

## Interpolationen in GeoSuite

Hinweise: Matrizen oder Vektoren sind nachstehend mit **fetten** Buchstaben bezeichnet.  
Das Zeichen  $\rightarrow$  verweist ins Kapitel 3 Definitionen.

### 1. Shepard's Methode

Quellen: (Shepard 1968), (ALGLIB IDW-Interpolation 2012)

Gegeben sind die Passpunkte  $\mathbf{x}_i$  (auch  $\rightarrow$ Knoten genannt) mit zugehöriger Restklaffung (alias Residue)  $z_i$  sowie irgendein Neupunkt  $\mathbf{x}$ . Die Passpunkte liegen im Ausgangs- und im Zielsystem mit Koordinaten vor, die Neupunkte nur im Ausgangssystem.

Wir führen im Neupunkt  $\mathbf{x}$  eine Gewichtung  $w_i$  ein:  $w_i(\mathbf{x}) = \left( \frac{1}{\|\mathbf{x} - \mathbf{x}_i\|} \right)^p$ ,  $p = 2$

$\|\mathbf{x} - \mathbf{x}_i\|$  steht für die Norm der Vektordifferenz  $\mathbf{x} - \mathbf{x}_i$ , was gleichbedeutend ist mit der Distanz zwischen Neupunkt  $\mathbf{x}$  und Passpunkt  $\mathbf{x}_i$  (im Ausgangssystem).

Die einfachste Form der inversen Distanzgewichtung (IDW) ist dann:

$f(\mathbf{x}) = \frac{\sum_i w_i(\mathbf{x}) \cdot z_i}{\sum_i w_i(\mathbf{x})}$   $f(\mathbf{x})$  ist der interpolierte Wert im Neupunkt  $\mathbf{x}$ , die sogenannte Interpolante oder Interpolierende (Funktion); in der Transformationsempfehlung ist sie mit  $U_k$  bezeichnet.

Die Shepard Methode entspricht in der Transformationsempfehlung (Bundesamt für Landestopografie swisstopo 2017) dem Kapitel **4.5.3. Interpolation nach gewogenem Mittel**, wenn man dort  $\lambda = 2$  und  $c = 0$  setzt.

Ein solcher IDW Algorithmus hat sowohl Vorteile als auch Nachteile. Seine Vorteile sind:

- ist sehr einfach zu implementieren
- es müssen keine Parameter eingestellt werden
- Fähigkeit, in mehrdimensionalen Räumen eingesetzt zu werden
- Fähigkeit, verstreute Daten zu interpolieren und in irgendeinem Koordinatenraster zu arbeiten.  $\rightarrow$  Multikollinearität ist nicht ein Problem, der Algorithmus kann arbeiten, selbst wenn alle Punkte in einem niedrig-dimensionalen Unterraum liegen (d.h. auf Gerade oder in Ebene). Wenn richtig implementiert, kann er sogar mit zusammenfallenden Knoten (d.h. Passpunkten) arbeiten.

Hingegen gibt es mehrere wichtige Nachteile:

- Niedrige Leistung bei grossen Datensätzen: Die  $\rightarrow$ Komplexität der Interpolation ist  $O(N)$ , d.h. die Laufzeit des Programms nimmt proportional zur totalen Anzahl Passpunkte  $N$  zu.
- der Algorithmus gibt entfernten Knoten zu viel Gewicht. Ihr Gesamtgewicht kann grösser sein als das Gewicht nahe gelegener Knoten. Dies ist in hoch-dimensionalen Räumen ausgeprägter.
- Die Methode von Shepard ist eine globale Methode, was für sich ein Problem ist. Die Interpolante ist zu empfindlich auf entfernte Ausreisser.
- Die Interpolante  $f(\mathbf{x})$  ist in  $\mathbf{x}_i$  flach, d.h. sie hat in den Knoten die Ableitung Null.

Aus diesen Gründen wird die originale Shepard Methode in der Praxis selten verwendet.

## 2. Modifizierte Shepard Methode

Quellen: (Renka 1988), (Thacker u.a. 2010), (ALGLIB IDW-Interpolation 2012), teilweise wörtlich übersetzt und ergänzt.

### 2.1 Modifizierter Shepard Algorithmus

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{N_w} W_i(\mathbf{x}) \cdot Q_i(\mathbf{x})}{\sum_{i=1}^{N_w} W_i(\mathbf{x})} \quad ; \quad N_w < N$$

Die totale Anzahl der Passpunkte ist  $N$ , jedoch werden nur die dem Neupunkte  $\mathbf{x}$  nächsten  $N_w$  Passpunkte verwendet.

### 2.2 Wie gross soll $N_w$ sein?

ALGLIB empfiehlt:  $N_w$  kontrolliert den Algorithmus lokal (*Anmerkung des Verfassers: d.h. wie weit soll die Gewichtung wirken, vgl. Kap. 2.3*). Zu grosse Werte machen die Interpolante zu global (die Berechnung wird langsam und die Interpolante unfähig, lokale Änderungen in der Interpolation zu reproduzieren). Andererseits führen zu kleine  $N_w$  zu spitzen und ungenauen Interpolanten (beispielsweise ist  $N_w = 1$  gerade eine Nearest Neighbour Interpolation). Ein guter Wert für  $N_w$  liegt normalerweise leicht höher als  $\max(1.5 \cdot N_q, 2^D + 1)$ ;  $D$  ist die Anzahl Dimensionen des Raumes, in geodätischen Fällen also 2 oder 3, womit  $N_w$  immer gleich  $1.5 \cdot N_q$  sein wird. (Renka 1988) erhielt für zweidimensionale Daten bei einem Wert von  $N_w = 19$  gute Resultate.

GeoSuite stellt  $N_w$  ein auf  $N_w \geq 2^D + 1$  und empfiehlt  $N_w \approx 2 \cdot N_q$

Parameter	Raumdimension		
	1	2	3
$N_w$	3	5	9

### 2.3 Gewichtsfunktion $W_i(\mathbf{x})$

$$W_i(\mathbf{x}) = \left( \frac{R_x - \|\mathbf{x} - \mathbf{x}_i\|}{R_x \cdot \|\mathbf{x} - \mathbf{x}_i\|} \right)^2 = \frac{1}{\|\mathbf{x} - \mathbf{x}_i\|^2} - \frac{1}{R_x^2}, \quad R_x = \max_{N_w} \|\mathbf{x} - \mathbf{x}_i\|$$

Der Einflussradius  $R_x$  ist somit die maximale Distanz zwischen dem Neupunkt  $\mathbf{x}$  und seinen  $N_w$  benachbarten Passpunkten  $\mathbf{x}_i$ . Ausserhalb dieses Radius' wird der Funktionswert von  $W_i$  zu Null.

### 2.4 Knotenfunktion $Q_i(\mathbf{x})$

Die **Knotenfunktion** (engl. nodal function)  $Q_i(\mathbf{x})$  tritt an die Stelle der Restklaffung  $z_i$ . Sie ist eine lokale Approximation für  $z_i$  im Passpunkte  $\mathbf{x}_i$ .  $Q_i(\mathbf{x})$  ist eine  $\rightarrow$ bivariate, wahlweise quadratische, lineare oder konstante Funktion. Die Koeffizienten der Funktion  $Q_i(\mathbf{x})$  resultieren aus einer gewichteten Kleinste-Quadrate-Einpassung auf einen Satz von  $N_q$  nächsten Nachbarn von  $\mathbf{x}_i$  mit der Bedingung  $Q_i(\mathbf{x}_i) = z_i$ .

Mathematisch ausgedrückt:  $Q_i(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + g$  ;  $\mathbf{A} \in \mathbb{R}^{D \times D}$ ,  $\mathbf{b} \in \mathbb{R}^D$ ,  $g \in \mathbb{R}$

Beispiel zur Berechnung dieser Knotenfunktion  $Q(\mathbf{x})$  (zweidimensionaler Fall):

Punkt  $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$       Koeffizienten  $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$  ;  $\mathbf{b} = \begin{bmatrix} e \\ f \end{bmatrix}$  und  $g$  ;

somit ausmultipliziert:  $Q_i = ax^2 + bxy + cxy + dy^2 + ex + fy + g$

da nicht anzunehmen ist, dass  $xy$  zwei unbekannte Koeffizienten  $b$  und  $c$  befriedigen muss, wird  $b = c$  gesetzt und daraus

$$Q_i = ax^2 + 2bxy + dy^2 + ex + fy + g$$

im  $i$ -ten Passpunkt  $\mathbf{x}_i$  ist zudem die Bedingung einzuhalten:  $Q_i(x_i) - z_i = 0$

Die 6 vorerst unbekanntes Koeffizienten  $[a \ b \ d \ e \ f \ g]^T$  sind also aus mindestens 3 Stützpunkten zu bestimmen. Dies erfolgt in einer "vermittelnden Ausgleichung mit Bedingung zwischen den unbekanntes Parametern (auch Gauss-Markov-Modell genannt)".

## 2.5 Wie gross soll $N_q$ sein?

$N_q$  kontrolliert einen anderen Aspekt der Nachbarschaft, nämlich die Anzahl der Knoten, die zur Bildung der eben beschriebenen Knotenfunktionen  $Q_i(\mathbf{x})$  verwendet werden. Eine gute Knotenfunktion muss durch  $(x_i, y_i)$  gehen und sie sollte die Funktion in der Umgebung näherungsweise approximieren. Ein zu kleiner  $N_q$  Wert wird eine gute Approximation verunmöglichen.

Ein zu grosser  $N_q$  Wert wird  $Q_i(\mathbf{x})$  zu einem globalen Verhalten führen; dies ist aber dort nicht sehr gut, wo es nötig wäre, nämlich im Einflussbereich von  $\mathbf{x}_i$ .

Ein guter  $N_q$  Wert ist gewöhnlich 1.5-mal grösser als die Anzahl freier Parameter der Knotenfunktion:  $1+D$  für eine lineare Knotenfunktion,  $(D+2) \cdot (D+1)/2$  für eine quadratische Knotenfunktion. (Renka 1988) erhielt für zweidimensionale Daten bei einem Wert von  $N_q = 13$  gute Resultate.

Gibt man keinen Wert vor, stellt GeoSuite  $N_q$  auf folgenden Wert ein:

Knotenfunktion	Parameter	Raumdimension		
		1	2	3
konstant:	$N_q$ wird ignoriert	--	--	--
linear:	$N_q \geq \text{Maximum von } (1.5 \cdot (D + 1)) ; 2^D + 1$	3	5	9
quadratisch:	$N_q \geq \text{Maximum von } (3/4 \cdot (D + 2) \cdot (D + 1)) ; 2^D + 1$	5	9	15

## 2.6 Vor- und Nachteile des Modified Shepard Algorithmus

Dieser Modified Shepard Algorithmus hat folgende Vorteile:

- Fähigkeit, in mehrdimensionalen Räumen eingesetzt zu werden (bei Geodäten ist der Raum allerdings meist auf 3 Dimensionen begrenzt).
- Fähigkeit, verstreute Daten zu interpolieren und in irgendeinem Koordinatenraster zu arbeiten. → Multikollinearität ist nicht ein Problem, der Algorithmus kann arbeiten, selbst wenn alle Punkte in einem niedrig-dimensionalen Unterraum liegen (d.h. Gerade, Ebene). Wenn richtig implementiert, kann er sogar mit zusammenfallenden Knoten (d.h. Passpunkten) arbeiten.
- Die erste Ableitung der Interpolante  $f(\mathbf{x})$  ist stetig.

- Gute Leistung, selbst bei grossen Datensätzen, was dank effizienter Nearest Neighbour Suchalgorithmen erreicht wird. Die  $\rightarrow$ Komplexität des Interpolationsaufbaus ist  $O(N \cdot \log N)$ , der Interpolation selber  $O(\log N)$ ; d.h. bei einer Verdoppelung von  $N$  läuft das Programm um eine konstante Zeit länger.
- Wenn quadratische oder lineare Knotenfunktionen verwendet werden, gibt es in der Nähe von Interpolationsknoten keine "flachen Flecken".
- Lokal wirkender Interpolationsalgorithmus.  $f(\mathbf{x})$  hängt nur von den nächsten Nachbarn von  $\mathbf{x}$  ab, was die Interpolationsqualität bedeutend verbessert.
- Leichte Generalisation auf Regressionsanalysen, d.h. Fähigkeit, mit stark rauschenden Daten zu arbeiten ("Rauschen" nennt man die zufälligen Abweichungen).
- Nur zwei Parameter müssen eingestellt werden:  $N_w$  und  $N_q$ .

Hingegen müssen auch mehrere Nachteile genannt werden:

- Die modifizierte Shepard Methode ist nur für grosse Datensätze (mit Hunderten von Punkten) und nur in niedrig-dimensionalen Räume (2 bis 5 Dimensionen) schneller als der Inverse Distance Weighting Algorithmus.
- Auch wenn der Algorithmus in irgendeinem Koordinatensystem arbeiten kann, nimmt seine Leistung ab, wenn Punkte in einem niedrig-dimensionalen Subraum liegen. In solchen Fällen werden langsame  $\rightarrow$ SVD-basierte kleinste Quadrate Lösungsalgorithmen anstelle von schnelleren  $\rightarrow$ QR-basierten verwendet.
- In einigen seltenen Fällen kann  $f(\mathbf{x})$  Diskontinuitäten haben. Eine Diskontinuität wird erscheinen, wenn mehr als  $N_w$  nächste Nachbarn von  $\mathbf{x}$  gleich weit entfernt von  $\mathbf{x}$  sind. In solchen Fällen wird ein  $\rightarrow$ k-NN Suchalgorithmus Nachbarn aufs Geratewohl wählen, was von Rundungsfehlern abhängt oder von der Reihenfolge, in der die Nachbarn aus der Suchstruktur herausgezogen wurden. Jedoch wird  $f(\mathbf{x})$  begrenzt bleiben und wenn man  $N_w$  vergrössert, wird die Wahrscheinlichkeit solcher Situationen abnehmen. Beispielsweise wird in kartesischen Koordinatensystemen  $N_w$  nach unten begrenzt, um solche Fehlerprobleme zu verhindern.

Jedoch überwiegen die Vorteile des Modifizierten Shepard Algorithmus die Nachteile, so dass er als ein Standardwerkzeug empfohlen werden kann für mehrdimensionale Interpolationen (entweder gestreuter oder regelmässig Daten).

## 2.7 Benutzer - Einstellungen der Knotenfunktion

Der Benutzer kann zwischen 4 Typen von Knotenfunktionen wählen (die Bedeutung der Farben ist analog der Verkehrsampeln):

- Die **konstante** Knotenfunktion (gelb) wird in der originalen Shepard Methode verwendet und bleibt nur für Evaluationszwecke im Angebot des Programms.
- **lineare** Knotenfunktion (grün): ihre Koeffizienten werden mittels Kleinste-Quadrate-Einpassung berechnet. Diese Funktion liefert bessere Genauigkeit als die konstante Funktion.
- **quadratische** Knotenfunktion (grün): ihre Koeffizienten werden mittels Kleinste-Quadrate-Einpassung berechnet. Diese Funktion liefert beste Qualität, solange genügend Daten vorliegen, um die Koeffizienten robust zu berechnen. Wenn die Daten spärlich vorliegen, wird vielleicht die lineare Knotenfunktion besser funktionieren.
- **"schnell" lineare** Knotenfunktion (rot): ihre Koeffizienten werden mit schnellen Algorithmen (Gradienten Interpolation) berechnet statt mit Kleinste-Quadrate basierten. Dies ist ein spezieller Knotenfunktionsstyp für Probleme, wo Geschwindigkeit wichtiger ist als Qualität. Koeffizienten solcher Knotenfunktionen minimierten einige Kleinste-Quadrate ähnliche Fehlerfunktion nicht, sie liefern aber gewöhnlich eine bescheidene Präzision. Hingegen ist diese Art von Knotenfunktionen weniger robust und tolerant auf zufällige Abweichungen als die vorgehenden. Sie sollten sie nur benutzen, wenn Sie ihre Qualität und Stabilität testen können.

## 2.8 Wie läuft der Algorithmus im Detail ab?

1. Passpunkte einlesen
2. Zu jedem Passpunkt  $\mathbf{x}_i$  werden die  $N_q$  nächsten Nachbarn gesucht ( $\Rightarrow$  Kap. 2.5),
3. und die Koeffizienten  $\mathbf{A}$ ,  $\mathbf{B}$  und  $g$  der Knotenfunktion  $Q_i(\mathbf{x})$  bestimmt. ( $\Rightarrow$  Kap. 2.4 und 2.7)
4. Zu jedem Interpolationspunkt  $\mathbf{x}$  werden die  $N_w$  nächsten Nachbarn gesucht. ( $\Rightarrow$  Kap. 2.2)
5. Die Gewichtsfunktion  $W_i(\mathbf{x})$  wird für diese Nachbarn berechnet. ( $\Rightarrow$  Kap. 2.3)
6. Die Interpolante  $f(\mathbf{x})$  wird als gesuchte Residue des Punktes  $\mathbf{x}$  berechnet. ( $\Rightarrow$  Kap. 2.1)

## 3. Definitionen

**Knoten:** als Informatikbezeichnung verwendet für die geodätischen Begriffe Passpunkt, Stützpunkt

**Multikollinearität:** die [Multikollinearität](#) ist ein Problem der [Regressionsanalyse](#) und liegt vor, wenn zwei oder mehr erklärende Variablen eine sehr starke [Korrelation](#) miteinander haben; in der Geometrie nennt man Punkte, die auf einer Geraden (oder Ebene) liegen, kollinear.

**Komplexität** (engl. Complexity): die [Komplexität](#) einer berechenbaren Funktion ist der zu ihrer Berechnung erforderliche Aufwand an Betriebsmitteln wie Speicherplatz, Rechenzeit, benötigte Geräte usw.

**bivariat:** Eine Funktion ist bivariat, wenn sie genau zwei unbestimmte Variablen enthält (z.B.  $f(x,y) = 3x^2 + y + y^3$ ). Der Funktionswert  $f$  hängt also von zwei ("bi") Variablen  $x$  und  $y$  ab.

**k-d tree:** In [computer science](#), a [k-d tree](#) (short for *k-dimensional tree*) is a [space-partitioning data structure](#) for organizing [points](#) in a  $k$ -dimensional [space](#).  $k$ -d trees are a useful data structure for several applications, such as searches involving a multidimensional search key (e.g. [range searches](#) and [nearest neighbor searches](#)).  $k$ -d trees are a special case of [binary space partitioning](#) trees.

**k-NN** Suchalgorithmus: ein  $k$ -dimensionaler Nearest Neighbour Suchalgorithmus.

**SVD:** singular value decomposition, Eigenwertzerlegung, Spektralzerlegung.  
Eigenwertzerlegung von  $\mathbf{A}$ :  $\mathbf{A} = \mathbf{S}\mathbf{D}\mathbf{S}^{-1}$ , wobei  $\mathbf{S} = [\mathbf{s}_1 | \mathbf{s}_2 | \dots | \mathbf{s}_m]$  und  $\mathbf{D} = \text{diag}(j_1, j_2, \dots, j_m)$ , wobei  $\mathbf{s}_i$  Eigenvektoren und  $j_i$  Eigenwerte sind mit  $j_1 \geq j_2 \geq \dots \geq j_m$ .

**QR-Zerlegung:** siehe z.B. <http://de.wikipedia.org/wiki/QR-Zerlegung>  
Zerlegung von  $\mathbf{A}$  in das Produkt  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  zweier anderer Matrizen, wobei  $\mathbf{Q}$  eine orthogonale ( $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ ) bzw. unitäre Matrix ( $\mathbf{Q}\mathbf{Q}^{-1} = \mathbf{I}$ ) und  $\mathbf{R}$  eine obere Dreiecksmatrix ist.

## 4. Raster Interpolationen

### 4.1 Bilineare Interpolation (auch bilineare Filterung genannt)

Quellen: [http://de.wikipedia.org/wiki/Bilineare\\_Filterung](http://de.wikipedia.org/wiki/Bilineare_Filterung), (Saner 2009)

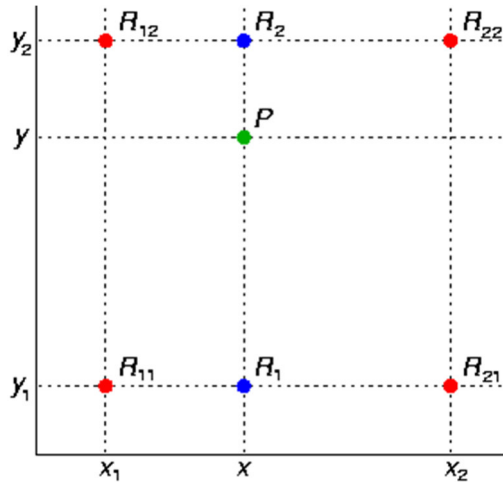


Abb. 1 bilineare Interpolation

Gegeben sind Restklaffungen  $R_{11}$ ,  $R_{12}$ ,  $R_{21}$  und  $R_{22}$  in den roten, gleich bezeichneten Punkten eines Rasters, später Rasterzelle genannt.

Um die Restklaffung in  $P$  näherungsweise (durch lineare Interpolation) zu bestimmen, wird zunächst in eine festgelegte Richtung (horizontal oder vertikal, hier horizontal) zwischen den Rasterpunkten linear interpoliert. Dabei sind  $R_1$  und  $R_2$  Stützpunkte zwischen den in horizontaler Richtung liegenden Rasterpunkten.

Nach dem gleichen Verfahren wird anschliessend zwischen den Restklaffungen dieser beiden Stützpunkte interpoliert.

Quelle: [http://de.wikipedia.org/wiki/Bilineare\\_Filterung](http://de.wikipedia.org/wiki/Bilineare_Filterung)

Für einen beliebigen Punkt  $(x_i, y_i)$  ergibt sich die interpolierte Residue zu

$$R_i = a + bx_i + cy_i + dx_i y_i$$

Man setzt die Koordinaten  $x_i$  und  $y_i$  eines Stützpunktes und seine gegebene Residue  $R_{ij}$  in die Gleichung ein. Für die vier Punkte gibt es vier lineare Gleichungen. In Matrizen geschrieben:

$$\mathbf{A} \cdot \mathbf{t} = \mathbf{R} \quad \text{bzw. ausgeschrieben:} \quad \begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_1 & y_2 & x_1 y_2 \\ 1 & x_2 & y_1 & x_2 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} R_{11} \\ R_{12} \\ R_{21} \\ R_{22} \end{bmatrix}$$

Dieses Gleichungssystem löst man nach den (unbekannten) Parameter  $a$ ,  $b$ ,  $c$ ,  $d$  auf; sie sind im Vektor  $\mathbf{t}$  zusammengefasst.  $\Rightarrow \mathbf{t} = \mathbf{A}^{-1} \cdot \mathbf{R}$

Dann kann man für einen beliebigen Punkt  $(x,y)$  die Residue rechnen:  $R = \begin{bmatrix} 1 & x & y & xy \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$

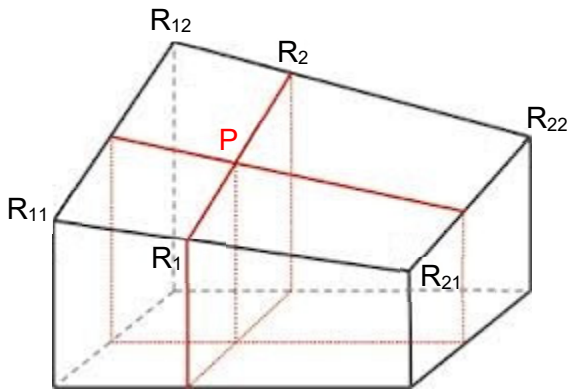


Abb. 2 Bilineare Interpolation einer NTV2-Masche (Quelle: <http://diegeodaeten.de/datumstransformation.html>)

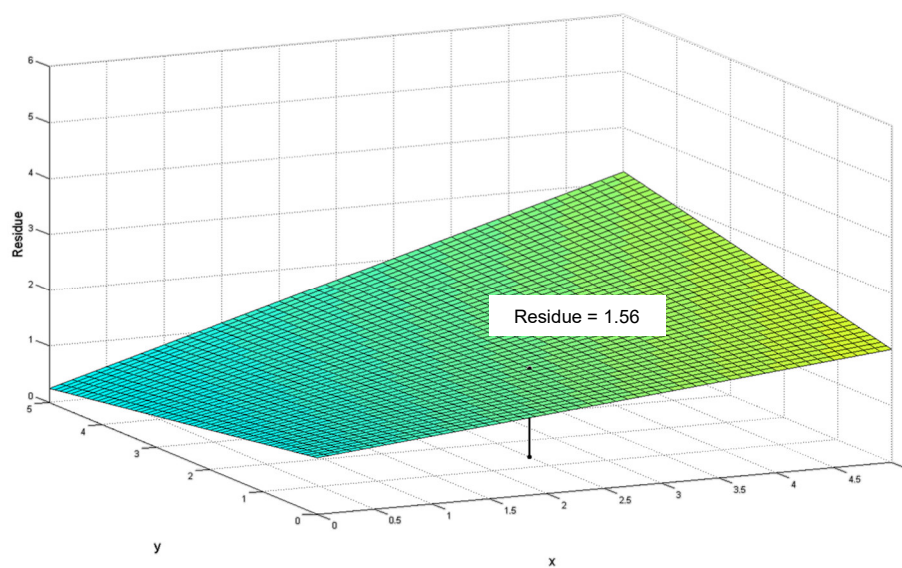


Abb. 3 bilineare Interpolationsfläche

Der Vorteil der bilinearen Interpolationsmethode liegt in ihrer Einfachheit, der Nachteil darin, dass sie am Rand der Rasterzelle abbricht: die Residuen gehen zu den benachbarten Rasterzellen nicht stetig über, sondern es gibt eine harte Bruchkante (siehe Abb. 4). Dies entspricht im Allgemeinen nicht dem realen Verhalten der Werte, wie sie durch die Interpolation modelliert werden sollen.

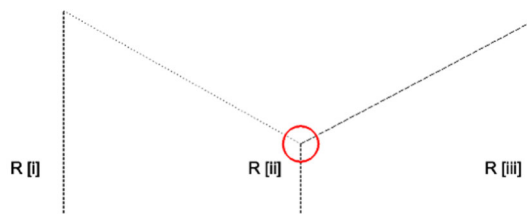


Abb. 4 Übergang der bilinearen Interpolationsfläche beim Maschenwechsel

Bei allen Bi-... Interpolationsarten muss das Rechenverfahren für die Restklaffungen der y- und der x-Koordinate gesondert durchgeführt werden.

## 4.2 Biquadratische Interpolation

Quelle: (Saner 2009)

Statt einer linearen Funktion wird ein Polynom 2. Grades verwendet:  $R_x = \sum_{i=0}^2 \sum_{j=0}^2 a_{ij} x^i y^j$

Gleich wie bei der bilinearen Methode lässt sich auch die Formel der biquadratischen Interpolation herleiten und für jeden Punkt  $(x_i, y_i)$  ergibt sich die Residue  $R_i$  zu:

$$R_i = a + bx_i + cy_i + dx_i^2 + ex_i y_i + fy_i^2 + gx_i^2 y_i + hx_i y_i^2 + ix_i^2 y_i^2$$

Hier braucht es neun Stützwerte  $R_A$  bis  $R_I$ , um eine Interpolationsfläche zu generieren (Abb. 5).

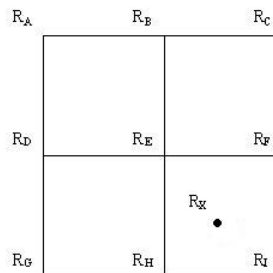


Abb. 5 Ausgangslage für eine biquadratische Interpolation

Für die neun bekannten Stützwerte ergibt sich folgendes Gleichungssystem:

$$\mathbf{A} \cdot \mathbf{t} = \mathbf{R} \quad \text{oder ausgeschrieben} \quad \begin{bmatrix} 1 & x_A & y_A & x_A^2 & x_A y_A & y_A^2 & x_A^2 y_A & x_A y_A^2 & x_A^2 y_A^2 \\ 1 & x_B & y_B & x_B^2 & x_B y_B & y_B^2 & x_B^2 y_B & x_B y_B^2 & x_B^2 y_B^2 \\ \vdots & & & & \dots & & & & \vdots \\ 1 & x_I & y_I & x_I^2 & x_I y_I & y_I^2 & x_I^2 y_I & x_I y_I^2 & x_I^2 y_I^2 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ \vdots \\ i \end{bmatrix} = \begin{bmatrix} R_A \\ R_B \\ \vdots \\ R_I \end{bmatrix}$$

aufgelöst ergibt sich der Lösungsvektor  $\mathbf{t} = \mathbf{A}^{-1} \cdot \mathbf{R} = [a \ b \ c \ d \ e \ f \ g \ h \ i]^T$ ,

daraus rechnet sich die Residue  $R_x$  eines Punktes  $(x, y)$  zu

$$R_x = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 & x^2 y & xy^2 & x^2 y^2 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ \vdots \\ i \end{bmatrix}$$



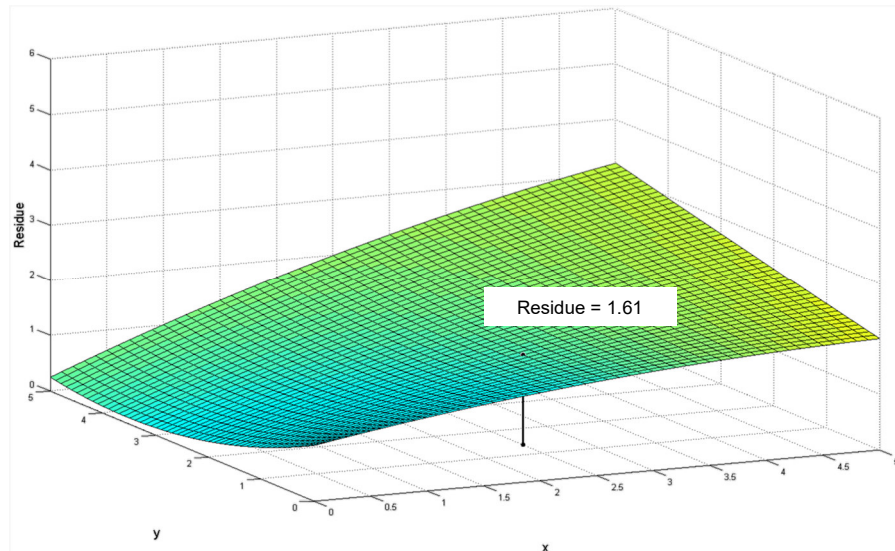


Abb. 6 biquadratische Interpolationsfläche

Durch die Verwendung von zusätzlichen Stützwerten verglichen mit der bilinearen Interpolation wird die eigentliche Interpolationsfläche dynamischer modelliert. Auf diese Weise entstehen bei den Übergängen der Interpolationsflächen zwischen Maschen weniger abrupte Wechsel (siehe Abb. 7), was dem wahren Verhalten der modellierten Werte (Residuen) eher entsprechen dürfte.

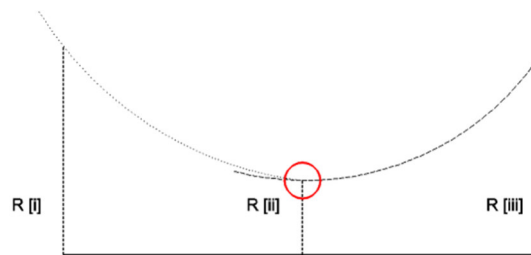


Abb. 7 Übergang der biquadratischen Interpolationsfläche beim Maschenwechsel

Diese Interpolationsart ist auch in der Transformationsempfehlung, Kap. 4.5.7 für den Fall  $i = j = 2$  beschrieben.

### 4.3 Bikubische Interpolation

Quelle: [http://en.wikipedia.org/wiki/Bicubic\\_interpolation](http://en.wikipedia.org/wiki/Bicubic_interpolation)

Die Interpolante  $R_x$  wird mit einem Polynom 3. Grades berechnet: 
$$R_x = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

Dabei braucht es 16 Stützwerte, um die vorerst unbekanntenen Koeffizienten  $a_{ij}$  zu bestimmen. Diese können sein:

- in Anwendungen der digitalen Bildverarbeitung:
  - die Residuen von 4 Rasterpunkten und
  - die 4 Werte der ersten Ableitung in x-Richtung (in jedem Rasterpunkt) und
  - die 4 Werte der ersten Ableitung in y-Richtung (in jedem Rasterpunkt) und
  - die 4 Werte der zweiten partiellen Ableitung in x- und y-Richtung (in jedem Rasterpunkt).
 Dies führt zu einer kontinuierlichen und glatten Interpolationsfläche auch über die einzelnen Rasterzellen hinweg. (Novak 1988)

- in geodätischen Anwendungen: die Residuen von 16 geodätischen Rasterpunkten

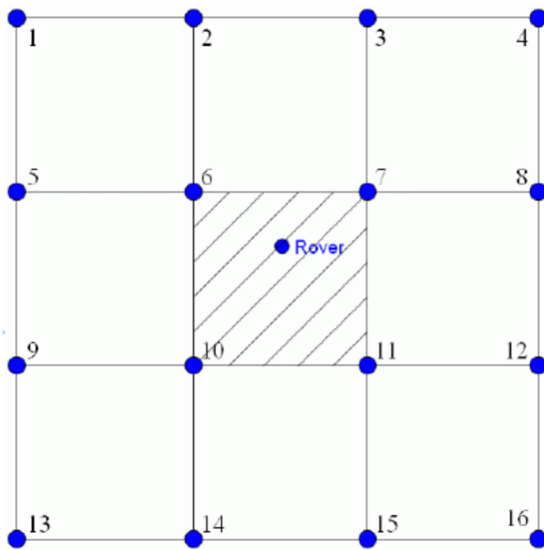


Abb. 8 Rasterpunkte mit Residuen, die zur biquadratischen oder bispline Interpolation der Interpolierenden im "Rover" Punkt verwendet werden dürfen. Quelle: (RTCM 10403.2 2013, S. 3-71-O)

Anstelle eines Polynoms 3. Grades nennt die Literatur auch kubische Splines. Ein kubischer Spline ist eine glatte Kurve, die durch die gegebenen Punkte im Koordinatensystem geht und eine minimale Gesamtkrümmung aufweist. Jedes Teilstück ist dabei durch eine kubische Parabel  $f(x) = a_i x^3 + b_i x^2 + c_i x + d_i$  mit geeigneten Koeffizienten  $a_i$ ,  $b_i$ ,  $c_i$  und  $d_i$  definiert. (Quelle: <http://www.arndt-bruenner.de/mathe/scripts/kubspline.htm>)

## 5. Literatur

- ALGLIB IDW-Interpolation (2012): inverse distance weighting interpolation/fitting. (= ALGLIB numerical analysis and data processing library). Online im Internet: URL: <http://www.alglib.net/interpolation/inversedistanceweighting.php> (Zugriff am: 26.10.2020).
- ALGLIB RBF Interpolation (2012): ALGLIB numerical analysis and data processing library. Fast RBF interpolation/fitting. Online im Internet: URL: <http://www.alglib.net/interpolation/fastrbf.php> (Zugriff am: 26.10.2020).
- Bundesamt für Landestopografie swisstopo (2017): Empfehlung für die Anwendung geometrischer Transformationsmethoden in der amtlichen Vermessung. 3. Aufl. Wabern: Geodäsie und Eidgenössische Vermessungsdirektion.
- Novak, Kurt (1988): Digitale Terrainmodelle, Interpolationsverfahren, Ableitung von Nachfolgeinformationen, Qualitätskontrolle. Photogrammetrische Verfahren im Zeichen der Reform der amtlichen Vermessung (RAV II). Zürich: ETH Zürich, Weiterbildungsveranstaltung IGP.
- Renka, Robert J (1988): „Multivariate Interpolation of Large Sets of Scattered Data.“ In: ACM Transactions on Mathematical Software, 14 (1988), 2, S. 139–148. Online im Internet: DOI: 10.1145/45054.45055
- RTCM 10403.2 (2013): RTCM Standard 10403.2 for Differential GNSS (Global Navigation Satellite Systems) Services - Version 3. RTCM, Arlington VA. Online im Internet: URL: <https://www.rtcn.org>
- Saner, Marco (2009): Echtzeit Bezugsrahmenwechsel mit swipos GIS/GEO. Bachelor Thesis. Fachhochschule Nordwestschweiz. Muttenz.
- Shepard, Donald (1968): „A two-dimensional interpolation function for irregularly-spaced data.“ In: Proceedings of the 1968 23rd ACM national conference. New York, NY, USA: ACM (= ACM '68), S. 517–524. Online im Internet: DOI: 10.1145/800186.810616
- Thacker, William I. u.a. (2010): „Algorithm 905: SHEPPACK: Modified Shepard Algorithm for Interpolation of Scattered Multivariate Data.“ In: ACM Transactions on Mathematical Software - TOMS, 37 (2010), 3, S. 1–20. Online im Internet: DOI: 10.1145/1824801.1824812

## 6. Anhang

Mailverkehr mit dem Autor des Programms GeoSuite vom 10./13.12.2012

Fragen Beat Sievers:

- Ist meine Vermutung richtig, dass Sie in GeoSuite/Transint die 'Modified Shepard's method' implementiert haben?
- Können Sie mir die Funktionen aus dem RBF ALGLIB Paket <sup>1</sup> nennen, die Sie in GeoSuite aufrufen?
- Können Sie mir evtl. den entsprechenden Auszug aus der GeoSuite Quellsoftware senden? Hilfreich wäre eine Zuordnung Bildschirm-Eingabe (Option) zum ALGLIB Prozedur.

Antwort Jérôme Ray (swisstopo):

- Oui, GeoSuite utilise « Modified Shepard's method » :  
<http://www.alglib.net/interpolation/inversedistanceweighting.php#header1>
- GeoSuite n'utilise pas RBF pour le moment.
- Ci-dessous les fonctions ALGLIB utilisées par GeoSuite :

```
// Initialize Shepard's interpolation
_shepardInterpolE = new alglib.idwinterpolant();
_shepardInterpolN = new alglib.idwinterpolant();

// Modified Shepard method for non-uniform datasets
if (_parameters.ShepardNodalFunction == -1)
{
    alglib.idwbuildmodifiedshepardr(inputPointsArrayE, nPts, 2, _parameters.ShepardRadius, out _shepardInterpolE);
    alglib.idwbuildmodifiedshepardr(inputPointsArrayN, nPts, 2, _parameters.ShepardRadius, out _shepardInterpolN);
}
else // Modified Shepard method for uniform datasets (standard)
{
    // nodalFuncType: 0=Constant, 1=Linear, -1=Fast Linear, 2=Quadratic
    alglib.idwbuildmodifiedshepard(inputPointsArrayE, nPts, 2, nodalFuncType, _parameters.ShepardNQ, _parameters.ShepardNW, out
    _shepardInterpolE);
    alglib.idwbuildmodifiedshepard(inputPointsArrayN, nPts, 2, nodalFuncType, _parameters.ShepardNQ, _parameters.ShepardNW, out
    _shepardInterpolN);

    // Get calculated/corrected parameters
    _parameters.ShepardNQ = _shepardInterpolE.innerobj.tree.kneeded;
    _parameters.ShepardNW = _shepardInterpolE.innerobj.nw;
}

...

// Interpolate points
for (int i = 0; i < inputPoints.Length; i++)
{
    double dE = alglib.idwcalc(_shepardInterpolE, new double[] { inputPoints[i].Coordinates[0], inputPoints[i].Coordinates[1] });
    double dN = alglib.idwcalc(_shepardInterpolN, new double[] { inputPoints[i].Coordinates[0], inputPoints[i].Coordinates[1] });

    ...
}
```

<sup>1</sup> (ALGLIB RBF Interpolation 2012); RBF = Radial Basis Functions