Short note

Least cost distance analysis for spatial interpolation

Jonathan A. Greenberg a,*,1, Carlos Rueda b,2, Erin L. Hestir a,1, Maria J. Santos a,1, Susan L. Ustin a,1

a Center for Spatial Technologies and Remote Sensing (CSTARS), Department of Land, Air and Water Resources, University of California, Davis, The Barn, 1 Shields Avenue, Davis, CA 95616, USA
b Monterey Bay Aquarium Research Institute, 7700 Sandholdt Rd., Moss Landing, CA 95039, USA

A R T I C L E    I N F O

Article info
Article history:
Received 17 March 2009
Received in revised form 19 May 2010
Accepted 21 May 2010
Available online 28 October 2010

Keywords:
Spatial interpolation
Least cost distance analysis
Network interpolation
Inverse distance weighting
Non-Euclidean distance

A B S T R A C T

Spatial interpolation allows creation of continuous raster surfaces from a subsample of point-based measurements. Most interpolation approaches use Euclidean distance measurements between data points to generate predictions of values at unknown locations. However, there are many spatially distributed data sets that are not properly represented by Euclidean distances and require distance measures which represent their complex geographic connectivity. The problem of defining non-Euclidean distances between data points has been solved using the network-based solutions, but such techniques have historically relied on a network of connected line segments to determine point-to-point distances. While these vector-based solutions are computationally efficient, they cannot model more complex 2- and 3-dimensional systems of connectivity. Here, we use least-cost-path analyses to define distances between sampled points; a solution that allows for arbitrarily complex systems of connectivity to be interpolated. We used least-cost path distances in conjunction with the inverse distance weighting interpolation for a proof-of-concept interpolation of water temperature data in a complex deltaic river system. We compare our technique to Euclidean distance interpolation, and demonstrate that our technique, which follows connectivity rules, yields are more realistic interpolation of water temperature.

© 2010 Published by Elsevier Ltd.

1. Introduction

Interpolation of geospatial data facilitates the creation of continuous raster surfaces from point-based measurements, and when used with spatial models it allows prediction of values at locations that do not have known information. Most geostatistical packages rely on Euclidean-based estimates of distances between known locations to develop the model. However, many spatially distributed data have complex geographic connectivity whose distances cannot be determined from Euclidean distances. For example, data in a stream network are often governed by the pattern of stream channel connectivity rather than straight-line distances (Peterson et al., 2007); examining genetic relatedness between populations isolated by distance may be governed by connectivity between preferred habitat rather than linear distances (Wright, 1943; Kimura and Weiss, 1964; Broquet et al., 2006). In such instances, non-Euclidean distance calculations for spatial interpolation are needed.

Interpolation using non-Euclidean distances is often achieved by calculating optimal path solutions to determine, initially, relevant non-Euclidean distances between locations of known values to construct the model. These distances from each unknown location to each known location are then used as inputs into the interpolation model. In the literature, derivation of distances between known and unknown locations is determined through a vector-based solution: the system is represented as a network of connected line segments (Ganio et al., 2005; Okabe et al., 2006; Peterson et al., 2006, 2007; Skøien et al., 2006). While vector-based solutions to deriving non-Euclidean distances for interpolation models are computationally efficient, they are constrained to systems which can be accurately described using 1-dimensional lines, e.g. road networks, narrow stream networks, etc. We present a more general solution for deriving non-Euclidean distances that can be applied to arbitrarily complex two- and three-dimensional surfaces and implementing these distances to derive an inverse-distance weighted (IDW) model.

2. Methods

2.1. Inverse distance weighting

We first review the inverse distance weighting model (Watson and Philip, 1985). The form of this model is

\[ Zest_j = \frac{\sum z_i}{(d_{ij} + s)^p} / \sum \frac{z_i}{(1 + s)^p} \] (1)

where \( Zest \) is estimated value at location \( j \); \( z_i \) measured sample value at location \( i \); \( d_{ij} \) distance between \( i \) and \( j \); \( s \) smoothing factor (default = 0); \( p \), weighting power (default = 1).
For each location $j$, $d_{ij}$ is the distance term that is traditionally calculated using an Euclidean solution (e.g. Pythagoras). For our analysis, however, we calculate this distance as the cumulative, raster-based least-cost-distance between location $j$ and known location $i$.

### 2.2. Least cost distance analysis

For a given cost surface raster, the least cost distance algorithm seeks to solve the optimized path from an arbitrary location $j$ that accumulates the least cost to a target location $i$. Each cell in a cost surface raster contains the cost to traverse that source cell to one of the eight surrounding grid cells. Higher values indicate higher costs. Through the solution for the optimum (least cost) path, the accumulated cost along the path can also be calculated. The raster cost surface can be defined in a myriad of ways, and barriers can be included (raster cells which cannot be traversed for any reason). Some least cost distance algorithms (e.g. ArcGIS’ path distance algorithm, ESRI, 2008) can also modify the accumulated costs as a function of the difference between adjacent cells (in a vertical context, the slope, in a horizontal context, the change in azimuth), which can introduce asymmetry into the least cost distance (e.g. $d_{ij}$ does not necessarily equal $d_{ji}$). With the exception of some cost surfaces that include barrier cells, a least cost distance analysis will always be able to derive a solution between any source and destination cell. The accumulated cost between $j$ and $i$, for our purposes, defines the distance $d_{ij}$ between these two locations.

The cost surface, therefore, must be well defined to generate reliable distances for use in the interpolation algorithm. In the simplest, trivial case, if we wished to replicate an Euclidean distance, we would set the entire cost surface to be the raster cell’s width, so the accumulated cost between any two points would be equal to the straight-line distance between these two points. $d_{ij}$ for any $i$ and $j$ would end up being the straight-line distance between these two points. If, for instance, our goal is to interpolate using “as-the-fish-swims” distances along a river network, we would set all of the cost cells within the river channels to be equal to the raster cell width, and assign all non-river cells to be barriers. The resolution of the cost surface must also be relevant to the scale of the analysis. Since the least cost paths are, fundamentally, calculated based on the eight connections surrounding a given grid cell, the accuracy of the raster approximation of the true least cost distance increases with higher resolution, but so too do
the computational costs. Additionally, the properties of the surface's connectivity can change with coarser resolution, as sites which may have been unconnected at higher resolutions (e.g. proximate, but unconnected regions) are merged into a single, connected region at coarser resolutions.

2.3. Integrating least cost distance analysis into spatial interpolations

(1) Given the set of $N$ known measurements $z_i$, each measurement located at the two-dimensional coordinate $i$, and a cost surface $C$, we run $N$ least cost distance analyses using the same cost surface $C$, but calculated as the accumulated cost to reach each location $i$. We define each individual accumulated cost surface calculated from a single location $i$ as $D_i$.

(2) For each location $j$, we can now determine the least cost distance $d_{ij}$ from each known measurement $i$ by examining position $j$ in each accumulated cost surface $D_i$.

(3) We apply the inverse distance weighting model (1) to each position $j$ in the stack of $D_i$ accumulated cost surfaces to determine the estimated value $Z_{estj}$.

2.4. Example of least cost distance analysis-based interpolation

We applied the least cost distance based IDW to produce spatially continuous estimates of average annual water temperatures in the Sacramento-San Joaquin River Delta (hereafter “the Delta”). The code was developed using Python, using GRASS GIS 6.3 (GRASS Development Team, 2007) to perform the GIS functions. The least cost distance analysis was performed using the r.cost library in GRASS GIS (Awaida and Westervelt, 2008). Annual average water temperatures were calculated from hourly temperature data collected by 35 temperature gauges deployed throughout the Delta and distributed by the California Data Exchange Center (Department of Water Resources, 2008, Fig. 2. Interpolated water temperatures using least cost distance analysis to define distances.)
see Fig. 1). We used 18 randomly selected gauges to derive the accumulated least cost distance surfaces using the cell size (10 m) as the cost for all cells falling within the water network, and assigning barrier values (cost = $\infty$) to the remaining land cells. Values from the remaining 17 gages were used to determine the best model parameters (smoothing, $s$, and power, $p$, see Eq. (1)) that minimized the sum-of-squares error between the predicted and measured values at these 17 test sites. An error of 1.8 was found to be the lowest error using a smoothing parameter ($s$) of 1425 and a power ($p$) of 1.0. For qualitative comparison, we also ran the same 18 sites through an Euclidean based IDW interpolation, optimizing in a similar fashion (smoothing = 1500, power = 1.2). Figs. 2 and 3 show the difference in predicted values between an interpolation using least cost path derived distances and Euclidean distances Delta-wide, and Fig. 4a and b shows representative subsets. Note how channels which are close to one another in Euclidean space, but do not connect to one another for a large distance, do not share similar values to one another, unlike the Euclidean-distance based interpolator.

3. Conclusions

Least cost distance analysis provides a generalizable approach to deriving non-Euclidean distance measurements for use in spatial interpolations. Using cost surfaces containing values of the cell sizes or barriers, this technique can replicate network-based interpolation techniques that most commonly use vector-based algorithms (e.g. Ganio et al., 2005; Okabe et al., 2006; Peterson et al., 2006, 2007; Skøien et al., 2006). However, unlike vector-based approaches, using a least cost distance approach can deal with arbitrarily complex cost surfaces and optimized paths across a two- or three-dimensional surface, not just those which can be represented by line networks.

One of the shortcomings of this method is the computational cost of the analysis—the least cost distance algorithm is the most computationally costly portion of the analysis, and one accumulated least cost distance surface must be calculated per known measurement. Advances in computing hardware and improvements in algorithm design, particularly the application of parallel computing (Hazel et al., 2006), however, can offset these
The degree to which this approach to deriving non-Euclidean distances can improve interpolations depends largely on the nature of the variables to be interpolated, and the way by which the cost surface is defined. The relatively simple cost surface presented herein, which consists of essentially two values: the width of a pixel or a barrier, is optimized for situations where network connectivity is added to what is essentially an Euclidean analysis (e.g. the accumulated costs of our example are in meters). A more complex application might be interpolating gene frequencies as a function of gene flow paths (Broquet et al., 2006). We might assume that a suite of biotic or abiotic variables may facilitate or hinder gene flow, but how to combine these variables into a single, meaningful cost surface may require some advanced preliminary analysis.

The approach described herein lays the groundwork for a non-Euclidean distance approach to interpolation. In future research, we are planning to investigate the impacts of scale, more complex cost surfaces, and anisotropy on the accuracy of our interpolations.

Fig. 4. a and b: Subset of interpolated temperature detailing the differences between using (a) least cost distance analysis to define distances and (b) Euclidean distances. Note how (a) shows changes in temperatures that correspond to river junctions and (b) shows the classic concentric-ring pattern of interpolation.

limitations. Alternatively, depending on the particular interpolation problem, cost surface cell resolutions can be coarsened to reduce the total number of cells to be analyzed.

References


