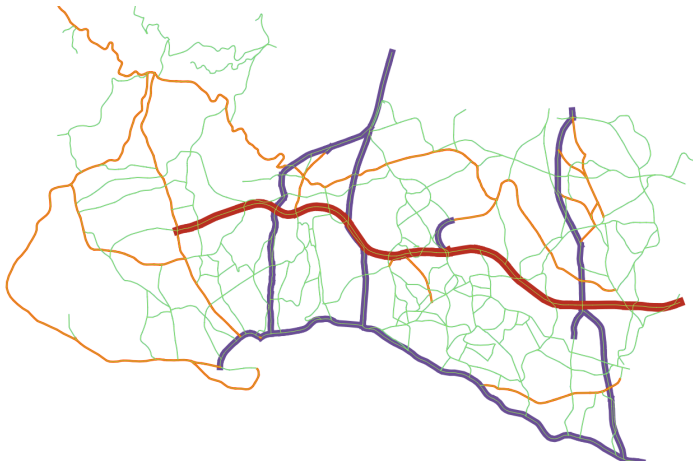
One of the main goals in database management is avoiding data duplication and redundancy. Codd suggested a series of transformations called **normal forms** to reduce redundancy of database tables. In particular, the proposed transformations aim to eliminate *functional dependencies* among non-key attributes in every table.

Those transformations are achieved by splitting tables into subtables that can be joined when required. While this prevents data duplication, it can also lead to a large number of tables and **complex database designs** that are hard to maintain.

Therefore, non-normalized table designs are often used in GIS. The goal is often to find a compromise between avoiding redundancy and limiting the complexity of the database.

Normalization: an example

Consider the problem of representing the following set of roads in a GIS. Each road is of a certain type (path, secondary road, main road, and highway). For each type of road a safety distance is defined.



Normalization: an example (cont.)

On the left is an attribute table for the road theme, with the attribute `safety_dst` (values in meters). However, the safety distance does not depend on the road: it only depends on the type.

A better database structure consists in splitting the table into a table without the attribute `safety_dst` and a new table, with only four rows, where the safety distances are defined. As a result, **tables are joined when needed and the database is easier to maintain.**

	type	safety_dst
1514	Secondary road	25.0
1515	Secondary road	25.0
1516	Highway	100.0
1517	Highway	100.0
1518	Highway	100.0
1519	Highway	100.0
1520	Path	10.0
1521	Path	10.0
1522	Secondary road	25.0
1523	Highway	100.0
1524	Secondary road	25.0

	type
1514	Secondary road
1515	Secondary road
1516	Highway
1517	Highway
1518	Highway
1519	Highway
1520	Path
1521	Path
1522	Secondary road
1523	Highway

type	safety_dst
Path	10.0
Secondary road	25.0
Main road	50.0
Highway	100.0

LAB: Exercises on structuring database tables

Propose spatial and non-spatial tables to structure each one of the following datasets within a GIS:

- 1 Land cover described by a hierarchical legend like Corine Land Cover 1st to 3rd levels (see next slide);
- 2 Location of river and levels of chemical contamination; Each river has a name and each segment of a river has a level of contamination; The levels are codes that range from 0 to 5, and correspond to the descriptions “no contamination”, “low contamination”, ..., “high contamination”; Note that the same river may have segments with different levels of contamination;
- 3 Location and characteristics of a set of trees; Each tree has unique ID (number code), a hierarchical taxonomic classification with two levels (scientific names of species and family), the species common name, height (m), and diameter at breast height (cm); Consider that there are many more trees than species.

LAB: Exercises on structuring database tables (cont.)

Level 1	Level 2	Level 3
1. ARTIFICIAL SURFACES	1.1 Urban fabric	1.1.1 Continuous urban fabric
		1.1.2 Discontinuous urban fabric
	1.2 Industrial, commercial and transport units	1.2.1 Industrial or commercial units
		1.2.2 Road and rail networks and associated land
		1.2.3 Port areas
		1.2.4 Airports
	1.3 Mine, dump and construction sites	1.3.1 Mineral extraction sites
		1.3.2 Dump sites
		1.3.3 Construction sites
	1.4 Artificial, non-agricultural vegetated areas	1.4.1 Green urban areas
		1.4.2 Sport and leisure facilities
2. AGRICULTURAL AREAS	2.1 Arable land	2.1.1 Non-irrigated arable land
		2.1.2 Permanently irrigated land
		2.1.3 Rice fields
	2.2 Permanent crops	2.2.1 Vineyards
		2.2.2 Fruit trees and berry plantations
		2.2.3 Olive groves
	2.3 Pastures	2.3.1 Pastures
	2.4 Heterogeneous agricultural areas	2.4.1 Annual crops associated with permanent crops
		2.4.2 Complex cultivation patterns
		2.4.3 Land principally occupied by agriculture, with significant areas of natural vegetation
		2.4.4 Agro-forestry areas

Figure: Part of the Corine Land Cover hierarchical legend (which has five Level 1 classes in total: 3–Forest; 4–Wetlands; 5–Water Bodies).

LAB: Possible solutions for problems in Slide 181

Ex. 1. One could use a database table associated to a **Polygon** feature class to represent the land cover parcels with a unique land cover class. `ID[integer]` can be an enumeration of the features, and `Code3[integer; 3]` is a 3 digit code like 111, 112, ..., 244, ... (see column “Level 3” in the previous Slide).

```
parcels(ID, Code3)
```

and a non spatial database table to represent the names and levels of the Corine Land Cover hierarchical legend

```
CorineLegend(Code3, NameLevel1, NameLevel2, NameLevel3).
```

The second table is essentially the Corine Land Cover legend, which is well established and stable. The table `CorineLegend` is not normalized since the attribute `NameLevel1` is functionally dependent on the attribute `NameLevel2`. Furthermore, the attribute `NameLevel2` is functionally dependent on the attribute `NameLevel3`. However, normalizing this table would require two extra tables and would make the database more complex with no relevant advantages.

LAB: Possible solutions for problems in Slide 181 (cont.)

Ex. 2. One could use a database table associated to a **Line** feature class to represent river segments, which domains are `ID[integer]` and `ContaminationCode[integer; 1]`

```
RiverSegments(ID, ContaminationCode),
```

and a non spatial database table to represent descriptions of the codes, where the domains are `Code[integer, 1]` and `Description[text string; 20]`:

```
ContaminationLevels(Code, Description)
```

The actual `ContaminationLevels` table is depicted below.

Code	Description
0	no contamination
1	low contamination
...	...
5	high contamination

LAB: Possible solutions for problems in Slide 181 (cont.)

Ex. 3. To design the database correctly, one should notice that scientific name of the **species**, **height** and **diameter** depend on each individual **tree**, while **family** and **common name** depend just on the **species**. With this in mind, we can consider a database table associated to a **Point** feature class to represent individual trees,

```
Trees(ID, SciName, height, DiameterBH),
```

and a non spatial database table to store family and common names

```
Species(SciName, SciFamily, ComName).
```

The actual `Species` table could be something like:

SciName	SciFamily	ComName
Pinus pinaster	Pinaceae	maritime pine
Quercus suber	Fagaceae	cork oak
...

LAB: Exercises on structuring database tables (cont.)

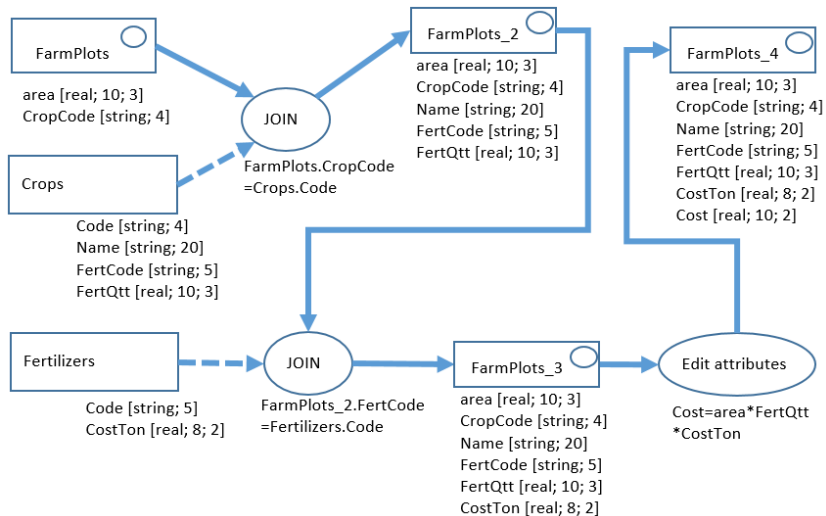
Suppose you want to manage the fertilizer needs and costs for a farm where the number of plots is quite larger than the types of crops. Your GIS project must contain information about:

- 1 The location and area (ha) of agriculture plots;
- 2 The name of the crop in each plot;
- 3 Type and quantity of fertilizer for each kind of crop (in $\text{ton} \times \text{ha}^{-1} \times \text{year}^{-1}$), which is independent of the location of the crop;
- 4 Fertilizer cost (in $\text{€} \times \text{ton}^{-1}$).

Answer to the following questions:

- 1 Propose spatial and non-spatial tables to structure the information you need within a GIS;
- 2 Given the design you have chosen for your GIS database, describe with a diagram the sequence of operations needed to compute the total annual cost of fertilizer for the farm.

LAB: A possible solution for the problem in Slide 186



The attribute `Cost` in the output stores the fertilizer annual cost for each plot. The answer to the problem is simply the sum of values of that attribute, i.e. the sum of the `Cost` column in the attribute table.

The selection operation ... and a bit of Logic

Definition (Selection)

The **selection** operation returns a **subset** of the rows of a database table. If the table is associated to a feature class this is equivalent to a subset of its geographic features.

Example: select the locations where the number of victims is larger than 3 (Slide 38).

In that example, the selection is defined with the expression `Count >3` which has a logical value (**TRUE** or **FALSE**) for each row of the attribute table `Cholera_deaths`. Since this is a **spatial database table**, each row corresponds to a unique geographic feature.

Hence, the **output** of the selection is:

- the **subset of rows** of the table where the attribute `Count` has value larger than 3; **or equivalently**,
- the **subset of locations** that satisfy the condition `Count >3`.

The selection operation ... and a bit of Logic (cont)

Any **logical (also called Boolean) expression**, i.e. which takes values **TRUE** or **FALSE** for each feature, can be used to perform a selection.

The operators **AND**, **OR** and **NOT** can be used to combine Boolean expressions into new Boolean expressions.

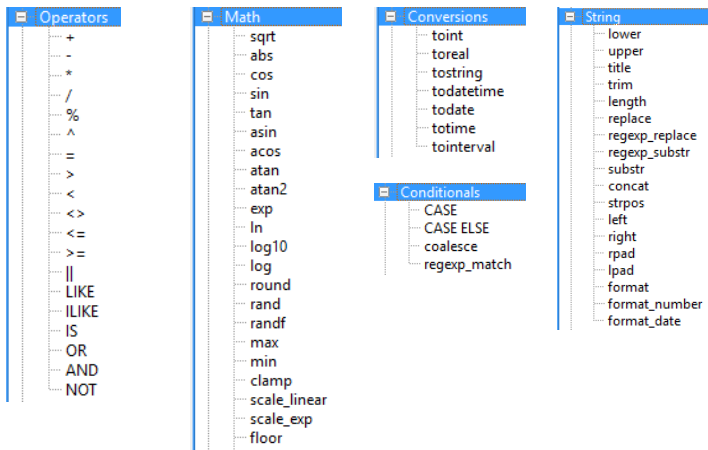
Examples from the spatial database table `cntry08_4` described in Slide 176:

- 1 `Emiss2009 > 100000 AND Variation>50` returns 25 of the 249 features of the feature class;
- 2 `"Emiss2009" < "Emiss2010"` returns 112 of the 249 features of the feature class;
- 3 `left("LONG_NAME",1)='A' AND "POP2007">10000000` returns just two features: the ones which `LONG_NAME` attribute value starts with 'A' and `POP2007` attribute value is larger than 10000000.

	ISO_2DIGIT /	ISO_3DIGIT	LONG_NAME	STATUS	POP2007	Name	OfName	Emiss2009	Emiss2010	Variation
18	AR	ARG	Argentine Republic	UN Member State	40301927	Argentina	the Argentine Republic	105701.0213	101635.4526	-3.846
131	EG	EGY	Arab Republic of Egypt	UN Member State	80264543	Egypt	the Arab Republic of Egypt	26790.9815	27609.5590	3.055

The selection by attributes operation in QGIS:

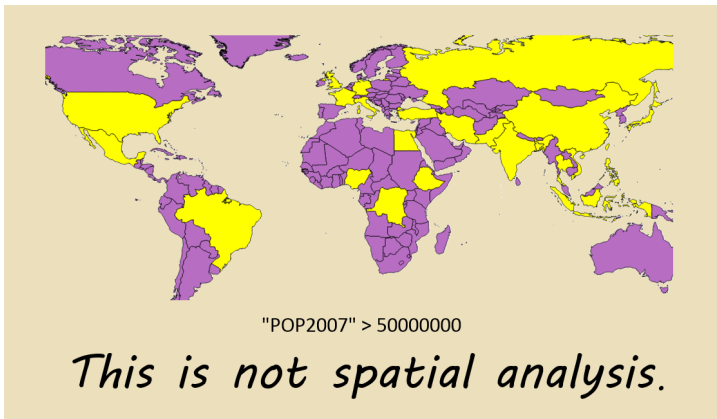
QGIS has a tool to define selection by attributes which is called “Select features using an expression”. A valid expression looks like the ones in the previous slide and can be build with a variety of functions like the following ones.



The image displays a screenshot of the QGIS expression builder's function list, organized into four main categories: Operators, Math, Conversions, and String. Each category is represented by a blue header bar with a list of functions below it.

- Operators**
 - + (addition)
 - (subtraction)
 - *
 - /
 - %
 - ^
 - =
 - >
 - <
 - <>
 - <=
 - >=
 - ||
 - LIKE
 - ILIKE
 - IS
 - OR
 - AND
 - NOT
- Math**
 - sqrt
 - abs
 - cos
 - sin
 - tan
 - asin
 - acos
 - atan
 - atan2
 - exp
 - ln
 - log10
 - log
 - round
 - rand
 - randf
 - max
 - min
 - clamp
 - scale_linear
 - scale_exp
 - floor
 - ceil
- Conversions**
 - toint
 - toreal
 - tostring
 - todatetime
 - todate
 - totime
 - tointerval
- Conditionals**
 - CASE
 - CASE ELSE
 - coalesce
 - regexp_match
- String**
 - lower
 - upper
 - title
 - trim
 - length
 - replace
 - regexp_replace
 - regexp_substr
 - substr
 - concat
 - strpos
 - left
 - right
 - rpadd
 - lpadd
 - format
 - format_number
 - format_date

Selection by attributes: an attribute query



The **selection by attributes** operation *it only depends on the attribute table contents*. It does not depend on the **geometry** and **location** of geographic features. GIS can do more than this: it can do **spatial analysis**.

Spatial Data Analysis

Spatial analysis

Spatial (or geographical) analysis is at the core of GIS. It turns raw data into useful information and reveals new evidence for more effective decision making.

What is particular about spatial analysis which wasn't present in the relational databases we discussed earlier?

It is the fact that **location is essential**.

Definition (Spatial analysis)

Spatial analysis is a set of methods whose results change when the locations of the objects being analyzed change. Longley *et al.* 2010

For instance, the calculation of the variation of emissions from 2009 to 2010 in the problem stated in Slide 167 is not spatial analysis since it does not depend on the location of the geographic features.

Standards for spatial data analysis

The OGC OpenGIS Implementation Standard for Geographic information / ISO 19125 defines:

Geometric objects which can be of type point, line, polygon, multi-point, etc, and are associated to a given Coordinate Reference System (see Slide 195);

Methods on geometric objects return properties like dimension, boundary, area, centroid, etc (see Slide 198);

Methods for testing spatial relations between geometric objects **equals**, **disjoint**, **intersects**, **touches**, **crosses**, **within**, **contains**, **overlaps** and **relate**, which returns **TRUE** or **FALSE** (see Slide 200);

Methods that support spatial analysis **distance**, which returns a **distance**, and **buffer**, **convex hull**, **intersection**, **union**, **difference**, and **symmetric difference**, which returns **new geometric objects** (see Slide 226).

Source: www.opengeospatial.org/standards/sfa

How to describe the location of geographic features: geometric objects

A **geometric object** is a spatial object representing a geometric set with respect to a given CRS.

*A geometric object may be the spatial representation of an object such as a **geographic feature** (i.e. the abstraction of real world phenomena) or a significant part of a feature.*

A collection of geometric objects is called a **geometry**. Therefore a **feature class is spatially represented by a geometry**.

The **geometry class hierarchy** defines the types of geometric objects. It has base subclasses Point, Curve, Surface and GeometryCollection and other subclasses like Line, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, where geometric classes of one type are aggregated from others. For instance, a **linestring** is defined from **points**, or a **multipolygon** is a collection of **polygons**.

How to describe the location of geographic features: geometric objects (cont.)








Geometry Types				
Type		Type		Common Usages
POINT		MULIPOINT		Tree, Pole, Hydrant, Valve
LINESTRING		MULTILINESTRING		Road, River, Railway, Pipeline
POLYGON		MULTIPOLYGON		Cadastre, Park, Administrative Boundary
COLLECTION				Graphics, Markups

Figure: Some examples of geometric objects. Each object has a type which is appropriated to represent some kind of geographic feature.

How to describe the location of geographic features: geometric objects (cont.)

The basic and composed types are described by their coordinates in the CRS: below are represented geometric objects of **dimension 0** (points), **1** (linestrings), and **2** (polygons). A *geometry collection* can contain objects of different dimensions.

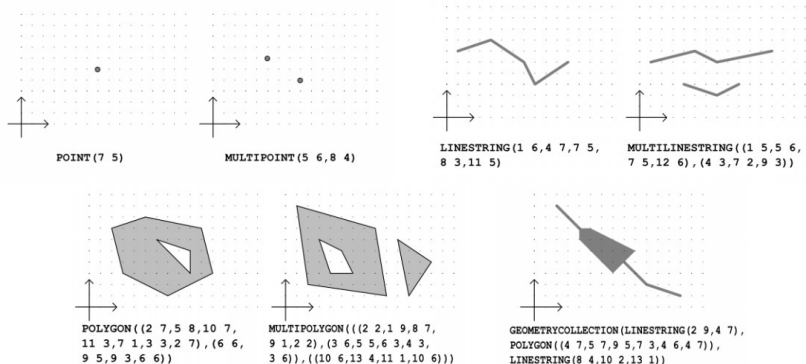


Figure: Illustration of the coordinates associated to several geometric objects.

Some basic properties of geometric objects: dimension, interior, boundary, exterior

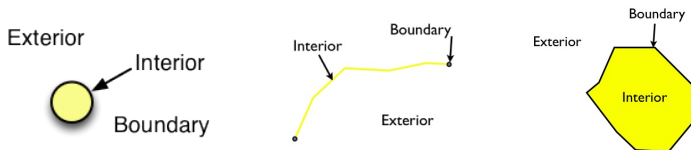
Points and Multipoints have **dimension 0**,

Curves like LineStrings and MultiLineStrings have **dimension 1**,

Surfaces like Polygons and MultiPolygons have **dimension 2**.

Geometric objects have an **Interior**, a **Boundary** and an **Exterior**:

- 1 If P is a Point, then $\mathbf{I}(P) = P$, $\mathbf{B}(P)$ is empty, and the exterior $\mathbf{E}(P)$ are all the points not in P ;
- 2 If C is a curve, then $\mathbf{B}(C)$ are the ends of the curve, $\mathbf{I}(C)$ are all the points of the curve except the ends, and the exterior $\mathbf{E}(C)$ are the remaining points;
- 3 If S is a surface, $\mathbf{B}(S)$ is the boundary, $\mathbf{I}(S)$ is the set of points of S which are not on the boundary and $\mathbf{E}(S)$ are the remaining points.

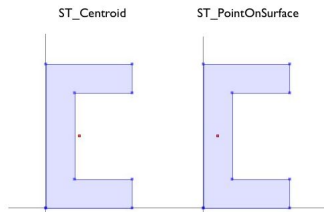


Some basic properties of geometric objects (cont.): area and centroid

Geometric objects of dimension 2 have an **area**, which can be determined with an appropriate function.

The area is measured with respect to the **Coordinate Reference System** of the object and therefore will depend on the map projection and on the distortions associated to the location (see Slide 88).

Among other basic properties, 2-dimensional geometric objects also have a **centroid**, which is the point of the center of mass of the object. However, the centroid might be located outside the geometric object. Hence, the OGC standard also defines the method `PointOnSurface` which returns a point that lies on the object.



Spatial relationships and Boolean spatial operators

Spatial databases are powerful because not only they store geometric objects, but because they also have the ability to test spatial relation between geometric objects.

Questions like “Which are the parks that have water fountains?” or “Does the train 202 crosses B Street?” can only be answered by comparing geometries representing the parks, fountains, subway lines and streets.

The OGC/ISO standard (see Slide 194) defines a set of Boolean operators (i.e. with value **TRUE** or **FALSE**) to test **spatial relationships** between geometric objects.

The operators are **equals**, **disjoint**, **intersects**, **touches**, **crosses**, **within**, **contains**, **overlaps** and **relate**.

Spatial relationships and Boolean spatial operators (cont.)

Given geometric objects a and b of dimension 0 (P), 1 (C) or 2 (S):

- 1 **Equal**: the objects coincide;
- 2 **Disjoint**: the objects do not intersect;
- 3 **Touches** applies to two objects as long as at least one has dimension larger than 0: a Touches b if they intersect at their boundaries;
- 4 For a/b of type P/C, P/S, C/C or C/S, a **Crosses** b if the geometries overlap but a is not within b nor b is within a ;
- 5 a **Within** b if a doesn't intersect the exterior of b ;
- 6 **Overlaps** applies to objects of the same dimension: It is true if a and b intersect but a is not within b nor b is within a ;
- 7 a **Contains** b if b Within a ;
- 8 a **Intersects** b if a and b are not Disjoint;
- 9 **Relates** allows to define any kind of spatial relation between a and b by testing the intersections between the interior, boundary and exterior of both objects.

Spatial relationships and Boolean spatial operators (cont.)

Overlap

Touch

Cross



Multipoint & Linestring



Linestring & Multipolygon



Multipoint & Polygon



Linestring & Multipolygon



Multipoint & Multipoint



Linestring & Linestring



Polygon & Polygon



Point & Linestring



Multipoint & Linestring



Linestring & Polygon



Linestring & Polygon



Point & Polygon



Multipoint & Polygon

Within/Contains

Disjoint

Intersects



Point & Multipoint



Multipoint & Multipoint



Point & Linestring



Multipoint & Linestring



Linestring & Linestring



Linestring & Polygon



Point & Polygon



Multipoint & Polygon



Point & Multipoint



Multipoint & Multipoint



Point & Linestring



Multipoint & Linestring



Linestring & Linestring



Linestring & Polygon



Multipoint & Polygon



Polygon & Polygon



Point & Multipoint



Multipoint & Multipoint



Point & Linestring



Multipoint & Linestring



Linestring & Polygon



Linestring & Polygon



Multipoint & Polygon



Linestring & Multipolygon

Geometries and spatial relationships: further details

The previous spatial relations generalize to **collections of geometric objects**, e.g. the geometric object a intersects geometry B if a intersects some geometric object b in B .

The formal definition of spatial relationships is based on the **Dimensionally Extended 9-Intersection model** DE9IM (Egenhofer, Clementini *et al.*) which uses the 9 intersections one can define between the interior, boundary and exterior of two geometric objects.

The details can be found in the above mentioned source www.opengeospatial.org/standards/sfa (Simple Feature Access - Part 1: Common Architecture).

A description of DE9IM and the derived definitions of spatial relationships are also available at en.wikipedia.org/wiki/DE-9IM.

Select by location: a spatial query

Recall the **selection** operation defined in Slide 188. Selection requires some condition which is **TRUE** or **FALSE** for each row of the database table.

Earlier, we saw how to write a Boolean expression using the **attributes** of the table – this was called a *selection by attributes*.

Select by location is based on Boolean expressions that involves **spatial relationships** (see Slide 200) between geographic features.

Example: In Snow's spatial analysis (see Slide 41) one could ask what are the influence areas around the pumps (Voronoi polygons) where no victims had been found. That query involves the **locations** of the victims and the locations of those polygons. The spatial relationship to use in that case would be **disjoint**.

Select by location: examples

Suppose you have the following database tables and feature classes for a given town:

- 1 A point feature class with the locations of buildings and number of residents, with the following schema:
`buildings(bldID, numRes);`
- 2 A point feature class with the location of subway stations
`subway_st(stID, stName);`
- 3 A line feature class with the location of streets
`streets(streetID, streetName);`
- 4 A polygon feature class of neighborhoods
`neighborhood(borID, borName).`

Write a diagram of GIS operations to answer each question:

- 1 What are the buildings in the neighborhood called “St Peter”?
- 2 What are the buildings in St Peter with more than 100 residents?
- 3 Which neighborhoods can you reach on “Bethlehem St”?
- 4 Which neighborhoods are *crossed* by “Bethlehem St” (i.e. which neighborhoods contain only part of that street) ?

Spatial joins

Earlier we defined and used *relational joins* to relate two database tables using a matching attribute.

GIS also allow us to define **spatial joins**, i.e. a join of attributes tables from two feature classes based on their spatial relationships.

The spatial join is between a **target feature class** and a **join feature class**. Each feature from the target feature class is compared with all features from the join feature class. The join feature for which the spatial relationship is **TRUE** passes its attributes to the target attribute table.

If, for some target feature, there is more than one join feature that satisfies the spatial relationship, then some criteria must be used (e.g. the first attribute value or the average of the attribute values) to collapse those join features' attributes into one attribute value to be passed to the target attribute table.

Spatial join: examples

Consider again the feature classes of Slide 205 .

Write a diagram of GIS operations to answer each of the following questions:

- 1 Create a table that indicates which neighborhood is associated to each subway station.
- 2 Which are the subway stations in St Peter's neighborhood? Provide an answer using a selection by attributes on the table described in the previous question;
- 3 Which are the subway stations in St Peter's neighborhood? This time around, provide an answer using a selection by location.

Joins and selections

In short , we have seen that the selection operations and joins can involve attributes or spatial relationships. The possible combinations are the following.

	join	selection
attributes	relational join	selection by attributes (attribute query)
spatial relationships	spatial join	selection by location (spatial query)
The operation returns:	an augmented attribute table, with new attributes	a subset of table rows, and corresponding geographic features

LAB: spatial joins and spatial queries with QGIS

The data for the following problems are world ecological data on major habitat types (WWF), major rivers and lakes within the world, EOS (Earth Observation System/NASA) validation sites data, world countries and continents. All spatial data geographic coordinates are latitude/longitude (the CRS is WGS84, i.e. EPSG:4326) and the spatial data format is shapefile.

Data are in folder `world_ecological_data`.

- 1 Feature class of world countries from 2008 `cntry08.shp`; The attribute table contains attributes `ISO_2DIGIT[string]`, `ISO_3DIGIT[string]`, `LONG_NAME[string]`, `STATUS[string]` and `POP2007[integer]`; Source:
- 2 Feature class `lakes` with attributes `NAME[string]`, `SUR_ELEV[integer]` and `DEPTH[integer]`; The elevation and depth units are feet (1ft = 30.48cm) and the “no data” value is -9999 for `SUR_ELEV` and -999 for `DEPTH`; Source:

www.baruch.cuny.edu/geoportal/data;

LAB: spatial joins and spatial queries with QGIS (cont.)

- 3 Feature class `rivers`; Attributes `NAME[string]` and `SYSTEM[string]`; Source: www.baruch.cuny.edu/geoportal/data;
- 4 Feature class `continents`; Attribute `CONTINENT[string]` that indicates the name of the continent; Source: www.baruch.cuny.edu/geoportal/data;
- 5 Feature class `ecoregions` which contains information about major WWF terrestrial biomes; Attributes `ECO_NAME[string;100]`, `WWF_MHTNUM[integer; 4]` and `SOURCEDATA[string; 50]`; The attribute `WWF_MHTNUM` is the code of major WWF terrestrial biomes and ranges from 1 to 14 (terrestrial biomes) and 98 (lakes); Source: worldwildlife.org/biomes;
- 6 Non spatial database table `WWF_major_habitat_types_ecological_land.txt`; This table (tab separated values) contains attributes `code[integer;2]`, and `name[string]` with the codes (between 1 to 14) and names of the major terrestrial biomes; Source: worldwildlife.org/biomes;

LAB: spatial joins and spatial queries with QGIS (cont.)

- 7 Non spatial database table `eos.table.txt`; This table (comma separated values) contains attributes `EOSsite[string]`, `latitude[real]`, `longitude[real]`, `precip[real]` (mean annual precipitation in mm), `temp[real]` (mean annual temperature in ° C), `LAI[real]` (average leaf area index), `Biome[string]`; Source: landval.gsfc.nasa.gov/coresite_gen.html.

LAB: spatial joins and spatial queries with QGIS (cont.)

The following problems are intended to review operations of selection and join, expressions in QGIS, and creating a point layer from coordinates. Every solution should be documented with an adequate diagram.

The new operations are available in QGIS in the following menus:

- 1 **Spatial join**: the tool **join attributes by location** in menu **Vector/Data Management Tools**; This tool is limited since it only uses the *intersects* spatial relationship;
- 2 **Spatial query**: in menu **Vector**; Alternatively, you can use the tool **Select by location** in menu **Vector/Research tools** but this last tool only uses *intersects*.
- 3 **List unique values** in menu **Vector/Analysis tools**.

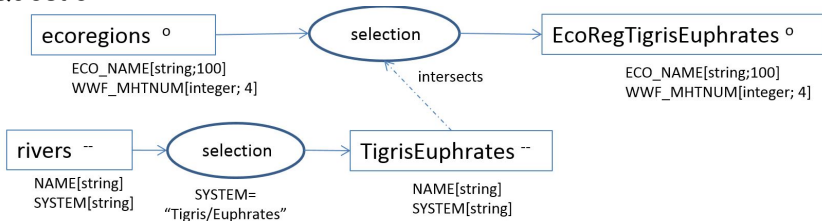
LAB: spatial joins and spatial queries with QGIS (cont.)

Themes: biomes, rivers and lakes:

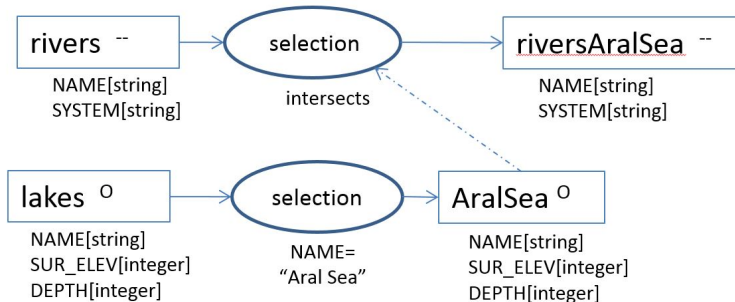
- 1 Determine the WWF terrestrial biomes that are intersected by rivers of the “Tigris/Euphrates” system;
- 2 Determine the rivers that flow into the “Aral Sea”; and the ones that flow into the “Caspian Sea”; What is the problem that occurs for this last question? How could you possibly solve it?
- 3 For each river, determine the lake it connects to, if any; The river “Waterhen” connects to two lakes: why are not both listed in the resulting table?
- 4 For each major river, determine the *mean* surface elevation of the lakes it connects to, if any;
- 5 Determine the lakes that are totally contained in *one* WWF major terrestrial biome. Which is the major biome name associated to each one of those lakes?

Some possible answers for the previous problem

1 Question 1:

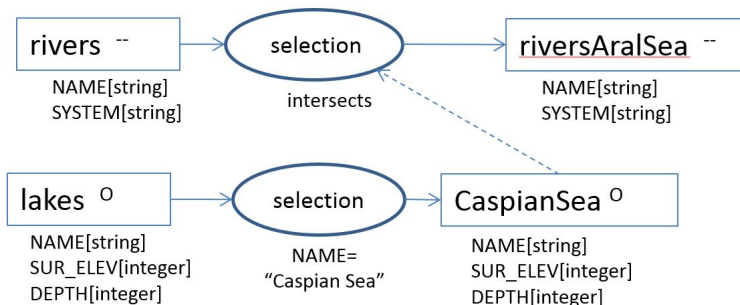


2 Question 2:



Some possible answers for the previous problem (cont.)

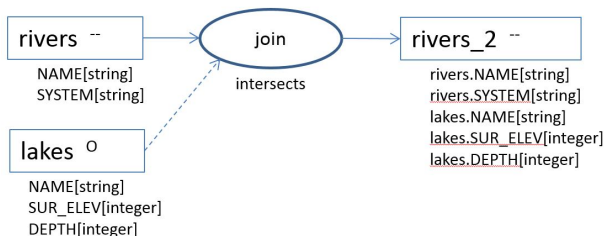
3 Question 2 (cont.):



In the second case, one of the rivers one would expect to select is not selected because the line geometric object associated to that feature doesn't actually intersect the geometric object associated to the "Caspian Sea". Therefore, those two features don't actually "intersect". To solve this, one has to edit the geometric object associated to the river feature so it touches the lake feature.

Some possible answers for the previous problem (cont.)

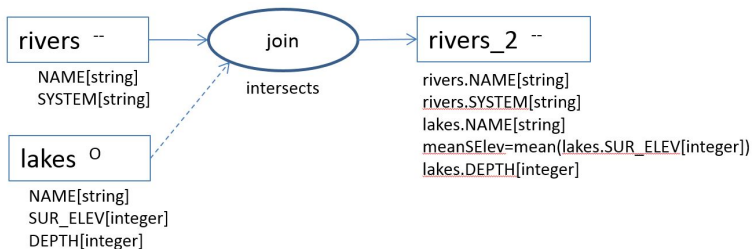
Question 3



When the spatial relationship (intersects) holds for more than one feature of the target feature class only one value is passed to the output attribute table (see Slide 206): in QGIS it is simply the first occurrence in the join feature table.

Some possible answers for the previous problem (cont.)

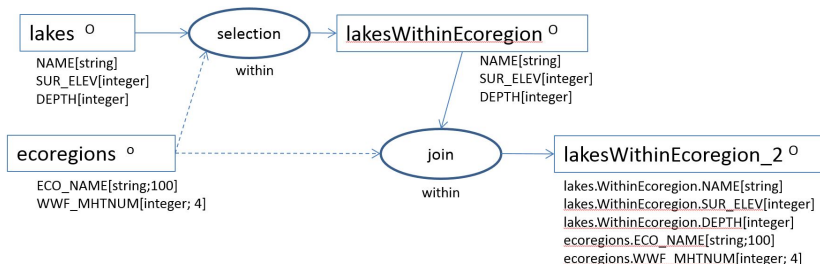
5 Question 4:



In this example, the attribute `SUR_ELEV` of `lakes` is numeric; Therefore, one can apply the function `mean` to the values for all the features that satisfy the spatial relationship “intersects”; In the output, each feature of `rivers` will have associated the resulting mean value (see Slide 206).

Some possible answers for the previous problem (cont.)

6 Question 5



This diagram contains a selection by location to determine a subset of lakes. To add the name of the WWF major terrestrial biome to the output attribute table, a spatial join is performed.

LAB: spatial joins and spatial queries with QGIS (cont.)

Themes: lakes and countries.

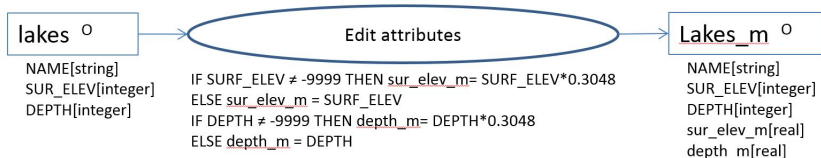
- 1 Convert the attributes `SURF_ELEV` and `DEPTH` from ft units into meters but keep the existing codes (-9999 and -999) to represent “no data”;
- 2 Select the lakes which surface elevation is higher than 1000m; Call the resulting feature class `highLakes`;
- 3 Perform a spatial join (tool **join attributes by location**) to join the attributes from `cntry08` to the attribute table of `highLakes`; call the resulting feature class `highLakesCountries`; The resulting table has 7 rows;
- 4 Perform now the opposite spatial join: join the attributes from `highLakes` to the attribute table of `cntry08`; Use the option **Only keep matching records**; Call the resulting feature class `countriesHighLakes`; The resulting table has 10 rows;
- 5 Compare the previous attribute tables: why do they have a different number of features, since the spatial relationship is the same (*intersects*)?

Some possible answers for the previous problem

- ❶ Question 1: This can be done creating a new attribute (say, `sur_elev_m` of real data type) in **field calculator** with the following expression.

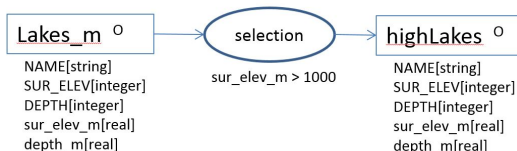
```
CASE
WHEN "SURF_ELEV" != -9999
THEN "SURF_ELEV"*0.3048
ELSE "SURF_ELEV"
END
```

The same approach can be followed for `DEPTH`; The diagram is:

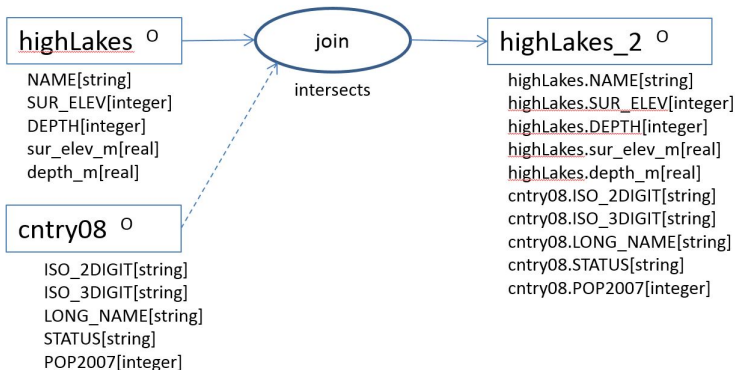


Some possible answers for the previous problem (cont.)

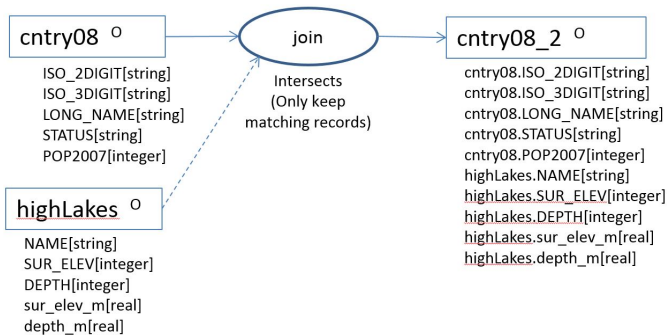
2 Question 2:



3 Question 3:



Some possible answers for the previous problem (cont.)



4 Question 4:

In Question 3 there are 7 lake features which are intersected by 10 different country features. The number of rows of the output attribute table is the number of features of the **target feature class** that spatially matches some feature in the **join feature class**. So, for Question 3, the target feature class is **highLakes** with 7 features while for Question 4 there are 10 target features is **cntry08** that intersect some feature in **highLakes**.

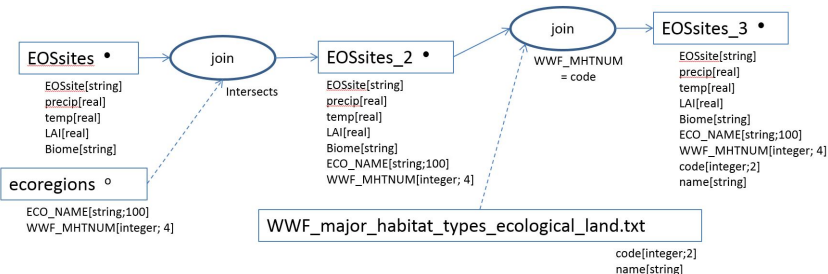
LAB: spatial joins and spatial queries with QGIS (cont.)

Themes: biomes and EOS sites:

- 1 Create a point feature class with the locations and attributes determined by table `eos.table.txt`; This feature class should be associated to the WGS84 CRS;
- 2 Determine the WWF terrestrial biomes associated to each EOS validation site and create a map of the EOS sites over the world labeled with the respective WWF biome code and the leaf area index; Choose a legend that indicates the biome names;
- 3 Which major terrestrial WWF biomes are represented in the set of EOS sites?

Some possible answers for the previous problem

- 1 Question 1: In QGIS a point feature class layer can be created from a text file which contains the coordinates of the points; this is done with the tool **Add delimited text layer**; The output is EOSsites with attributes precip, temp, LAI and Biome;
- 2 Questions 2 and 3



To answer the questions, both WWF biome codes and WWF major terrestrial biome names are necessary; The first *spatial join* associates the biome to each EOS site; the second *relational join* adds the name to the output attribute table.

Spatial analysis: computing distances and new geometric objects

Up to this point, spatial analysis was only used to do queries on the data and to relate (join) information from spatial location.

The OGC/ISO standard (see Slide 194) also defines a set of operators which return the **distance** between geometric objects or return **new geometric objects** from existing ones according to some spatial relation. The standard operators are: **Distance**, **Buffer**, **ConvexHull**, **Intersection**, **Union**, **Difference** and **Symmetric Difference**.

The results of some of these methods, namely **Distance** and **Buffer**, depend on the **distance** between any two points and therefore **vary with the Coordinate Reference System associated to the geometry**.

For instance, the distance measured on a geographic CRS (latitude/longitude) is in general quite different than the distance measured on projected coordinates (See example in Slide 104).

Spacial analysis: computing distances and new geometric objects (cont.)

Given two geometric objects a and b :

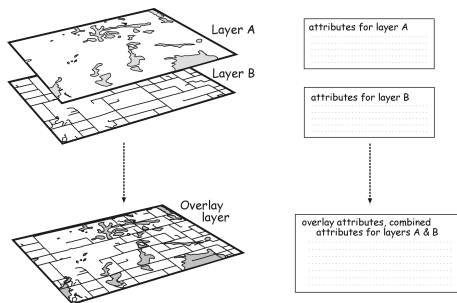
- 1 **Distance** between a and b returns the shortest **distance** between points of a and points of b ;
- 2 **Buffer** of a with distance d returns a geometric object that represents all points whose **distance** to a is less or equal to d ;
- 3 **ConvexHull** of a returns a geometric object that represents the convex hull of all points in a ;
- 4 **Intersection** of a and b returns the geometric object that represents all the points that belong to a and b ;
- 5 **Union** of a and b returns the geometric object that represents all the points that belong either to a or to b ;
- 6 **Difference** between a and b returns the geometric object that represents all the points that belong to a but not to b ;
- 7 **Symmetric Difference** of a and b returns the geometric object that represents all the points that either (1) belong to a but not to b , or (2) belong to b but not to a .

Map overlay

We call **map overlay** to a generic spatial operation in which two or more maps or layers (i.e. **feature classes**) registered to a common coordinate system are superimposed, either digitally or on a transparent material, for the purpose of showing the relationships between features that occupy the same geographic space.

Map overlay involves:

- 1 the spatial analysis of geometric objects (see Slide 226);
- 2 the more general spatial analysis of **collections** of geometric objects;
- 3 the processing of the **attributes tables** of the feature classes.

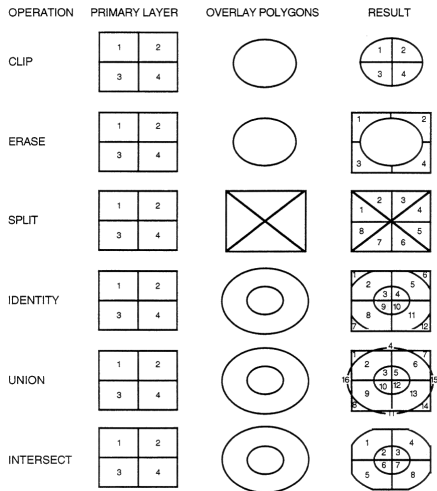


Map overlay (cont.)

The basic methods listed on Slide 226 are then combined to define overlay operations that deal with both the **geometries** and the **attribute tables** of the feature classes.

For instance:

- 1 A **union overlay** combines the geographic features and attribute tables of both inputs into a single new output;
- 2 An **intersect overlay** defines the area where both inputs overlap and retains the attributes for each;
- 3 A **clip overlay** defines the area where both inputs overlap but only retains the attributes of the primary layer.



QGIS: spatial analysis

GIS software (e.g. QGIS, ArcGIS, GeoMedia, etc) allow users to perform a multitude of analysis over feature classes. Each software offers different sets of tools, with diverse parameters and outputs formats. Some fundamental tools like **buffer**, **intersection**, **dissolve** or **clip**, are ubiquitous, although distinct softwares may offer a different set of options for each of those tools.

Next, some fundamental spatial analysis tools available in QGIS are described. The documentation is available at

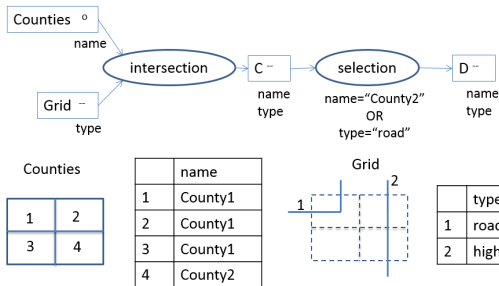
<http://www.qgis.org/en/docs/index.html>.

Distance Matrix tool in menu **Vector/Analysis Tools** is only defined for Point feature classes: it returns a table with three attributes:

- 1 one selected attribute of the first feature class (A);
- 2 one selected attribute of the second feature class (B);
- 3 the distances between all pairs of points (a, b), with a from feature class A and b from feature class B.

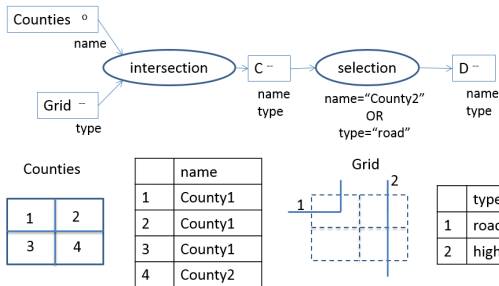
Understanding spatial operations: intersection

Consider the following diagram that represents operations over the input feature classes Counties and Roads (the dotted line indicates the position of Counties). Describe all features and attribute tables of output feature classes C and D.

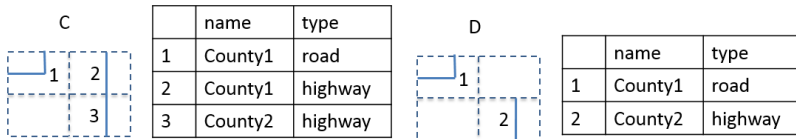


Understanding spatial operations: intersection

Consider the following diagram that represents operations over the input feature classes Counties and Roads (the dotted line indicates the position of Counties). Describe all features and attribute tables of output feature classes C and D.



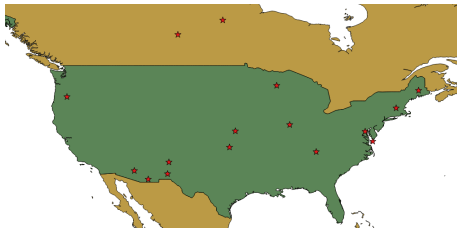
Solution: feature classes and attribute tables of C and D.



QGIS: spatial analysis (Intersect)

Intersect tool in menu **Vector/Geoprocessing tools**: given two feature classes (Point, Line or Polygon, simple or multiple) it returns a new feature class that represents their **intersection**.

In the example, the Point feature class of EOS sites is intersected with the Polygon feature class representing the USA. Only the EOS sites **over the USA polygons** belong to the output. All attributes from both inputs are retained.

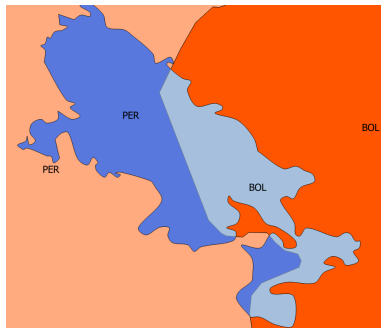


	EOSite	latitude	longitude	precip	temp	LAI	Biome	ECO_NAME	_MH1	SOURCEDATA	_2D1C	_3D1G	LONG_NAME	STATUS	POP2007
0	ARM/CART, SGP	36.64000...	-97.50000...	1361.690...	14.3599...	3.000...	grassland 93%/96%	Central Mixed-Grass Pr...	8	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
1	BARC, USDA ARS	39.03000...	-76.84999...	2202.000...	12.7799...	5.000...	urban/non-vegetate...	Chesapeake Bay Lowla...	4	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
2	Bondville	40.00699...	-88.29099...	914.3999...	10.8300...	4.500...	broadleaf cropland 9...	Central Tallgrass Prairie	8	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
3	Cascades	44.24900...	-122.1800...	2202.000...	8.60000...	10.00...	needleleaf forest 100...	West Cascades	5	M. Goering/Febr...	US	USA	United States of ...	UN Member State	301139947
4	Harvard Forest L...	42.53799...	-72.17100...	1117.000...	8.90000...	6.000...	broadleaf forest 99.1...	Lower New England / ...	4	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
5	Howland	45.20000...	-68.73300...	1040.000...	5.80000...	5.000...	broadleaf forest 93...	Northern Appalachian ...	4	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
6	Jornada LTER	32.60699...	-106.8690...	264.0000...	14.4000...	0.500...	shrubs 100%/79%	Chihuahuan Desert	13	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
7	Konza Prairie LTER	39.08200...	-96.56000...	840.0000...	12.9000...	4.170...	grassland 46%/42%	Osage Plains/Flint Hills ...	8	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
8	Maricopa Agricult...	33.07000...	-111.9699...	190.5000...	19.8599...	8.000...	grassland 45%/19%	Sonoran Desert	13	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
9	SALSA San Pedro	31.73999...	-109.8499...	324.0000...	17.6000...	4.500...	grassland 90%/63%	Apache Highlands	13	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
10	Sevilleta LTER	34.34400...	-106.6710...	242.0000...	13.3000...	1.000...	shrubs 80%/82%	Arizona-New Mexico M...	5	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
11	Walker Branch	35.95799...	-84.28799...	1435.000...	13.9000...	6.000...	broadleaf forest 74...	Cumberlands And Sout...	4	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947
12	Wisconsin: Park F...	45.94599...	-90.27200...	810.0000...	6.60000...	8.400...	broadleaf forest 58...	Superior Mixed Forest	4	Bailey, R, modifie...	US	USA	United States of ...	UN Member State	301139947

QGIS: spatial analysis (**Union**)

Union tool in menu **Vector/Geoprocessing tools**: given two feature classes (Point, Line or Polygon, simple or multiple) it returns a new feature class that represents **features that belong to at least one of the feature classes**.

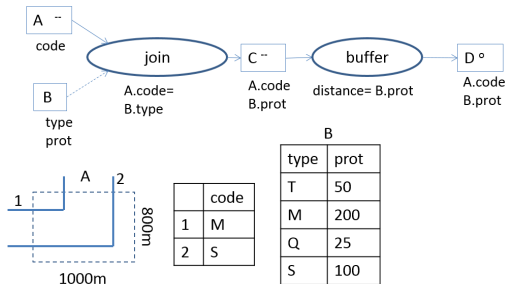
The result of the Union of the feature class `PeruBolivia` and the feature class `LakeTiticaca` contains four features and all attributes from both tables. Only the features that correspond to the intersection have non Null values for the attributes of `LakeTiticaca`.



	ISO_2DIGIT	ISO_3DIGIT	LONG_NAME	STATUS	POP2007	NAME	SURF_ELEV	DEPTH
0	BO	BOL	Republic of Bolivia	UN Member State	9119152	Lake Titicaca	3810	990
1	BO	BOL	Republic of Bolivia	UN Member State	9119152	NULL	NULL	NULL
2	PE	PER	Republic of Peru	UN Member State	28674757	Lake Titicaca	3810	990
3	PE	PER	Republic of Peru	UN Member State	28674757	NULL	NULL	NULL

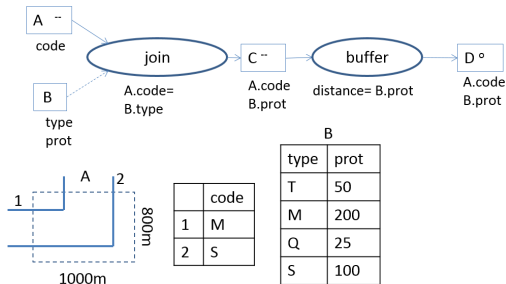
Understanding spatial operations: **buffer**

Consider the following diagram that represents operations over the input feature classes A and B (the dotted line indicates the scale and positions over the CRS). Describe all features and attribute tables of output feature classes C and D.

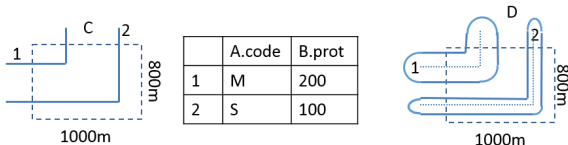


Understanding spatial operations: **buffer**

Consider the following diagram that represents operations over the input feature classes A and B (the dotted line indicates the scale and positions over the CRS). Describe all features and attribute tables of output feature classes C and D.



Solution: feature classes C and D and attribute table of both C and D.



QGIS: spatial analysis (**Buffer**)

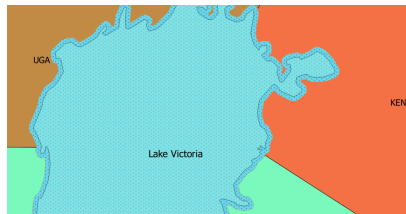
Buffer(s) tool in menu **Vector/Geoprocessing tools**: given a feature class it returns a new feature class. It has parameters:

- 1 **Segments to approximate**: the higher the number, the smoother the boundary of the resulting geometry;
- 2 **Buffer distance**: can be a fixed value or can be defined by an attribute of the input feature class; if the value is negative, QGIS returns an *inwards buffer*;
- 3 **Dissolve buffer results** collapses all output geometric objects, possibly into types MultiPoint, MultiLineString or MultiPolygon;

The attribute table of the output feature class have the same set of attributes as the input:

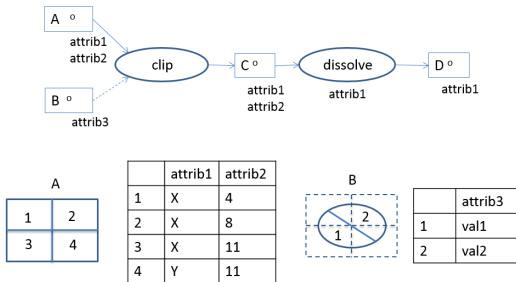
	NAME	SURF_ELEV	DEPTH
0	Lake Victoria	1134	277

- 1 if the **dissolve** option is **off**, the attribute values are preserved;
- 2 if the **dissolve** option is **on**, the attributes of the single row of the output have **NULL** value.



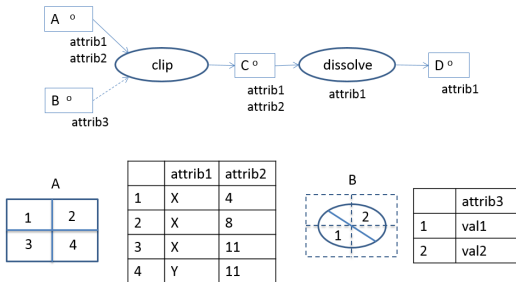
Understanding spatial operations: clip and dissolve

Consider the following diagram that represents operations over the input feature classes A and B (the dotted line indicates the position of B). Describe all features and attribute tables of output feature classes C and D.

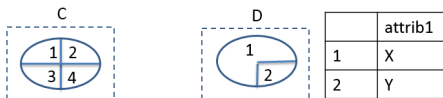


Understanding spatial operations: clip and dissolve

Consider the following diagram that represents operations over the input feature classes A and B (the dotted line indicates the position of B). Describe all features and attribute tables of output feature classes C and D.



Solution: The feature classes and their attribute tables are represented below. The table of C is identical to the table of A.



QGIS: spatial analysis (**Intersect** versus **Clip**)

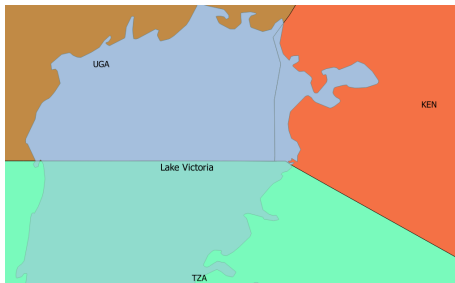
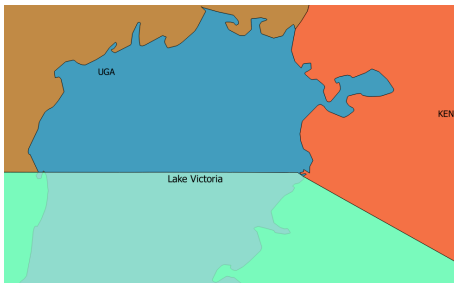
Clip tool in menu **Vector/Geoprocessing tools** returns the same spatial extent than **Intersect**. However, only the attributes of the primary feature class are retained. This means that in general the output of **Clip** contains fewer features than the output of **Intersect**. In the example, the primary layer is LakeVictoria and the other layer is UgandaKenya.

Clip

	NAME	SURF_ELEV	DEPTH
0	Lake Victoria	1134	277

Intersect

	NAME	SURF_ELEV	DEPTH	ISO_2DIGIT	ISO_3DIGIT	LONG_NAME	STATUS
0	Lake Victoria	1134	277	UG	UGA	Republic of Uganda	UN Member
1	Lake Victoria	1134	277	KE	KEN	Republic of Kenya	UN Member



QGIS: spatial analysis (**Dissolve**)

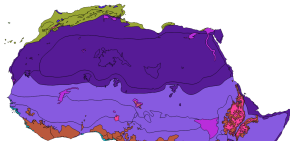
Dissolve tool in menu **Vector/Geoprocessing tools**: The input is a feature class of dimension 1 or 2. The output is a new feature class which is spatially identical but has possibly a lower number of features. The tool has one option **dissolve field**, which can be one of the attributes or **dissolve all**. If one attribute is selected then: (A) all features that have the same value for that attribute are grouped into a new unique feature; (B) the output attribute table has the same attributes but only one value (the first) per group.

Input

	ECO_NAME	WVF_MHTNL	SOURCE DATA
23	Northeastern Congolian Low...	1	Olson, 2001
30	Cape Verde Islands Dry For...	2	Olson, 2001
32	Zambesian Cryptosepalum D...	2	Olson, 2001
31	Madagascar Dry Deciduous ...	2	Olson, 2001
106	Madeira Evergreen Forests	4	Olson, 2001
107	Mediterranean Conifer And ...	5	Olson, 2001
51	St. Helena Scrub And Woodl...	7	Olson, 2001
41	Mandara Plateau Mosaic	7	Olson, 2001

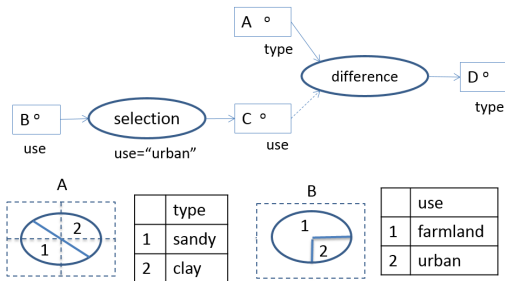
Output

	ECO_NAME	WVF_MHTNUM	SOURCE DATA
4	Albertine Rift Montane Forests	1	Olson, 2001
6	Cape Verde Islands Dry Forests	2	Olson, 2001
8	Madeira Evergreen Forests	4	Olson, 2001
7	Mediterranean Conifer And Mixed Forests	5	Olson, 2001
9	Angolan Miombo Woodlands	7	Olson, 2001
10	East African Halophytics	9	Olson, 2001
0	Angolan Montane Forest-Grassland Mosaic	10	Olson, 2001



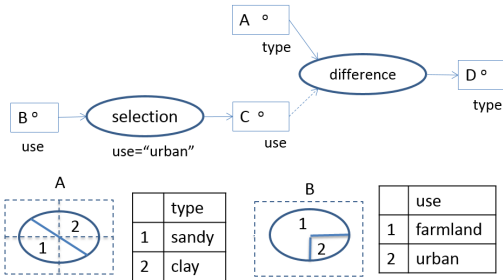
Understanding spatial operations: difference (aka erase)

Consider the following diagram that represents operations over the input feature classes A and B (the dotted lines indicates the relative positions). Describe all features and attribute tables of output feature classes C and D.



Understanding spatial operations: difference (aka erase)

Consider the following diagram that represents operations over the input feature classes A and B (the dotted lines indicates the relative positions). Describe all features and attribute tables of output feature classes C and D.



Solution: feature classes and attribute tables of C and D.

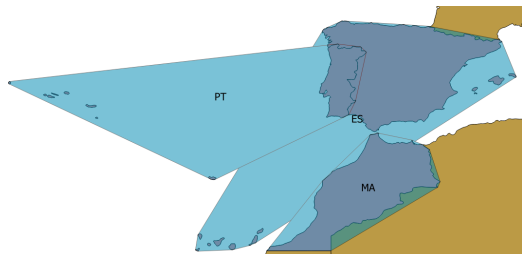


QGIS: spatial analysis (**Convex Hull**)

Convex hull(s) tool in menu **Vector/Geoprocessing tools**: given a feature class (Point, Line or Polygon, simple or multiple) it returns a new feature class. It has parameters:

- 1 **Create single minimum convex hull**: creates a single convex hull for all features; The attribute table has one row, with the attribute values of one of the features;
- 2 **Create convex hull based on input field**: (A) groups the features according to a selected attribute; (B) for each group computes its convex hull; (C) according to the order of the groups, removes the intersections; Each feature corresponds to one row.

In the example, the order of the table rows (Morocco, Portugal, Spain) determines the result of the operation. Here, the attribute `ISO_2digit` is used to group the features.



QGIS: spatial analysis – calculating **area** and **perimeter**

Given a feature class of dimension 2, the **area** and **perimeter** of the geometric objects can be computed with the following tools:

- 1 Tool **Export/add geometry columns** in menu **Vector/Geometry tools**; The area can then be computed over either:
 - 1 the Layer CRS;
 - 2 the Project CRS, or
 - 3 the ellipsoid.

If the CRS of the feature class is geographic (i.e. latitude/longitude coordinates) then the option to choose is **ellipsoid**. If the coordinates are projected then the CRS of the layer should be used, but some errors in the calculation of areas and perimeters can arise due to distortions induced by the projection.

- 2 Using **field calculator** in editing mode; The area is returned by the expression `$area`.

LAB: Spatial analysis with QGIS

The data for this problem is described in Slide 209.

Themes: biomes and continents

- 1 Determine the WWF terrestrial biomes in Africa using the data sets `ecoregions` and `continents`; Name the output `ecoregionsAfrica` (use **Clip** tool);
- 2 Determine the total area per type of major WWF terrestrial biome (attribute `WWF_MHTNUM`) and build a table with the names and total areas of those biomes for Africa (use **Dissolve** tool).

Themes: biomes and rivers

- 1 Again, determine the total area per type of major WWF terrestrial biome in Africa, but just for the regions which are located within 100 km of the major rivers. To project the data, you can use Africa Albers Equal Area Conic (EPSG:102022); to project in QGIS you can **save** the layer **as** a new shapefile with the selected CRS.

Spatial connectivity: Singlepart and Multipart geographic features

There are different types of geometric objects as discussed in Slides 195 through 197. An important distinction is between:

- 1 **Singlepart** objects like Point, LineString or Polygon, which are **spatially connected** objects;
- 2 **Multipart** objects like MultiPoint, MultiLineString or MultiPolygon are aggregates of singlepart objects and can be **spatially disconnected**.

Geometric properties, attributes (e.g. attribute values, area) and spatial relationships are determined with respect to a *feature*, regardless of being singlepart or multipart.

When spatial connectivity is relevant, features should correspond to single spatial units, i.e.,

- 1 Each feature should be itself spatially connected; This can be done with the **Dissolve** tool which remove internal boundaries;
- 2 Features should be spatially disconnected from each other; This can be done with the **Multipart to Singlepart** tool (in QGIS this is available in menu **Vector/Geometry tools**).

LAB: Spatial analysis with QGIS

The data for the following problems are County level data from Cascais, Portugal. All spatial data coordinates from CRS EPSG:102164 (projected coordinates).

Data are in folder `Cascais`.

- 1 Polygon feature class describing the limits of the county `CascaisCountyLimit`;
- 2 Polygon feature class with the description of the land use `LandUse` with attributes `code[string]` and `descriptio[string]`;
- 3 Polygon feature class with the description of the soil type `SoilType` with attribute `COD1[string]`;
- 4 Polygon feature class `Slopes` that indicates the locations with flat and moderate slopes, with attribute `SlopeType[string]` which takes values "up to 5%" and "between 5 and 16%";

LAB: Spatial analysis with QGIS (cont.)

- 5 Line feature class with the description of the water streams
`StreamsNetwork` with attribute `streamType[string]` with values "first" or "other";
- 6 Line feature class with the road network `roads` with attribute `type`; the type can be "Path", "Secondary road", "Main road" or "Highway";
- 7 Non spatial database table `SoilProductivity.csv` with two attributes separated by ";": `code`, which matches the attribute `COD1` in feature class `SoilType`, and `Productivt`, which can take values "low", "medium" or "high".

The goal of the analysis is to find adequate areas in the County of Cascais to build a recreational facility given a set of spatial constraints.

LAB: Spatial analysis with QGIS (cont.)

Solve the following problems and document your steps with a diagram.

- ❶ As seen above, the `roads` feature class contains roads of four different types. Create a non spatial table which indicates a safety distance (in meters) for each type of road. The values should be 10, 25, 50 and 100, for the 4 different types of roads. Suggestion: use the tool **List unique values** in menu **Vector/Analysis tools**.
- ❷ Determine the locations that satisfy the following requirements:
 - ❶ The current land use should be agriculture, bush, empty, or “forest and bush”; and
 - ❷ If the use is agriculture, bush or empty, the slope should be less than 5%; However, if the current use is “forest and bush” the slope can be as high as 16%; and
 - ❸ The soil productivity should be low or medium according to table `SoilProductivity.csv`; and
 - ❹ The locations should be away from roads, at a distance at least equal to the safety distance stated above; and
 - ❺ Spatially connected areas (A) must be intersected by a stream of “first” type, *and* (B) must have at least 30 ha.

LAB: Some hints for the previous problem

- 1 It is better to first do a selection by attributes on `LandUse` before intersecting with `Slopes` to reduce the computing time of the intersection operation;
- 2 Restriction 2: This can be done with a selection based on the following expression.

```
(  
  ("descriptio" = 'agriculture' OR  
  "descriptio" = 'bush' OR  
  "descriptio" = 'empty')  
  AND "SlopeType" = 'up to 5%'  
)  
OR  
( "descriptio" = 'forest and bush' )
```

- 3 Restriction 4: when using the **Buffer** tool, chose the option **dissolve** to speed-up the upcoming computation.

Raster Data Analysis

Raster Spatial Data

Raster data represent the space as a **regular grid**. This is typically the best form of representation of spatial phenomena when the goal is to represent every point in space, i.e. the **space itself** – see Slide 5.

Raster data are typically encountered in:

- 1 Image data from remote sensing;
- 2 Data compilation by grid squares;
- 3 Sampling of a continuous spatial distribution;
- 4 Spatial interpolation of punctual data.

Raster data have an *a priori* **spatial resolution**, which can be problematic since objects on the surface (e.g. a field, a road) may require different levels of resolution. The **boundaries** of raster cells are **too regular** to represent precisely most of geographical features.

On the other hand, **map overlay operations over raster data** tend to be computationally **easier** than map overlay over vector feature classes.

Raster representation

All raster data have essentially the same structure: a **regularly spaced array of data**. Each element of the array is called a **grid cell** and has a numerical value or a no-value (No data, Null, ...).

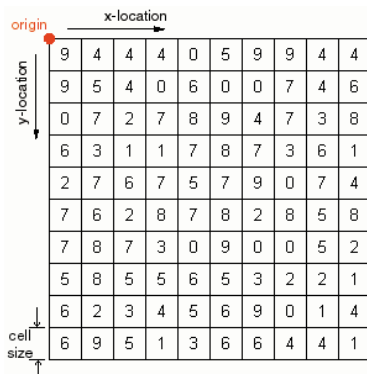


Figure: Illustration of the general structure of raster data. In the simplest case, geographic raster data are georeferenced by the coordinates of a single corner of the array and the cell size (spatial resolution).

Raster versus Vector representation

In image processing in general, vector representation is considered superior to represent **shapes** since scale can be changed without loss of precision and contrast.



RASTER IMAGE

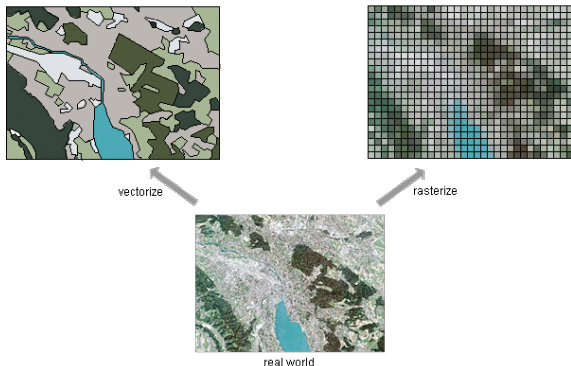


VECTOR IMAGE

However, many images are our days acquired digitally, which means that the **native format** of the information is actually raster.

Raster versus Vector representation of geographic features

In general, vector representation of *objects in space* is more accurate and flexible. Representation into raster format is called **rasterization**; it depends on a fixed spatial resolution, and in principle implies loss of information (e.g. one road cannot be represented precisely unless the spatial resolution is, say, below 10cm). Representation of geographic information into vector format is called **vectorization**; it is as precise as the accuracy of the point coordinates that define geometric objects.



Raster Data Formats

Raster data consists of spatially coherent, digitally stored numerical data, collected from sensors, scanners, or in other ways numerically derived.

Raster data values, as read in from a file, are organized by software into **two dimensional arrays**, the indices of the arrays being used as coordinates (note: there may also be additional indices for multispectral data).

There are **many different raster formats** used by different softwares and raster image producers. Most GIS can read a multitude of raster formats and export raster data in a variety of formats.

Currently, the most popular and versatile raster data format is **Tag Image File Format** (TIFF) , with filename extension `.tiff` or `.tif`. The TIFF file format allows for a flexible set of information fields (tags). It consists of a single file up to 4Gb and it is a **lossless** format.

Raster Geographical Data Models: the GeoTIFF format

The GeoTIFF data format is the TIFF format plus **geolocation tags**. The geolocation is defined as follows.

- 1 **Raster space R** or image space, used to reference the pixel values in an image: it is a regular grid with (0,0) denoting the upper-left corner; The grid has a given number of **columns** and number of **rows**; Each cell value can be thought as associated to an area, i.e. the first cell value is associated to the cell with center (0.5,0.5), or to a point, i.e. the first cell value is associated to (0,0);
- 2 **Model space M**, used to reference points on the Earth: it has a Coordinate Reference System (CRS) which may include vertical coordinates.
- 3 A **transformation between R and M**, which can be defined as one tie point and pixel size (in x and y), a transformation matrix, a list of ground control points (typically 3 corners of the image), or rational polynomial coefficients for orthorectification.

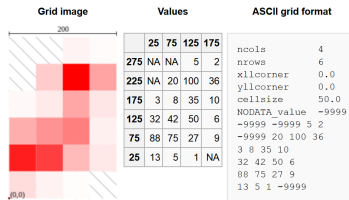
The **resolution** is the size of a cell after transformation. The **data type** of the data can be integer or “float” and of a given size (bits).

For details see <http://trac.osgeo.org/geotiff/>

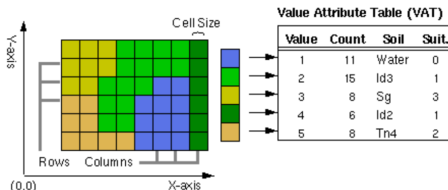
Raster Geographical Data Models: the ESRI grid format

There are two types of ESRI grid formats:

- 1 ASCII grid, which is contained in a single file. Geolocation is present as the coordinates of the upper-left corner of the image and spatial resolution (single cell dimension for both x and y):



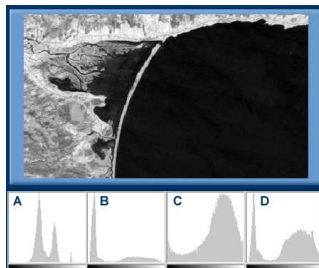
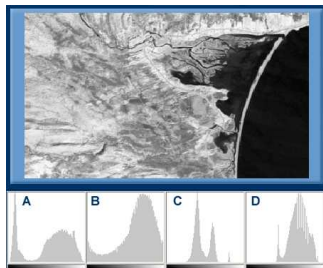
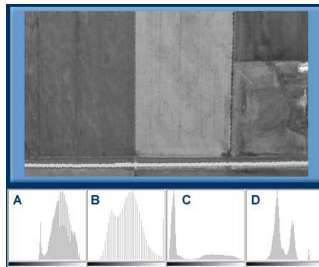
- 2 ESRI grid, which is made up of several files. The data is represented as a VAT table. The CRS can be defined.



Raster data sets: histograms

An **image histogram** is a powerful technique to explore the contents of a raster data set. It represents the frequency distribution (y -axis) of cell values (x -axis).

Example: Choose the right histogram for each image.



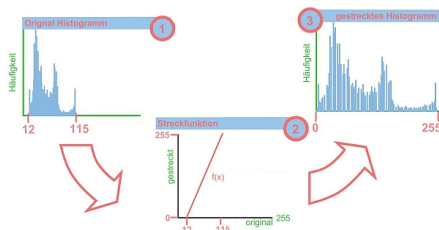
Solution: top D; bottom A and B

Raster data sets: histogram and contrast enhancement

A raster dataset with **grid cell values** $\{z_1, \dots, z_n\}$ has range $[z_{\min}, z_{\max}]$. The j -th grid cell, with value z_j , is **displayed** with a some **intensity** I_j on the screen which ranges from I_{\min} (say, 0) to I_{\max} (say, 1). Usually, I_{\min} is displayed as **dark** and I_{\max} is displayed as **light**.

With no contrast enhancement, the intensity I is a linear function of z over the whole $[z_{\min}, z_{\max}]$; hence, the histogram of the I 's has the same shape as the histogram of the z 's. **Contrast enhancement** is defined by a (increasing, non-linear) function $I = f(z)$.

The simplest case is to select a new range $[z_{\text{new min}}, z_{\text{new max}}]$ and define f continuous such that $f(z) = I_{\min}$, when $z < z_{\text{new min}}$, $f(z) = I_{\max}$, when $z > z_{\text{new max}}$, and f linear in between.



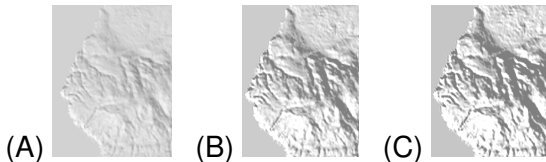
Raster data sets: contrast enhancement in QGIS

GIS usually permit contrast enhancement for displaying raster layers. QGIS (layer **Properties/style**) allows the simple piecewise linear $I = f(z)$ function described in the previous slide.

To define $z_{\text{new min}}$ and $z_{\text{new max}}$ two alternatives are available:

- 1 $z_{\text{new min}}$ and $z_{\text{new max}}$ are **quantiles** q_{min} and q_{max} of the distribution of grid cell values z ; in QGIS this is called **cumulative count cut**;
- 2 $[z_{\text{new min}}, z_{\text{new max}}]$ can be defined as $\bar{z} \pm k \times s_z$ where \bar{z} and s_z are respectively the **mean** and **standard deviation** of values z , and k is selected by the user.

Below, the same raster data set is depicted (A) without contrast enhancement, (B) using the (default) $q_{0.02}$ and $q_{0.98}$ quantiles, and (C) with excessive enhancement (too much saturation) with $q_{0.05}$ and $q_{0.95}$.



Raster data sets: color compositing

The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors.

Spatially matching raster data sets can be displayed as a unique image using the RGB model. This is typically done with **multispectral imagery**. In that case the color composites are known as:

- 1 **True color**, when the RGB image bands correspond to the spectral regions of $0.6\text{--}0.7\mu\text{m}$, $0.5\text{--}0.6\mu\text{m}$ and $0.4\text{--}0.5\mu\text{m}$, respectively;
- 2 **False color**, otherwise.

Before merging bands into a RGB composite, each band may be enhanced as described in Slide 257.

In QGIS this is done in the layer's **Properties/style**.

Basic operations on raster data sets

GIS include many functions that can be applied to a single raster or a set of overlaid rasters. **Map overlay** for raster representations are usually performed with that kind of functions.

Since a raster data layer is essentially a matrix of values, operations on those structures are typically simpler than the computational geometry algorithms used for spatial queries.

The necessary computational resources tends to increase along the following types of operations.

- **local operations** applied to each cell individually, e.g. arithmetic or logical operations, comparisons and *reclassification*;
- **focal operations** that apply to the local neighborhood of each cell, e.g. local filters, local slope and aspect, hill shading (usually for digital elevation models);
- **zonal ou global operations** that depends of the cell values of the whole raster, e.g. minimum value, image segmentation, or least-cost path.

Arithmetic and logical operations on raster data sets

Each cell of a raster data set has a **numeric value**, which can be integer or decimal, also called `float`. Common arithmetic and logical operations on numbers can naturally be extended to **arrays of numbers**.

Geographic raster data sets are essentially geolocated arrays of numbers and therefore operations can be performed cell by cell.

However, if one want to apply those operations to two distinct rasters with possibly distinct spatial resolutions, it is necessary to use some **resampling** technique to match both rasters to a common array.



Figure: Two rasters with distinct alignment and resolution. In Slide 275 a technique is described to align distinct rasters.

Arithmetic and logical operations on raster data sets

Suppose that `input1` and `input2` represent two raster data sets with the same spatial extent.

Comparisons, e.g. `input > 0`, and **Boolean expressions** like

```
(input1 > 0 AND input1 <= 20) OR input2 != 32
```

return a new raster where each cell value is either **TRUE** (which by convention is evaluated as **1**) or **FALSE** (evaluated as **0**).

Arithmetic expressions like

```
1 + exp((input1 - input2)^2)
```

return a new raster of numerical values.

When the grid cell value is **not defined** (e.g. `No data`), the output grid cell value is not defined neither.

Reclassification with arithmetic and logical operations

Reclassification is an operation over a raster data set which goal is to categorize the cell values.

Example: if the input raster data set `input` has values, say, between 0 and 255, the goal can be to apply the following function to cell values.

input	output
0	0
]0,100[1
[100, 150[2
larger or equal than 150	3

The reclassification above is defined with the following expression.

```
0 * (input=0) +  
1 * (input>0 AND input<100) +  
2 * (input>=100 AND input<150) +  
3 * (input>=150)
```

Note: this uses the convention that **TRUE** is **1**, and **FALSE** is **0**.

LAB: Raster analysis with QGIS

The following problems will use SRTM data on elevation and Landsat 8 data for bands 2 (Blue) , 3 (Green), 4 (Red), 5 (NIR), and band 8 (panchromatic). The images are over areas around Lisbon.

Data are in folder `Raster_Lisbon`.

- ❶ Landsat 8 (sensor OLI) data for March 21, 2014: bands 2 to 5 and 8 (panchromatic). The data is in GeoTIFF format and CRS EPSG:32629 – WGS 84 / UTM zone 29N; The files are
 - ❶ LC82040332014078LGN00_B2.TIF to LC82040332014078LGN00_B5.TIF; spatial resolution of 30m;
 - ❷ B8_clip.TIF which is band 8, LC82040332014078LGN00_B8.TIF clipped to a x range from 454527m to 498615m and to a y range from 4249424m and 4313984m; spatial resolution of 15m;
 - ❸ Metadata are in LC82040332014078LGN00_MTL.txt;
- ❷ SRTM elevation data. Data in GeoTIFF format: resolution 3-arc second, and CRS EPSG:4326–WGS 84; Negative values indicate water; File `n38_w010_3arc_v2.tif` (see Slide 80).

Source <http://earthexplorer.usgs.gov/>

LAB: Raster analysis with QGIS – Part 1: Metadata, histograms, and color composites

Select for the Project CRS EPSG:32629–WGS 84/UTM zone 29N.

- 1 Create a new QGIS project with all data sets;
- 2 Check the CRS, resolution and spatial extent of each raster data set;
- 3 Observe the histogram **layer properties/histogram** for one of the Landsat layers; Verify that value 0 indicates non available data for these images;
- 4 Raster data sets can be stacked to create multi band layers; This can be done with the tool **Merge** in menu **Raster/Miscellaneous** with the option **layer stack**; Create a stack with bands 2 to 5 and name it `composite`;
- 5 Notice that the resulting multi band layer is displayed in color; In **Properties/Style** change the display to obtain a real color composite (RGB=432) and a false color composite (RGB=543).

LAB: Raster analysis with QGIS – Part 2: Calculate average NDVI per elevation class

The goal of this problem is to determine the average NDVI (normalized difference vegetation index) for elevation classes 0–25m, 25–50m, 50–100m, 100–200m and higher for the study area.

- 1 Use **Raster calculator** in the **Raster menu** to compute the approximate value of the vegetation index
$$\text{NDVI} = (\text{NIR} - \text{Red}) / (\text{NIR} + \text{Red})$$
 using bands 4 (Red) and 5 (NIR). Name the output **NDVI** and use the following expression.

```
to real(LC82040332014078LGN00_B5@1-LC82040332014078LGN00_B4@1) /  
to real(LC82040332014078LGN00_B5@1+LC82040332014078LGN00_B4@1)
```
- 2 Notice that in **Raster calculator** each individual band of composite is available as `composite@1` to `composite@4`; In alternative to the above, one could have computed

```
to real(composite@4-composite@3)/to real(composite@4+composite@3)
```
- 3 Project `n38_w010_3arc_v2.tif` onto the CRS EPSG:32629 – WGS 84/UTM zone 29N using **Save as** on the layer, name the output `DEM.tif` and add it to the project;

LAB: Raster analysis with QGIS – Part 2: average NDVI (cont.)

- 4 Identify the spatial resolution of `DEM` in **Properties** or in the **Information** tool in menu **Raster/Miscellaneous**; Verify that the resolution in the x direction is around 73m and in the y direction is around 92m (compare with calculations on Slide 84);
- 5 Reclassify `DEM` according to the following function:

all values up to 0	→	0;
]0,25]	→	1,
]25,50]	→	2,
]50,100]	→	3,
]100,200]	→	4
larger or equal to 200	→	5;

Name the result `DEM_6classes`; This can be done with **Raster calculator** in the **Raster** menu using the technique exemplified in Slide 263;

- 6 Convert `DEM_6classes` to a polygon vector layer using tool **Polygonize** in menu **Raster/Conversion**; name the output `DEM_poly_6classes` and the single output attribute `class`;

LAB: Raster analysis with QGIS – Part 2: average NDVI (cont.)

- 7 Select the polygons that satisfy the condition `class>0`;
- 8 Dissolve the result grouping features with the same `class` value; The output should be a Polygon feature class, with 5 multipart polygons; Name it `DEM_dissolve`;
- 9 To conclude the exercise, we want to extend the attribute table of `DEM_dissolve` with the NDVI statistics for each (multipart) polygon; This can be done with the tool **Zonal statistics** in menu **Raster/Zonal statistics**, with inputs `NDVI` and `DEM_dissolve` (suggestion: you can use `ndvi_` as the attributes' prefix);

Discussion of the results: in the resulting attribute table of `DEM_dissolve`, the attribute `ndvi_mean` contains the average of all grid cell values of layer `NDVI` that fall into the corresponding feature (polygon associated to the elevation class of `DEM_dissolve`) . As expected, the NDVI values tend to be larger for higher elevations.

Raster datasets: digital elevation models and terrain analysis

The previous problems illustrated the use of raster data sets to model elevation. Most GIS include operations to derive variables like **slope**, **aspect** and **flow direction** from a digital elevation model in raster format.

The main idea is to estimate for each cell center (x, y) the **plane tangent to the surface** with slope S_x along the x direction and slope S_y along the y direction. Using 8 neighbors, the usual estimates are:

$$S_x = \frac{(z_3 - z_1) + 2(z_6 - z_4) + (z_9 - z_7)}{8 \times \text{resolution along } x}, \quad S_y = \frac{(z_1 - z_7) + 2(z_2 - z_8) + (z_3 - z_9)}{8 \times \text{resolution along } y}.$$



For this example where the resolution is 20m the estimates are:

① $S_x = (1 + 2 \times 3 + 3)/(8 \times 20) = 0.0625$, and

② $S_y = (39 + 2 \times 35 + 37)/(8 \times 20) = 0.9125$.

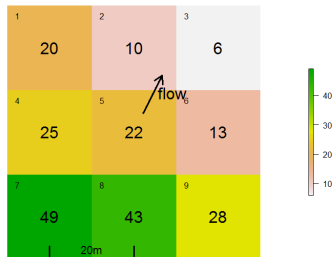
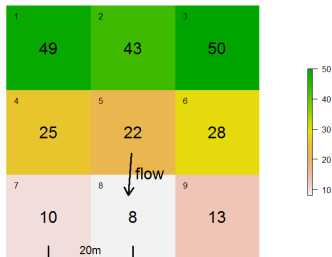
Raster datasets: digital elevation models and terrain analysis (cont.)

The following variables are derived from the estimates S_x and S_y .

- 1 The **flow direction**, i.e. the direction of steepest descent, given by the vector $(-S_x, -S_y)$;
- 2 The **slope** is the slope along the flow direction $S = \sqrt{S_x^2 + S_y^2}$;
- 3 The **aspect** is the flow direction expressed as the **azimuth** (relative to the North, in degrees) or a cardinal (N,E,S,W) or intercardinal (NE, SE, SW, NW) **direction**. The azimuth is $\arctan(S_x/S_y)$ plus 0° (1st Quadrant), 180° (2nd or 3rd Quadrants), or 360° (4th Quadrant). If $S_x = S_y = 0$ the aspect is undefined (flat terrain).

azimuth	direction	azimuth	direction
337.5°– 22.5°	N	22.5°– 67.5°	NE
67.5°– 112.5°	E	112.5°– 157.5°	SE
157.5°– 202.5°	S	202.5°– 247.5°	SW
247.5°– 292.5°	W	292.5°– 337.5°	NW

Raster datasets: digital elevation models and terrain analysis (cont.)



In this example (see Slide 269),

- ① $S_x=0.0625$, $S_y=0.9125$,
- ② $S=\sqrt{0.0625^2+0.9125^2}=0.9146379$,
- ③ Since $\arctan(S_x/S_y)=3.918^\circ$, the aspect is 183.918° , i.e. S.

In this example,

- ① $S_x=-0.36875$, $S_y=-0.73125$,
- ② $S=\sqrt{0.36875^2+0.73125^2}=0.8189647$,
- ③ Since $\arctan(S_x/S_y)=26.76^\circ$, the aspect is 26.76° , i.e. NE.

Raster datasets: digital elevation models and terrain analysis (cont.)

In addition to the previous variables, it is possible to derive **hillshading** and **visibility** from a DEM.

Hillshading is proportional to the estimated cosine of the angle of incidence of light over the surface (θ_i). This depends of the position of the light source (typically the Sun) which is determined by the Solar Zenith angle (θ_{sun}) and the Solar Azimuth angle (ϕ_{sun}). Given the aspect (ϕ_{aspect}) and the slope (θ_{slope}) in degrees,

$$\cos \theta_i = \cos \theta_{\text{sun}} \cos \theta_{\text{slope}} + \sin \theta_{\text{sun}} \sin \theta_{\text{slope}} \cos(\phi_{\text{sun}} - \phi_{\text{aspect}}).$$

The maximum value of hillshading corresponds to $\theta_i = 0$, i.e. when the Sun radiation is normal to surface at the location.

Visibility is a binary variable that indicates if a position (grid cell) is or not visible from a location with coordinates (x, y, Z) . Usually the location is indicated as a position (x, y) and an elevation above ground h , with $Z = h + z(x, y)$, where $z(x, y)$ is the elevation given by the DEM for the grid cell that contains (x, y) .

LAB: Raster analysis with QGIS – Part 3

In the previous problem (Part 2, Slide 266), Landsat data and elevation data have different CRS, resolution and extent. In particular, the grid cells are not aligned. This was not problematic since there was no need to perform arithmetic operations involving both data sets.

Next we consider an hypothetical problem where the **alignment of the grids is relevant**. Suppose one wants to compute a **risk index for landslide** that depends on the amount of vegetation (measured by the NDVI) and the slope at the surface.

The risk index is computed as the **slope (measured as a ratio) times the NDVI factor**. This factor is 1 if NDVI larger than 0.4, 2 if NDVI is between 0.25 and 0.4, 3 if it is between 0.1 and 0.25 and finally 5 if NDVI is lower than 0.1. This calculation is to be made in an area where both Landsat bands and elevation data are available.

The risk index is meaningless over **water**, where `n38_w010_3arc_v2` is negative. It is also meaningless over **urban areas** and where **clouds** contaminate the Landsat images. Those areas can be identified from Landsat band 8 (panchromatic).

LAB: Raster analysis with QGIS – Part 3.1: re-projecting and resampling

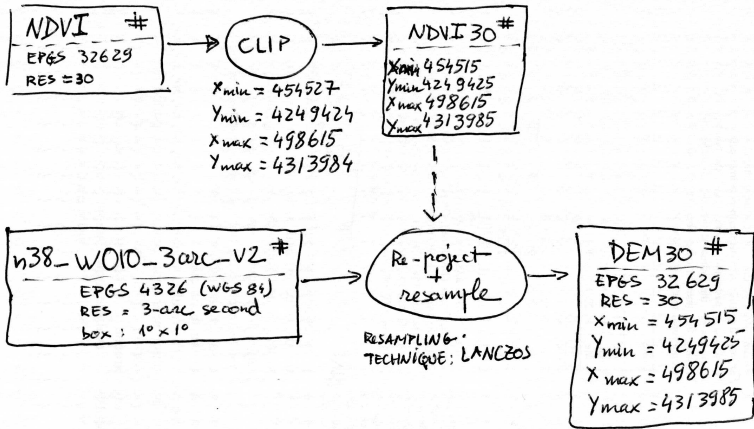
- 1 The first step is to consider an area where all data is available;
Let's consider the rectangle in CRS EPSG:32629
WGS 84/UTM zone 29N where x ranges from 454527m to 498615m and y ranges from 4249424m and 4313984m;
- 2 To clip NDVI (see Slide 266) to that region we can use the tool **Clipper** in menu **Raster/Extraction** using the set of coordinates above; Name the output NDVI30;
- 3 Compare NDVI30 and NDVI; the pixels should be perfectly aligned; Check that the spatial extent of NDVI30 is $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ equal to
454515 4249425 498615 4313985;
- 4 The next goal is to clip and resample the elevation data so it's aligned with NDVI30; Actually, instead of using DEM from Slide 266, it's better to re-project the original elevation data set n38_w010_3arc_v2.tif, adding some parameters to change the resolution and spatial extent as well;

LAB: Raster analysis with QGIS – Part 3.1: re-projecting and resampling (cont.)

- 5 Open the tool **Warp** in menu **Raster/Projections**; the input is `n38_w010_3arc_v2.tif`; Indicate the output name `DEM30` and the target CRS `EPSG:32629 - WGS 84/UTM zone 29N`; Select the resampling method as `Lanczos` and No data values as `0`; **do not [OK] yet!**;
- 6 The tricky part is to align the raster cells with `NDVI30`; QGIS **Warp** tool by itself is too restrictive to do this, but one can add options to the **gdal** function `gdalwarp` directly; Click the pencil symbol on the lower box of the **Warp** tool to start editing and add after `gdalwarp`
`-te 454515 4249425 498615 4313985`
`-tr 30 30`
which define the extent and resolution of the raster to be created. Then [OK];
- 7 Verify that `DEM30` is now aligned with `NDVI30` as expected.

LAB: Raster analysis with QGIS – Part 3.1: re-projecting and resampling (cont.)

PART 3.1.



LAB: Raster analysis with QGIS – Part 3.2: Derive Slope

- 8 In fact, the new digital elevation model looks more precise than the original one; This is because the resampling above leads to a more detailed elevation surface since it uses a smooth **spatial interpolation** technique; for a better rendering of DEM30 use tool **Relief** in menu **Raster/Terrain Analysis**; call the result Relief30; apply the same tool to DEM and compare the results;
- 9 From DEM30 compute the slope with tool **Slope** also in menu **Raster/Terrain Analysis**, which outputs a grid of slopes **measured in degrees**; Name the output SlopeDeg30;
- 10 Convert SlopeDeg30 to slopes in ratio, using the relation $\text{slope}_{\text{ratio}} = \tan(\text{slope}_{\text{angle}})$; first you need to convert degrees into radians before you apply the function tan; name the output SlopeRatio30;
- 11 Use DEM30 to exclude grid cells over the water and combine this with SlopeRatio30 to obtain the slopes as ratios over land; the expression to use is `SlopeRatio30 * (DEM30>=0)`; name the output SlopeRatioLand30;

LAB: Raster analysis with QGIS – Part 3.3: Compute risk index

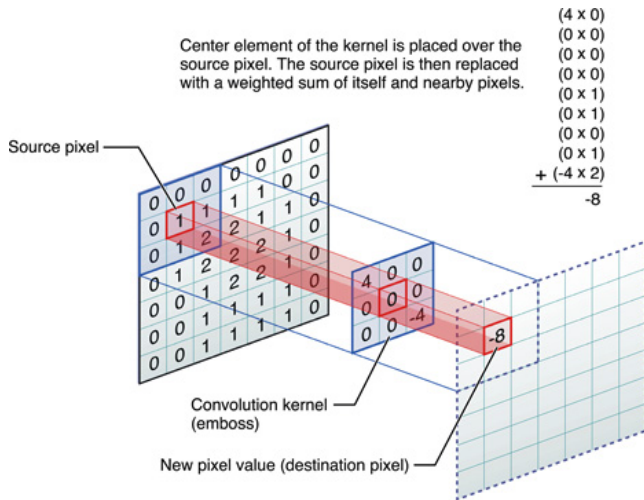
- 12 NDVI30 needs to be reclassified according to Slide 273; this can be done with the technique described in Slide 263; name the output `NDVI30factor`;
- 13 Finally, the risk index can be computed as
`NDVI30factor * SlopeRatioLand30`.

Notes:

- 1 GDAL stands for Geospatial Data Abstraction Library; The options for `gdalwarp` are available at
<http://www.gdal.org/gdalwarp.html>
- 2 A overview of raster tools in QGIS can be found in
http://www.qgis.org/en/docs/user_manual/plugins/plugins_gdaltools.html.
- 3 Spatial interpolation techniques will be the topic of a forthcoming lecture.

Raster datasets: image filters

A **linear filter** is a standard *focal* image processing technique: a linear filter replaces each grid cell value by a weighted sum of its neighbors. The **kernel** defines the neighborhood and the weights.



Raster datasets: linear filters

Linear filters can be designed to address problems in image processing like the following.

- 1 Image smoothing;
- 2 Computation of image derivatives and gradients;
- 3 Approximate (discretize) continuous filters (e.g. Gaussian filters);
- 4 Template matching, i.e. determine parts of the image that are similar to a given template.

Exercise. In Slide 269 the images S_x and S_y were derived from a DEM. Show that S_x and S_y can be defined from the DEM using linear filters.

Solution: S_x and S_y can be defined with the following 3×3 linear filters:

$$S_x : \begin{bmatrix} -\frac{1}{8 \times 20} & 0 & \frac{1}{8 \times 20} \\ -\frac{2}{8 \times 20} & 0 & \frac{2}{8 \times 20} \\ -\frac{1}{8 \times 20} & 0 & \frac{1}{8 \times 20} \end{bmatrix} \quad S_y : \begin{bmatrix} \frac{1}{8 \times 20} & \frac{2}{8 \times 20} & \frac{1}{8 \times 20} \\ 0 & 0 & 0 \\ -\frac{1}{8 \times 20} & -\frac{2}{8 \times 20} & -\frac{1}{8 \times 20} \end{bmatrix}$$

Raster datasets: image filters (cont.)

Filters can be used for many other operations on rasters. When an image exhibits “salt-and-pepper noise”, different kinds of filters can be used to reduce that noise. Below, the image on the right is “filtered”.



In particular the following **non-linear filters** are used to that end:

- 1 the **median** filter, which replaces the central grid cell value by the median of its neighbors;
- 2 the **majority** filter, which replaces the central grid cell value by the most frequent value among its neighbors.

LAB: Raster analysis with QGIS – Part 3.4: Exclude “bright” areas

In the description of the problem (see Slide 273) we noted that **urban areas** and **clouds** should be removed from the analysis. To attempt to do this with the available information, we will use the panchromatic Landsat band to **identify “bright” pixels**, since typically urban areas and clouds have high reflectance in the visible range of wavelengths.

Before the selection of bright areas, it is convenient to perform an **image correction for illumination**. In fact, less illuminated areas on the image have a lower signal regardless of the land cover. A simple approach to illumination correction consists in dividing the signal by $\cos \theta_i$ (where θ_i is the incident solar angle, as in Slide 272). However, this correction is known to over correct the signal where slope is high. A better correction is the **Minnaert correction** below.

$$S_{\text{cor}} = S_{\text{raw}} \frac{\cos \theta_{\text{slope}}}{(\cos \theta_{\text{slope}} \cos \theta_i)^k} \quad (1)$$

where S_{raw} and S_{cor} are the original and corrected signals and k is the Minnaert constant which depends on the slope θ_{slope} .

LAB: Raster analysis with QGIS – Part 3.4: “bright” areas (cont.)

To apply Equation 1 to our data we need:

- 1 A DEM that matches `B8_clip`; Proceed as in Slide 276 but use a resolution of 15m instead of 30m; Name the output `DEM15`;
- 2 Reclassify with "`DEM15@1`" > 0 and convert 0 to No data with tool **Raster / Conversion / Translate** using the option **No data** to create a mask `Land_mask`;
- 3 Multiply `land_mask` by `DEM15` to get a new DEM where the water grid cells have No data value; Name it `DEM15land`;
- 4 θ_{slope} , which is returned by tool **Slope** in menu **Raster/terrain Analysis** (value in degrees) from `DEM15land`; Name the output `Slope15`;
- 5 $\cos \theta_i$; The tool **Hillshade**, in the same menu, derives from `DEM15land` a layer with values $255 \times \cos \theta_i$; We need to use the sun elevation and azimuth at the time the panchromatic image was acquired; This information is available in the metadata file and is 46° for sun elevation and 147° for the solar azimuth; Name the output `Hillshade15`;

LAB: Raster analysis with QGIS – Part 3.4: “bright” areas (cont.)

- ⑥ The Minnaert constant k depends on the slope as discussed in Lu *et al. Photogrammetric Engineering & Remote Sensing* 74, (2008); The value of k can be defined by

$1.121523 - 0.07263274 * \text{"Slope15@1"} +$
 $0.002189456 * \text{"Slope15@1"}^2 - 0.00002057283 * \text{"Slope15@1"}^3$; According to this relation k decreases from slope=0° (where $k = 1.12$) to high slope values (where k approaches 0); Name the output K15;

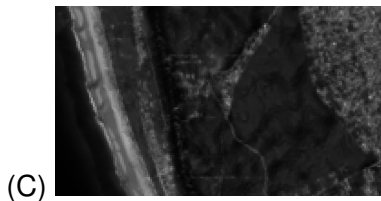
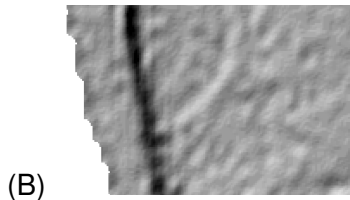
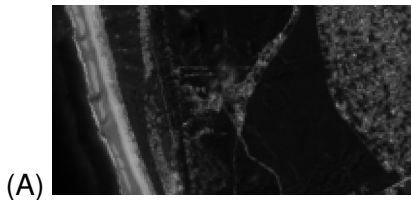
- ⑦ In our case the *raw* signal is Landsat band 8 (layer B8_clip); Apply Equation 1 to B8_clip, Slope15, Hillshade15 and K15; In raster calculator, the expression is:

$\text{"band8_clip@1"} * \cos(3.1416 * \text{"Slope15@1"} / 180.0) /$
 $(\cos(3.1416 * \text{"Slope15@1"} / 180.0) * \text{"Hillshade15@1"} / 255.0)^{\text{"K15@1"}};$

Name the output B8cor.

LAB: Raster analysis with QGIS – Part 3.4: “bright” areas (cont.)

Below, (A) a clip of the raw image, (B) $\cos \theta_i$ indicating the amount of illumination, and (C) the corrected image. The less illuminated area (dark stripe in B) is actually very steep so k is low and prevents an overcorrection.



LAB: Raster analysis with QGIS – Part 3.4: “bright” areas (cont.)

At this point we have a corrected version of the panchromatic band. Next, “extract” the “non-bright” pixels and vectorize the result.

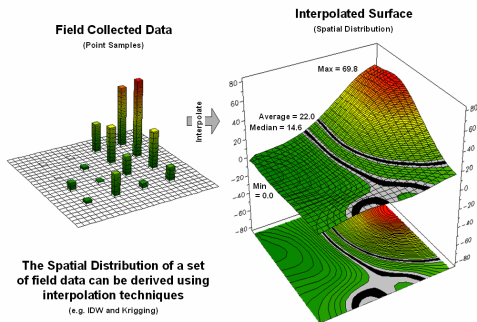
- 8 Explore the layer `B8cor` to select a cut-off point to discriminate “bright” and “non-bright” areas (use 11000 as the threshold) ;
- 9 Reclassify the corrected panchromatic image using that cut-off value to obtain the “non-bright” areas; Name the output `nonBright15` which should have values `No data` (water), 0 (for bright areas) and 1 (for non-bright areas);
- 10 Aggregate grid cells of `nonBright15` using a 9×9 **majority filter** with a 50% threshold; You can do this with tool **Majority filter** available in the **Processing toolbox** that can be open from the **Processing** menu;
- 11 Vectorize with tool **Raster / Conversion / Polygonize** and select just the polygons that correspond to “non-bright” areas;
- 12 You can check that polygons are shaped as grid cell boundaries; to attenuate this effect use tool **Simplify** in menu **Vector / Geometry tools / Simplify geometries** with a tolerance of 20m.

Spatial Interpolation

Spatial Interpolation

Definition (Spatial interpolation)

It is the interpolation of measurements at point locations to the whole space. More precisely, given a set of measurements $\{z_1, \dots, z_n\}$ at spatial locations $\{(x_1, y_1), \dots, (x_n, y_n)\}$, spatial interpolation determines a function $z = f(x, y)$ that interpolates the z_i measurements to all locations within the range of (x_i, y_i) locations.

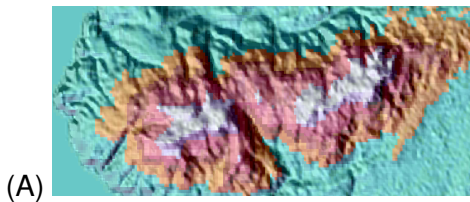


The *point samples* can be elevation, temperature, precipitation or other continuous variables.

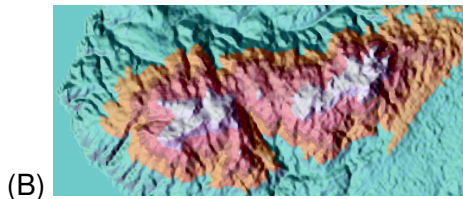
Figure: www.innovativegis.com/

Spatial Interpolation (cont.)

In Slide 274 a first example of spatial interpolation was discussed. GDAL **warp** tool was used to re-project SRTM elevation data (see Slide 80) at a native resolution of 3-arc degrees (figure A) to a resolution of 30m (figure B). In the process a *smooth interpolation* (**Lanczos**) was applied to the original data set.



DEM at the native 3-arc degrees original resolution, i.e., approximately 90m between elevation measurements.



DEM obtained after resampling A to a grid with a spatial resolution of 30 m. The resampling is *smooth* and preserves the original elevation measurements.

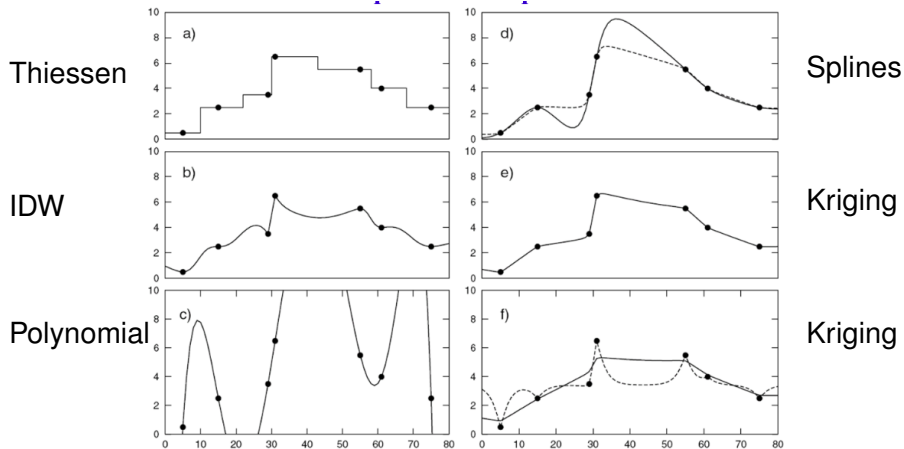
Spatial interpolation methods

Different interpolation methods yield data of varying quality and can strongly influence modeling results.

- **linear**, e.g. interpolation over triangular irregular networks (TIN);
- **polynomial**;
- **splines**;
- **kernel**, e.g. Lanczos interpolation;
- **Thiessen** or **nearest neighbor**;
- **inverse distance weighting** (IDW), a generalization of nearest neighbor interpolation;
- **kriging** and related geostatistics methods;
- ...

Comparison of interpolation techniques

The figure below illustrates a set of distinct interpolation techniques in one dimension. The dots represent discrete measurements and the graphs represents the interpolated values. Most techniques illustrated below **preserve the observation at the original locations** but lead to different interpolated values for locations in between.



Some interpolation techniques for spatial data sets

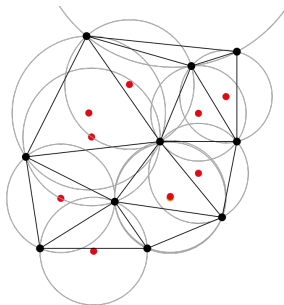
We will discuss the following techniques (which are available in QGIS).

- 1 **Nearest neighbor**: this technique of interpolation associates to each point (x, y) the measurement z_i of the closest point (x_i, y_i) ; Spatially, this corresponds to attribute the value z_i to all locations with the **Voronoi polygon** (see Slide 39) of (x_i, y_i) ; The major drawback of this approach is that it leads to a **discontinuous** spatial interpolation;
- 2 **Linear interpolation** over an irregular triangular network; The assumptions of are simple like for the previous technique; the interpolation is continuous but not smooth (see Slide 293);
- 3 **Moving window**; This large family of techniques uses a moving window (or kernel) over the area to compute the convolution of the kernel with the available measurements; The shape and the distribution of weights of the kernel define the output: if the kernel is uniform the technique is called **moving average**; if the kernel depends on the inverse of the distance to its center the technique is called **inverse distance weighting** (see Slide 299).

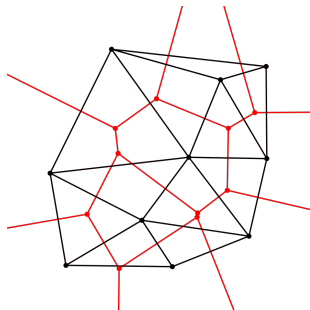
Linear interpolation and triangular irregular networks (TIN)

Given a set of (x_i, y_i) points (**black dots** in figure A), the **Delaunay triangulation** fills their convex hull with triangles. Each triangle is defined by three points such that their circumcircle doesn't contain a fourth point. The triangles' circumcenters (**red dots**) define the **Voronoi polygons** of the set of points (red **polygons** in B).

For each triangle with vertices with u, v, w indices, the **linear interpolation** is given by the linear function $z = a + b x + c y$ such that $z_i = a + b x_i + c y_i$ for all $i = u, v, w$.



(A)



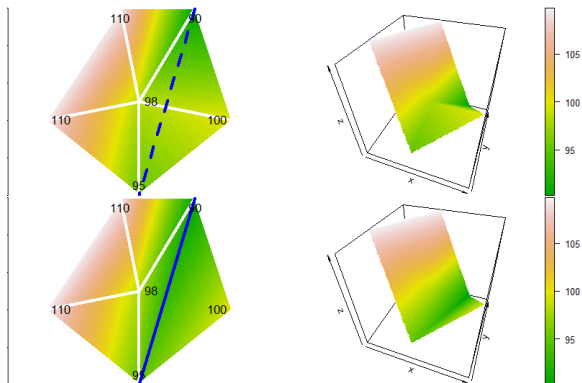
(B)

Linear interpolation and triangular irregular networks (cont.)

Linear interpolation is a widely used technique for topographic elevation data. The linear interpolation is done with respect to the set of **points where the elevation is measured**. Additional constraints defined by **breaklines** prevent Delaunay triangles to cross designated line features like contour lines, ridges or water streams.

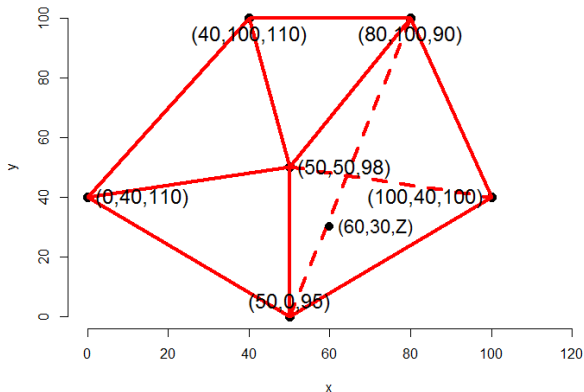
Top: No restrictions on the triangulation. The **water stream** (dashed) just provides elevation measurements.

Bottom: The **water stream** also defines a **breakline** that alters the triangulation. A better elevation model is derived.



Linear interpolation and triangular irregular networks (cont.)

Exercise: Consider the two triangulations illustrated in the previous slide. Estimate the elevation Z of the point marked with • ($x = 60, y = 30$) for each of those triangulations, i.e. write the system of equations that return the solution for each case.



Linear interpolation and triangular irregular networks (cont.)

Solution: In slide Slide 294 two cases were considered: the water stream is not a breakline (top), or it is a breakline (bottom).

- ① In the first case, • belongs to the triangle with vertices (50,50,98), (100,40,100) and (50,0,95). Therefore, the value of Z is given by the following system of linear equations.

$$\begin{cases} 98=a+ b \times 50+c \times 50 \\ 100=a+b \times 100+c \times 40 \\ 95=a+ b \times 50+ c \times 0 \\ Z=a+ b \times 60+c \times 30 \end{cases} \quad \text{with solution } Z = 97.32\text{m}$$

- ② For the second case, • belongs to the triangle with vertices (80,100,90), (100,40,100) and (50,0,95). The solution of the new system is consistent with the occurrence of the water stream.

$$\begin{cases} 90=a+ b \times 80+c \times 100 \\ 100=a+b \times 100+ c \times 40 \\ 95=a+ b \times 50+ c \times 0 \\ Z=a+ b \times 60+ c \times 30 \end{cases} \quad \text{with solution } Z = 93.86\text{m}$$

Linear interpolation and triangular irregular networks (example)

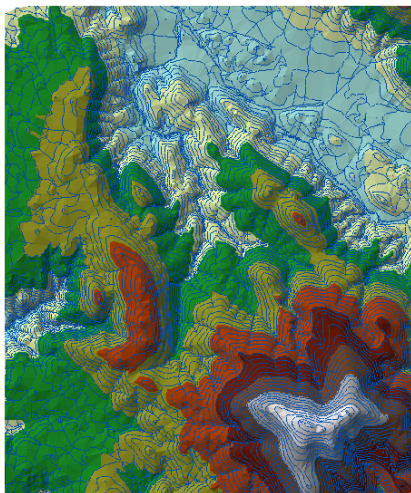
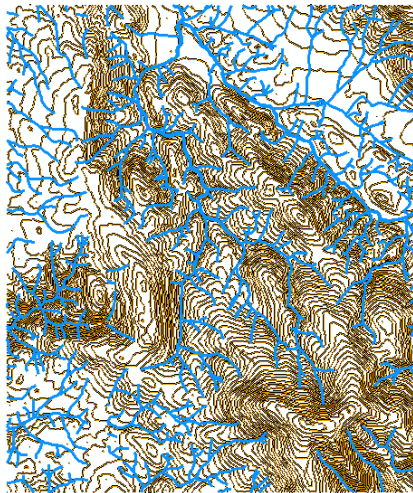


Figure: On the left, locations with known elevation (**water streams**, **contour lines**). On the right, an irregular triangular network that interpolates elevation over the whole region.

LAB: Linear interpolation and triangular irregular networks

Consider a set of topographic data at a very fine scale in an area of 5km by 6km in Eastern Portugal, projected to the CRS EPSG 20790 (Portuguese grid). In QGIS, use these data to define a Digital Elevation Model (DEM) with a spatial resolution of 10m for the study area. **Firstly**, use the available elevation information just as points, with no breaklines; **Secondly**, consider that countour lines and water streams are **breaklines**. Compare the results and determine where the differences are larger.

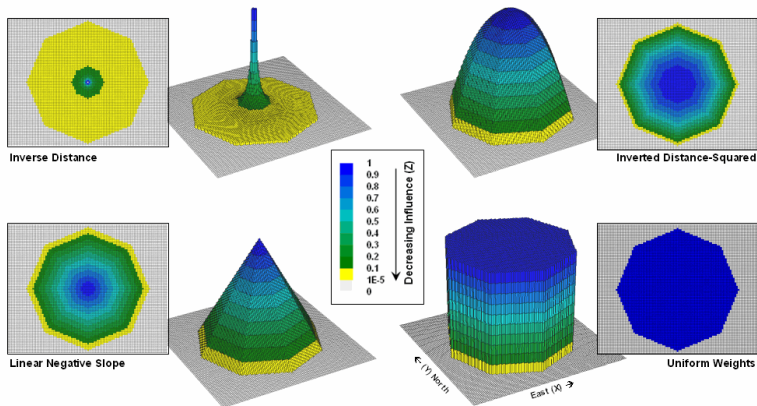
Data are in folder `Spatial_Interpolation`.

- 1 `ContourLines` is a line vector data set, where each line has the attribute `altitude` which values are meters;
- 2 `WaterStreams` is a line vector data set; Each geometric object is a 3D line, i.e. each vertex has coordinates (x, y, z) , where z is the elevation in meters.

In QGIS the TIN linear interpolation tool is available in menu **Raster / Interpolation**. The elevation of `WaterStreams` is given by its “Z-coordinate”.

Spatial interpolation with a moving window

A **kernel** k is a 2-dimensional distribution like any of the illustrated below. Its support is the domain where $k > 0$. The z estimate at location (x_0, y_0) is given by the following integral which is called a **convolution of k and f** : $\int \int k(x - x_0, y - y_0) \times f(x, y) dx dy$.

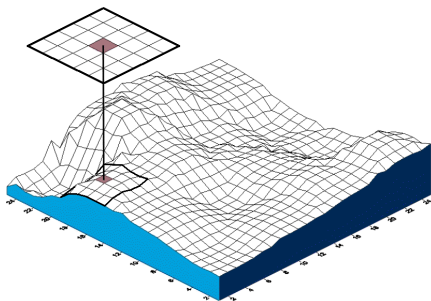


Spatial interpolation with a moving window (cont.)

Given a set of measurements $\{z_1, \dots, z_n\}$ at spatial locations $\{(x_1, y_1), \dots, (x_n, y_n)\}$ consider the function $f(x, y) = z_i$ when (x, y) is one of the locations and $f = 0$ otherwise.

For a given kernel $k \geq 0$, at any location (x_0, y_0) the convolution of k with f is a **weighted average** of all measurements that **fall inside the support of k** . The weights are given by the shape of k .

The kernel is “moved” across the space to compute the convolution at each location (x, y) . The result is a spatial interpolation of the available measurements.



Spatial Interpolation: an example of application

Ecology data sources for GIS are frequently derived by some kind of spatial interpolation technique. Among available data sources, many being freely-available, for inclusion in GIS, it is not always clear which data are native (as for most of remote sensing data) and which data are the result of spatial interpolation of point measurements.

Four types of data sources that provide GIS readable data.

- **Terrestrial** – *e.g.* Atlas of the Biosphere, South And Central America Environmental Data, MODIS-Derived Satellite Data;
- **Climatic** – *e.g.* IPCC Data Distribution Centre, Climatic Research Unit of the University Of East Anglia, WorldClim Global Climate Data;
- **Elevation, altitude, bathymetric and topographic** – *e.g.* SRTM 90m Digital Elevation Data and other sources at Earth Explorer;
- **Marine** – *e.g.* Permanent Service For Mean Sea Level, World Oceans Atlas.

Spatial Interpolation: an example of application

Ecology data sources for GIS are frequently derived by some kind of spatial interpolation technique. Among available data sources, many being freely-available, for inclusion in GIS, it is not always clear which data are native (as for most of remote sensing data) and which data are the result of spatial interpolation of point measurements.

Four types of data sources that provide GIS readable data.

- **Terrestrial** – *e.g.* Atlas of the Biosphere, South And Central America Environmental Data, MODIS-Derived Satellite Data;
- **Climatic** – *e.g.* IPCC Data Distribution Centre, Climatic Research Unit of the University Of East Anglia, **WorldClim Global Climate Data**;
- **Elevation, altitude, bathymetric and topographic** – *e.g.* SRTM 90m Digital Elevation Data and other sources at Earth Explorer;
- **Marine** – *e.g.* Permanent Service For Mean Sea Level, World Oceans Atlas.

Spatial Interpolation: the case of WorldClim

WorldClim is a climatic data set based on punctual **temperature** and **precipitation** time series.

It is a raster data set with ≈ 1 km nominal resolution (30 arc sec), derived from a set of weather station data, which is widely used in ecological modeling. The point data come from the following sources.

- Global Historical Climate Network Dataset (GHCN): over than 20000 weather stations for precipitation and over 7000 for temperature;
- World Meteorological Organization climatological normals: over 4000 and 3000 stations;
- FAOCLIM: over 27000 and 20000 stations;
- Other regional databases.

Overall, and excluding duplicate records, Worldclim is based on over 47000 locations for precipitation, 25000 for mean temperature, and 14000 for min and max temperatures.

Spatial Interpolation: some limitations of WorldClim

- 1 Although the set of weather stations is dense in parts of the world, the spatial distributions of stations is very irregular.

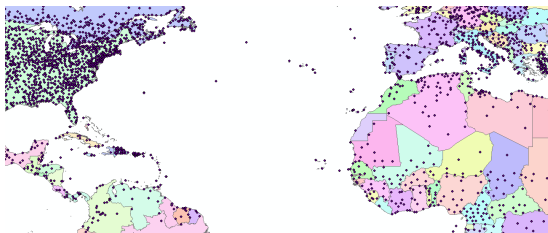


Figure: Spatial distribution of weather stations of the Global Historical Climate Network Dataset.

- 2 Worldclim is a **raster** dataset that covers all land areas – for most locations it only provides **estimates** of the climatic variables values.

LAB: Spatial interpolation with inverse distance and uniform windows

The goal of this exercise is to test some moving window interpolation tools available in QGIS. The available information is average temperature at several location in the Iberian Peninsula from the Global Historical Climate Network Dataset. Use the tool **Grid (interpolation)** in menu **Raster / Analysis** and (A) select algorithm **inverse distance to power**; (B) select algorithm **moving average**. In each case choose an output spatial resolution (1km), a convenient kernel support, and compare the results. Adjust the extent coordinates.

Data are in folder `Spatial_Interpolation`.

- 1 A point feature class `GHCN_stations_iberia`; the attributes `jan`, `fev`, ... are the average temperatures (in 0.01°C) measured at weather stations of the Global Historical Climate Network Dataset;
- 2 A polygon feature class `Iberian_Peninsula`.

Spatial Interpolation: Co-variables

In the previous examples, each variable was modeled independently. However, this is a weak approach for variables like elevation or precipitation which depend on variables other than location (x, y). namely, temperature and precipitation depend on elevation, aspect, and other environmental variables.

For instance, Worldclim raster layers are obtained by fitting a second-order spline using **latitude**, **longitude** and **elevation** as independent variables (these are called co-variables).

The techniques used earlier lack the possibility of using co-variables in spatial interpolation. To overcome this, other available techniques, not discussed here, are based on some assumption about an underlying spatial statistical model to provide estimates and uncertainty measures over those estimates.

- 1 geostatistical techniques;
- 2 geographic regression.