A Spatiotemporal Interpolation Method Using Radial Basis Functions for Geospatiotemporal Big Data

Travis Losser and Lixin Li
Department of Computer Sciences
Georgia Southern University
Statesboro, Georgia 30460–7997
Email: tlosser1@gmail.com, lili@georgiasouthern.edu

Reinhard Piltner
Department of Mathematical Sciences
Georgia Southern University
Statesboro, Georgia 30460–8093
Email: rpiltner@georgiasouthern.edu

Abstract—This research designs and implements the Radial Basis Function (RBF) spatiotemporal interpolation method to assess the trend of daily PM$_{2.5}$ concentration for the contiguous United States over the year of 2009, at both the census block group level and county level. This research also compares the performance of the RBF spatiotemporal interpolation with the Inverse Distance Weighting (IDW) spatiotemporal interpolation. Traditionally, when handling spatiotemporal interpolation, researchers tend to treat space and time separately and reduce the spatiotemporal interpolation problems to a sequence of snapshots of spatial interpolations. In this paper, PM$_{2.5}$ data interpolation is conducted in the continuous space-time domain by integrating space and time simultaneously under the assumption that spatial and temporal dimensions are equally important when interpolating a continuous changing phenomenon in the space-time domain. The RBF-based spatiotemporal interpolation methods are evaluated by leave-one-out cross validation. More importantly, this study explores computational issues (computer processing speed) faced during implementation of spatiotemporal interpolation for huge data sets. Parallel programming techniques and an advanced data structure named k-d tree are adapted in this paper to address the computational challenges.

I. INTRODUCTION

Particulate matter (PM) is the term used to describe condensed phase (solid or liquid) particles suspended in the atmosphere. A growing body of research has pointed towards the smaller particles, in particular PM less than 2.5 $\mu$m in diameter, as a metric more closely associated with adverse health effects than other metrics [1], [2], [3]. PM$_{2.5}$ is the mass concentration of airborne particles with an aerodynamic diameter of less than 2.5 $\mu$m, expressed in $\mu g/m^3$, where the volume of air is its volume at ambient conditions. The size of 2.5 $\mu$m was chosen because of its significance for the penetration of human lungs [4]. This paper focuses on monitoring the trend of daily PM$_{2.5}$ concentrations by using the deterministic RBF (Radial Basis Function) interpolation. This paper also compares the performance of the RBF interpolation method with the commonly used IDW (Inverse Distance Weighting) interpolation method.

A. Experimental PM$_{2.5}$ Data

To assess the trend of air pollution over the course of a year in the contiguous United States, daily PM$_{2.5}$ concentration is experimented in this paper. Three sets of data were used. The first dataset stores PM$_{2.5}$ measurements that were obtained from U.S. Environmental Protection Agency (EPA). It contains 146,125 PM$_{2.5}$ concentration measurements collected at 955 monitoring sites on 365 days in 2009. This dataset has the following attributes: id, year, month, day, x, y, and PM$_{2.5}$ concentration measurement, where x and y are the longitude and latitude coordinates of the monitoring sites. The locations of the monitoring sites are illustrated as red dots in Figure 1.

Fig. 1. 955 monitoring sites with PM$_{2.5}$ concentration measurements.

The second dataset contains the centroid locations of 3,109 counties, and the third dataset contains the centroid locations of 207,630 census block groups that are to be interpolated. A census block group is a geographical unit used by the United States Census Bureau that is between the census tract and the census block.

B. Literature Review on Spatial and Spatiotemporal Interpolation in GIS

Spatial interpolation is the procedure of estimating the value of properties at unsampled sites within the area covered by existing observations. Spatial interpolation methods are well developed and widely adopted in various GIS (Geographic Information System) applications [5], [6], [7]. The rationale behind spatial interpolation is the observation that points close together in space are more likely to have similar values than points far apart, which is known as Tobler’s First Law of Geography [8]. There are a number of spatial interpolation algorithms such as IDW (Inverse Distance Weighting) [9], Kriging [10], shape functions [11], spline [12], and trend surface [13].

A popular GIS software tool that is used in industry for performing spatial interpolations is ArcGIS Spatial Analyst [14]. Many studies used ArcGIS to evaluate different spatial
interpolation methods. One of these studies used ArcGIS for the comparison of spatial interpolation methods for mapping soil pH by depths [15]. The study used Kriging, IDW, and RBF to estimate the pH measurement in unsampled points and create a continuous dataset that could be represented over a map of the entire study area. The study showed that the RBF and Kriging interpolators performed best at depths A and B respectively.

There has been an increasing demand for spatiotemporal interpolation [16], [17], [18], [19] in the recent years. Spatiotemporal interpolation involves estimation of the unknown values at unsampled location-time pairs with a satisfying level of accuracy [16]. However, when applying traditional spatial interpolation methods for spatiotemporal data, such as using the spatial interpolation tools in ArcGIS, researchers face many challenges. One of the major challenges is that traditional GIS researchers tend to treat space and time separately when interpolation needs to be conducted in the continuous spatiotemporal domain. A better approach is to integrate space and time, so that time is treated as another dimension in space [20], [21], [22]. Therefore, this paper designs and implements an RBF-based spatiotemporal interpolation that uses time as another dimension in space to assess the trend of daily PM$_{2.5}$ concentrations for the contiguous United States in 2009.

II. RADIAL BASIS FUNCTION INTERPOLATION

A. Overview of Radial Basis Functions

In recent decades, the use of radial basis functions (RBFs) has gained popularity for interpolating data and for approximating solutions of partial differential equations [23], [24], [25], [26], [27], [28], [29]. In computer graphics and medical imaging, RBFs showed some advantages in modeling complicated surfaces [30], [31]. By using radial basis functions, it became possible to deal with higher dimension problems in a similar way as dealing with two- and three-dimensional problems. After choosing N points in a domain under consideration, we can easily construct N linearly independent basis functions by using distances measured from the chosen points. In 2D, the distance $r$ from a point $(x, y)$ is used: $r = \sqrt{(x - x^P)^2 + (y - y^P)^2}$. For a N-dimensional problem with points $P = (p_1, p_2, ..., p_N)$ and $Q = (q_1, q_2, ..., q_N)$, the euclidian distance is $r = \|P - Q\| = \sqrt{(P - Q)^T(P - Q)}$.

From a catalog of RBFs we first choose a type of radial basis function $\phi(r)$ and then add some low order polynomial functions in $P$ to get the interpolation function [31]:

$$f(x^P) = \sum_{q=1}^{N} \lambda_q \phi(\|x^P - x^q\|) + P(x^P).$$

For a 3D problem, the polynomial part can be chosen as $P = p^x x + p^y y + p^z z + p^l l$.

B. Radial Basis Functions

Some of the most commonly used radial basis functions include the following ($r = \|x^P - x^q\|$ and $\sigma > 0$ for each radial basis function):

1) Gaussian Function [28]:

$$\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right).$$

2) Multi-Quadric Function [25], [28]:

$$\phi(r) = (r^2 + \sigma^2)^{1/2}.$$

3) Inverse Multi-Quadric Function [28]:

$$\phi(r) = (r^2 + \sigma^2)^{-1/2}.$$

4) Thin Plate Spline Function [31], [32]:

$$\phi(r) = r^2\ln(r).$$

5) Wendland Compactly-Supported Function [31], [33]:

$$\phi(r) = \left(1 - \frac{r}{\sigma}\right)^d (4\sigma + 1).$$

The width or scaling parameter $\sigma$ is used for controlling the smoothness properties of the radial basis function [28], [32], [33]. Each radial basis function is given its own width $\sigma_j$ whose value is determined during the training process. A common choice for the width parameter is

$$\sigma_j = \frac{d_{\text{max}}}{\sqrt{2M}}$$

where $d_{\text{max}}$ is the maximum distance between points. This ensures that the individual RBFs are neither too wide, nor too narrow, for the given training data [32], [33].

Radial basis function may be used to perform interpolation of a set of $N$ data points in a multi-dimensional space. This requires that all the $D$ dimensional input vectors $x^P = \{x^P_i : i = 1, \ldots, D\}$ be mapped onto the corresponding target outputs $w^P$. The goal of RBF interpolation is to find a function $f(x)$ such that [28]

$$f(x^P) = w^P, \forall p = 1, \ldots, N.$$
One option to solve the system of equations is to make use of the Singular Value Decomposition (SVD) [28]. SVD is used to compute the pseudo-inverse, $\varphi^+$ of the matrix $\varphi$. The SVD decomposes:

$$\varphi = USV^T,$$

where $S$ is an $N \times M$ matrix with non-zero entries $s_i$ along the main diagonal. The pseudo-inverse is computed:

$$\varphi^+ = VS^+U,$$

where $S^+$ is an $M \times N$ matrix with non-zero entries $s_i^+$:

$$s_i^+ = \begin{cases} \frac{1}{s_i} & \text{for } s_i > \epsilon, \\ 0 & \text{otherwise}. \end{cases}$$

Here $\epsilon$ is a threshold that is chosen based on the round-off error of the computer being used. If one of the $s_i$’s is very small (essentially zero), then the matrix is not full rank and the SVD method ignores one of the columns.

The pseudo-inverse is then used to calculate the weights of the hidden layer using the following equation:

$$\lambda = \varphi^+ w.$$ 

Once the weights are determined, there is a function $f(x^p)$ that represents a continuous differentiable surface that passes exactly through each input data point. The function $f(x^p)$ is represented as a sum of $N$ radial basis functions, each with a different center $x_i$ and weighted by an appropriate coefficient $\lambda_i$. The function $f(x^p)$ is then used to interpolate the values of the output data points [28].

The problem with using all the input data points to train the RBF network is that it requires $O(n^2)$ storage to represent the system of linear equations. This quickly becomes prohibitive to store the square matrix $\varphi$ on any computer as the number of input data points used for training the RBF network increases [31]. When the number of $M$ basis functions equals the number of input data points $N$, the interpolation using radial basis functions is exact because the RBF mapping passes through every data point [28]. The number of $M$ basis functions need not equal the number of input data points $N$. In general it is better to have $M$ much less than $N$ to reduce the amount of storage space required for the radial basis functions. This technique produces an approximate solution rather than an exact interpolation because the number of basis function has been reduced in order to reduce the computational complexity and storage requirements [28], [31].

One approach for determining the number of $M$ radial basis functions is to find the $k$-nearest neighbors in the input dataset that are closest to the output point that is being interpolated [31]. The $k$-nearest neighbors are determined by calculating the Euclidean distance between the input point with a known measurement and the output point that is to be interpolated. The measured points that have the closest distances to the interpolation point are selected as the nearest neighbors. This approach only uses the $k$-nearest neighbors for setting up the matrix $\varphi$ to solve for the weights used in the interpolation calculation [31].

III. IMPLEMENTATION OF SPATIOTEMPORAL INTERPOLATION METHOD USING RADIAL BASIS FUNCTIONS FOR GEOSPATIOTEMPORAL BIG DATA

A. Parallel Computing Techniques

Parallel computing techniques were used for the RBF interpolation to improve the performance of interpolating the datasets of the experimental PM$_{2.5}$ data. The daily PM$_{2.5}$ concentration values in 2009 for the centroids of individual counties and census block groups need to be interpolated. This leads to $3,109 \times 365 = 1,134,785$ interpolations at the county level, and $207,630 \times 365 = 75,784,950$ interpolations at the census block group level. Since the interpolation results are very large, it is beneficial to use parallel computing techniques by splitting the work of the RBF spatiotemporal interpolations between multiple threads in order to perform multiple interpolations simultaneously.

During the interpolation, each thread takes one line of location data from either the county level dataset or the census block group level dataset, then performs the interpolation for all the possible time instances for that line of data. For the experimental PM$_{2.5}$ data, the datasets to be interpolated contain the centroid locations of individual counties or census block groups in the contiguous United States. Since the time domain is (2009, month, day), all the possible time instances are the 365 days in 2009. After a thread finishes the interpolation work for the line of the location data, it retrieves some more work from the interpolation queue. This is repeated until the interpolation queue is empty. Each thread can perform the interpolation work without conflicting with other threads because the points being interpolated are not dependent on the interpolation values for other points. Therefore, a performance advantage is gained by using threading methodology to split and share the whole interpolation work among threads.

B. K-Nearest Neighbor Search Using a K-D Tree

The most time consuming part of this RBF interpolation implementation is finding the $k$-nearest neighbors. The $k$-nearest neighbors are determined by calculating the Euclidean distance between the point with a known measurement and the point that is to be interpolated. The measured points that have the closest distances to the interpolation point are selected as the nearest neighbors.

The major advantage for finding the $k$-nearest neighbors in the RBF interpolation is the possibility of reducing the number of $M$ radial basis functions, so that the number of radial basis functions is much less than the number of input data points. Reducing the number of radial basis functions reduces the amount of storage that is required to solve the system of linear equations for the weights and also improves performance because there is less computational complexity. Using only the $k$-nearest neighbors in the RBF interpolation also removes any input points that are far away from the point being interpolated. Input data points that are far away may drive the results of the interpolation operations toward skewed or inaccurate interpolation results.

Therefore, in addition to parallel computing, the k-d tree (or multidimensional binary search tree) [34] data structure has been adapted to improve the performance of the k-nearest
neighbors search in our study. The idea behind the $k$-nearest neighbors search is to explore the partition of the k-d tree that is closer to the query point first because it will likely contain the nearest neighbors. The algorithm starts at the root node and walks down the k-d tree recursively as if it were searching the tree for the query point. The path of walking down the tree to search the query point $q$ is illustrated by arrowed lines in Figure 2.

The original nearest neighbor searching algorithm using the k-d tree data structure can be found in [35]. It finds one nearest neighbors. In our study, we adapted the original algorithm, and made it more efficient for searching ($k$) nearest neighbors of a given data point instead of just finding one nearest neighbor. The modified algorithm uses a data structure called a bounded priority queue (BPQ) that stores the list of $k$ nearest neighbors with their distances to the query point $q$. The bounded priority queue has a fixed upper bound on the number of elements (or points) that can be stored, which is the number of nearest neighbors $k$. Whenever a new element is added to the queue, if the queue is at capacity, the element with the highest priority value (i.e. the longest distance) is ejected from the queue. The pseudocode is illustrated in Algorithm 1.

C. Using Radial Basis Functions for Spatiotemporal Interpolation

To develop an efficient spatiotemporal interpolation method suitable for the daily PM$_{2.5}$ data, the RBF-based spatiotemporal interpolation method treats time as an equivalent to the spatial dimensions, using the so-called “extension approach” [16]. The RBF-based interpolation is adapted to utilize the following spatiotemporal interpolation formula:

$$f(x, y, ct) = \sum_{q=1}^{N} \lambda_q \varphi(d_i) + P(x, y, ct)$$

where $c$ is a factor with the unit [spatial distance unit/time unit], $f(x, y, ct)$ is the interpolation value to calculate at the unmeasured location $(x, y)$ and time instance $t$, $N$ is the number of nearest neighbors with measured values surrounding $(x, y, ct)$, $w_q$ are the weights that were calculated using the measured values of each nearest neighbor, and $d_i$ is the spatiotemporal Euclidean distance between $(x_i, y_i, ct_i)$ and $(x, y, ct)$. The spatiotemporal Euclidean distance $d_i$ is calculated using the following formula:

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + c^2(t_i - t)^2}.$$
The scaled time values $c \times t$ as shown in the third column in Table I, where $t$ is the naive choice of time using incremental values with one increment per day.

IV. RESULTS OF USING RADIAL BASIS FUNCTIONS FOR INTERPOLATING GEOSPATIOTEMPORAL BIG DATA

A. Computational Performance by Using Parallel Computing and K-D Trees

By applying the parallel programming techniques and the k-d tree for finding the $k$-nearest neighbors as described in Section III, the execution time of the RBF spatiotemporal interpolation shows that good performance is achieved.

A test was run on a system that was equipped with an Intel Core i7-3630QM CPU running at 2.40 GHz with 16GB of RAM and 8 available processors for servicing thread requests. Using the location dataset to be interpolated at the county level, for 1,134,785 (3,109 × 365) PM$_{2.5}$ interpolation results, the process took 40 seconds with 2 nearest neighbors and 45 seconds with 3 nearest neighbors as illustrated in Figure 3. The runtime results for the interpolations at the county level were the same for each of the five radial basis functions used (i.e. Multi-Quadric, Inverse Multi-Quadric, Gaussian, Thin Plate Spline, and Compactly Supported RBF). A test was also run on the same system using the dataset to be interpolated at the census block group level. For 75,784,950 (207,630 × 365) PM$_{2.5}$ interpolation results, the process took 1,560 seconds (26 minutes) with 2 nearest neighbors and 1,818 seconds (30 minutes) with 3 nearest neighbors as illustrated in Figure 3. The runtime results for the interpolations at the census block group level were the same for each of the five radial basis functions used (i.e. Multi-Quadric, Inverse Multi-Quadric, Gaussian, Thin Plate Spline, and Compactly Supported RBF).

B. Leave-One-Out Cross Validation Results

Leave-one-out cross validation performs $N$ experiments for a dataset with $N$ samples. For each experiment, $N-1$ samples are used for training and the remaining sample is used for testing. That means that for $N$ separate times, the function approximator is trained on all the data except for one point and an interpolation is made for that point. Leave-one-out cross validation has unbiased performance estimation, but has a very large variance that can cause unreliable estimates [37].

Leave-one-out cross validation (LOOCV) was used for determining an optimal radial basis function and the optimal number of nearest neighbors for interpolating the PM$_{2.5}$ concentration values. The effect of changing the radial basis function and the number of nearest neighbors $N$ was investigated by previewing the output of the RBF interpolation methods and calculating the Root Mean Square Percentage Error (RMSPE). The RBF interpolation method that minimizes RMSPE value is determined to be the best interpolation method. ArcGIS also finds optimal parameters by minimizing RMSPE when implementing traditional spatial RBF methods. The RMSPE is calculated using the following formula:

$$\text{RMSPE} = \sqrt{\frac{\sum_{j=1}^{N} (O_j - I_j)^2}{N}} * 100$$

where $N$ is the number of observations, $O_j$ is the actual measured value from the PM$_{2.5}$ dataset and $I_j$ is the interpolated PM$_{2.5}$ value.

Figure 4 illustrates the leave-one-out cross validation RMSPE results for the PM$_{2.5}$ dataset using each of the radial basis functions without the optional polynomial added to the RBF interpolation. The thin plate spline function was not used in this test because it requires the polynomial be used in the RBF interpolation. Without the polynomial, the thin plate spline radial basis function has no solution. Each radial basis function was tested at 1, 2, 3, 4, 5, 6, and 7 nearest neighbors for the leave-one-out cross validation.

Figure 4 shows that the best RBF method for the PM$_{2.5}$ data is the Gaussian radial basis function, because this function minimizes the RMSPE error statistic. Table II shows the RMSPE results for the Gaussian radial basis function without using the optional polynomial in the RBF interpolation. The best $k$-nearest neighbors to use for the Gaussian radial basis function are 1, 2, and 4 nearest neighbors at RMSPE values of 87.35699, 81.24684, and 92.38953 respectively. Therefore, the Gaussian radial basis function was chosen to interpolate the PM$_{2.5}$ dataset.
to interpolate the PM$_{2.5}$ data, the RBF methods have similar RMSPE results for the leave-one-out cross validation. However, the RMSPE results are not as low as the Gaussian radial basis function without the optional polynomial added to the RBF interpolation. Therefore, the Gaussian radial basis function without the optional polynomial added to the RBF interpolation was chosen to interpolate the PM$_{2.5}$ dataset.

V. COMPARISON RESULTS WITH IDW-BASED SPATIOTEMPORAL INTERPOLATION

A. IDW-based Spatiotemporal Interpolation

The IDW-based spatiotemporal interpolation method that uses the extension approach [16] is compared with the RBF-based interpolation method in our study. The IDW method is a simple and intuitive deterministic interpolation method based on Tobler’s First Law of Geography [8]. It is one of the most commonly used interpolation method in GIS. Many studies have used IDW-based interpolation methods [38], [39], [40].

IDW interpolation assumes that each measured point has a local influence that diminishes with distance, and weights the points closer to the interpolated location greater than those farther away. The traditional spatial IDW method is adapted to utilize the following spatiotemporal interpolation formula based on the extension approach:

$$w(x, y, ct) = \sum_{i=1}^{N} \lambda_i w_i,$$
$$\lambda_i = \frac{\left(\frac{d_i}{\text{dist}_{max}}\right)^p}{\sum_{k=1}^{N} \left(\frac{d_k}{\text{dist}_{max}}\right)^p}$$  \hspace{1cm} (1)

where $c$ is a factor with the unit [spatial distance unit/time unit], $w(x, y, ct)$ is the interpolation value to calculate at the unmeasured location $(x, y)$ and time instance $t$, $N$ is the number of nearest neighbors with measured values surrounding $(x, y, ct)$, $\lambda_i$ are the weights assigned to each measured value $w_i$ at $(x_i, y_i, ct_i)$, $d_i$ is the spatiotemporal Euclidean distance between $(x_i, y_i, ct_i)$ and $(x, y, ct)$, and $p$ is the exponent that influences the weighting of $w_i$. Weighting value $\lambda_i$ ranges from 0 to 1 and is a function of the inverse of the spatiotemporal distance between a nearest neighbor and the unmeasured point. It is worth to note that the sum of all the $\lambda_i$’s ($i \in [1, N]$) is 1. The spatiotemporal Euclidean distance $d_i$ is calculated using the following formula by getting the sum of three distances squared under the root sign:

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + c^2(t_i - t)^2}.$$  \hspace{1cm} (2)

B. Leave-One-Out Cross Validation Results

Leave-one-out cross validation (LOOCV) was used for determining an optimal IDW method for interpolating the PM$_{2.5}$ concentration values based on the number of nearest neighbors $N$ and the exponent $p$. The effect of changing $p$ and $N$ was investigated by previewing the output of the IDW interpolation methods and calculating error statistics using LOOCV. Leave-one-out cross validation was performed on the following 45 IDW methods with each method having a different number of nearest neighbors $N$ and a different exponent $p$, with $N \in \{3, 4, 5, 6, 7\}$ and $p \in \{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$. Figure 6 illustrates the RMSPE results for the 45 IDW methods that were evaluated in the leave-one-out cross validation, for the PM$_{2.5}$ data.

---

TABLE II. RMSPE RESULTS FOR THE GAUSSIAN RBF INTERPOLATION.

<table>
<thead>
<tr>
<th>Neighbors</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>87.35699</td>
</tr>
<tr>
<td>2</td>
<td>81.24684</td>
</tr>
<tr>
<td>3</td>
<td>89.41231</td>
</tr>
<tr>
<td>4</td>
<td>92.38953</td>
</tr>
<tr>
<td>5</td>
<td>112.24791</td>
</tr>
<tr>
<td>6</td>
<td>104.22233</td>
</tr>
<tr>
<td>7</td>
<td>114.57411</td>
</tr>
</tbody>
</table>

Figure 4. Leave-one-out cross validation results without the optional polynomial added to the RBF interpolation.

Figure 5 illustrates the leave-one-out cross validation RMSPE results for the PM$_{2.5}$ dataset using each of the radial basis functions with the optional polynomial added to the RBF interpolation. Each radial basis function was tested at 1, 2, 3, 4, 5, 6, and 7 nearest neighbors for the leave-one-out cross validation.

Figure 5 shows that when using the optional polynomial to interpolate the PM$_{2.5}$ data, the RBF methods have similar RMSPE results for the leave-one-out cross validation. However, the RMSPE results are not as low as the Gaussian radial basis function without the optional polynomial added to the RBF interpolation. Therefore, the Gaussian radial basis function without the optional polynomial added to the RBF interpolation was chosen to interpolate the PM$_{2.5}$ dataset.

V. COMPARISON RESULTS WITH IDW-BASED SPATIOTEMPORAL INTERPOLATION

A. IDW-based Spatiotemporal Interpolation

The IDW-based spatiotemporal interpolation method that uses the extension approach [16] is compared with the RBF-based interpolation method in our study. The IDW method is a simple and intuitive deterministic interpolation method based on Tobler’s First Law of Geography [8]. It is one of the most commonly used interpolation method in GIS. Many studies have used IDW-based interpolation methods [38], [39], [40].

IDW interpolation assumes that each measured point has a local influence that diminishes with distance, and weights the points closer to the interpolated location greater than those farther away. The traditional spatial IDW method is adapted to utilize the following spatiotemporal interpolation formula based on the extension approach:

$$w(x, y, ct) = \sum_{i=1}^{N} \lambda_i w_i,$$
$$\lambda_i = \frac{\left(\frac{d_i}{\text{dist}_{max}}\right)^p}{\sum_{k=1}^{N} \left(\frac{d_k}{\text{dist}_{max}}\right)^p}$$  \hspace{1cm} (1)

where $c$ is a factor with the unit [spatial distance unit/time unit], $w(x, y, ct)$ is the interpolation value to calculate at the unmeasured location $(x, y)$ and time instance $t$, $N$ is the number of nearest neighbors with measured values surrounding $(x, y, ct)$, $\lambda_i$ are the weights assigned to each measured value $w_i$ at $(x_i, y_i, ct_i)$, $d_i$ is the spatiotemporal Euclidean distance between $(x_i, y_i, ct_i)$ and $(x, y, ct)$, and $p$ is the exponent that influences the weighting of $w_i$. Weighting value $\lambda_i$ ranges from 0 to 1 and is a function of the inverse of the spatiotemporal distance between a nearest neighbor and the unmeasured point. It is worth to note that the sum of all the $\lambda_i$’s ($i \in [1, N]$) is 1. The spatiotemporal Euclidean distance $d_i$ is calculated using the following formula by getting the sum of three distances squared under the root sign:

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + c^2(t_i - t)^2}.$$  \hspace{1cm} (2)

B. Leave-One-Out Cross Validation Results

Leave-one-out cross validation (LOOCV) was used for determining an optimal IDW method for interpolating the PM$_{2.5}$ concentration values based on the number of nearest neighbors $N$ and the exponent $p$. The effect of changing $p$ and $N$ was investigated by previewing the output of the IDW interpolation methods and calculating error statistics using LOOCV. Leave-one-out cross validation was performed on the following 45 IDW methods with each method having a different number of nearest neighbors $N$ and a different exponent $p$, with $N \in \{3, 4, 5, 6, 7\}$ and $p \in \{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$. Figure 6 illustrates the RMSPE results for the 45 IDW methods that were evaluated in the leave-one-out cross validation, for the PM$_{2.5}$ data.
It can be concluded from Figure 6 that the best IDW methods for the PM$_{2.5}$ data are the ones with three nearest neighbors, because they minimize the RMSPE from the cross validation error statistics. As the exponent $p$ approaches five, the RMSPE results are almost identical. Therefore, the best IDW method for interpolating the PM$_{2.5}$ data are ($n = 3, 4, 5, 6, 7$ with $p = 5$).

Figure 7 illustrates the comparison between the best IDW interpolation method and the best RBF interpolation method. Figure 7 shows that the Gaussian radial basis function without the optional polynomial added to the RBF interpolation is significantly better than the IDW interpolation method because the Gaussian RBF interpolation minimizes the RMSPE cross validation error statistic. Therefore, the best interpolation method to use for interpolating the PM$_{2.5}$ data is the Gaussian radial basis function without the optional polynomial.

C. Computational Performance

To compare the computational performance between the IDW interpolation method and the RBF interpolation method, a test was run on a system that was equipped with an Intel Core i7-3630 QM CPU running at 2.40 GHz with 16GB of RAM and 8 available processors for servicing thread requests. The test was run using the datasets to be interpolated at the county level and the census block group level. For 1,134,785 (3,109×365) PM$_{2.5}$ interpolation results, the process took 45 seconds with 3 nearest neighbors for the IDW interpolation method and 45 seconds with 3 nearest neighbors for the RBF interpolation method as illustrated in Figure 8. For 75,784,950 (207,630 x 365) PM$_{2.5}$ interpolation results, the process took 1,598 seconds (26.6 minutes) with 3 nearest neighbors using the IDW interpolation method and the process took 1,818 seconds (30 minutes) with 3 nearest neighbors using the RBF interpolation method as illustrated in Figure 8. Therefore, the IDW interpolation method has a slightly better computational performance than the RBF interpolation method when interpolating the PM$_{2.5}$ data at the census block group level.

![Fig. 6. Leave-one-out cross validation results for the 45 IDW interpolation methods.](image)

![Fig. 8. RBF run times at the census block group level for both 2 and 3 nearest neighbors.](image)

![Fig. 7. Comparison between the best IDW interpolation method and the best RBF interpolation method.](image)

VI. CONCLUSION

The RBF interpolation method was studied to determine how well it can interpolate the large experimental PM$_{2.5}$ datasets when compared to the IDW interpolation method. In this research, we design and implement an RBF-based spatiotemporal interpolation method that treats time as an equivalent to the spatial dimensions [16] and produce daily PM$_{2.5}$ interpolation results at both the census block group level and county level.

This study has made several contributions. First, the study explores computational issues encountered when implementing spatiotemporal interpolation for large datasets and presents researchers with the appropriate techniques. Multi-threaded parallel computing techniques together with the adapted k-d tree data structure have been designed and implemented. The second contribution of this paper concerns finding optimal radial basis function for the RBF-based spatiotemporal interpolation method. The study shows how leave-one-out cross validation can be used to select the appropriate radial basis function for interpolating the PM$_{2.5}$ data by using RMSPE (Root Mean Square Percentage Error). Third, the RBF-based spatiotemporal interpolation method is compared with the IDW-based method. Our experiment shows that although the IDW interpolation method has a slightly better computational performance than the RBF interpolation method when interpolating the PM$_{2.5}$ data at the census block group level,
the RBF method significantly outperforms the IDW method with smaller RMSPE result.

ACKNOWLEDGMENT

The authors of this paper would like to thank Dr. Xingyou Zhang at the Centers for Disease Control and Prevention for providing the valuable experimental PM$_{2.5}$ data for this study.

REFERENCES