A Directional Gradient-Curvature Method for Gap Filling of Gridded Environmental Spatial Data with Potentially Anisotropic Correlations

Milan Žukovič\textsuperscript{a,b}, Dionissios T. Hristopoulos\textsuperscript{c,*}

\textsuperscript{a}Institute of Physics, Faculty of Science, Pavol Jozef Šafárik University, Park Angelinum 9, 040 01 Košice, Slovakia
\textsuperscript{b}SORS Research a.s., Mojzesova 38, 040 01 Košice, Slovakia
\textsuperscript{c}Geostatistics Research Unit, Department of Mineral Resources Engineering, Technical University of Crete, Chania 73100, Greece

Abstract

We introduce the Directional Gradient-Curvature (DGC) method, a novel approach for filling gaps in gridded environmental data. DGC is based on an objective function that measures the distance between the directionally segregated normalized squared gradient and curvature energies of the sample and entire domain data. DGC employs data-conditioned simulations, which sample the local minima configuration space of the objective function instead of the full conditional probability density function. Anisotropy and non-stationarity can be captured by the local constraints and the direction-dependent global constraints. DGC is computationally efficient and requires minimal user input, making it suitable for automated processing of large (e.g., remotely sensed) spatial data sets. Various effects are investigated on

*Corresponding author.

\textit{Email addresses:} milan.zukovic@upjs.sk (Milan Žukovič), dionisi@mred.tuc.gr (Dionissios T. Hristopoulos)

\textit{URL:} http://www.mred.tuc.gr/home/hristopoulos/dionisi.htm (Dionissios T. Hristopoulos)

\textit{Preprint submitted to Atmospheric Environment}
synthetic data. The gap-filling performance of DGC is assessed in comparison with established classification and interpolation methods using synthetic and real satellite data, including a skewed distribution of daily column ozone values. It is shown that DGC is competitive in terms of cross validation performance.

Keywords: correlation, anisotropy, spatial interpolation, stochastic estimation, optimization, simulation

1. Introduction

Atmospheric data, whether they are obtained by means of ground or remote sensing methods, often include data gaps. Such gaps arise due to different reasons, e.g. incomplete time series, spatial irregularities of sampling pattern, equipment limitations or sensor malfunctions (Jun & Stein, 2004; Lehman et al., 2004; Albert et al., 2012; Bechle et al., 2013). For example, remote sensing images may include obscured areas due to cloud cover, whereas gaps also appear between satellite paths where there is no coverage for a specific period (Emili et al., 2011). The impact of missing data on the estimate of statistical averages and trends can be significant (Sickles & Shadwick, 2007). There is an interest in the development of new methods for filling gaps in atmospheric data and their comparison with existing imputation methods (Junninen et al., 2004). Particularly for frequently collected, massive remotely sensed data, the efficient filling of the gaps is a challenging task. Traditional geostatistical interpolation methods such as kriging, e.g. (Wackernagel, 2003), can be impractical due to high computational complexity, restriction to Gaussian data, as well as var-
ious subjective choices in variogram modeling and interpolation search radius (Diggle & Ribeiro, 2007). In particular, computationally efficient methods are needed for filling gaps in very large data sets (Cressie, 2008; Hartman & Hössjer, 2008).

In the following, we consider a set of sampling points $G_s = \{\vec{s}_i, i = 1, \ldots, N\}$, where $\vec{s}_i = (x_i, y_i) \in \mathbb{R}^2$. The points are scattered on a rectangular grid $\tilde{G}$ of size $N_G = L_x \times L_y$, where $L_x$ and $L_y$ are respectively the horizontal and vertical dimensions of the rectangle (in terms of the unit length), such that $N_G > N$. Let $G_p = \{\vec{s}_p, p = 1, \ldots, P\}$ be the set of prediction points, representing locations of missing values, such that $\tilde{G} = G_s \cup G_p$.

The data, $Z(G_s) = \{z_i, \forall \vec{s}_i \in G_s\}$, are considered as a realization of the continuous random field $Z(\vec{s}_i)$. To reduce the dimensionality of the configuration space, we discretize the continuously valued field. For applications that do not require high resolution, e.g., environmental monitoring and risk management, $Z(G_s)$ can be discretized into a small (e.g., eight) number $N_c$ of levels (classes). Continuous distributions are obtained at the limit $N_c \to \infty$. In the current study, the spatial prediction of missing values is posed as a spatial classification problem for ranked numerical data. Continuous interpolation is approximated by considering an arbitrarily high number of levels.

The discretization classes $C_q, q = 1, \ldots, N_c$ correspond to the intervals $C_q = (t_q, t_{q+1}]$ for $q = 2, \ldots, N_c - 1$, $C_1 = (-\infty, t_2]$, and $C_{N_c} = (t_{N_c}, \infty)$. The classes are defined with respect to threshold levels $t_k, k = 2, \ldots, N_c$. All the classes have a uniform width except $C_1$ and $C_{N_c}$ which extend to negative and positive infinity respectively, to include values outside the observed interval $[z_{\min}, z_{\max}]$. More general class definitions can be investigated. The
class identity field $I(\vec{s})$ takes integer values $q = 1, \ldots, N_c$ that represent the respective class index. In particular, $I(\vec{s}_i) = q$ implies that $z_i \in C_q$. The prediction problem is equivalent to assigning a class label at each point in $G_p$. A map of the process $Z$ can be generated consisting of equivalent-class (isolevel) contours.

2. The Directional Gradient-Curvature Model

The Directional Gradient-Curvature (DGC) model is inspired by Spartan spatial random fields (SSRF) (Hristopulos, 2003), which are based on short-range interactions between the field values. The SSRF model is parametric and represents stationary, continuous and isotropic Gaussian random fields. To relax these assumptions, we introduce an almost non-parametric approach that aims at matching short-range correlations in $G_s$ with those of $\tilde{G}$. This idea was recently successfully applied to spatial random fields still assuming spatial isotropy (Zukovič & Hristopulos, 2012). The present model extends this approach by relaxing even the isotropic assumption through incorporating anisotropic dependence. In particular, the correlations used in DGC represent the normalized squared gradient and curvature energies of the discretized class identity field along different directions. Let $a_n$ be the lattice step in direction $\vec{e}_n$. The local square gradient and curvature terms in each of the $d$ directions used are given by

$$G_n(I; \vec{s}_i) = \frac{[I(\vec{s}_i + a_n\vec{e}_n) - I(\vec{s}_i)]^2}{a_n^2}, \quad n = 1, \ldots, d$$ (1)
The normalized squared gradient, $C_n(I_g)$, and curvature, $C_n(I_s)$ energies in a direction $e_n$, are defined as averages of the above over the grid $\tilde{G}$. $I_g = [I(s_1)\ldots I(s_{NG})] \forall s_i \in \tilde{G}$, represents the values of the class identity field on the entire grid, $I_p = [I(s_1)\ldots I(s_{Ng})] \forall s_i \in G_p$, is the class field at the prediction sites, and $I_s = [I(s_1)\ldots I(s_{Ng})] \forall s_i \in G_s$ are the class identity values at the sampling sites. The matching of the gradient and curvature constraints on $G_s$ and $\tilde{G}$ is based on the following objective functional:

$$U(I_p|I_s) = \sum_{n=1}^{d} \left[ w_1 \phi(C_n(I_g), C_n(I_s)) + w_2 \phi(C_n(I_g), C_n(I_s)) \right], \quad (3)$$

$$\phi(x, x') = \begin{cases} 
(1 - x/x')^2, & x' \neq 0 \\
x^2, & x' = 0.
\end{cases} \quad (4)$$

In the above, $w = (w_1, w_2)$ represent gradient and curvature weights ($w_1, w_2 \geq 0, w_1 + w_2 = 1$), and $d$ is the number of the directions used. We use (4) to measure the deviation between the $G_s$- and $\tilde{G}$-based values instead of $\phi = (x - x')^2$, because the gradient and curvature can have very different magnitudes, depending on the units used; this means that one term may dominate in the optimization. By using normalized constraints we compensate for the possible disparity of magnitudes between gradient and curvature. If one of the sample quantities is zero, the second line of (4) is
used to avoid a singular denominator.

We select \( w = (0.5, 0.5) \) and \( d = 4 \), representing four directions with the following angles (with respect to the positive x-axis): \( 0^\circ, 45^\circ, 90^\circ, \) and \( 135^\circ \).

Given the above, the classification problem is equivalent to determining the optimal configuration \( \hat{I}_p \) that corresponds to the minimum of (3):

\[
\hat{I}_p = \arg \min_{I_p} U(I_p | I_s).
\] (5)

The optimization of (3) is conducted numerically. The choice of the initial configuration is important to obtain a reliable, fast and automatic algorithm: it should prevent the optimization from getting trapped in poor local minima and minimize the relaxation path in configuration space to the equilibrium, and it should also minimize the need for user intervention. The sampling points retain their values \( I_s \). Assuming a certain degree of spatial continuity, common in geospatial data sets, the initial state of \( I_p \) is determined based on the sample states in the immediate neighborhood of the individual prediction points. The neighborhood of \( \vec{s}_p \) is determined by an \( m \times m \) stencil \( (m = 2l + 1) \) centered at \( \vec{s}_p \). Then, the initial value at a prediction point is assigned by majority rule, based on the prevailing value of its sample neighbors inside the stencil. The stencil size is chosen automatically, reflecting the local sampling density and the distribution of class identity values. Namely, it is adaptively set to the smallest size that contains a finite number of sampling points with a prevailing value. If no majority is reached up to some neighborhood size \( m_{\text{max}} \times m_{\text{max}} \), the initial value is assigned (i) randomly from the range of the labels with tie votes or (ii) from the entire range of labels \( 1, ..., N_c \), if majority is not reached due to absence of sampling points within the maximum stencil.
The above method of initial state assignment will be referred to as majority rule with adaptable stencil size (MRASS).

The updating of class identity states on $G_p$ uses the “greedy” Monte Carlo (MC) method (Papadimitriou & Steiglitz, 1982), which unconditionally accepts a new state if the latter lowers the cost function. The greedy MC algorithm may cause the termination of the DGC algorithm at local minima of the objective function [3]. Targeting exclusively global minima (e.g. by simulated annealing) unduly emphasizes exact matching of the energies on the entire domain with those in the sample domain; however, the latter are subject to sampling fluctuations and measurement errors.

The algorithm performs a random walk through the grid $G_p$. It terminates if $P$ consecutive update trials do not produce a single successful update. If the computational budget is a concern, the algorithm can terminate when a pre-specified maximum number of Monte Carlo steps is exceeded. In either case, the generated realization is accepted only if the residual value of the cost function is below a user-defined tolerance level $tol$. Otherwise, the realization is rejected and a new one is generated. The algorithm generates $M$ different realizations. The median values from all the accepted realizations at each missing-value point represent the prediction of the algorithm. Associated confidence intervals are also derived.

The main steps of the procedure described above are summarized by means of the following algorithm.

1. Define the number of realizations $M$, the number of classes $N_c$, the maximum stencil size $m_{\text{max}}$, the residual cost function tolerance $tol$ and the maximum number of Monte Carlo steps $i_{\text{max}}$ (optional).
2. Discretize $Z(G_s)$ to obtain the sample class identity field $I_s$.

3. Calculate the directional sample energies $\overline{G}_n(I_s), \overline{C}_n(I_s), n = 1, \ldots, d$.

4. Initialize the simulated realization index $j = 1$.

5. While $j \leq M$ repeat the following steps:
   - Assign initial values $\hat{I}_p^{(0)}$ to the prediction points in $G_p$ based on MRASS.
   - Calculate the initial energy values $\overline{G}_n(I_s^{(0)}), \overline{C}_n(I_s^{(0)}), n = 1, \ldots, d$, and the objective function $U^{(0)} = U(\hat{I}_p^{(0)}|I_s)$.
   - Initialize the simulated state index $i = 0$, and the rejected states index $i_r = 0$.
   - While $(i_r < P) \land (i \leq i_{\text{max}})$ repeat the following updating steps:
     - Generate a new state $\hat{I}_p^{(i+1)}$ by randomly, i.e., with probability 0.5, adding $\pm 1$ to the state $\hat{I}_p^{(i)}$, maintaining the condition $1 \leq \hat{I}^{(i+1)}(\tilde{s}_j) \leq N_c, \forall \tilde{s}_j \in G_p$.
     - Calculate $\overline{G}_n(I_g^{(i+1)}), \overline{C}_n(I_g^{(i+1)}), n = 1, \ldots, d$.
     - Calculate $U^{(i+1)} = U(\hat{I}_p^{(i+1)}|I_s)$.
     - If $U^{(i+1)} < U^{(i)}$ accept the new state $\hat{I}_p^{(i+1)}$; $i_r \rightarrow 0$; else $\hat{I}_p^{(i+1)} = \hat{I}_p^{(i)}$; $U^{(i+1)} = U^{(i)}$; $i_r \rightarrow i_r + 1$; end.
     - $i \rightarrow i + 1$;
   - If $U^{(i)} < tol$ store the realization $\hat{I}_p^{(i)}(j) = \hat{I}_p^{(i)}$; $j = j + 1$;
   - Else return to 5 (a); end.

---

1 The algorithm checks if the number of samples for calculating $\overline{G}_n(I_s), \overline{C}_n(I_s)$ is sufficient for obtaining reliable estimates.
6. Evaluate the statistics from the realizations $\hat{I}_p^*(j)$, $j = 1, \ldots, M$.

The DGC method lies between interpolation and conditional simulation. Interpolation methods provide a single optimal configuration of the missing values, e.g., kriging is based on the minimization of the mean square error. Conditional simulation based on Markov Chain Monte Carlo methods aims to sample the entire configuration space and reconstruct the joint conditional probability density function of the missing data. DGC on the other hand samples the configuration space that corresponds to local minima of the objective function. Since DGC returns multiple realizations, we can characterize it as a stochastic method. However, in DGC the sampling of the configuration space is restricted to the subspace of local minima. The afforded dimensionality reduction is responsible for the computational efficiency of the method.

3. DGC validation methodology

In this section we conduct numerical experiments, in which a portion of the data is set aside to be used for validation of the classification/interpolation algorithms tested. The performance of DGC is evaluated by calculating the misclassification rate $F^* = 1/P \sum_{\bar{s}_p \in G_p} \left[ 1 - \delta(I(\bar{s}_p), \hat{I}(\bar{s}_p)) \right]$, where $I(\bar{s}_p)$ is the true class identity value at the validation points, $\hat{I}(\bar{s}_p)$ is the classification estimate and $\delta(I, I') = 1$ if $I = I'$, $\delta(I, I') = 0$ if $I \neq I'$. The gap-filling of DGC is compared with the $k$-nearest neighbor (KNN) (Dasarathy, 1991) and fuzzy $k$-nearest neighbor (FKNN) (Keller et al., 1985) classification algorithms. We chose the $k$ values that minimize the cross validation errors to obtain the lowest achievable errors by KNN and FKNN. The KNN and
FKNN algorithms are applied using the Matlab® function \texttt{fknn} \cite{Akbas2007}.

The interpolation performance is compared with the inverse distance weighted (ID) \cite{Shepard1968}, nearest neighbors (NN), bilinear (BL), bicubic (BC), and biharmonic spline (BS) \cite{Sandwell1987} methods. For the Gaussian synthetic data we also include the ordinary kriging (OK) method \cite{Wackernagel2003}. Given the Gaussian distribution and knowledge of the covariance parameters, OK provides optimal predictions and thus also a standard for comparing DGC estimates. The NN, BL, BC and BS interpolation algorithms were implemented by means of the Matlab® function \texttt{griddata}. For ID we used the Matlab® function \texttt{fillnans} \cite{Howat2007}. Finally, for OK we used the routines available in the Matlab® library \texttt{vebyk} \cite{Sidler2009}.

Let \( \hat{Z}(\vec{s}_p) \) be the estimate of the continuous field calculated from the back transformation

\[
\hat{Z}(\vec{s}_p) = \left[ t_{I_2(\vec{s}_p)} + t_{I_2(\vec{s}_p)+1} \right] / 2, \quad p = 1, \ldots, P. \tag{6}
\]

If \( Z(\vec{s}_p) \) is the true value at \( \vec{s}_p \) the estimation error is \( \epsilon(\vec{s}_p) = Z(\vec{s}_p) - \hat{Z}(\vec{s}_p) \).

For \( N_c >> 1 \) we calculate the following prediction errors: average absolute error

\[
\text{AAE} = (1/P) \sum_{\vec{s}_p \in G_p} |\epsilon(\vec{s}_p)|, \tag{7}
\]

average relative error

\[
\text{ARE} = (1/P) \sum_{\vec{s}_p \in G_p} \epsilon(\vec{s}_p) / Z(\vec{s}_p), \tag{8}
\]
average absolute relative error

\[ \text{AARE} = \frac{1}{P} \sum_{\vec{s}_p \in G_p} |e(\vec{s}_p)|/Z(\vec{s}_p), \]  

(9)

root average squared error

\[ \text{RASE} = \sqrt{\frac{1}{P} \sum_{\vec{s}_p \in G_p} e^2(\vec{s}_p)}, \]  

(10)

and linear correlation coefficient \( R \).

If \( S \) sample configurations are considered, the mean values of the validation measures (i.e., the MAAE, MARE, MAARE, MRASE, and MR) are calculated by averaging over the sample configurations. To focus on the local performance of DGC, we use the respective “local” errors, i.e., MAE, MRE, MARE, and RMSE, in which the spatial average is replaced by the mean over predictions obtained from \( M \) different simulations. Furthermore, we record the optimization CPU time, \( T_{\text{cpu}} \), and the number of Monte Carlo steps (MCS).

The computations are performed in Matlab® programming environment on a desktop computer with 3.25 GB RAM and an Intel® Core™ 2 Quad CPU Q9650 processor with an 3 GHz clock.

4. Results

4.1. Synthetic Data

DGC performance is first studied on synthetic data sampled on regular grids. The data are simulated from the Gaussian random field \( Z \sim \)
Table 1: Mean misclassification rate $\langle F^* \rangle$ [%] and standard deviation $S_{F^*}$ for synthetic Gaussian data with anisotropic Matérn covariance obtained by the DGC, KNN and FKNN algorithms.

| Levels | $p$ [%] | $N_c = 8$ | $N_c = 16$ |
|--------|---------|-----------|-----------|
|        |         |           |           |
| Model  | DGC KNN FKNN | DGC KNN FKNN | DGC KNN FKNN | DGC KNN FKNN |
| $\langle F^* \rangle$ | 18.9 29.1 27.5 | 26.9 35.6 34.8 | 26.4 51.4 51.4 | 38.6 58.1 57.9 |
| $S_{F^*}$ | 1.6 1.4 1.4 | 2.2 1.2 1.2 | 2.1 1.6 1.5 | 2.5 1.1 1.0 |

$N(m = 50, \sigma = 10)$ with Whittle-Matérn covariance given by $c_Z(\vec{r}) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} h^\nu K_\nu(h)$, where $h = \sqrt{r_1^2/\xi_1^2 + r_2^2/\xi_2^2}$ and $\vec{r} = (r_1, r_2)$ is the lag distance between two points. $K_\nu$ is the modified Bessel function of the second kind and of order $\nu$, where $\nu = 2.5$ is the covariance smoothness parameter. The principal axes of anisotropy are aligned with the coordinate axes. The correlation length in the vertical direction is set to $\xi_2 = 2$ and in the horizontal direction to $\xi_1 = 4$. The field is sampled on a square grid $\tilde{G}$, with $N_G = 50 \times 50$ nodes using the spectral method (Drummond and Horgan, 1987). Missing data samples $Z(G_s)$ of size $N = N_G - \lceil (p/100\%) N_G \rceil$ are generated from the complete sets by randomly removing $P = \lceil (p/100\%) N_G \rceil$ values, which are used as validation points. For different degrees of thinning (typically $p = 33\%$ and $66\%$), we generate $S = 100$ different sampling configurations. The predictions at the removed points are calculated and compared with the true values.

The classification results for the synthetic data are summarized in Table 1. The misclassification rate obtained by DGC is considerably smaller than the KNN and FKNN rates in all cases, although DGC shows somewhat larger sample-to-sample fluctuations. The mean CPU time required by DGC ranges between 0.96 and 1.11 seconds and the mean number of Monte Carlo steps...
between $10^4$ and $5 \times 10^4$. The DGC interpolation performance is evaluated in Table 2 using $N_c = 1000$ classes. In terms of validation errors (smallest errors and largest $R$), for the uniformly thinned data ($p = 33\%, 66\%$) OK ranked best. As mentioned above, for Gaussian data with known covariance parameters OK is expected to give optimal predictions. The known directional correlation lengths also allowed identifying a region of influence around the prediction points, thus optimizing search neighborhoods and consequently the OK CPU time. Nevertheless, the OK CPU time was the highest. For $p = 33\%$ the DGC performance ranked second and for $p = 66\%$ it was com-

![Figure 1: DGC interpolation results for synthetic Gaussian data with anisotropic Matérn covariance based on $M = 100$ simulation runs on a single sample generated by 66\% thinning. Subfigures include (a) original field, (b) thinned sample, (c) interpolated data based on the median values from $M$ runs, (d) comparison of the empirical cdfs of the original and interpolated data, (e) spatial distribution of the 95\% confidence interval (c.i.) widths, and (f) root mean squared errors of predictions.](image-url)
parable to BS, with the other models performing worse than DGC. We note that DGC values in Tables 1 and 2 are based on $M = 1$ simulation run for each of $S = 100$ sample realizations. Increased values of $M$ (e.g. $M = 100$) only marginally improved the validation results.

To account for more realistic patterns of missing data in remote sensing, e.g. due to cloud cover, we investigate a sample realization in which a solid block of data (rectangle of $16 \times 8$ pixels) is missing (see Fig. 2). The block is deliberately chosen to include a small area with extreme values to test the ability of DGC to predict a “hotspot”. For this study, DCG performs better than the other methods. To compensate for using one sample ($S = 1$), we run $M = 100$ simulations. Therefore the DGC CPU time is considerably higher compared to the (a) and (b) cases ($M = 1$).

The dramatic increase of the OK CPU time is caused by the augmented search neighborhood necessary to span the missing data gap and the cubic dependence of kriging on the number of points in the search neighborhood. Generally, the DGC CPU time is proportional to $N_c$ and increases with $p$, reflecting the increased dimension of the configuration space and number of variables involved in the optimization. For $p = 66\%$ the optimization involves approximately $10^6$ Monte Carlo steps. As shown in Fig. 1, multiple simulation runs allow estimating the interpolation uncertainty, with respect to the subspace of configurations that correspond to local minima of the objective function (3).
Figure 2: Interpolation results for synthetic Gaussian data with an isotropic Matérn covariance. Sample data are generated by removal of a block of data in the area marked by the dashed rectangle. The DGC results are based on $M = 100$ simulation runs. Subfigures show data interpolated by (a) OK and (d) DGC, variance by (b) OK and (e) DGC, and absolute errors by (c) OK and (f) DGC.

4.2. Real Data

4.2.1. Radioactive Potassium Concentration

The first real data set represents soil concentration of radioactive potassium measured by gamma-ray spectrometry over part of Canada (Anonymous, 2008), on a grid with $N_G = 256 \times 256$ nodes extending in latitude from 56S to 57N and in longitude from −100W to −98E, with a resolution of 250 m. The data have been preprocessed to correct for background and airplane flight height. The potassium concentrations are in units of % and their summary statistics are as follows: $N_G = 65536$, $z_{\min} = 0.39$, $z_{\max} = 3.26$, $\bar{z} = 1.60$, $z_{0.50} = 1.61$, $\sigma_z = 0.52$, skewness coefficient equal to 0.10, and kurtosis coef-
Table 2: Interpolation validation measures for Gaussian data with anisotropic Matérn covariance, using (a,b) $S = 100$ samples generated by 33% and 66% random thinning, respectively, and (c) $S = 1$ sample generated by removal of a solid block of data. The DGC uses $N_c = 1000$ and the results are based on $M = 1$ simulation run in (a) and (b) and $M = 100$ simulation runs in (c).

|       | MAE    | MAE [%] | MAARE [%] | MRAE [%] | MRASE | MIR [%] (a) | MIR [%] (b) | MIR [%] (c) | $\langle T_{cpu} \rangle$ |
|-------|--------|---------|-----------|----------|-------|------------|------------|------------|------------------|
| DGC   | 0.17   | 0.60    | 1.28      | -0.01    | -0.00 | 1.76       | 0.35       | 1.04       | 2.45             |
| NN    | 2.08   | 2.09    | 5.01      | -0.19    | -0.17 | 8.12       | 4.25       | 2.62       | 2.65             |
| BL    | 0.63   | 1.01    | 4.95      | -0.12    | -0.21 | 6.49       | 1.29       | 2.10       | 9.41             |
| BC    | 0.43   | 0.78    | 4.92      | -0.06    | -0.12 | 6.82       | 0.89       | 1.61       | 9.32             |
| BS    | 0.32   | 0.55    | 5.09      | -0.02    | -0.06 | 9.19       | 0.65       | 1.13       | 9.61             |
| ID    | 1.04   | 1.44    | 5.14      | -0.29    | -0.33 | 7.33       | 2.15       | 2.96       | 9.85             |
| OK    | 0.06   | 0.31    | 2.49      | -1E-5    | -0.01 | 4.66       | 0.12       | 0.65       | 4.71             |

A plot of the data in Fig. 3(a) displays clear signs of anisotropy. Samples $Z(G_s)$ were generated from the original data by random thinning with $p = 33\%$ and 66\%.

Classification ($N_c = 8, 16$) and interpolation ($N_c = 1500$) results for $p = 33\%$ are listed in Table 3. The prediction performance of DGC is superior to other models except for the BS. The outstanding performance of the latter is likely due to the smooth spatial variation of the radioactivity data. An example of prediction results based on $M = 100$ simulation runs for one sample realization with $p = 66\%$ is shown in Fig. 3. The DGC classification CPU time with $N_c = 8, 16$ was 2.6 and 3.3 seconds respectively. The computational time for DGC interpolation is comparable to that of BS, but one order higher than NN, BL, and BC times. The limiting factor in DGC are the MC simulations that involve up to $\sim 10^7$ Monte Carlo steps (see Fig. 4(a)). The histogram in Fig. 4(b) gives the distribution of the objective function residuals for 100 accepted configurations and verifies that all of them correspond to small ($< 2 \times 10^{-4}$) values.
Figure 3: DGC interpolation of the radioactivity data obtained from $M = 100$ simulation runs using one sample set generated by 66% thinning. The plots are as those in Fig. 1.

Table 3: Classification ($N_c = 8, 16$) and interpolation ($N_c = 1500$) results for the radioactivity data with $p = 33\%$ thinning. DGC is compared with k-nearest neighbor (KNN) and fuzzy k-nearest neighbor (FKNN) models for classifications, and with the nearest neighbor (NN), bilinear (BL), bicubic (BC), biharmonic spline (BS) and inverse distance (ID) models for interpolation.

| Classification | Interpolation | Interpolation |
|----------------|---------------|---------------|
| Model          | $N_c = 8$    | $N_c = 16$    | $N_c = 1500$ |
|                | $\langle F^* \rangle$ | $S_{F^*}$ | $\langle F^* \rangle$ | $S_{F^*}$ | MAAE | MARE [\%] | MAARE [\%] | MRASE | MR [\%] | $\langle T_{cpu} \rangle$ |
| DGC            | 4.01 0.21    | 8.64 0.28    | 9.4e−4 2.2e−3 | 6.8e−2 1.56e−3 | 100.00 | 39.94 |
| KNN            | 4.96 0.16    | 11.44 0.22   | - -       | - -       | - -       | - -       |
| FKNN           | 4.22 0.16    | 10.14 0.23   | - -       | - -       | - -       | - -       |
| NN             | - -         | - -          | 2.3e−2 0.165 | 1.64 3.5e−2 | 99.78 1.41 |
| BL             | - -         | - -          | 3.7e−3 5.9e−2 | 0.27 5.9e−3 | 99.78 1.40 |
| BC             | - -         | - -          | 1.7e−3 2.2e−2 | 0.12 2.9e−3 | 100.00 1.46 |
| BS             | - -         | - -          | 4.7e−4 1.2e−3 | 3.4e−2 7.6e−4 | 100.00 32.82 |
| ID             | - -         | - -          | 1.3e−2 0.160 | 0.90 1.7e−2 | 99.95 183.64 |
4.2.2. Ozone Layer Thickness

The second real-world data set represents daily column ozone measurements on June 1, 2007 (Acker and Leptoukh, 2007). The data are on a 1° × 1° grid with $N_G = 180 \times 360$ nodes extending in latitude from 90S to 90N and in longitude from 180W to 180E. The data set includes naturally missing (and therefore unknown) values. The data are in Dobson units with the following summary statistics: $N = 48501$, $z_{\text{min}} = 158$, $z_{\text{max}} = 596$, $\bar{z} = 311.58$, $z_{0.50} = 308$, $\sigma_z = 46.05$, skewness coefficient equal to 0.31, and kurtosis coefficient equal to 2.30. The gaps are mainly due to limited coverage on the particular day, generating conspicuous stripes of missing values in the south-north direction. Since the true values at these locations are not known, validation measures are not evaluated. Instead, the interpolation quality is assessed empirically, based on the visual continuity between the
observed data and the predictions.

In the DGC reconstructed image, as shown in Fig. 5(b), some traces of the stripe pattern due to overestimation in low-value areas (averaging effect) can still be observed. However, this effect is somewhat less pronounced than in other interpolation methods, presented in Fig. 6. Indeed, histograms of the predicted values, see Fig. 7, show a larger proportion of DGC predictions in bins below the sample average $\bar{z} = 311.58$ compared to the other methods.

Figure 5: DGC interpolation results for ozone data, obtained from $M = 100$ simulation runs on one set of the original data with missing values: Original data (a), interpolated data based on the median values from $M$ runs (b), and spatial distribution of the 95% confidence interval (c.i.) widths (c).

Figure 6: Interpolation results for ozone data, using the nearest-neighbor (NN) (a), the inverse distance (ID) (b), and the biharmonic spline (BS) (c) methods.
Figure 7: Histograms of the ozone values predicted by the respective methods: DGC, NN, ID, and BS.

5. Discussion and Conclusions

We presented and investigated the DGC method for the prediction of missing data on rectangular grids. DGC is based on stochastic simulation conditioned by sample data with a global objective function that accounts for anisotropic correlations. The constraints involve normalized directional gradient and curvature energies in specified directions. The simulation samples the configuration subspace that leads to local minima of the objective function.

For reliable application of DGC sufficiently high sampling density and number of data for the calculation of sample constraints are desirable. We evaluated the average numbers $n_p$ of nearest-neighbor sampling-point pairs per direction and $n_t$ of compact triplets of sampling points per direction. The
first, \( n_p \), is equal to the number of terms involved in \( G_n(I_s) \), while the second, \( n_t \), is the number of terms in \( C_n(I_s) \). For uniform random thinning \( n_p \) and \( n_t \) depend only on the degree of thinning \( p \) and the domain size \( L \). For \( p = 66\% \) we obtained \( (n_p, n_t) = (280, 92) \) for \( L = 50 \) and \( (n_p, n_t) = (7523, 2542) \) for \( L = 256 \) without significant differences between different directions. These values are sufficient for reliable estimates of \( G_n(I_s) \) and \( C_n(I_s) \). However, smaller grids or higher thinning degrees can lead to insufficient sampling.

Regarding sensitivity of DGC to noise, we have run tests on simulated random field realizations to which Gaussian white noise is added. DGC seems more sensitive to noise than other interpolation methods (e.g., BL, BC, BS), resulting in a larger increase of cross-validation errors with increasing noise variance. This effect is caused by the fact that methods like BL, BC, and BS perform some smoothing of the noise by means of the weighted average over extended neighbors. DGC on the other hand focuses on correlations over a small local neighborhood, which are sensitive to noise. Hence, in its current formulation DGC is more useful for smooth data distributions, such as the ones studied herein. For noisy data, improvements can be made by developing directional kernel-based estimators for the square gradient and curvature in the spirit of (Elogne & Hristopulos, 2008; Hristopulos & Elogne, 2009) or by incorporating an initial filtering stage to reduce noise (Brownrigg, 1984; Yin, 1996).

DGC does not rely on assumptions about the probability distribution of the data, it is reasonably efficient computationally, and it requires very little user input (i.e., the number of simulation runs, the number of class levels and the size of the maximum stencil for initial state selection). Potential applica-
tions include filling of data gaps in satellite images and restoration of damaged digital records. For applications in the interpolation of data sampled on an irregular grid, DGC needs to be extended to account for the lack of grid structure. This can be accomplished using kernel functions with adjustable bandwidth as shown in (Elogne & Hristopulos, 2008; Hristopulos & Elogne, 2009).

DGC shares conceptual similarities with interpolation methods based on splines, which are generated by minimizing an objective function formed by the linear combination of the square gradient and the square curvature (Wessel, 2009). A special case of the splines-based approach is the BS method used above for comparison purposes. DGC does not require minimization of the square gradient and curvature but requires matching the sample and entire-grid values of these constraints. Splines-based methods require solving a linear system involving the Green’s function of the interpolation operator; the numerical complexity of this calculation scales with the third power of the system size. DGC does not require such a costly operation, because the objective functional is defined using local couplings. Finally, in contrast with splines interpolation, DGC introduces a stochastic element by sampling the configuration subspace of local minima of the DGC objective function. On the other hand, splines-based methods can handle irregularly spaced samples, while DGC is currently restricted to grid data.

Acknowledgments

This work is funded by the European Commission, under the 6th Framework Programme, Contract N. 033811, DG INFSO, action Line IST-2005-
2.5.12 ICT for Environmental Risk Management. The views expressed herein are those of the authors and not necessarily those of the European Commission.

Ozone layer thickness data used in this paper were produced with the Giovanni online data system, developed and maintained by the NASA Goddard Earth Sciences (GES) Data and Information Services Center (DISC).

References

Acker, J.G., Leptoukh, G., 2007. Online Analysis Enhances of NASA Earth Science Data, Eos, Trans. AGU, Vol. 88, No. 2 (9 January 2007), 14-17. Online at: http://disc.sci.gsfc.nasa.gov/giovanni/

Akbas, E., 2007. Fuzzy k-NN (http://www.mathworks.com/matlabcentral/fileexchange/13358-fuzzy) MATLAB Central File Exchange. Retrieved August 10, 2008.

Albert, M.F.M.A., Schaap, M., Manders, A.M.M., Scannell, C., O’Dowd, C.D., de Leeuw, G., 2012. Uncertainties in the determination of global sub-micron marine organic matter emissions. Atmospheric Environment, 57(9), 289-300.

Anonymous, 2008. National Gamma-Ray Spectrometry Data Base, Geoscience Data Repository, Geological Survey of Canada, Earth Sciences Sector, Natural Resources Canada, Government of Canada. Online at: http://www.nrcan.gc.ca/earth-sciences/home
Bechle, M.J., Millet, D.B., Marshall, J.D. Remote sensing of exposure to NO2: satellite versus ground based measurement in a large urban area. Atmospheric Environment, Available online 13 December 2012, doi: 10.1016/j.atmosenv.2012.11.046.

Brownrigg, D.R.K., 1984. The weighted median filter. ACM Communications, 27(8), 807-818.

Cressie, N., 2008. Fixed rank kriging for very large spatial data sets. Journal of the Royal Statistical Society Series B, 70(1), 209-226.

Dasarathy, B.V. (Ed.), 1991. Nearest neighbor norms: NN pattern classification techniques. IEEE Computer Society Press, Los Alamitos, CA.

Diggle, P.J., Ribeiro, Jr., P.J., 2007. Model-based Geostatistics. Series: Springer series in statistics. Springer, New York.

Drummond, I.T., Horgan, R.R., 1987. The effective permeability of a random medium. Journal of Physics A, 20, 4661-4672.

Elogne S.N., Hristopulos D.T., 2008. Geostatistical applications of Spartan spatial random fields. In: Proceedings of the 6th International Conference on Geostatistics for Environmental Applications, Rhodes, Greece: October 2006. Eds. A. Soares and M.J. Pereira and R. Dimitrakopoulos, Vol. 15, pp. 477-488, Springer, Berlin.

Emili, E., Popp, C., Wunderle, S., Zebisch, M., Petitta, M., 2011. Mapping particulate matter in alpine regions with satellite and ground-based measurements: An exploratory study for data assimilation. Atmospheric Environment, 45(26), 4344-4353.
Hartman, L., Hössjer, O., 2008. Fast kriging of large data sets with Gaussian Markov random fields. Computational Statistics and Data Analysis, 52, 2331 - 2349.

Howat, I.M., 2007. Filling NaNs in array using inverse-distance weighting. (http://www.mathworks.com/matlabcentral/fileexchange/15590-fillnans), MATLAB Central File Exchange. Retrieved June 15, 2009.

Hristopulos, D.T., 2003. Spartan Gibbs random field models for geostatistical applications, SIAM Journal in Scientific Computation, 24, 2125-2162.

Hristopulos, D.T., Elogne, S.N., 2009. Computationally efficient spatial interpolators based on Spartan spatial random fields, SIAM Journal in Scientific Computation, 57(9), 3475-3487.

Jun, M., Stein, M.L., 2004. Statistical comparison of observed and CMAQ modeled daily sulfate levels. Atmospheric Environment, 38(27), 4427-4436.

Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J., Kolehmainen, M., 2004. Methods for imputation of missing values in air quality data sets. Atmospheric Environment, 38(18), 2895-2907.

Keller, J.M., Gray, M.R., Givens, J.A.Jr., 1985. Fuzzy k-nearest neighbor algorithm. IEEE Transactions on Systems, Man and Cybernetics, 15, 580-585.

Lehman, J., Swinton, K., Bortnick, S., Hamilton, C., Baldridge, E., Eder, B., Cox, B. 2004. Spatio-temporal characterization of tropospheric ozone across the eastern United States. Atmospheric Environment, 38(26), 4357-4369.
Papadimitriou, C.H., Steiglitz, K., 1982. *Combinatorial Optimization*. Prentice Hall, New Jersey.

Sandwell, D.T., 1987. Biharmonic Spline Interpolation of GEOS-3 and SEASAT Altimeter Data. *Geophysical Research Letters*, 14, 139-142.

Shepard, D., 1968. A two-dimensional interpolation function for irregularly-spaced data. Proceedings of the 1968 ACM National Conference, 517-524.

Sickles, J.E., Shadwick, D.S., 2007. Effects of missing seasonal data on estimates of period means of dry and wet deposition. *Atmospheric Environment*, 41(23), 4931-4939.

Sidler, R., 2009. vebyk: Value Estimation BY Kriging. (http://www.mathworks.com/matlabcentral/fileexchange/4566-vebyk), MATLAB Central File Exchange. Retrieved June 15, 2009.

Wackernagel, H., 2003. *Multivariate Geostatistics*. Springer, New York.

Wessel, P., 2009. A general-purpose Green’s function-based interpolator. *Computers & Geosciences*, 35(6), 1247-1254.

Yin, L., Yang, R.K., Gabbouj, M., Neuvo, Y., 1996. Weighted median filters: A tutorial. *IEEE Transactions on Circuits and Systems II*, 43(3), 157192.

ˇZuković, M., Hristopulos D.T., 2012. Reconstruction of missing data in remote sensing images using conditional stochastic optimization with global geometric constraints, *Stoch Environ Res Risk Assess*, Available online 4 Spetember 2012, doi: 10.1007/s00477-012-0618-5.