DOI 10.4430/bgta0360

Improved automated methods for near real-time mapping - application in the environmental domain

M. Iurcev, F. Pettenati and P. Diviacco

Istituto Nazionale di Oceanografia e Geofisica Sperimentale, Trieste, Italy

(Received: 29 October 2020; accepted: 17 April 2021; published online: 23 September 2021)

ABSTRACT Large networks of fixed or mobile sensors based on Internet of Things technologies generate a large amount of data, that requires specific methods for preprocessing, averaging, interpolation and graphic representation. In this, bidimensional scalar data are taken into account; we consider the issues of window averaging, interpolation, and convex hull definition. The considered scenario is a fully automated, near real-time acquisition and visualisation system. The aim of this work is to compare the quality and the computational cost of various interpolation methods, while highlighting the possible risks behind the need to introduce experts' choices in some of them. In this, some algorithmic optimisations and comparisons are presented, based on both simulated and real data. We present some software solutions for the problems of window averaging and buffering of the convex hull and some tests on interpolation methods like Inverse Distance Weighting and Natural Neighbours to determine which is the most appropriate for our real-time measurements.

Key words: spatial interpolation, near real-time, geostatistics, convex hull, natural neighbours.

1. Introduction

The "near real-time" acquisition and representation of environmental data has become a topic of increasing importance, especially after the wide proliferation of IOT (Internet of Things) systems [see for example the applications to agricultural monitoring in Fastellini and Schillaci (2020)].

The expression "near real-time" is without any well-established definition, hence we will simply consider the case of a delay shorter than 5-10 minutes for the whole process of data acquisition, transmission, storage, processing, and visualisation.

When massive data acquisition is made through a network of sensors (e.g. fixed weather stations, IOT mobile devices, crowdsourcing systems) or during a field survey (e.g. research vessels), then data processing and rendering may raise several issues and caveats, well studied and documented in geostatistics and other long-established disciplines. Furthermore, in a near real-time environment there are even more aspects to consider, like the computational time and complexity of processing and technological issues like transmission systems, efficient data storage and retrieval, server-side software and client-side web rendering.

The final step of the whole dataflow is the mapping of the spatial data.

Analysis and representation of spatial data on a map is commonly implemented by means of a Geographic Information System (GIS). Modelling spatial data, or better pattern recognition using contour maps, is an important step for the interpretation of data themselves. Moreover, in real time problems, along with the ever increasing use of new technologies, a huge need of methods of data visualisation and analysis in real time is emerging. An example, a typical case of near real time is the ShakeMap application in seismology, the tool to build maps of strong motion data soon after an earthquake, which needs an interpolation on a grid, using the seismic stations available data (Worden *et al.*, 2010) and the empirical ground motion model (GMM). In meteorological monitoring (Hutchinson, 1998; Boer *et al.*, 2001) a choice of suitable spatial interpolation is surely a key issue.

There are many methods of interpolation useful for real-time realisations. In general, we can distinguish two categories: deterministic and statistics methods. They are generally developed for interpolation on a regular grid, but not necessarily; there are methods that directly use triangulation between observed data points. The GMT package (Wessel *et al.*, 2019) for example, widely used in geosciences, uses both approaches. The triangulation method is very fast, as the contour is done directly on the structure of the 2D distribution of the data, but this comes at the expense of the quality of the map. Conversely, interpolation on regular grid methods produce smooth maps. As far as quality is concerned, linear methods are very reliable, such as bilinear interpolation or Kriging. Defined as the best method for modelling the spatial uncertainty, or better defined the BLUE (Best Linear Unbiased Estimator), it is the basis of geostatistics (Goovaerts, 1997; Isaaks and Srivastava, 1989; Wackernagel, 1998; Chilès and Delfiner, 1999), the theory based on the spatial correlation of data. However, this method needs preliminary data processing and various input parameters, making it not very easy to implement for real-time problems (e.g. Pebesma *et al.*, 2011). Boer *et al.* (2001) used it to model monthly climate data. Also the most used method of GMT package, Spline in Tension has two arbitrary parameters: the tension (T from 0 to 1) and the search ray. By the modality of this parameter you achieve a (T = 0) biharmonic to (T = 1) harmonic function configuration (Wessel and Bercovici, 1976). This method was used in the old version of ShakeMaps, with arbitrary T value; actually the new version uses a conditional MultiVariate Normal approach [MVN: Worden *et al.* (2010)]: the assessment of the average of bidimensional distribution, in which a variable of interest is interpolated through mean and a deviation, with the conditional of another variable of observations. The deviation is a linear combination of variations depending on the source and the geological conditions (intrinsic in the GMMs). The Natural Neighbour (NN) method (Sibson, 1980) is a linear combination of observed data and weights, which are normalised intersections between areas of Voronoi polygons. The advantage of this method is that it does not require input parameters, but has a much more cumbersome software implementation, due to the tessellation that underlies it.

1.1. Background and objectives

The context we are considering is an acquisition system of low-cost environmental sensors, sending continuous measures of physical quantities such as water, air or soil temperature, pressure, relative humidity, carbon dioxide, ozone or particulate matter (PM10, PM2.5) levels. The measures are georeferenced at least with GPS WGS84 latitude and longitude and marked with an UTC timestamp. They are scalar, but many considerations can be easily extended to vectorial data. Similar considerations are valid for a lot of different quantities, like environmental noise, sun exposure, biological parameters as seawater chlorophyll or measurements such as traffic density.

We just assume to have availability of a massive flux of data, which is incoming to our server from fixed or mobile sensors, validated and stored in a database.

Many similar systems have been developed in recent years, based on IOT and low-cost sensors for air pollution monitoring, especially in large urban areas, for example by Parmar *et al.* (2017), Duangsuwan *et al.* (2018) and Dhingra *et al.* (2019). Near real-time data monitoring is often implemented, whereas near real-time interpolation and mapping is far less common: a valid example can be found in Schneider *et al.* (2017).

Initially we want to represent our data only after some preprocessing like validation and window averaging. Subsequently, an interpolation of the samples will provide information in the whole acquisition region, not only in the sampling locations. There is a huge variety of distinct interpolation techniques, but we define two basic requirements to be met: the first requirement is the independence from human intervention, since the entire process should be an automated activity, running night and day on a server. In the second place, sufficient speed and an acceptable computational impact are needed. We want to test and compare two deterministic interpolation methods, IDW (Inverse Distance Weighting) and NNI (Natural Neighbour Interpolation).

We eventually define an area of confidence (boundary), preventing the map from showing interpolated data on distant regions, where the significance of the interpolation is too low. This is especially relevant when exposing data to a common audience, out of the scientific community, to avoid misunderstandings or acceptance of fake data as true.

Every result must be georeferenced and readily available on the server through an interactive web interface.

There are additional requirements, related to technological, economic or ethical aspects: the software environment must rely on open source protocols, programming languages and systems, e.g. a LAPP architecture (Linux - Apache - Postgres - PHP). Tailor-made coding with a high degree of flexibility and availability of scientific libraries suggests the use of Python (or similar languages, like R) with its standard calculus libraries (like NumPy, SciPy, etc.). Full GIS software like Surfer and Arcmap cannot be easily integrated into such an environment, because they are mainly designed for human interaction and only partially for automation and system integration. Moreover, they are commercial softwares (not open source), often available only for Windows and customisable only within given limits.

Lastly, the assessment of computational speed and performance requires the same conditions for all the algorithms: same hardware and same software environment. We compared Python programs with the same computer, data set and programming language.

2. Does automatic necessarily mean unbiased?

Since, as mentioned before, the system has to run unmanned, we wanted to understand the implications of such a choice; essentially whether such a system can be considered impartial, unbiased and not prejudiced. Comparing it, in fact, to other methods such as for example the Bayesian network, that generally is referred to as volitional and not automatic, can an automated method guarantee what is commonly called objectivity?

We understand that this largely depends on what we mean with the term objectivity. Several authors, such as for example Karsken (1992), Spade (1994) or Baise (2020), noticed that its meaning changed quite a lot from how we commonly use it now and has even been reversed by some thinkers.

Modern philosophy of science shows us that starting from the observation itself up to the theorisation phase, the standard scientific method is biased by several inconsistencies that undermine its traditional authoritative position. Cognitive penetrability and theory-ladenness perspectives stemming from the studies of Hanson (1958), MacPhersons (2012) and Zeimbekis (2013) maintain that there is no way to avoid that knowledge of *x* is biased by previous knowledge of *x*. In addition, following a path linking Duhem, Quine and Lakatos, once theories are built, they are not really questioned by tests; on the contrary, experiments themselves become only the projection of such theories. Science, then, becomes less logic and more socio-logic (Diviacco, 2012, 2014) as a network of supporters is necessary for a vision to exist (Whitley, 2000). Following Latour and Woolgar (1986), science is a social construct so that in this framework it is not easy to define what is unbiased. Methods that do not involve human intervention can, therefore, avoid introducing further arbitrariness but at the same time, since they cannot decontaminate the data, they process from the intrinsic cognitive penetrability and theory-ladenness data contain and their results cannot be considered unbiased.

3. Window averaging

Data collected from automated acquisition can be very copious and irregularly spaced. Furthermore, other issues must be considered:

a) too dense point clusters may raise computational issues during interpolation, like ill conditioned matrices or numerical precision errors, which are difficult to control;
b) when a physical quantity is measured there are statistical fluctuations, due to environmental causes or low precision of the sensor, especially in the special case of low-cost sensors, which are not relevant for the final user;
c) several physical quantities, like air temperature or humidity in a fixed location, are time-dependent, but their variations are normally quite slow across the day;
d) a reduction of data set size before interpolation is very convenient in order to reduce the computational effort;
e) spatial and temporal data representation is more understandable if we show the aggregated data and not the single acquisitions;
f) in the particular case of crowdsourcing paradigm may be better to hide the individual location of the contributors and show only aggregated data, for privacy reasons.

Data aggregation based on moving window averaging (e.g. Isaaks and Srivastava, 1989) is a simple approach to deal with all these issues. We used an overlapping rectangular moving window, defined by a fixed reference grid, with 200×200 m^2 cells, in combination with a time quantisation of one UTC hour. The spatial grid is based on the coordinate system WGS84/UTM zone 33N (EPSG:32633), well fitting for surveys in Italy and based on metric coordinates. A slice of data within one UTC hour (e.g. from 2020-01-01 14:00:00 to 2020-01-01 15:00:00) and one single grid cell is a datacube.

Of course, the choice of datacube's size depends on the specific context. 200×200 m^2 × 1 hour is a reasonable option for the spatial and temporal resolution of air quality parameters like temperature or particulate matter, but the mechanism could be improved with an adaptive grid. Nevertheless, a fixed reference grid of datacubes has an advantage in terms of implementation and processing simplicity, visualisation conciseness and ease of representation of time evolving spatial phenomena. Besides, other statistical properties on the datacube are calculated, like the number of samples or the standard deviation.

In Fig. 1 the location of single samples acquired within one hour is shown. In Fig. 2 the cells after averaging (measure of atmospheric relative humidity) are displayed.

Wider time intervals were also considered (2, 4 ... up to 24 hours), for a better averaging of slow changing parameters, like PM10 pollutants.

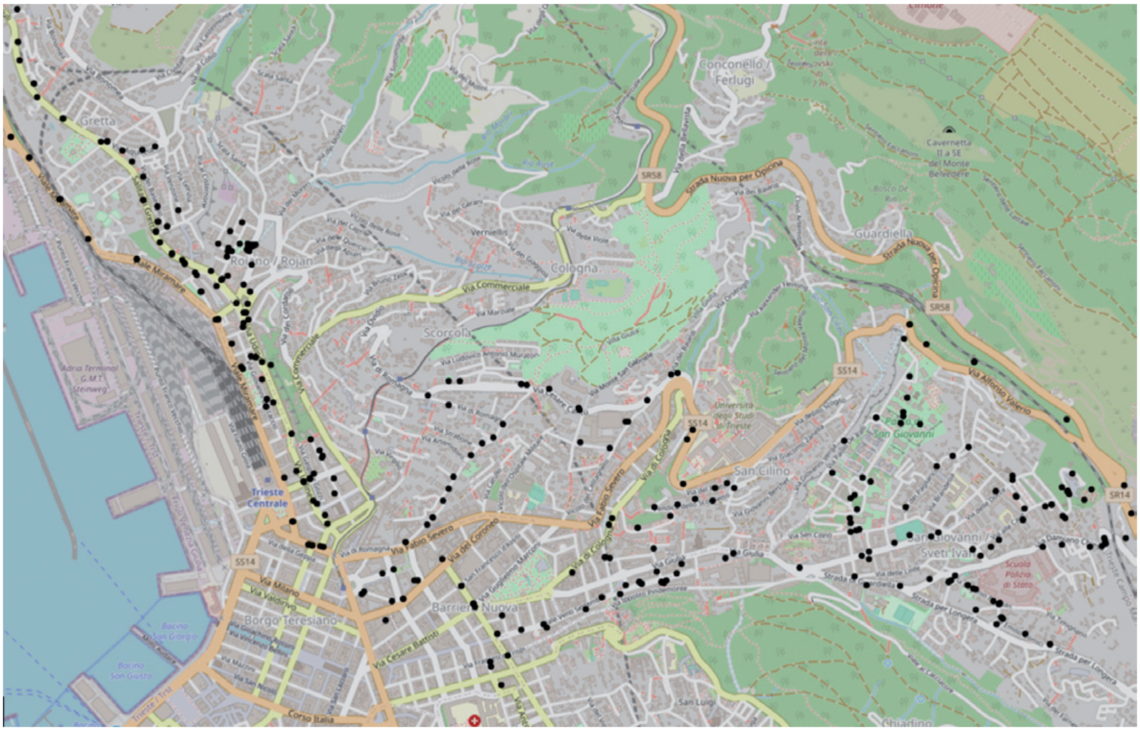

Fig. 1 - Sample points.

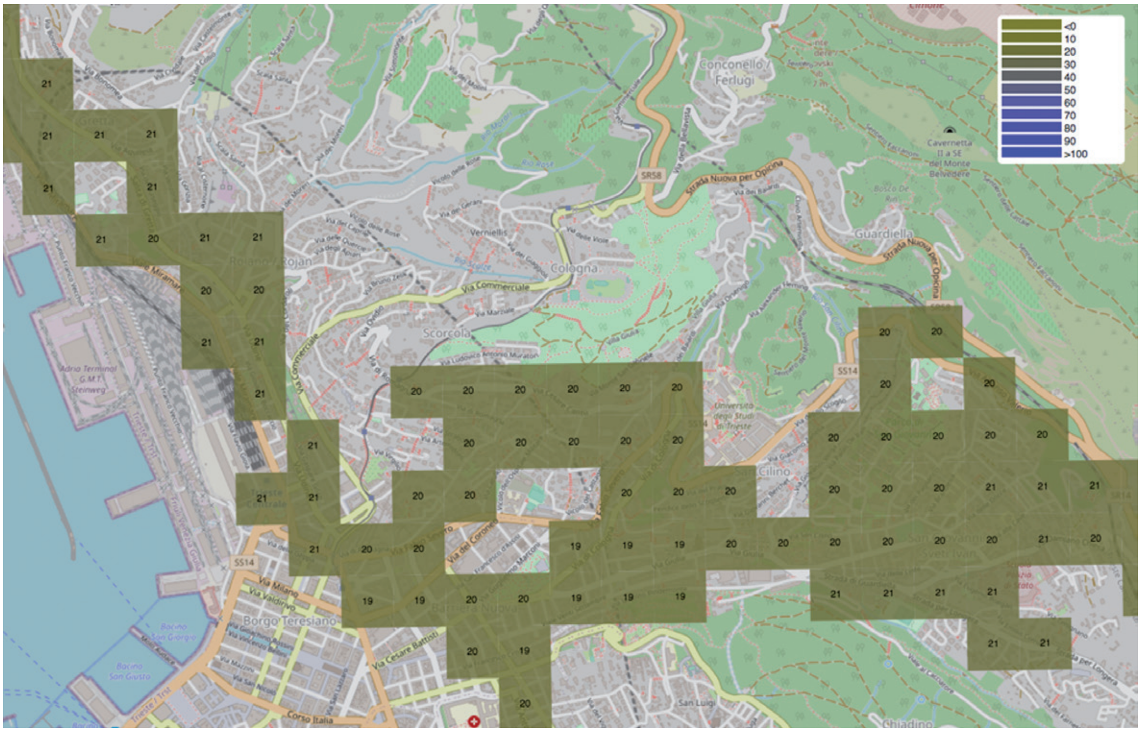

Fig. 2 - Averaged cells.

3.1. Rectangular vs. hexagonal grids

Rectangular grids for window averaging are an obvious and rational choice, but other kinds of tessellation are possible, like the hexagonal grid. Hexagons are much less used than squares, primarily for their more cumbersome implementation, but, as well analysed by Birch *et al.* (2007), a 'beehive' geometry introduces some favourable characteristics.

Maps in a hexagonal grid tend to be less ambiguous than maps in a rectangular grid, especially in the corners. Rectangular lattices are also distracting because human vision is especially sensitive to vertical and horizontal lines and generally speaking the hexagonal grid should be preferred when it has advantages for the construction or representation of nearest neighbourhood, movement or connectivity.

We extensively implemented and tested hexagonal grids alongside rectangular grids, as displayed in Fig. 3. It is not easy to compare the computational effort required by the two methods, since it is very dependent on the software architecture. For instance, we chose to calculate and store the rectangular or hexagonal cells into a PostGIS database, as polygons. This operation is done only once, so it has no impact on further system performance. Later, PostGIS queries allow us to select points, which are contained in specific cells, through complex SQL operations. At this point, the cell geometry may affect the overall performance, simply because we are using either 4-side or 6-side polygons.

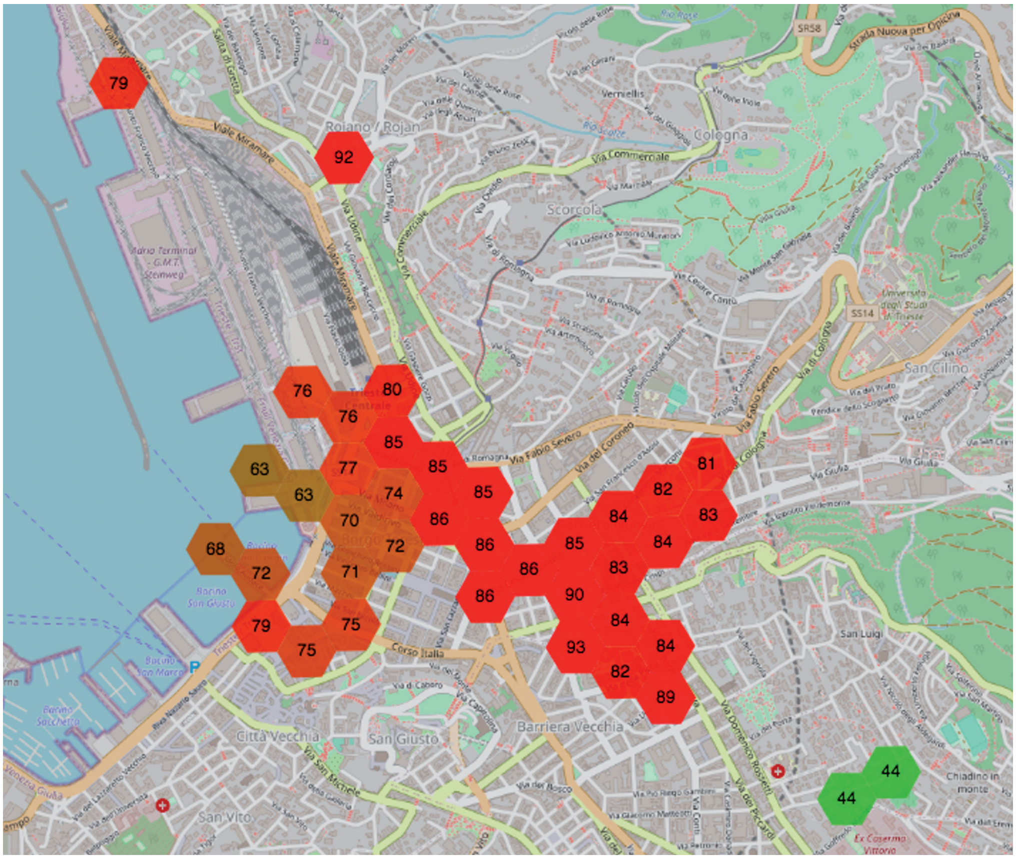

Fig. 3 - Hexagonal averaging grid.

4. Deterministic vs. geostatistical interpolation

The issue of geospatial data interpolation may be addressed in several ways.

Generally speaking, we call $X = \{\boldsymbol{x}_1 ... \boldsymbol{x}_N\}$ the set of interpolating points, corresponding to our measured values $\{f(\boldsymbol{x}_1) ... f(\boldsymbol{x}_N)\}$. We are looking for methods to calculate $f_{INT}(\boldsymbol{x})$ (interpolant) in a generic point $\boldsymbol{x}$.

Function $f(\boldsymbol{x})$ may be known or unknown. In case of measurement of a physical quantity, it is generally unknown. Interpolation is hence a problem of estimation. Interpolating points $\boldsymbol{x}$ beyond the range of X is called extrapolation and is subject to greater uncertainty, therefore it is necessary to define a reasonable interpolation domain surrounding X, a boundary. This issue will be addressed later.

A relevant subdivision of interpolation techniques is between deterministic and geostatistical. While the deterministic method uses directly a mathematical approach based on some spatial relation as distance between points (IDW, see below) or tessellation area (NNI, see below), the latter (for instance Kriging) relies on modelling the measures as realisation of a stochastic process $Z(\boldsymbol{x})$, hence: $f_i = Z(\boldsymbol{x}_i)$; the former simply considers the spatial relationship between $\boldsymbol{x}_i$. In the geostatistical approaches (Wackernagel, 1998), the spatial relationship is realised by the variograms, statistical tools that allow to assess the spatial autocorrelation.

4.1. Inverse Distance Weighting interpolation (IDW)

IDW is a deterministic method, first introduced by Barnes (1964) and then resumed by Shepard (1968).

Assuming that $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ are the interpolating points (data) and $\boldsymbol{x}$ a generic point, the interpolant is:

$$f_{IDW}(\boldsymbol{x}) = \frac{\sum_{k=1}^{N} w_k(\boldsymbol{x})\, f(\boldsymbol{x}_k)}{\sum_{k=1}^{N} w_k(\boldsymbol{x})} \tag{1}$$

where the weights are:

$$w_k(x) = \parallel x - x_k \parallel^{-\mu}. \tag{2}$$

A typical exponent range is $2 \leq \mu \leq 4$ for a smooth result, giving more influence to $\boldsymbol{x}_k$, which are closer to the interpolated point $\boldsymbol{x}$. Here we chose the values 3 and 4.

Shepard's method is very simple to implement, quite fast and produces more than acceptable graphic representations. It is used by many scientific analysis software, like Ocean Data View (https://odv.awi.de/) for fast rendering of interpolated spatial data in a marine environment, when heavier computational methods are not preferable (like DIVA interpolation, based on a finite element mesh).

IDW can be further improved assuming that, as locations get farther from the prediction location, the measured values have less spatial autocorrelation with the prediction location. Then, a search neighbourhood can be defined, eliminating from the calculus the samples outside a certain circular range. This technique also improves the computational impact, by reducing the number of samples used in the formula. The size and the shape (if not circular) of a search neighbourhood are parameters to adjust, along with the choice of an exponent, then requiring a human intervention or some adaptive algorithm.

4.2. Natural Neighbour Interpolation (NNI)

A basic definition of a planar ordinary Voronoi decomposition is:

$$let\ X := \{x_i, \ldots., x_N\} \subset R^2\ where\ 2 < n < \infty\ and\ x_i \neq x_j\ for\ i \neq j. \tag{3}$$

We call *Voronoi polygon* associated with $\boldsymbol{x}_i$ the region given by:

$$V(x_i) = \{x|\ \|x - x_i\| < \|x - x_j\|\ for\ i \neq j\}. \tag{4}$$

Whereas the set given by:

$$V = \{V(x_i), \ldots., V(x_N)\} \tag{5}$$

is the Voronoi decomposition associated with set *X* (Okabe *et al.*, 2000). Intuitively, every region contains all points, which are closest to $\boldsymbol{x}_i$ than to every other point of set *X*.

Fig. 4 shows the Voronoi decomposition of a limited area.

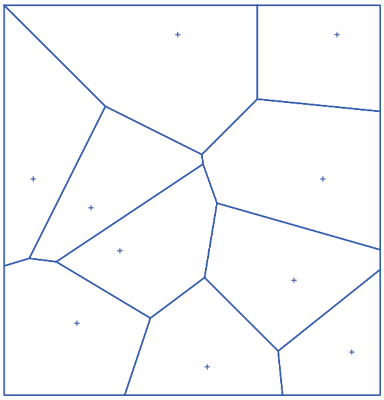

Fig. 4 - Voronoi decomposition.

NNI interpolation is based on a local interpolant, which represents the value of *f(**x***)* in an arbitrary point ***x**** as a weighted and linear function of the values *f(*$\boldsymbol{x}_i$*)* in the 'nearby' points $\boldsymbol{x}_i$

$$f(\boldsymbol{x}^*) = \sum_{i=1}^{N} w_i \cdot f(\boldsymbol{x}_i) \tag{6}$$

where w_i are the weights, with $w_i \neq 0$ only for 'nearby' points.

Sibson (1980) used proximal Voronoi cells as a definition of 'nearby'. As well described in Okabe *et al.* (2000), Sibson's method defined the weight as normalised ratios between areas:

$$w_i := \frac{A_i}{\sum_j A_j} = \frac{A_i}{A_{TOT}} \tag{7}$$

where A_i is the area of $V(\mathbf{x}_i) \cap V(\mathbf{x}^*)$ (for the details see next paragraph), while A_{TOT} is the area of the Voronoi cell relative to ***x****, as displayed in Fig. 5.

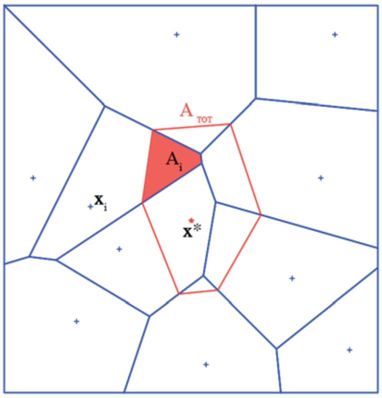

Fig. 5 - Voronoi cell for the point of interpolation.

NNI aims to take into account clustering of data and distance of the interpolation from data via a geometrical (thus deterministic) approach, implying a linear decline of spatial correlation with distance.

4.3. Software improvement of the NNI algorithm

The Python module SciPy (scientific library for mathematics, science and engineering, based on Qhull library) includes all the tools for Voronoi and Delaunay decomposition. As we implemented the NN method, the first choice was to use this library (scipy.spatial.Voronoi) in two different steps:

a) just once for the initial Voronoi decomposition $\mathcal{V}$, relative to our data set;

b) for every single interpolation point ***x****, when we need to compute the new Voronoi cell $V(\mathbf{x}^*)$ and all its areas of intersections with $\mathcal{V}$.

The first step has little impact on performance, because it runs only once. The second step can be critical. A straightforward implementation would be recalculating the whole Voronoi decomposition for every ***x****, simply adding ***x**** to the set of points $\{\mathbf{x}_i\}$, just in order to find the new Voronoi polygon $V(\mathbf{x}^*)$.

In effort to reduce the computation time for step b, a different algorithm has been developed and implemented, which does not use of SciPy library during step b and does not require any intersection to be calculated between convex polygons $V(\mathbf{x}_i) \cap V(\mathbf{x}^*)$.The method is based on the basic properties and definitions of Voronoi decomposition, as displayed in Fig. 6.

The objective is to find f^*, the interpolated value in ***x****.

We assume that: $\mathbf{x}_i$ are the samples; $f(\mathbf{x}_i)$ are the measured values; $V_i = V(\mathbf{x}_i)$ the Voronoi polygons.

The core idea is a variation of the perpendicular bisector method (e.g. Isaaks and Srivastava, 1989): for every Voronoi polygon V_i we find the perpendicular bisector between $\boldsymbol{x}_i$ and $\boldsymbol{x}^*$ (the point to be interpolated). We look at the intersection between V_i and the bisector h. If it is empty, then V_i is not a neighbour, else the intersection between V_i and the half-plane H (defined by the bisector) containing $\boldsymbol{x}^*$ is exactly the partial area A_i needed for the NNI formula. Particular attention must be paid to degenerate cases where bisector h passes through one or two vertices of polygon V_i. In pseudocode:

```
$A_{TOT} = 0$
$f^* = 0$
for each $V_i$:
        if $\boldsymbol{x}_i = \boldsymbol{x}^*$ then END: $f^* = f(\boldsymbol{x}_i)$
        find $\boldsymbol{x}_h = 1/2\ (\boldsymbol{x}_i + \boldsymbol{x}^*)$ (middle point)
        let $h$ be the line orthogonal to segment $\boldsymbol{x}_i\boldsymbol{x}^*$, passing through $\boldsymbol{x}_h$
        for each edge $s_j$ of $V_i$:
            if exists, find the intersection point between $h$ and $s_j$
            $\boldsymbol{a}$ is the first one found, $\boldsymbol{b}$ is the second one
        if $\boldsymbol{a} \simeq \boldsymbol{b}$ or there is no intersection, then skip $V_i$
        define the set P = {$\boldsymbol{a}, \boldsymbol{b}$}
        let $H$ be the half-plane defined by $h$ and containing $\boldsymbol{x}^*$
        for each vertex $\boldsymbol{v}_j$ of $V_i$:
            if $\boldsymbol{v}_j \in H$ then add $\boldsymbol{v}_j$ to the set P
        order the points in P (vertices of a convex polygon)
        $A_i$ = area(P) is the partial area relative to $\boldsymbol{x}_i$
        $f^* = f^* + f(\boldsymbol{x}_i) / A_i$
        $A_{TOT} = A_{TOT} + A_i$
```

Eventually we can calculate the interpolated value as described above:
$f^* = f^* / A_{TOT}$

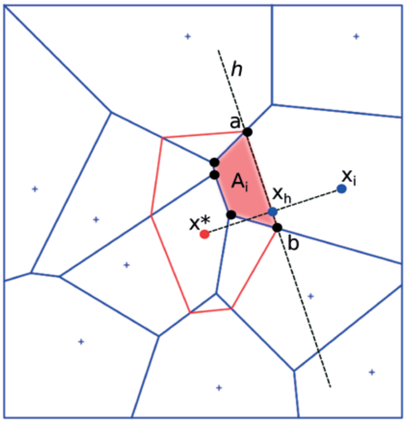

Fig. 6 - Improved NNI method.

Considering the distribution of the number of vertices for Voronoi polygons, which is Gaussian with an average value of 6 [at least in a Poisson Voronoi decomposition, see Okabe *et al.* (2000)], then the number of iterations and operations inside the external loop (over index *i*) is always limited. All the operations can be implemented as simple additions and multiplications.

The time complexity order is therefore *O(N)*, where *N* is the number of sample points $\boldsymbol{x}_i$ and we may conclude that NNI can be implemented with a linear algorithm like IDW.

Interpolation methods like IDW or NNI, based on Euclidean distance $d = sqrt(\Delta x^2 + \Delta y^2)$ should not be directly applied on geographic coordinates (longitude and latitude <lon, lat>), which are anisotropic far from equator. This would bring inconsistent results also in a small geographic area with neglectable curvature, because $\Delta x_m/\Delta y_m \neq \Delta lon/\Delta lat$, where $<x_m, y_m>$ are some metric coordinates. The data must be first reprojected to a metric projection (e.g. UTM or Pseudo Mercator) and, then, interpolated. In the following example, we show a small data set with Voronoi decomposition and the additional Voronoi polygon relative to an interpolation point $\boldsymbol{x}*$. The geometry of Voronoi polygons was calculated respectively after and before the reprojection. Fig. 7 shows an example of the difference between the two cases, which thereafter produces a significant difference in NN linear interpolation, with a different topology of Voronoi tessellation and different intersection areas. The IDW method has circular symmetry around data points, hence the difference between the two cases is a simple vertical stretch, because a circle of equidistant points becomes an ellipse.

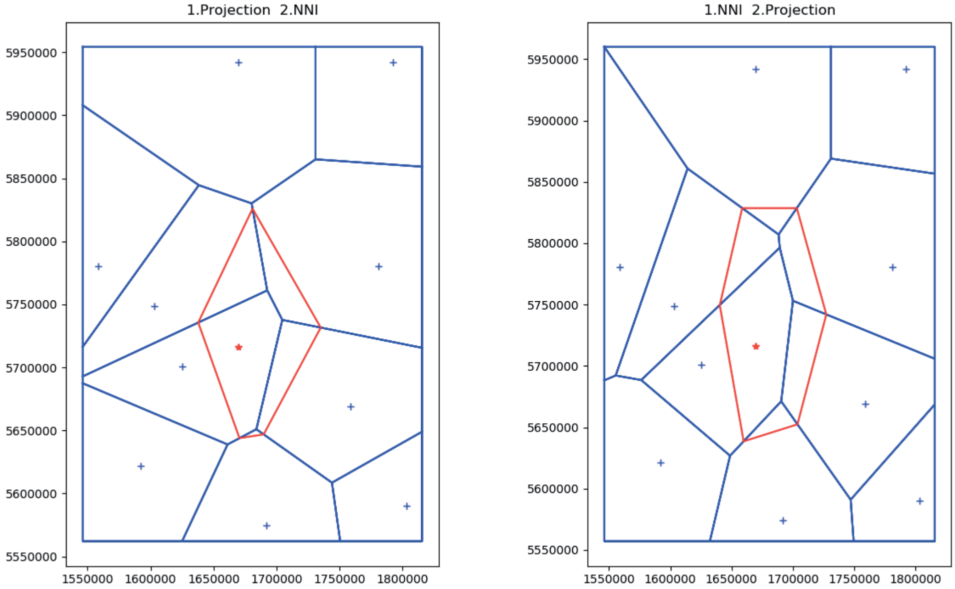

Fig. 7 - Dependency of Voronoi geometry on anisotropic coordinates.

Furthermore, if the data set covers a wide geographical area, we should also consider the curvature and a definition of distance on a spherical surface.

4.4. Kriging

Let us consider the stochastic process $Z(\boldsymbol{x})$ and assume it is a Stationary Random Function (SRF). Defined its mean as the first moment:

$$m = \boldsymbol{E}[Z(\boldsymbol{x})]. \tag{8}$$

We define as second-order moment, the covariance function (Chilès and Delfiner, 1999):

$$cov[Z(\boldsymbol{x}), Z(\boldsymbol{x}+\boldsymbol{h})] = C(\boldsymbol{h}) = \boldsymbol{E}[(Z(\boldsymbol{x}) - \boldsymbol{E}(Z(\boldsymbol{x}))) (Z(\boldsymbol{x}+\boldsymbol{h}) - \boldsymbol{E}(Z(\boldsymbol{x}+\boldsymbol{h})))]. \tag{9}$$

If the process is second-order stationary, we can also assume $\boldsymbol{E}[Z(\boldsymbol{x})] = \mu$ (constant mean) and the variance of the increments $Var[Z(\boldsymbol{x}) - Z(\boldsymbol{x}+\boldsymbol{h})]$ to be finite and depending only on distance $\boldsymbol{h}$ (condition of isotropy). The correlation between $Z(\boldsymbol{x})$ *and* $Z(\boldsymbol{x}+\boldsymbol{h})$ is a function named correlogram

$$\rho = C(\boldsymbol{h})/C(0). \tag{10}$$

The covariance and the correlogram show how the correlation depends on distance $\boldsymbol{h}$. The covariance is an even function and according to Schwartz's inequality it is bounded by $\boldsymbol{h}$ at the origin (Chilès and Delfiner, 1999):

$$C(\boldsymbol{h}) = C(-\boldsymbol{h}) \qquad |C(\boldsymbol{h})| \leq C(0). \tag{11}$$

Hence we may define the theoretical semivariance, a second-order moment, function of *h:*

$$\gamma(h) = \tfrac{1}{2}Var[Z(x+h) - Z(x)] = E[\big(Z(x+h) - Z(x)\big)^2]. \tag{12}$$

The semivariance is estimated through the analysis of empirical variogram:

$$\gamma_e(h \pm \delta) := \frac{1}{2|N(h \pm \delta)|} \sum_{(i,j)\in N(h\pm\delta)} \left(z_i - z_j\right)^2 \tag{13}$$

where z_i are the sampled values, separation distances are divided into bins $h \pm \delta$, and $N(h \pm \delta)$ is the number of sample pairs with a distance included in each bin. By analysing γ_e, different parameters are estimated:

- variogram model: an equation fitting the empirical result (most used are exponential, Gaussian or spherical, see Fig. 8);
- nugget: intersection of the curve with *y*-axis;
- sill: asymptotic height of the curve;
- range: distance (on *x*-axis) where the difference between curve and sill becomes negligible.

Kriging is then defined as a linear estimator, where the unknown value z_0 is estimated by a linear combination of the *n* data plus a shift parameter λ_0:

$$z_0^* = \lambda_0 + \sum_{i=1}^{N} \lambda_i \, z_i. \tag{14}$$

Let us interpret the z_i data values as outcomes of random variables Z_i, and the unknown value to estimate z_0 as outcome of a random variable Z_0. To solve this equation we have to select the constants and the weights to minimise the error variance $z_0^* - z_i$, in sense of mean square . The minimum is ensured by the positive property of the covariance function that is a convex function (Chilès and Delfiner, 1999). The formula in matrix notation, in the case of Simple Kriging is

$$\boldsymbol{K\lambda} = \boldsymbol{k_0} \Longrightarrow \lambda = K^{-1}\,\boldsymbol{k_o}. \qquad (15)$$

If the data covariance matrix $\boldsymbol{K}$ is a positive definite matrix, this yields one and only solution. $\boldsymbol{K}$ and $\boldsymbol{k_0}$ are estimated using the variogram.

We may observe a few facts. First, Kriging is perfectly suitable for the interpolation of environmental measures and the literature about this method is abundant [an example is described by Tyagi and Singh (2013), who applied the method to PM10 data].

Second, the stochastic method of Kriging is difficult to implement as an automated procedure, and normally requires human intervention and analysis. Recently, interesting works were presented, which explore techniques of automated refinement of the variogram model, like in Barca *et al.* (2017), where the whole interpolation process is only partially unsupervised (e.g. Pebesma *et al.*, 2011).

Nonetheless, deterministic methods like IDW or NNI are simpler choices for a fully automated implementation, relying only on the spatial properties of the samples.

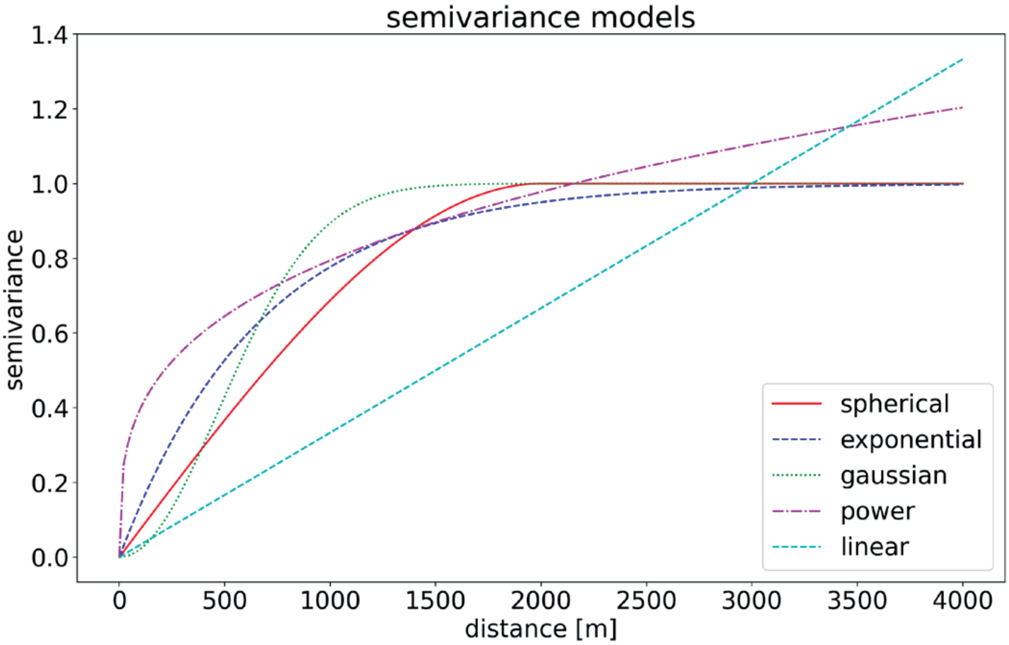

Fig. 8 - Semivariance (variogram) models for Kriging.

In order to show how different can be the variograms, we used a data set of 648 measurements (acquired within one hour) and plotted the variogram for two different quantities: pressure (Fig. 9) and PM10 (Fig. 10).

Range and sill exhibit different values, which need to be evaluated and normalised. In addition, the pressure diagram resembles a Gaussian model, whereas the PM10 diagram is better represented by a spherical model.

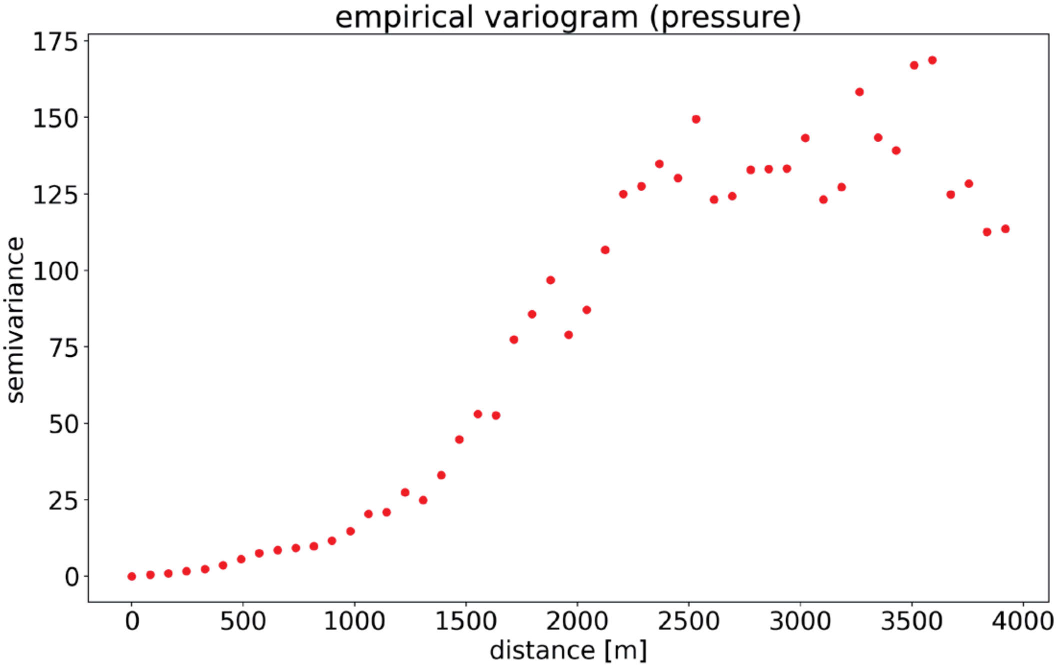

Fig. 9 - Empirical variogram for pressure measures.

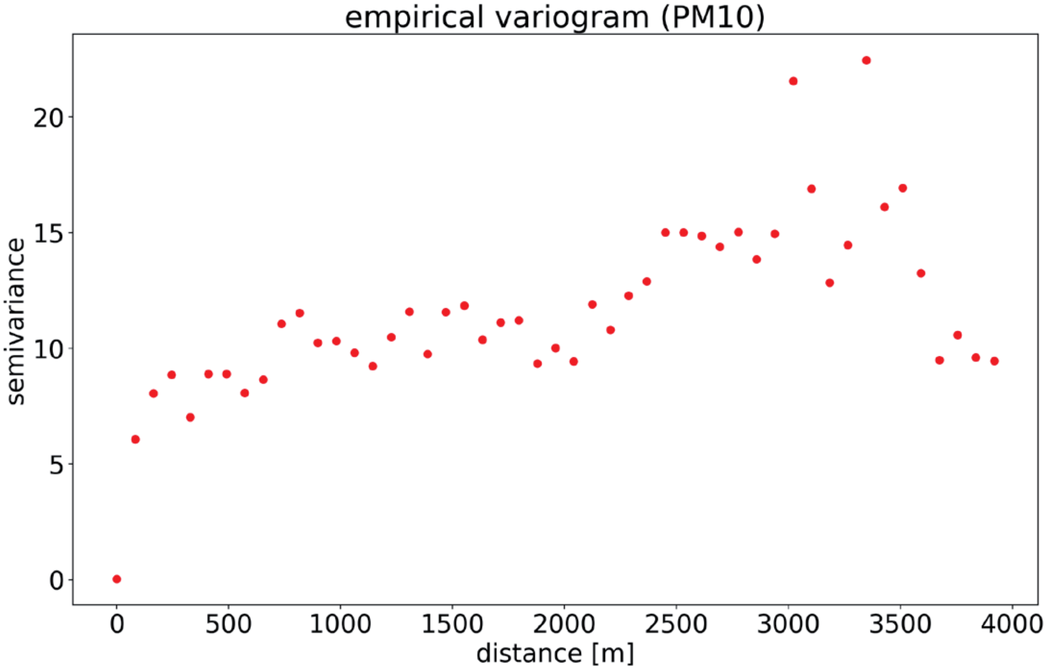

Fig. 10 - Empirical variogram for PM10 measures.

Further considerations must be taken into account, like the use of anisotropic Kriging, where the variogram is not only a function of distance *h*, but has a significant bias along a direction (defined by its "azimuth" angle), or the more general case of "trended" data, which requires the "detrending" of data before the interpolation, after the recognition of a trending surface (linear or of higher order), superimposed to data. All these considerations require estimations and processing that are not easy to implement as an unsupervised method.

In conclusion, in our investigation we dismissed Kriging methods. The reason is the high degree of complexity involved in automatisation, required for the interpolation of near real-time data.

5. Comparison

The evaluation of interpolation techniques and their comparison may rely on different aspects. Foster and Evans (2008) conveniently divide quantitative evaluation methodologies into two main classes: those which require a full and correct output to be known *a priori*, and those for which the correct outputs are unknown. They define them as simulation-validation and cross-validation. Using an *a priori* approach, or simulation-validation, and testing interpolations of functions $f(\boldsymbol{x})$ ($\mathbb{R}^2 \rightarrow \mathbb{R}$), there are possibly infinite test functions to choose, with very different characteristics. The test function $f(\boldsymbol{x})$ is one possible realisation of the random variable $Z(\boldsymbol{x})$ for the considered stochastic process. We are not including time-dependency, for sake of simplicity.

For instance, the distribution of air temperature, air humidity or PM10 all have unique traits, depending on the physical quantity we are measuring, on the dynamic evolution and many other boundary conditions like geographical features, water bodies or anthropic structures. We may assume some level of differentiability, but some statistical properties like spatial stationarity cannot be assured. As an example, air pressure is strictly related to altitude, and pollution to urban areas; this implies that bivariate covariance may vary significantly from one area to another. We based the comparisons on two different test data sets. The first one is an analytical function (Franke's test function), the second one is a "real world" data set, a DTM surface.

A similar approach to comparison among interpolation techniques can be found in Bater and Coops (2009), who instead take into account high resolution LiDAR data.

5.1. Franke's test function

In his work, Franke (1979) was concerned with methods for solving the scattered data interpolation problem. He made a comparison of 29 methods for a solution of this problem. Several of these methods were variations of IDW interpolation. NNI was not included, since it would be defined many years later. In order to test the spatial interpolation techniques, he proposed some test functions $f(\boldsymbol{x})$, non-polynomial and with different characteristics. The principal and subsequently most cited one is Franke's function:

$$
\begin{aligned}
f(x, y) = {} & \frac{3}{4}\, exp\left[-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right] + \frac{3}{4}\, exp\left[-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right] \\
& + \frac{1}{2}\, exp\left[-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}\right] - \frac{1}{5}\, exp\left[-(9x\ -4)^2 - (9y\ -7)^2\right].
\end{aligned}
\tag{16}
$$

Franke (1979) describes it as a surface consisting of two Gaussian peaks and a sharper Gaussian dip superimposed on a surface sloping towards the first quadrant. It is represented in Fig. 11. The displayed surface is undersampled only for visual clarity.

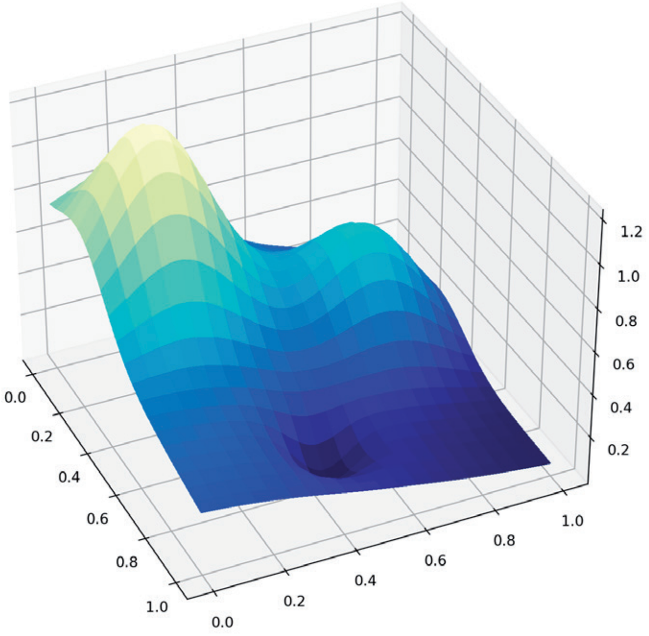

Fig. 11 - Franke's test function.

5.2. DTM as test surface for environmental data

Considering the lowest atmosphere layer (altitude = 0÷11 km) the barometric formula is:

$$P(h) = P_b \cdot \left[\frac{T_b}{T_b + L_b \cdot (h - h_b)}\right]^{\frac{g_0 \cdot M}{R^* \cdot L_b}} \quad (17)$$

with:

P_b = static pressure: 1013.25 hPa;
T_b = standard temperature: 298.15 K = 25 °C;
L_b = standard temperature lapse rate: -6.49 K/km;
h = height above sea level [m];
R^* = universal gas constant: 8.3144598 J/(mol·K);
g_0 = gravitational acceleration: 9.80665 m/s^2;
M = molar mass of Earth's air: 0.0289644 kg/mol.

The formula is charted in Fig. 12. Considering a first-order approximation from h = 0 (blue dashed line), we may appreciate that the formula is almost linear for altitude lower than 1000 m. The pressure drops approximately by 11.3 Pa/m.

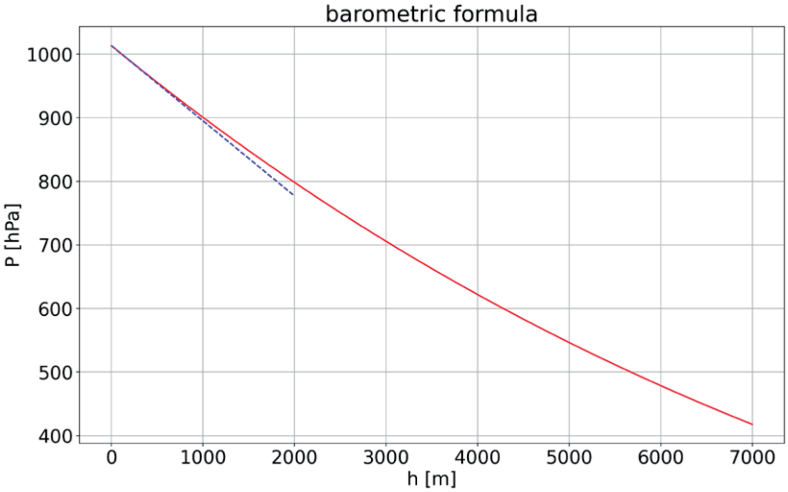

Fig. 12 - Barometric formula.

Atmospheric pressure is one of the physical quantities we are considering (measuring and processing). Assuming that pressure is in linear relationship with altitude, we may infer that DTM (digital terrain model) is a valid and significant test set for the considered interpolation methods, since the geostatistical properties of (smoothed) altitude are coherent with the properties of pressure.

The Carta Tecnica Regionale ("Regional Topographic Map" published by the Autonomous Region Friuli Venezia Giulia in Italy) offers DTM with a resolution of 10 m. A "low resolution" DTM was used for its availability but also because environmental parameters normally have quite smooth variations; for this reason a very detailed surface (e.g. a 0.5 m terrain model) should not bring an improvement within this specific context. We used a rectangular slice relative to an area within the Trieste municipality (Fig. 13).

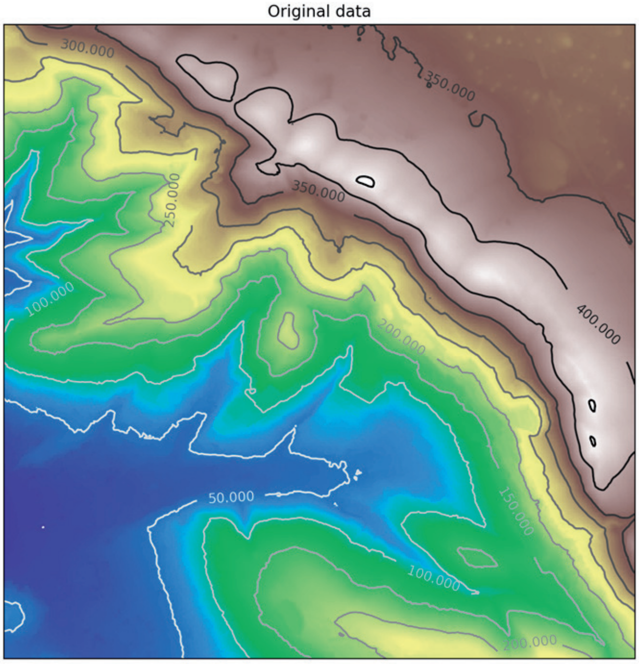

Fig. 13 - Original DTM data.

6. Boundary and buffering

The simplest method for a geospatial interpolation is to extend the computation to a rectangular bounding box. This approach can be partially misleading when the shape of the point cloud is not well fitting in a rectangle. We may have points external to the cloud and too far from it to provide meaningful results. To solve this issue and to better represent the interpolation on a map, a basic approach is to use the convex hull of the point cloud. The convex hull of a point set is the smallest convex polygon containing the whole set. If we imagine the point cloud nails planted on a wooden table, the convex hull would be a rubber band surrounding all the nails.

The area of the convex hull can be furtherly extended outwards within a limited distance. This operation is generally denominated "buffering" and provides some additional area of extrapolation, surrounding the convex hull at fixed distance *d*, in first approximation. Distance *d* may be defined as a fraction of maximum distance between the points. The basic concept of convex hull to help assess the potential reliability of prediction in unsampled areas may be further analysed and extended to a generalised independent variable hull (gIVH), as reported by Conn *et al.* (2015).

The implemented algorithm moves through the ordered points P_i of the convex hull and for every point:

- find the segment P_iA_i of length *d*, rotated of π/2 clockwise in respect to $P_{i-1}P_i$;
- find the segment P_iB_i of length *d*, rotated of π/2 counterclockwise in respect to P_iP_{i+1};
- fInd the segment P_iC_i of length *d*, bisecting the angle 2α (defined by points $A_iP_iB_i$).

The new buffered perimeter is .... $A_{i-1}\,C_{i-1}\,B_{i-1}\,A_i\,C_i\,B_i\,A_{i+1}\,C_{i+1}\,B_{i+1....}$ as in Fig. 14.

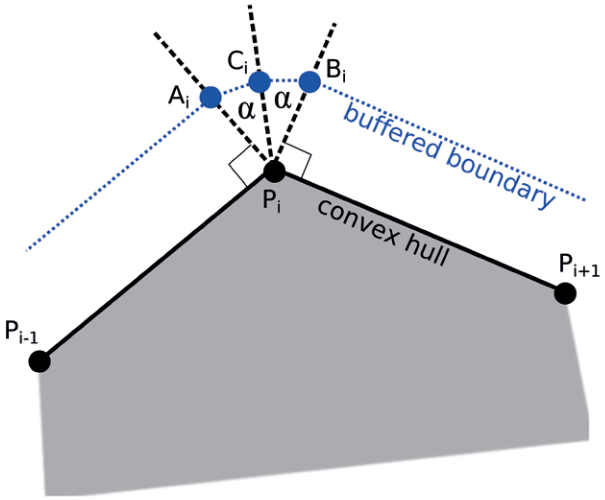

Fig. 14 - Recursive method for convex hull buffering.

The polygonal A_i C_i B_i is an approximation of the circular arc at constant distance, but the method can be improved recursively by finding the bisectors of angles α, then the bisectors of the bisectors, *ad libitum* (see Fig. 15).

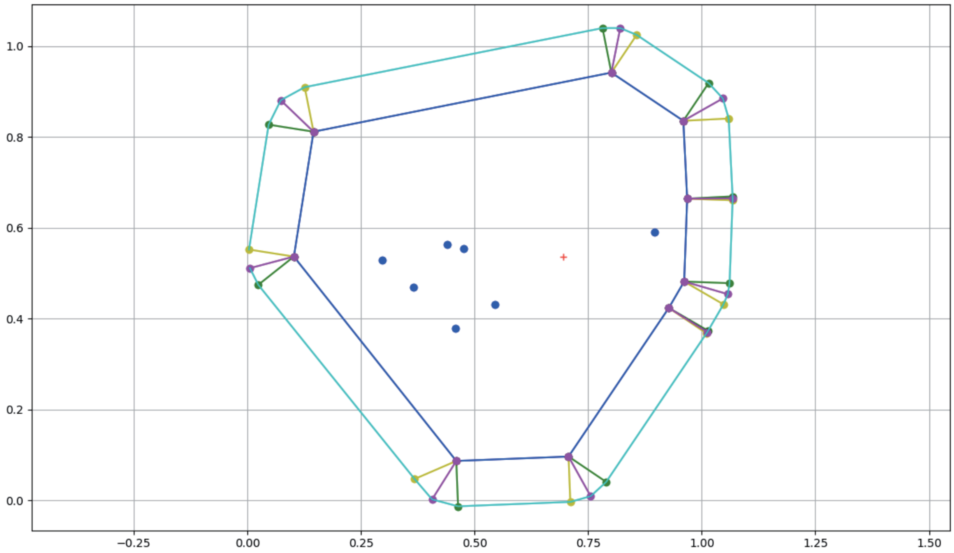

Fig. 15 - Example of convex hull buffering.

Eventually, two Python libraries were implemented, in order to provide all necessary tools. They rely only on NumPy and SciPy.

7. Results

Two different test data sets were taken into account: the FF (Franke's function) and the DTM at 10 m.

The FF was evaluated on 100×100 points with x and y between 0.0 and 1.0. A random subset of 100 points (1%) was interpolated by using IDW3 (IDW with exponent p = 3.0), IDW4 (exponent p = 4.0), and NNI.

The DTM was evaluated on 383×386 points, which correspond to a rectangular area. The original data in latitude and longitude WGS84 (EPSG:4326) were converted to WGS84 Pseudo-Mercator projection (EPSG:3857, in metres). A random subset of 1000 points (0.67%) was interpolated, by using IDW3, IDW4, and NNI. For each test set we compared all the interpolated values (u_i) with the original ones (v_i), evaluating errors and bivariate statistics between original and interpolated values. The analysis yielded histogram plot, mean, variance, RMSE (root mean square error) and skewness coefficient of the errors, quartiles Q1, Q2 (median), Q3, Q3-Q1 interquartile distance, minimum and maximum of the absolute errors, scatter plot, Pearson's correlation coefficient, moment of inertia and linear regression (slope and offset) for the pairs (v_i, u_i) (e.g. Isaaks and Srivastava, 1989). Tables 1 and 2 show the numerical results. Figs. 16 and 17 show histograms and the scatterplots with superimposed lines of linear regression.

Table 1 - Results with Franke's test function.

	IDW exp=3	IDW exp=4	NNI
errors			
mean	0.004977	0.003617	-3.326e-05
variance	0.001914	0.001762	0.0006603
RMSE	0.04404	0.04213	0.0257
skewness coefficient	2268	9682	1.724e+04
absolute errors			
quartile Q1	0.006653	0.006837	0.002672
median Q2	0.01912	0.01859	0.008087
quartile Q3	0.04611	0.04069	0.02203
Q3 - Q1	0.03945	0.03385	0.01935
min	0	0	0
max	0.1826	0.2209	0.1226
bivariate statistics			
Pearson's correlation coefficient	0.9886	0.9893	0.9963
moment of inertia	0.09696	0.08873	0.03301
linear regression offset	0.024	0.01071	0.01359
linear regression slope	0.9531	0.9825	0.9664

Table 2 - Results with DTM test.

	IDW exp=3	IDW exp=4	NNI
errors [m]			
mean	0.9094	0.4536	0.2558
variance	167.3	109	48.91
RMSE	12.97	10.45	6.998
skewness coefficient	-0.000111	9.325e-05	0.0004675
absolute errors [m]			
quartile Q1	1.89	1.641	0.832
median Q2	5.335	4.463	2.446
quartile Q3	12.1	10.05	6.0550
Q3 - Q1	10.21	8.408	5.223
min	0	0	0
max	71.69	55.48	44.72
bivariate statistics			
Pearson's correlation coefficient	0.9956	0.9969	0.9986
moment of inertia	3.245e+04	2.107e+04	9451
linear regression offset	9.287	3.557	2.558
linear regression slope	0.9606	0.9854	0.9892

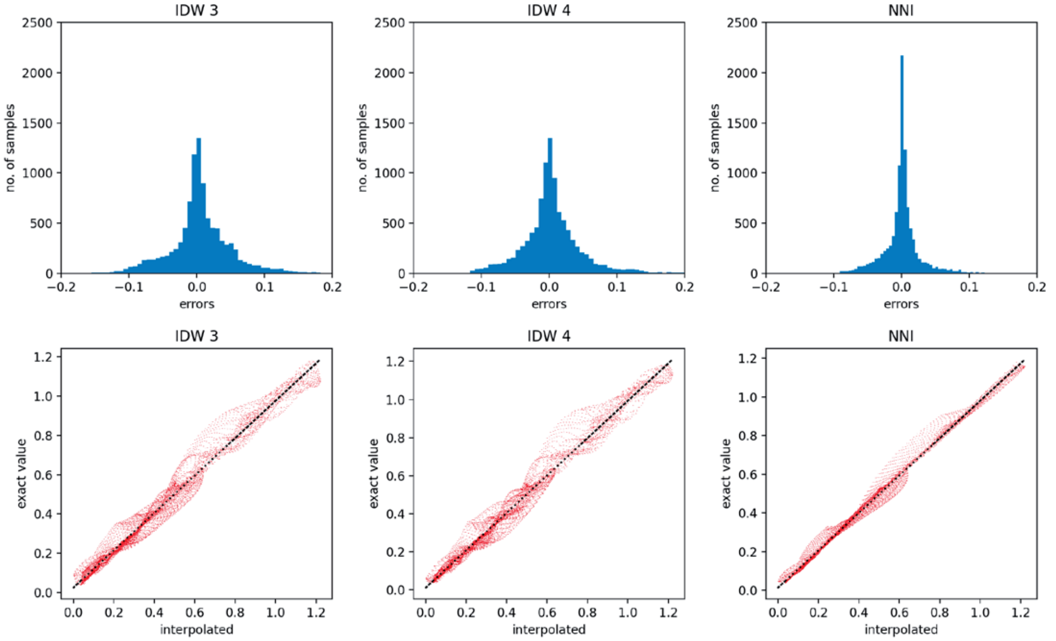

Fig. 16 - Error histograms and scatterplots of Franke’s function test.

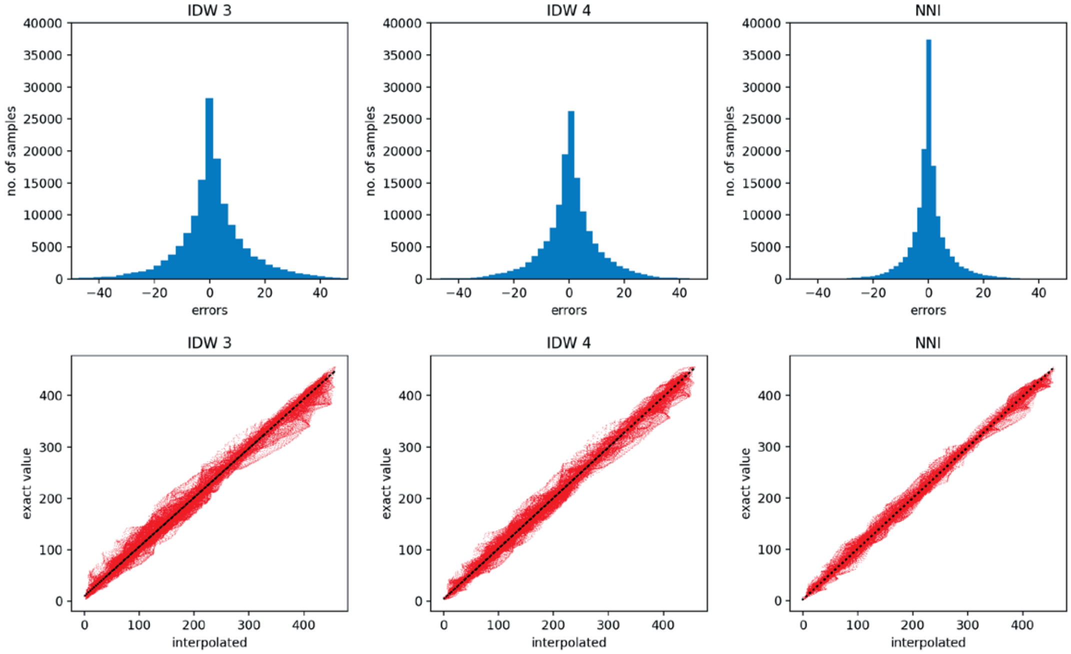

Fig. 17 - Error histograms and scatterplots of DTM test.

The interpolated surfaces are displayed in Fig. 18 (IDW) and Fig. 19 (NNI), after colouring and contouring.

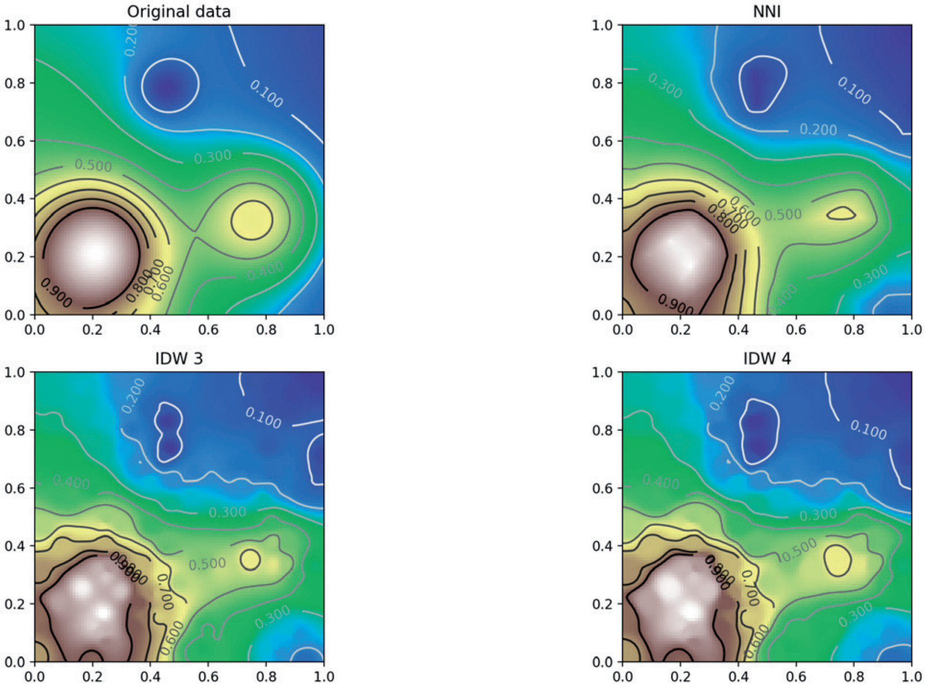

Fig. 18 - Contour plots of Franke’s function test.

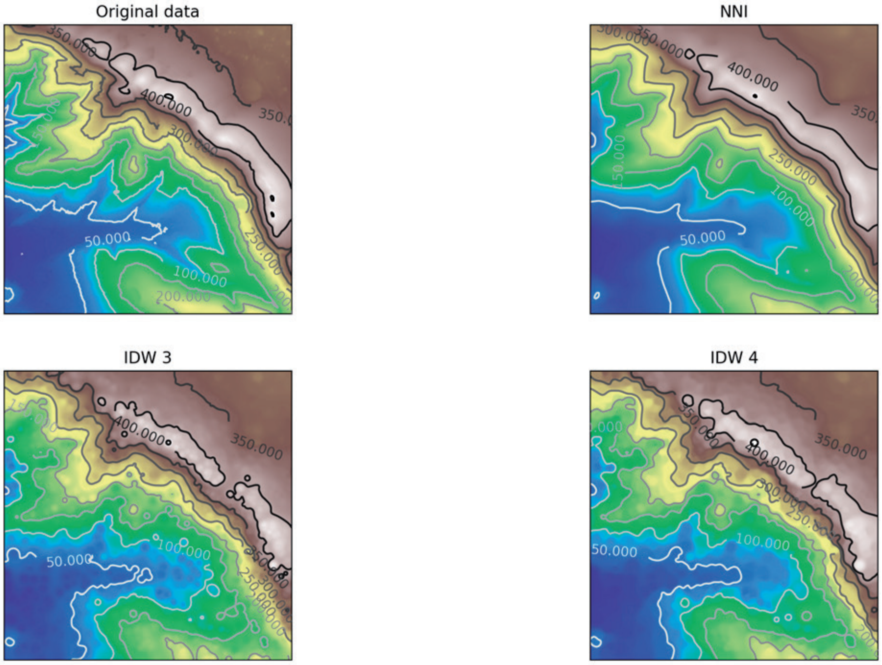

Fig. 19 - Contour plots of DTM test.

A hillshade plot for the DTM test is represented in Fig. 20. IDW interpolations present some recognisable graphic artifacts, the circular patterns around the samples, well known in literature as “bull’s eye effect”, due to the radial characteristic of IDW equation. The phenomenon is well illustrated in Fig. 21, presenting a zoomed detail.

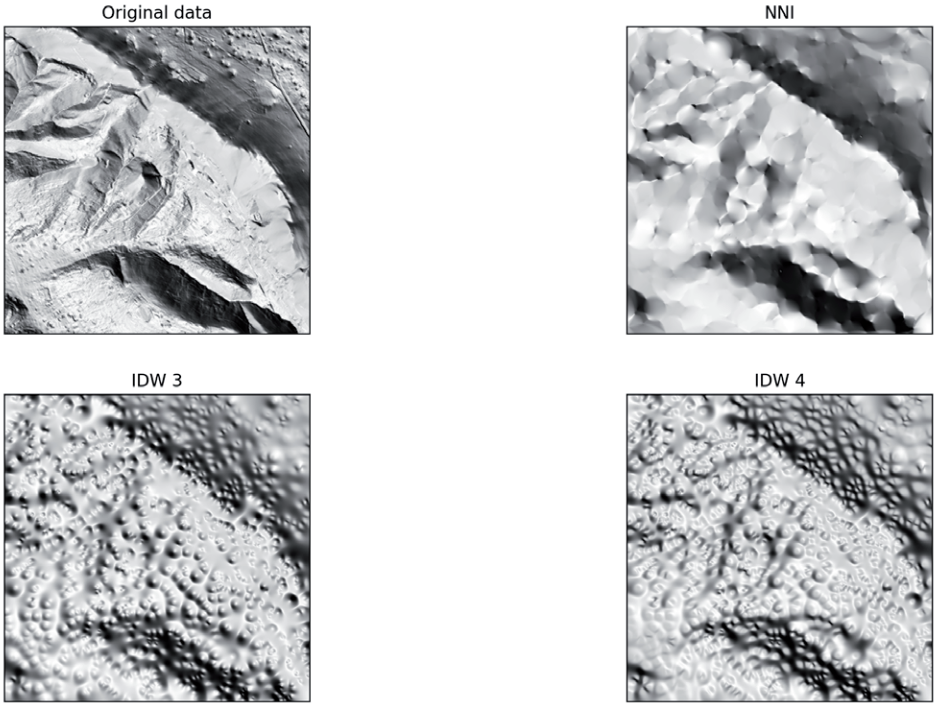

Fig. 20 - Hillshade plot of DTM test.

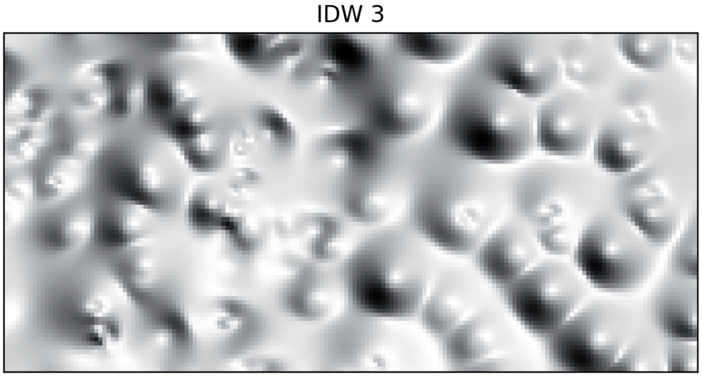

Fig. 21 - Hillshade detail of DTM test (IDW3).

We also evaluated a specific computational aspect of the interpolation algorithms. We tested the IDW algorithm, the NNI (using SciPy for the incremental Voronoi decomposition) and the optimised version of NNI algorithm, in order to assess their speed, assumed the algorithms operate in the same conditions (same hardware, same software environment, same programming language, and same data sets).

Fig. 22 shows the results for a comparison test between IDW algorithm (with exponent = 3), basic NNI, and optimised NNI (implemented with our algorithm, previously described).

The number of considered data samples was always 120. The tests were executed on a MacBook Air (1.6 GHz i5, 8GB RAM and SSD).

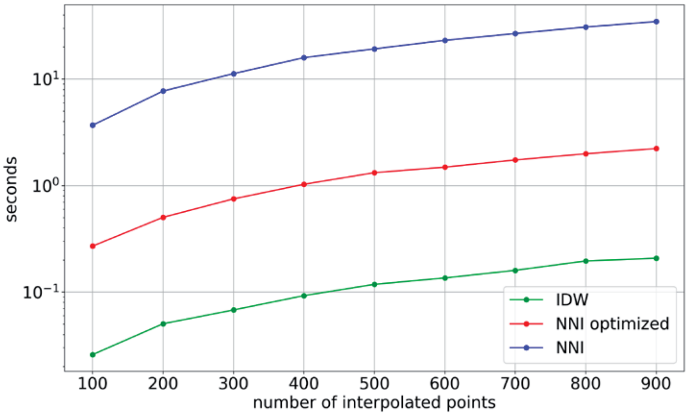

Fig. 22 - Speed comparisons.

8. Implementation

The last testbed is the practical implementation of a near real-time system collecting environmental data, processing and mapping them on maps through an intuitive web interface.

The web interface in Fig. 23 displays the aggregated data after a selection query. Datacubes are relative to an interval of one hour and PM10 measures; in each datacube the median of the samples and other statistical information are calculated. The cells are coloured according to a configuration file, as also reported in the map legend. The reference grid layer is shown as an example. The map is interactive, allowing the user to click and explore every single cell. All data are updated in near real-time, every 10 minutes.

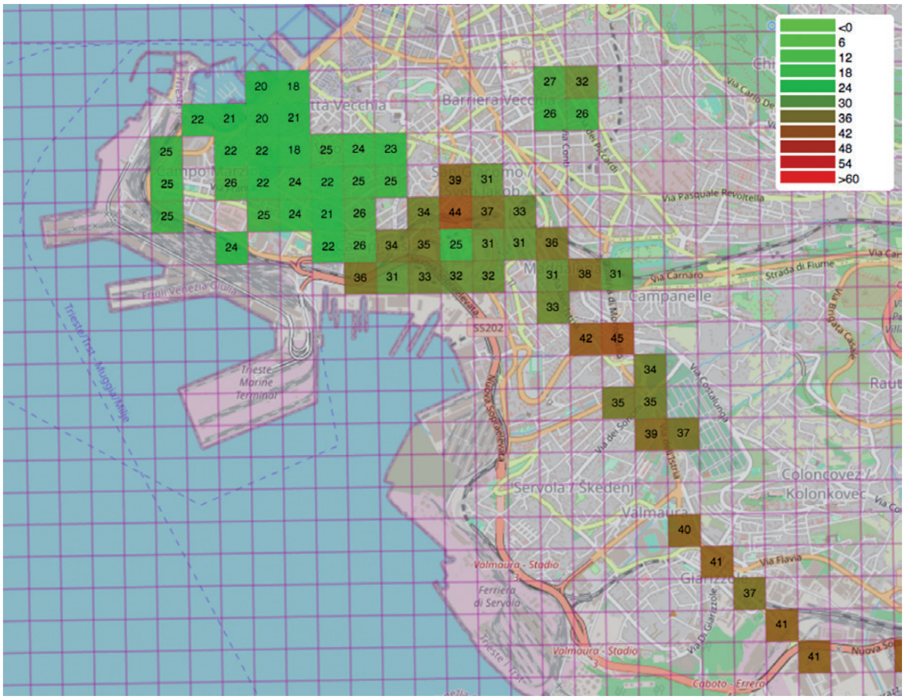

Fig. 23 - Web presentation of PM10 averaged values.

The result of interpolation over the same data set is shown in Fig. 24. IDW was chosen as an interpolation method, because NNI was too slow for our specific software and hardware environment.

PNG transparent files generated by the Python scripts, along with a small TXT file, which contains all georeferencing information, previously stored on the server, are requested through HTTPS. The web interface displays the image as an additional layer, showing interpolation and contour. The image is clipped with a transparency mask along the computed boundary (the buffering). The parameters for linear colour maps are defined in a configuration file, for every represented physical quantity. The colour maps are used to define cell colours (discrete view) and interpolated maps (continuous view).

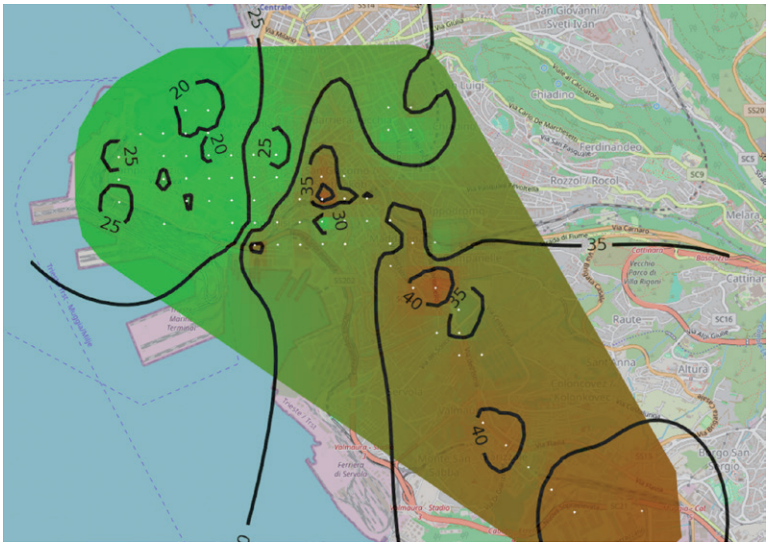

Fig. 24 - Web presentation of interpolated PM10 values.

9. Discussion

The numerical results of Tables 1 and 2 highlight better performance for NNI method in comparison to IDW method and a slight improvement for IDW method with a higher exponent (4 compared to 3). Only the analysis with linear regression in the case of Franke’s test function exhibits some better results of IDW in respect to NNI (slope is closer to 1.0 and offset to 0.0).

In addition, the IDW method tends to produce some artifacts, like bull’s eye effect. Anyway, the IDW method is still a valid choice, especially if we need a very fast algorithm. Speed comparisons display a significant improvement of speed, of one order of magnitude, of IDW compared to NNI, which is a relevant asset in a near real-time environment. IDW can potentially be improved by more advanced techniques, like search neighbourhood criteria. Both interpolation methods being deterministic, do not need any human intervention or sophisticated auto-adaptive algorithms, which is a desirable aspect for a fully automated system. The choice of the interpolation method eventually depends on the computational resources, which are available. If the processing speed is sufficient in order to ensure a near real-time response (or even better, if parallelisation is available, e.g. by means of GPU), the NNI method would be a better choice.

Schneider (2017) is a valid example of further improvement by means of data assimilation methods or, more specifically, data fusion methods. A modelled distribution can be used to adjust the observed distribution (in this case, concentration of NO_2). Data assimilation is one of the major fields of interest for a future enhancement of our research.

For a better definition of a boundary where the interpolated values are most significant and accurate or for a proxy of uncertainty, further improvements can be made by considering the prediction variance and similar geostatistical parameters.

10. Conclusions

We considered the differences between several approaches to the interpolation of bidimensional data in order to implement an automatic method based only on an overall assumption of data weighting upon distance that aims at removing researchers' bias and arbitrariness from the process.

We analysed the issue of data gridding and a first approximation approach to the problem of boundary, by presenting a simple method to define the "buffering" of the convex hull of measured data.

Subsequently, we chose two deterministic interpolation methods, IDW and NNI, excluding the Kriging method for its implementation complexity. The former is totally parameter-free, whereas the latter requires at least the choice of an exponent. We presented a comparison between the methods, based on the use of an analytical test function (the Franke's function) and the use of DTM data. After considering the linear dependence of atmospheric pressure on altitude, at least close to the sea level, we noticed the strong relation between the digital terrain model and some environmental measures. Thus a test with DTM possibly offers better adherence to a "real world" scenario than an abstract mathematical model.

The errors were evaluated through Pearson's correlation index, RMSE and error distribution around zero. The results highlighted a moderate improvement for NNI, which has a better error assessment.

Nonetheless, speed tests on point sets of increasing size showed that NNI is roughly one hundred times slower than simpler IDW. Then, we presented an implementation of NNI algorithm, which offers adequate speed, ten times faster than implementing NNI with "as-is" SciPy Python libraries. Our implementation is optimised for the recalculation of the Voronoi area relative to every point of interpolation.

We may infer that IDW (as first approximation) or NNI (an optimal solution, which is however demanding more computational power and speed, approximately one order of magnitude) allow us to implement near real-time interpolation of spatial environmental data in a deterministic way, without needing of parameter selection by skilled users or AI algorithms.

We may infer that IDW is a valuable simple alternative to more complex methods like NNI, in case of near real-time computing. The high speed (10 times faster than NNI in our implementation), the ease of coding and the more than discrete results in terms of relative errors provide a favourable assessment for this algorithm.

Both IDW and NNI algorithms, like most interpolation methods, have a useful property: the computations in different points are totally independent. This implies the opportunity of a trivial parallelism, which can be easily implemented using multiprocessing or a GPU environment like CUDA/NVIDIA.

Finally, we showed a successful example of near real-time interpolation of environmental data, based on all the aforementioned considerations. Both gridded data (with rectangular cells) and IDW interpolated data (with contour lines, labelling, proper colour maps and boundary defined by a buffered convex hull) were generated on-the-fly on a server. The results were displayed in a web page, dynamically interacting with the user.

The method we developed proves to be effective and prevents further introductions of bias in the data. However, its products cannot be considered totally bias-free, given the deterministic weighting scheme of NNI and the possible presence of biases introduced in the preceding phases, potentially stacked ever since the very first phase of the experimental design.

REFERENCES

Baise A.; 2020: *The objective–subjective dichotomy and its use in describing probability.* Interdisciplin. Sci. Rev., 45, 174-185, doi: 10.1080/03080188.2019.1705559.

Barca E., Porcu E., Bruno D. and Passarella G.; 2017: *An automated decision support system for aided assessment of variogram models.* Environmen. Modell. Softw., 87, 72-83, doi: 10.1016/j.envsoft.2016.11.004.

Barnes S.L.; 1964: *A technique for maximizing details in numerical weather-map analysis*. J. Appl. Meteorol., 3, 396-409.

Bater C.W. and Coops N.C.; 2009: *Evaluating error associated with lidar-derived DEM interpolation*. Comput. Geosci., 35, 289-300, doi: 10.1016/j.cageo.2008.09.001.

Birch C.P.D., Oom S.P. and Beecham J.A.; 2007: *Rectangular and hexagonal grids used for observation, experiment and simulation in ecology.* Ecol. Modell., 206, 347-359, doi: 10.1016/j.ecolmodel.2007.03.041.

Boer E.P., de Beursl K.M. and Hartkamp A.D.; 2001: *Kriging and thin plate splines for mapping climate variables*. Int. J. Appl. Earth Obs. Geoinf., 3, 146-154, doi: 10.1016/S0303-2434(01)85006-6.

Chilès J.P. and Delfiner P.; 1999: *Geostatistics*: *modeling spatial uncertainty, 2nd ed.* Wiley & Sons, Hoboken, NJ, USA, 726 pp., ISBN: 978-0-470-18315-1.

Conn P.B., Johnson D.S. and Boveng P.L.; 2015: *On extrapolating past the range of observed data when making statistical predictions in ecology.* PLoS ONE, 10, e0141416, doi: 10.1371/journal.pone.0141416.

Dhingra S., Madda R.B., Gandomi A.H., Patan R. and Daneshmand M.; 2019: *Internet of things mobile-air pollution monitoring system (IoT-Mobair)*. IEEE Internet of Things Journal, 6, 5577-5584, doi: 10.1109/JIOT.2019.2903821.

Diviacco P.; 2012: *Addressing conflicting cognitive models in collaborative e-research: a case study in exploration geophysics.* In: Collaborative and Distributed E-Research, Innovations in Technologies, Strategies and Applications, IGI Global press, Hershey, PA, USA, pp. 247-275, doi: 10.4018/978-1-4666-0125-3.ch012.

Diviacco P.; 2014: *Reconciling knowledge and collaborative e-research*. In: Diviacco P., Fox P., Pshenichny C. and Leadbetter A. (eds), Collaborative Knowledge in Scientific Research Networks, IGI Global press, Hershey, PA, USA, pp. 1-20, doi: 10.4018/978-1-4666-6567-5.ch001.

Duangsuwan S., Takarn A. and Jamjareegulgarn P.; 2018: *A development on air pollution detection sensors based on NB-IoT network for smart cities*. In: Proc. 18th International Symposium on Communications and Information Technologies (ISCIT), Bangkok, Thailand, pp. 313-317, doi: 10.1109/ISCIT.2018.8587978.

Fastellini G. and Schillaci C.; 2020: *Precision farming and IoT case studies across the world*. In: Castrignanò A., Buttafuoco G., Khosla R., Mouazen A., Moshou D. and Naud O. (eds), Agricultural Internet of Things and Decision Support for Precision Smart Farming, Academic Press, Cambridge, MA, USA, pp. 331-415, doi: 10.1016/B978-0-12-818373-1.00007-X.

Foster M.P. and Evans A.N.; 2008: *Performance evaluation of multivariate interpolation methods for scattered data in geoscience applications.* In: Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Boston, MA, USA, pp. IV565-IV568, doi: 10.1109/IGARSS.2008.4779784.

Franke R.; 1979: *A critical comparison of some methods for interpolation of scattered data.* Naval Postgraduate School, Monterey, CA, USA, Technical Report, NPS-53-79-003, <hdl.handle.net/10945/35052>.

Goovaerts P.; 1997: *Geostatististics for natural resources evaluation*. Applied Geostatistical Series, Oxford University Press, Oxford, UK, 496 pp., ISBN: 9780195115383.

Hanson N.R.; 1958: *Patterns of discovery: an inuiry into the conceptual foundations of science*. Cambridge University Press, Cambridge, UK, 252 pp., ISBN: 9780521051972.

Hutchinson M.F.; 1998: *Interpolation of rainfall data with thin plate smoothing splines*. J. Geog. Inf. Decis. Anal., 2, 139-151.

Isaaks E.H. and Srivastava R.M.; 1989: *An introduction to applied geostatistics*. Oxford University Press, New York, NY, USA, 561 pp., doi: 10.1016/0098-3004(91)90055-I.

Karskens M.; 1992: *The development of the opposition subjective versus objective in the 18th century.* Archiv für Begriffsgeschichte, 35, 214-256, <www.jstor.org/stable/24363024>.

Latour B. and Woolgar S.; 1986: *Laboratory life: the construction of scientific facts*. Princeton University Press, Princeton, NJ, USA, 296 pp.

MacPherson F.; 2012: *Cognitive penetration of colour experience: rethinking the issue in light of an indirect mechanism*. Philos. Phenomen. Res., 84, 24-62, doi: 10.1111/j.1933-1592.2010.00481.x

Okabe A., Boots B., Sugihara K. and Chiu S.N.; 2000: *Spatial tessellations: concepts and applications of Voronoi diagrams, 2nd ed.* John Wiley & Sons Ltd., Hoboken, NJ, USA, 696 pp., doi: 10.1002/9780470317013.

Parmar G., Lakhani S. and Chattopadhyay M.K.; 2017: *An IoT based low cost air pollution monitoring system*. In Proc. International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), Bhopal, India, pp. 524-528, doi: 10.1109/RISE.2017.8378212.

Pebesma E., Cornford D., Dubois G., Heuvelink G.B.M., Hristopulos D., Pilz J., Stöhlker U., Morin G. and Skøien J.O.; 2011: *INTAMAP: the design and implementation of an interoperable automated interpolation web service*. Comput. Geosci., 37, 343-352.

Schneider P., Castell N., Vogt M., Dauge F.R., Lahoz W.A. and Bartonova A.; 2017: *Mapping urban air quality in near real-time using observations from low-cost sensors and model information*. Environ. Int., 106, 234-247, doi: 10.1016/j.envint.2017.05.005.

Shepard D.; 1968: *A two-dimensional interpolation function for irregularly-spaced data.* In: Proc. 23rd ACM National Conference, New York, NY, USA, pp. 517-524, doi: 10.1145/800186.810616.

Sibson R.; 1980: *A vector identity for Dirichlet tessellation*. Math. Proc. Cambridge Philos. Soc., 87, 151-155.

Spade P.; 1994: *Five texts on the Mediaeval problem of universals: Porphyry, Boethius, Abelard, Duns Scotus, Ockham*. Hackett Publishing Co., Indianapolis, IN, USA, 320 pp.

Tyagi A. and Singh P.; 2013: *Applying Kriging approach on pollution data using GIS software.* Int. J. Environ. Eng. Manage., 4, 185-190.

Wackernagel H.; 1998: *Multivariate geostatistics*: *an introduction with applications, 2nd.* Springer-Verlag, Berlin-Heidelberg, Germany, 307 pp., doi: 10.1007/978-3-662-03550-4.

Wessel P. and Bercovici D.; 1976: *Interpolation with spline in tension: a Green's function approach*. Math. Geol., 30, 77-93.

Wessel P., Luis J.F., Uieda L., Scharroo R., Wobbe F., Smith W.H.F. and Tian D.; 2019: *The generic mapping tools version 6*. Geochem. Geophys. Geosyst., 20, 5556-5564, doi: 10.1029/2019GC008515.

Whitley R.; 2000: *The intellectual and social organization of the sciences, 2nd ed*. Oxford University Press Inc., New York, NY, USA, 106 pp.

Worden C.B., Wald D.J, Allen T.I., Lin K., Garcia D. and Cua G.; 2010: *A revised ground-motion and intensity interpolation scheme for ShakeMap*. Bull. Seismol. Soc. Am., 100, 3083-3096.

Zeimbekis J.; 2013: *Color and cognitive penetrability*. Philos. Stud., 165, 167-175, doi: 10.1007/s11098-012-9928-1.

Corresponding author: Massimiliano Iurcev
Istituto Nazionale di Oceanografia e di Geofisica Sperimentale
Borgo Grotta Gigante 42c, 34010 Sgonico (TS), Italy
Phone: +39 040 2140316; e-mail: miurcev@inogs.it