Paper-ID: VGI_200322

Netzwerk-Voronoi-Diagramme

Margot Graf [1], Stephan Winter [2]

[1] *Große Sperlgasse 41/28, 1020 Wien*
[2] *Institut für Geoinformation, Technische Universität Wien, Gusshausstr. 27-29, 1040 Wien*

VGI – Österreichische Zeitschrift für Vermessung und Geoinformation **91** (3), S. 166–174

2003

BibTEX:

```
@ARTICLE{Graf_VGI_200322,
Title =  {Netzwerk-Voronoi-Diagramme},
Author =  {Graf, Margot and Winter, Stephan},
Journal =  {VGI -- {\"O}sterreichische Zeitschrift f{\"u}r Vermessung und
   Geoinformation},
Pages =  {166--174},
Number =  {3},
Year =  {2003},
Volume =  {91}
}
```


Netzwerk-Voronoi-Diagramme

Margot Graf und Stephan Winter, Wien

Zusammenfassung

In dieser Arbeit wird das Netzwerk-Voronoi-Diagramm vorgestellt und seine Implementierung beschrieben. Dazu wird Dijkstra's kürzester-Wege-Algorithmus so modifiziert, dass er kürzeste Wegzeiten von mehreren Voronoi-Generatoren gleichzeitig berechnet. Auf diese Weise erhält man Partitionierungen der Knoten eines Netzwerks. Über diese Knoten-Netzwerk-Voronoi-Diagramme hinaus werden dann auch Kanten den gegebenen Generatoren zugeordnet. Für Kanten-Netzwerk-Voronoi-Diagramme wird eine spezielle Behandlung von Richtungen und unsymmetrischen Kosten im Netzwerk vorgeschlagen. Flächen-Netzwerk-Voronoi-Diagramme bleiben unberücksichtigt, da die betrachteten Anwendungen auf nicht-planaren Netzen beruhen. Drei Anwendungen demonstrieren den Nutzen solcher Netzwerk-Voronoi-Diagramme.

Abstract

This paper presents the Network-Voronoi-Diagram and describes its implementation. Dijkstra's shortest path algorithm is modified in way that it calculates shortest paths from several Voronoi generators at the same time. The result is a partition of the nodes of the network. Additionally the arcs of the network are attributed to the generators, considering especially their direction and unsymmetric costs. Faces are excluded in the diagrams because the considered networks are not planar. Three applications demonstrate the contribution of Network Voronoi Diagrams compared to Voronoi Diagrams.

1. Einleitung

Stellen Sie sich vor, Sie suchen für Ihre Tochter eine neue Schule. Unter den wichtigeren Kriterien befindet sich sicher die Länge des Schulweges. Als räumliches Suchproblem formuliert, suchen Sie, ausgehend von Ihrer Wohnung, die nächstgelegene Schule unter allen Schulen in Ihrer Stadt. Die klassische Methode zur Lösung dieses Problems stellt das Voronoi-Diagramm dar. Das Voronoi-Diagramm grenzt das zu jeder Schule nächstgelegene Gebiet ein; es wird üblicherweise auf einer ebenen Fläche in der L_2-Metrik bestimmt (Abb. 1). Ihre Wohnung findet sich also in genau einem (Einzugs-)Gebiet einer Schule.

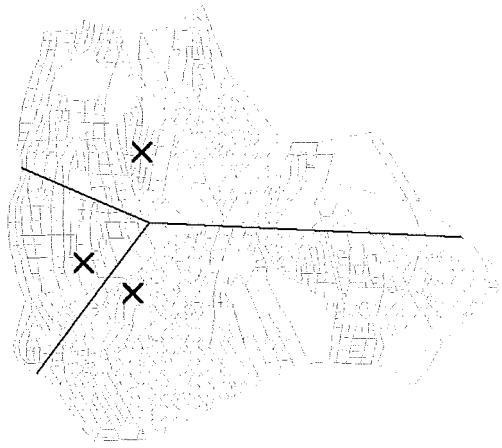


Abb. 1: Drei Schulen und ihre jeweiligen Einzugsgebiete in einem Voronoi-Diagramm.

Für manche Aufgaben der Einzugsgebietsplanung reicht diese Methode nicht aus. Sobald nämlich die Bewegungsmöglichkeiten auf ein Netzwerk eingeschränkt sind, stellt sie nur eine Näherungslösung dar. Mehrere Annahmen des Voronoi-Diagramms sind im städtischen Raum verletzt:

- Die Distanzen sind nicht euklidisch zu betrachten, sondern entlang des Verkehrsnetzwerkes. Muß Ihre Tochter um einen Häuserblock herumwandern, kann sich die Länge ihres Schulwegs selbst bei kurzer Luftlinie erheblich ändern.
- Die Distanzen können unsymmetrisch sein. Für Ihre Tochter ist bei Gefälle die eine Richtung beschwerlicher als die andere, und sollte sie den Bus benutzen, werden die Routen und damit die Fahrzeiten für Hin- und Rückweg durch Einbahnstraßenregelungen variieren.
- Die Distanzen können inhomogen sein. Ihre Tochter wird Teile des Schulwegs zu Fuß zurücklegen, andere Teile mit öffentlichen Verkehrsmitteln. Eine wenige Minuten lange U-Bahnfahrt überbrückt aber Distanzen, die zu Fuß zurückzulegen sehr viel länger brauchen würde

Welche Schule ist also wirklich die nächstgelegene, das heißt, welche Schule kann unter Berücksichtigung des städtischen Verkehrsangebots in kürzester Zeit erreicht werden? Um diese Frage zu lösen, wird hier und da in der Literatur ein Netzwerk-Voronoi-Diagramm vorgeschlagen. Das Netzwerk-Voronoi-Diagramm betrachtet allein die Kosten in einem Netz, hier also die Reisezeiten entlang des Verkehrsnetzes. Es teilt das Netz selbst in Voronoi-Gebiete ein, das heißt, in Mengen von Knoten und Kanten, die je einem Voronoi-Generator (hier: einer Schule) nächstgelegen sind (Abb. 2).

Abb. 2: Das Netzwerk-Voronoi-Diagramm von drei Generatoren auf einem Straßennetz.

Relevante Probleme für Netzwerk-Voronoi-Diagramme sind vielfältig. In einer Stadt spielen Einzugsgebiete zum Beispiel für Geschäfte, für Infrastruktureinrichtungen oder für die Rettung eine wichtige Rolle. Die Anwendungen lassen sich in drei Gruppen klassifizieren:

- Im statischen Netzwerk-Voronoi-Diagramm werden Fragen nach dem nächstgelegenen Generator beantwortet. Welche Schule ist die nächstgelegene zur Wohnung? Oder invers: welche Rettungsstation ist die zum Unfall nächstgelegene?.
- Das Netzwerk-Voronoi-Diagramm kann Veränderungen für planerische Zwecke simulieren. Zur Auswahl eines neuen Standorts wird in ein Netzwerk-Voronoi-Diagramm ein neuer Generator eingefügt (zur Schliessung gelöscht). Zur Optimierung des Standorts wird die Lage des Generators variiert, und die (lokalen oder auch globalen!) Veränderungen des Diagramms betrachtet. Fragen wären: Wie verändern sich Einzugsgebiete, wenn eine weitere Schule eröffnet? Wie verändern sich Einzugsgebiete, wenn eine Baustelle den Verkehr demnächst für einige Monate beeinträchtigt?
- Das Netzwerk-Voronoi-Diagramm kann dynamisch auf Realzeit-Probleme reagieren. Falls in einem Netzwerk Knoten oder Kanten ausfallen, oder ihre Gewichte sich ändern, wie verändern sich Einzugsbereiche? Wenn zum Beispiel in einem Gebäude ein Notausgang durch Feuer blockiert ist, zu welchem Notausgang soll man nun von welchem Bereich des Gebäudes aus leiten?

In dieser Arbeit wird das Netzwerk-Voronoi-Diagramm vorgestellt und seine Implementierung beschrieben. Dazu wird Dijkstra's Kürzester-Wege-Algorithmus so modifiziert, dass er kürzeste Wegzeiten von mehreren Generatoren gleichzeitig berechnet. Auf diese Weise berechnen wir Wälder von kürzesten-Wege-Bäumen, die eine Partitionierung der Knoten des Netzwerks bilden. Über diese Knoten-Netzwerk-Voronoi-Diagramme hinaus ordnen wir auch Kanten den gegebenen Generatoren zu. Für Kanten-Netzwerk-Voronoi-Diagramme werden Richtungen und unsymmetrische Kosten im Netzwerk behandelt. Flächen-Netzwerk-Voronoi-Diagramme bleiben unberücksichtigt, da die betrachteten Verkehrsnetze nicht planar sind. Drei Anwendungen demonstrieren den Nutzen solcher Netzwerk-Voronoi-Diagramme. Der Algorithmus und die flexible Behandlung von Kanten sind in dieser Weise bisher nicht publiziert worden.

2. Literatur

Im folgenden Abschnitt werden die Unterschiede der Berechnung zwischen dem klassischen Voronoi-Diagramm und einem Netzwerk-Voronoi-Diagramm dargestellt. Die relevante Literatur wird dabei berücksichtigt.

2.1. Voronoi-Diagramme

Voronoi-Diagramme [1], [2] werden oft im Zusammenhang mit dem sogenannten *Post-Office-Problem* beschrieben [3]. Auf einer ebenen Fläche gibt es mehrere Postämter. Jedem Postamt soll nun ein Gebiet zugeordnet werden, so dass für jeden Punkt innerhalb dieses Gebietes die Distanz zum zugehörigen Postamt kleiner als zu einem anderen Postamt ist. Das Postamt entspricht einem Voronoi-Generator, die Grenzen der einzelnen Gebiete stellen ein Voronoi-Diagramm dar. Für die Einteilung der Gebiete kann man entsprechend der Aufgabenstellung

verschieden definierte Distanzen verwenden, zum Beispiel die *Euklidische Distanz* (L_2-Metrik) oder die *Manhattan-Distanz* (L_1-Metrik).

Die klassische und einfachste Form des Voronoi-Diagrammes sieht man in Abb. 1. Hier wurde als zugrundeliegende Metrik der Fläche die L_2-Metrik verwendet. In der L_2-Metrik wird die Distanz für zwei Punkte A und E nach der Formel von Pythagoras (2.1) berechnet. Diese Distanz wird als euklidische Distanz bezeichnet:

$$L_2(A,E) = \sqrt[2]{(x_A - x_E)^2 + (y_A - y_E)^2} \qquad (2.1)$$

Diese Wahl bedeutet, dass man sich auf der Fläche in alle Richtungen mit gleicher Geschwindigkeit fortbewegt. Die Punkte gleicher Distanz von einem Startpunkt liegen daher auf einem Kreis. Die Größe dieses Kreises ist abhängig von der Geschwindigkeit, mit der man sich von einem Punkt entfernt und der Zeit, die seit dem Start vergangen ist. Daher wird für diesen Kreis die Bezeichnung Isochrone verwendet. Isochrone verbinden Punkte gleicher zeitlicher Distanz von einem oder mehreren Startpunkten. Zeichnet man für die Voronoi-Generatoren die Isochronen für einen bestimmten Zeitpunkt ein, so entstehen dort, wo sich Isochrone zweier Generatoren treffen, Kanten des Voronoi-Diagrammes (Abb. 3).

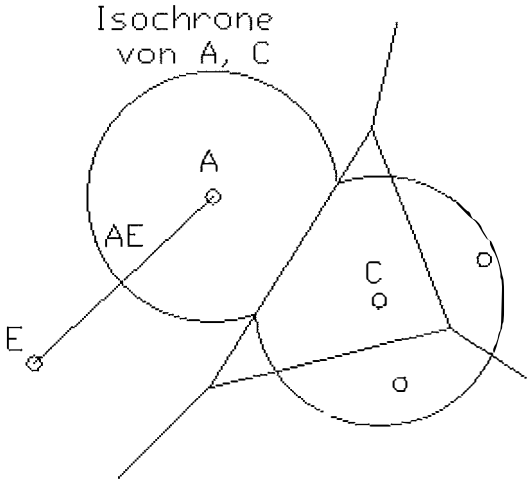


Abb. 3: Die euklidische Distanz AE und die Isochronen für die zwei Voronoi-Generatoren A und C.

Diese Art der Berechnung des Voronoi-Diagrammes ist häufig in Geographischen Informationssystemen implementiert, obwohl die in einem Verkehrsnetz tatsächlich auftretenden Reisekosten, die typischerweise von der L_2-Metrik erheblich abweichen, nicht adäquat berücksichtigt werden. Je inhomogener die Netzstruktur oder die Gewichte im Netz sind, desto schlechter wird die Näherung der realen Kosten durch ein Voronoi-Diagramm. Im Kapitel 4, den Anwendungen, wird der Unterschied zwischen einem Voronoi-Diagramm und einem Netzwerk-Voronoi-Diagramm, das die tatsächlich im Netzwerk herrschenden Kosten berücksichtigt, gezeigt.

Die Aufteilung der Fläche, wie sie beim klassischen Voronoi-Diagramm vorgenommen wird, ist in Verkehrsnetzen oft nicht sinnvoll. Verkehrsnetze sind in der Regel nicht planar, haben also nicht immer von Kanten eingeschlossene Flächen. Ferner ist der Zugang zum Netzwerk eingeschränkt und nur an bestimmten Stellen möglich. Für das Straßennetz sind es Häuserblöcke und -eingänge, im öffentlichen Verkehrsnetz sind es die Haltestellen, an denen ein Betreten und Verlassen des Netzes möglich ist. Desweiteren sollen hier verschiedene Fortbewegungsmodi und -geschwindigkeiten modelliert werden. Daher wird das Netzwerk-Voronoi-Diagramm vorgestellt.

2.2. Netzwerk-Voronoi-Diagramme

Ein Netzwerk ist durch die Angabe von *Knoten* und ihren gewichteten und gerichteten Verbindungen, den *Kanten*, definiert. Das Augenmerk bei der Bildung eines Netzwerk-Voronoi-Diagrammes liegt daher nicht auf der Aufteilung der Fläche, sondern dem Aufteilen der Knoten und Kanten zu den Voronoi-Generatoren. Generell gibt es in einem Netzwerk drei verschiedene Arten von Voronoi-Diagrammen [4].

1. Das Knoten-Netzwerk-Voronoi-Diagramm: Hier wird jeder Netzwerk-Knoten dem zugehörigen Generator zugeteilt, sodass die Kosten im Netzwerk von einem Knoten zu dem zugehörigen Generator kleiner ist als zu einem anderen Generator.
2. Das Kanten-Netzwerk-Voronoi-Diagramm: Hier wird jede Kante dem zugehörigen Generator zugeteilt, oder gegebenenfalls so geteilt, dass die Distanz von einem Punkt auf der Kante zum zugehörigen Generator kleiner ist als zu einem anderen Generator.
3. Das Flächen-Netzwerk-Voronoi-Diagramm: Die Begrenzung des Voronoi-Diagramms in der Fläche trennt die Punkte mit minimaler Distanz zu den nächstgelegenen Kanten unterschiedlicher Generatoren. Dieses Diagramm stimmt in der Regel nicht mit dem (flächenhaften) Voronoi-Diagramm der L2-Metrik überein, wie es oben beschrieben wurde. Es kann in einem Netzwerk nur dann bestimmt werden, wenn das Netzwerk planar ist.

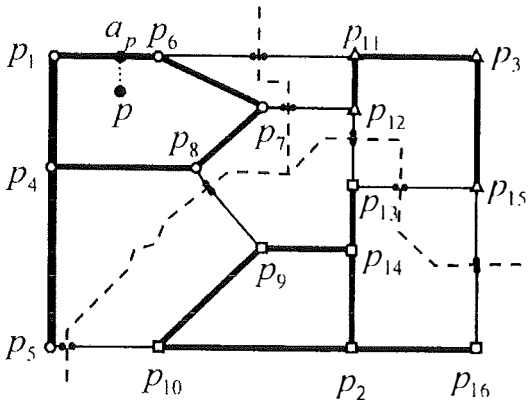


Abb. 4: Die Aufteilung der Knoten, Kanten und Flächen zu den 3 Voronoi-Generatoren p_1, p_2 und p_3 (aus [4]).

Für die Bildung des Netzwerk-Voronoi-Diagrammes verwendet man einen Algorithmus, der auf dem Kürzesten-Wege-Algorithmus von Dijkstra beruht [5], [6], [7]. Der Dijkstra-Algorithmus bestimmt in einem zusammenhängenden Netz die kürzesten Wege von einem ausgewählten Knoten zu allen übrigen Knoten [8]. Er muss für unsere Zwecke also so modifiziert werden, dass er kürzeste Wege von mehreren Generatoren gleichzeitig berechnet. Jeder Knoten im Netzwerk erhält so eine Wegzeit zum nächstgelegenen Generator, und es entsteht eine Menge von nicht-überlappenden, aber vollständigen Kürzeste-Wege-Bäumen.

In Abbildung 4 wird davon ausgegangen, dass das Gewicht einer Kante identisch ist mit der euklidischen Distanz von Anfangs- und Endknoten. Das gezeigte Netzwerk ist außerdem ungerichtet, das heißt eine Kante repräsentiert eine symmetrische Verbindung der Knoten in beide Richtungen. Für das Kanten-Netzwerk-Voronoi-Diagramm wird daher bei Okabe et al., wenn Anfangs- und Endknoten einer Kante zwei verschiedenen Voronoi-Generatoren angehören, auf dieser Kante der Punkt bestimmt, von dem die euklidische Distanz zu den zwei Generatoren gleich ist. An diesem Punkt entsteht eine Begrenzung des Kanten-Netzwerk-Voronoi-Diagrammes. In dieser Arbeit wird diese Methode erweitert, um auch mit Netzen adäquat umzugehen, in denen Kanten gerichtet oder Kosten unsymmetrisch sind. So kann man jetzt zum Beispiel die unterschiedlichen Geschwindigkeiten verschiedener Fortbewegungsmodi innerhalb eines Netzes oder Einbahnstraßenregelungen modellieren.

Die Bildung des Kanten-Netzwerk-Voronoi-Diagrammes ist für räumliche Fragestellungen in der Regel ausreichend. Durch die Einteilung einer Kante zu zugehörigen Voronoi-Generatoren ist z. B. in einem Straßennetz auch die Zugehörigkeit für jedes Gebäude entlang einer Straße, bzw. in einem Gebäudenetz die Zugehörigkeit für jedes Zimmer entlang eines Ganges bestimmt. Somit kann für jeden Einstiegs- und Ausstiegspunkt aus dem Netz der Voronoi-Generator bestimmt werden.

3. Methode

In diesem Kapitel werden die Modifikationen des Dijkstra-Algorithmus gezeigt, mit deren Hilfe Knoten- und Kanten-Netzwerk-Voronoi-Diagramm berechnet werden können.

3.1. Berechnung des Knoten-Netzwerk-Voronoi-Diagramms

Wir berechnen das Knoten-Netzwerk-Voronoi-Diagramm für Generatoren, die eine Untermenge der Knoten des Netzes darstellen. Für die Zuordnung der übrigen Knoten zu den Voronoi-Generatoren wurde der Dijkstra-Algorithmus modifiziert.

Der modifizierte Algorithmus berechnet von jedem Voronoi-Generator die (Reise-)Kosten zu benachbarten Knoten. Er weist diesen Knoten die gefundenen Kosten zu. In den folgenden Iterationen wird jeweils der Knoten mit niedrigsten Kosten als ‚gefunden' markiert und von ihm die Berechnung der Kosten zu benachbarten Knoten fortgesetzt. Wurde einer der benachbarten Knoten bereits früher von einem anderen Knoten erreicht, so wird ihm der niedrigere Wert der zwei

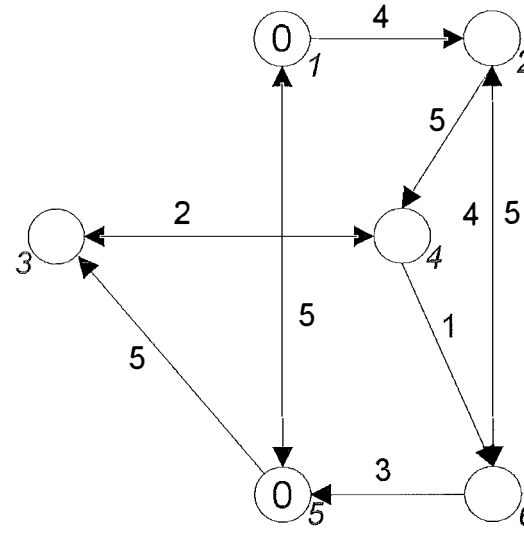


Abb. 5: Beispiel eines einfachen Netzwerkes.

berechneten Kosten zugewiesen. Der Algorithmus endet, wenn alle Knoten gefunden worden sind. Um jedem Knoten einen Voronoi-Generator zuordnen zu können, wurde der Algorithmus so modifiziert, dass bei jedem Markieren eines Knoten festgestellt wird, von welchem Knoten er gefunden worden ist, und welchem Voronoi-Generator der Vorgängerknoten zugeordnet war. Dem Knoten wird dann ebenfalls dieser Voronoi-Generator zugeordnet.

Man betrachte das folgende Beispiel. Gegeben ist ein Netzwerk mit sechs Knoten und elf Kanten, dargestellt in Abb. 5. Gesucht ist die Einteilung der Knoten zu den zwei gegebenen Voronoi-Generatoren 1 und 5.

Der Aufbau dieses Netzwerkes mit Hilfe einer Adjazenzmatrix dargestellt. Das Gewicht einer Kante *k*, die vom Knoten *m* zu *n* führt, steht in der Adjazenzmatrix an der Stelle [*m,n*]. Gibt es keine Verbindung von *m* zu *n* steht in der Adjazenzmatrix an der Stelle [*m,n*] der Wert unendlich:

$$Adjazenzmatrix = \begin{bmatrix} \infty & 4 & \infty & \infty & 5 & \infty \\ \infty & \infty & \infty & 5 & \infty & 5 \\ \infty & \infty & \infty & 2 & \infty & \infty \\ \infty & \infty & 2 & \infty & \infty & 4 \\ 5 & \infty & 5 & \infty & \infty & \infty \\ \infty & 4 & \infty & \infty & 3 & \infty \end{bmatrix}$$

Für den Algorithmus werden die drei Vektoren *Distanz, Knoten*, und *KV (= Knotenvoronoi)* initialisiert. Der Vektor *Distanz* enthält die momentan bekannten Kosten von den nächstgelegenen Generatoren zu den Knoten. Zur Initialisierung werden die Kosten zu den Generatoren selbst mit 0 angesetzt, und alle übrigen Kosten sind noch unbekannt, also unendlich. Der (binäre) Vektor *Knoten* enthält die Information, ob für einen Knoten die minimale Distanz bereits gefunden worden ist; dann ändert sich der Wert an der entsprechenden Stelle von 1 zu 0. Der Vektor *KV* enthält für jeden Knoten den zugehörigen Voronoi-Generator. Zur Initialisierung sind nur die Generatoren selbst bekannt. Für das Beispiel lauten die drei initialisierten Vektoren:

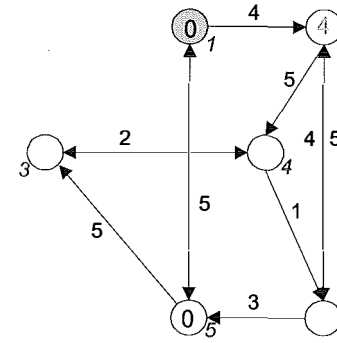
Distanz = [0 ∞ ∞ ∞ 0 ∞]

Knoten = [1 1 1 1 1 1]

KV = [1 ∞ ∞ ∞ 5 ∞]

Der Algorithmus wiederholt nun die oben beschriebenen Schritte, bis der Vektor *Distanz* keine endlichen Werte mehr enthält und alle Knoten gefunden worden sind. Für das hier gezeigte Netzwerk endet die Berechnung nach sechs Durchläufen. Für jeden Durchlauf wird zuerst der Knoten *minindex* mit dem minimalen Wert im Vektor *Distanz* und die ihm benachbarten Knoten j (aus der Adjazenzmatrix) bestimmt. Für die Knoten *minindex* und *j* muss außerdem gelten, dass ihr Wert in *Knoten* 1 beträgt.

1. Durchlauf: *minindex* = 1, *j* = {2, 5}

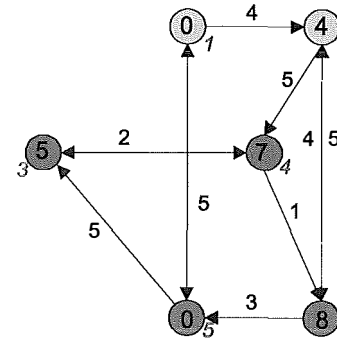


Distanz = [0 4 ∞ ∞ 0 ∞]

Knoten = [0 1 1 1 1 1]

KV = [1 1 ∞ ∞ 5 ∞]

Nach sechs Durchläufen erhält man folgendes Ergebnis:



Distanz = [0 4 5 7 0 8]

Knoten = [0 0 0 0 0 0]

KV = [1 1 5 5 5 5]

Nachdem so das Knoten-Netzwerk-Voronoi-Diagramm berechnet wurde, kann nun das Kanten-Netzwerk-Voronoi-Diagramm bestimmt werden.

3.1.1. Berechnung des Kanten-Netzwerk-Voronoi-Diagramms

Jede Kante wird nach dem Voronoi-Generator ihres Anfangs- und Endknotens untersucht. Für die Einteilung einer Kante zu einem Voronoi-Generator müssen daher vier verschiedene Fälle berücksichtigt werden (Abb. 6):

1. Anfangs- und Endknoten der Kante gehören zum gleichen Voronoi-Generator im Knoten-

Netzwerk-Voronoi-Diagramm. Dann wird die gesamte Kante ebenfalls diesem Voronoi-Generator zugeordnet.

2. Anfangs- und Endknoten gehören zu verschiedenen Voronoi-Generatoren.
   a) Die Kante ist nur in einer Richtung begehbar, das heißt es gibt keine Verbindung vom End- zum Anfangsknoten. Dann wird die gesamte Kante dem Voronoi-Generator des Anfangsknotens zugeteilt.
   b) Die Kante ist in zwei Richtungen zu selben Kosten begehbar (oder es gibt eine Kante vom Anfangs- zum Endknoten und eine Kante vom End- zum Anfangsknoten mit selbem Gewicht), und zusätzlich ist das Betreten des Netzes nur am Anfangs- bzw. Endknoten möglich. Solche Kanten treten zum Beispiel im öffentlichen Verkehrsnetz auf, wo Anfangs- bzw. Endknoten die Haltestellen darstellen. Dann wird die gesamte Kante dem Voronoi-Generator des Knotens zugeteilt, der die niedrigeren Kosten von einem Voronoi-Generator besitzt.
   c) Die Kante ist in zwei Richtungen begehbar, allerdings möglicherweise zu unterschiedlichen Kosten, und das Betreten der Kante ist an jeder Stelle möglich. Dann muss die Kante an jenem Punkt geteilt werden, an dem die Kosten von den jeweiligen Voronoi-Generatoren gleich groß sind. Eine Teilstrecke x_A wird dem Voronoi-Generator des Anfangsknotens zugeteilt, die andere Teilstrecke x_E dem Voronoi-Generator des Endknotens. Am Teilungspunkt wird ein Knoten zugefügt, der keinem Generator zugeteilt ist.

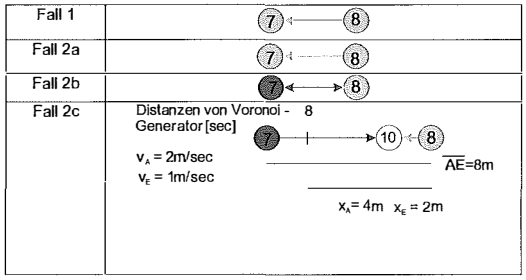


Abb. 6: Die Einteilung einer Kante zu den Voronoi-Generatoren ihrer beiden Knoten.

Man betrachte wiederum das Netz aus Abbildung 5. Für dessen sechs Knoten wurde bereits das Knoten-Netzwerk-Voronoi-Diagramm ermittelt und im Vektor *KV* festgehalten. Im Vektor *Distanz* sind die Kosten von den Voronoi-Generatoren zu den zugehörigen Knoten enthalten. Die Matrix *Koordinaten* enthält die Koordinaten der Knoten.

$$Koordinaten = \begin{bmatrix} 2 & 10 \\ 5 & 10 \\ \dots & \dots \\ \dots & \dots \\ 2 & 0 \\ 5 & 0 \end{bmatrix}$$

Distanz = [0 4 5 7 0 8]
KV = [1 1 5 5 5 5]

In diesem Netzwerk gibt es elf Kanten. Die Anfangs- und Endknoten dieser Kanten sind in den Vektoren *A* und *E* festgehalten. *A* und *E* werden folgendermaßen gebildet: Die Adjazenzmatrix wird Zeile für Zeile nach endlichen Werten untersucht. Zum Beispiel steht der erste endliche Wert an der Stelle [1, 2], daher ist *A*[1] = 1, und *E*[1] = 2 – die erste Kante führt daher vom Punkt 1 zum Punkt 2. Nachdem die Vektoren *A* und *E* gebildet sind, kann man die Vektoren *KV*[*A*] und *KV*[*E*] bilden. Sie enthalten die Voronoi-Generatoren der Anfangs- bzw. Endknoten.

k = 11
A = [1 1 2 2 3 4 4 5 5 6 6]
E = [2 5 4 6 4 3 6 1 3 2 5]
KV[*A*] = [1 1 1 1 5 5 5 5 5 5 5]
KV[*E*] = [1 5 5 5 5 5 5 1 5 1 5]
Kantenvoronoi = [∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞]

Die Berechnung erfolgt nacheinander für *i = 1, 2,..., k.* Der Vektor *Kantenvoronoi* wird daher zuerst an seiner ersten Stelle berechnet, dann an der zweiten und so fort.

Als Ergebnis erhält man

- für jede Kante den zugehörigen Voronoi-Generator, ersichtlich im Vektor *Kantenvoronoi:*
  Kantenvoronoi = [1 1 1 1 5 5 5 5 5 5 5]
- den geänderten Vektor *E*, indem die neuen Endpunkte für geteilte Kanten eingetragen sind:
  E = [2 7 4 8 4 3 6 7 3 8 5]
- den geänderten Vektor Distanz, der die Distanz für die neuen Punkte enthält:
  Distanz = [1 0 4 5 7 2 0 8 2.5 8.5]
- den geänderten Vektor Koordinaten, der auch die neu berechneten Koordinaten dieser Endpunkte enthält:

$$Koordinaten = \begin{bmatrix} 2 & 10 \\ 5 & 10 \\ \dots & \dots \\ \dots & \dots \\ 2 & 0 \\ 5 & 0 \\ 2 & 5 \\ 5 & 1.1 \end{bmatrix}$$

Schliesslich erhält man als Ergebnis das Kanten-Netzwerk-Voronoi-Diagramm in Abbildung 7.

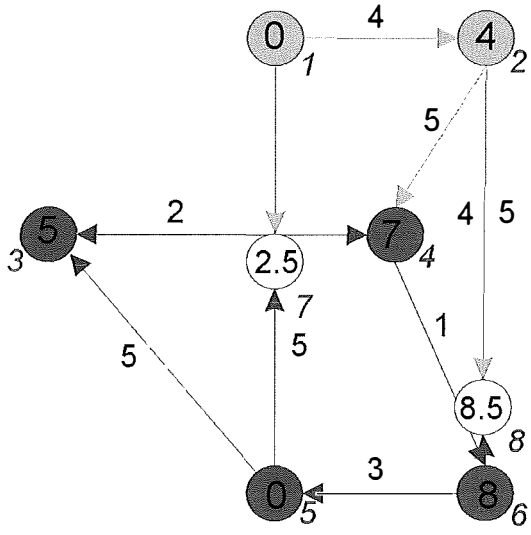


Abb. 7: Das Kanten-Netzwerk-Voronoi-Diagramm zum Netz in Abbildung 5.

4. Implementierung und Tests

Der oben beschriebene Algorithmus zur Berechnung von Knoten- und Kanten-Netzwerk-Voronoi-Diagramm wurde in IDL implementiert, der Interactive Data Language von Research Systems, Inc.. Diese Programmiersprache eignet sich besonders für *Rapid Prototyping* und Testen von Algorithmen, die auf die beigefügte Bibliothek von Matrizen- und Bildverarbeitungsoperationen zurückgreifen. Im vorliegenden Fall waren die Berechnung des klassischen Voronoi-Diagramms, die Isochronen-Berechnung und die einfache Visualisierung des Netzes der Bibliothek entnommen.

Für Tests der Implementierung sind zwei Datensätze herangezogen worden:

- Ein Straßennetzwerk der Innenbezirke von Wien war am Institut vorhanden. Zusätzlich wurden in dieses Netzwerk die U-Bahnlinien U1 bis U4 integriert, um den Algorithmus auch in einem öffentlichen Verkehrsnetzwerk zu testen.
- Ein Netzwerk der Flure des Universitätsgebäudes Gusshausstraße 27–29 wurde aus vorhandenen 2D-CAD-Dateien der einzelnen Stockwerke modelliert, die freundlicherweise von der Bundesimmobilienverwaltung zur Verfügung gestellt wurden.

In Abbildung 8 wurde für vier Voronoi-Generatoren das Netzwerk-Voronoi-Diagramm des Straßennetzes berechnet und durch verschiedenfärbige Gebiete dargestellt. Zusätzlich wurde das klassische Voronoi-Diagramm mit einer schwarzen Linie eingezeichnet. Die unterschiedlichen Graustufen stellen die Reisezeiten von einem Generator zu den Knoten dar: Punkte innerhalb des weißen Gebietes erreicht man in einer Minute, die nächsten Graustufen stellen Intervalle von 6, 11, 16, ... Minuten dar. Dabei wurde eine durchschnittliche Geschwindigkeit von 20 km/h angenommen.

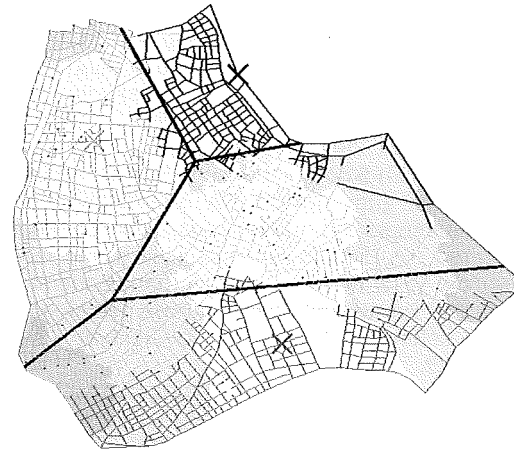


Abb. 8: Netzwerk-Voronoi-Diagramm (farbig) und Voronoi-Diagramm (schwarze Linien) von vier Generatoren.

Der Unterschied zwischen dem klassischen Voronoi-Diagramm und einem Netzwerk-Voronoi-Diagramm wird besonders deutlich, wenn ein Voronoi-Generator in einem inhomogenen Teil des Straßennetzes liegt: Der blaue Generator in Abbildung 8 liegt in einem Gebiet, das im Westen durch Einbahnen und im Osten durch ein Gebiet mit wenig Straßen erschlossen ist. Dadurch verschlechtert sich die Erreichbarkeit der benachbarten Gebiete, und das Einzugsgebiet verkleinert sich zugunsten des orangefarbenen Generators.

In Abbildung 9 wurden für die Standorte von fünf Wiener Schulen die Einzugsgebiete in einem Straßennetz für Fußgänger mit der Möglichkeit der Benützung der U-Bahn berechnet. Die Graustufen stellen Reisezeitintervalle von jeweils fünf Minuten dar. Man erhält die maximale Entfernung von einem Punkt in einem Netzwerk zu 30 Minuten, die durchschnittliche Reisezeit beträgt 14,4 Minuten. Zusätzlich kann man auch die Größe des Einzugsgebietes berechnen: Dem roten Generator sind zum Beispiel 98km Straßenlänge zugeordnet. Das klassische Voronoi-Diagramm ist in Abbildung 9 noch gezeigt, obwohl das Netz hier nicht mehr planar und durch verschiedene Fortbewegungsmodi stark verzerrt ist.

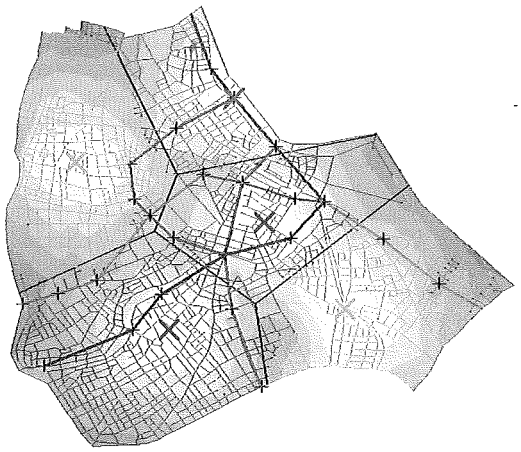


Abb. 9: Netzwerk-Voronoi-Diagramm (farbig) und Voronoi-Diagramm (schwarze Linien) von fünf Generatoren im symmetrischen Fußgängernetz mit U-Bahnen (farbige Linien).

In einem Gebäudenetzwerk spielt die Berechnung des Netzwerk-Voronoi-Diagrammes z. B. im Katastrophenfall eine wichtige Rolle. Angenommen im Institutsgebäude in der Gusshausstrasse bricht beim Haupteingang ein Feuer aus, und Rauch und Feuer breiten sich rasch aus. Das Gebäude muss schnellstens evakuiert werden, aber der Bereich des Haupteingangs ist durch das Feuer gesperrt. Welche Ausgänge sind für die Personen, die sich noch im Gebäude aufhalten, die nächsten?

Aus Abbildung 10 erkennt man für jeden Punkt im Gebäude den nächstgelegenen Ausgang. Durch die Sperre des Haupteinganges wird vor allem das Einzugsgebiet des schwarzfarbenen Ausgangs vergrößert. Im Normalfall ist er für 308m (15%) der Gänge im Gebäude der nächstgelegene. Wenn der Haupteingang versperrt ist erhöht sich der Wert auf 548m (27%).

Wege aus den Zimmern und Wege zwischen den Zimmern sind in diesem Netz vernachlässigt. Für dieses (nicht-planare) Netzwerk lässt sich kein sinnvoller Vergleich mit einem klassischen Voronoi-Diagramm mehr ziehen.

5. Diskussion und Ausblick

Das klassische Voronoi-Diagramm wurde für Netzwerke mit Knoten und Kanten übertragen. Der Kürzeste-Wege-Algorithmus von Dijkstra konnte für die Berechnung des Netzwerk-Voronoi-Diagrammes modifiziert werden. Der modifizierte Algorithmus berechnet die kürzesten Wegzeiten von ausgewählten Knoten – den Voronoi-Generatoren – zu allen anderen Knoten im Netzwerk und findet für jeden Knoten den nächstgelegenen Generator. Für die Punkte entlang einer Kante des Netzwerkes konnte ebenfalls der nächstgelegene Generator berechnet werden. Somit war das Voronoi-Diagramm in einem Netzwerk für die Knoten und Kanten vollständig bestimmt. Die Modifikation wurde an einem Beispiel erläutert und zum Testen implementiert.

Anwendungen ergeben sich vor allem für inhomogen gewichtete oder multi-modale Netzwerke, für die ein erheblicher Genauigkeitsgewinn gegenüber dem klassischen Voronoi-Diagramm erreicht wird, und für dreidimensionale Netzwerke. Drei Beispiele für solche Netzwerke

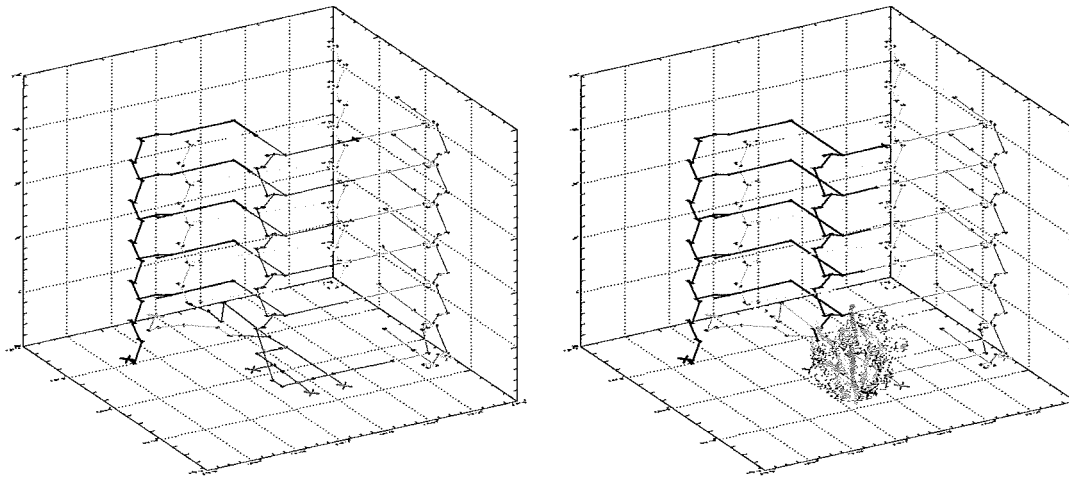


Abb. 10: Netzwerk-Voronoi-Diagramm für ein Gebäudenetz. Links: Generatoren an jedem Ausgang. Rechts: ein Generator fällt aus; das Netzwerk-Voronoi-Diagramm verändert sich daraufhin stark.

sind ein Straßennetz, ein öffentliches Verkehrsnetz und ein Gebäudenetz. In diesen Netzen wurden einige der möglichen Anwendungen gezeigt: die Berechnung des Einzugsgebietes von öffentlichen Einrichtungen im Straßennetz, die Dauer von Schulwegen bei vorgegebenen Schulstandorten und verschiedenen Fortbewegungsmöglichkeiten, und die Evakuierung eines Gebäudes im Katastrophenfall. Die Ergebnisse konnten anschaulich gezeigt werden, für das planare Straßennetz auch im Vergleich mit einem klassischen Voronoi-Diagramm.

Der Algorithmus kann auf jedes Netzwerk angewendet werden. An seiner Implementierung könnte man einige Änderungen anbringen, die die Laufzeit des Algorithmus verbessern. Es sollte auch berücksichtigt werden, dass es in einem Straßennetzwerk Generatoren geben kann, die sich nicht an einem Knoten befinden, und dass es dort offene Gebiete, wie große Plätze und Parks, ohne Knoten und Kanten gibt.

Für jeden Punkt entlang einer Kante wurde der nächstliegende Generator berechnet. Umgekehrt wurden aber die Generatoren immer an der Stelle eines Knoten angenommen. Die Möglichkeit, dass ein Voronoi-Generator auf einer Kante liegen kann, wurde nicht berücksichtigt. Der Algorithmus könnte leicht so modifiziert werden, dass man auch den Generator als Punkt auf einer Kante bestimmt, die Distanz zum Endknoten (oder zu den zwei Endknoten) dieser Kante berechnet, und dann den Algorithmus mit der so gewonnenen Distanz startet. Eine zweite Möglichkeit wäre, an der Stelle eines Generators einen zusätzlichen Knoten einzufügen und von ihm die Verbindungen zu den Endknoten neu zu definieren und in die Adjazenzmatrix einzutragen.

Für Gebiete innerhalb einer Stadt, in denen man sich in jede beliebige Richtung fortbewegen kann, wie zum Beispiel auf großen Plätzen, könnte eine Berechnung des klassischen Voronoi-Diagrammes diese Gebiete aufteilen und den Generatoren zuordnen. Die Zuordnung könnte man so vornehmen: Zu einem Platz führen verschiedene Straßen. Für den Knoten einer Straße, der an den Platz grenzt, wurde die Distanz zu einem Generator bereits berechnet. Von allen Knoten des Netzwerkes, die am Rande eines Platzes sind, starten die Isochronen zu verschiedenen Zeitpunkten, und es ergeben sich für die Schnittlinien der Isochrone von verschiedenen Generatoren hyperbolische Begrenzungen des Voronoi-Diagramms.

Berechnet man das Netzwerk-Voronoi-Diagramm mit dem beschriebenen Algorithmus, könnte man gleichzeitig die kürzesten Wege von den einzelnen Knoten und Kanten des Netzwerkes zu den Voronoi-Generatoren berechnen. So könnte man häufig frequentierte Straßensegmente erkennen, wenn viele der erhaltenen kürzesten Wege diese Straßensegmente enthalten.

Zusätzlich zur Berechnung des Einzuggebietes und seiner Größe ist es für viele Anwendungen wichtig, wie viele Menschen in diesem Gebiet wohnen. Wenn diese Daten in einem GIS gespeichert sind (z.B. die Anzahl der Menschen pro Gebäude oder pro Häuserblock), können diese Daten in Kombination mit dem Einzugsgebiet die Anzahl der Menschen pro Einzugsgebiet liefern. Das gleiche gilt für ein Gebäudenetzwerk: Die Kombination des Einzugbereiches für einen bestimmten Ausgang mit der Anzahl der Personen, die sich in den Zimmern entlang der zugeordneten Verbindungsgänge aufhalten, liefert die Anzahl der Menschen, die im Katastrophenfall diesen Ausgang benützen werden. Die Kombination mit Attributdaten wie Kapazität oder Breite, erlaubte oder mögliche Geschwindigkeit, oder Zeitraum und Standort von Baustellen führt zu einer Verbesserung der Berechnung der kürzesten Wegzeiten und der steuernden Maßnahmen.

Abschließend kann gesagt werden, dass die Berechnung von Voronoi-Diagrammen, wie sie in dieser Arbeit beschrieben wurde, eine Basis für viele weitere Berechnungen, Modellierungen und Analysen darstellt.

Literatur

[1] *Aurenhammer, F.:* Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys, 1991, 23(3): 345–405.

[2] *Aurenhammer, F.; Klein, R.:* Voronoi diagrams. In: Sack, J.-R.; Urrutia, J. (Hrsg.), Handbook of Computational Geometry, Elsevier Science Publishing, Amsterdam, 2000, Seiten 201–290.

[3] *de Berg, M.; van Kreveld, M.; Overmars, M.; Schwarzkopf, O.:* Computational Geometry. Springer, Berlin, 2000, 367 Seiten.

[4] *Okabe, A.; Boots, B.; Sugihara, K.; Chiu, S.N.:* Spatial Tesselations: Concepts and Applications of Voronoi Diagrams. John Wiley & Sons, Chichester, United Kingdom, 2000.

[5] *Erwig, M.:* The Graph Voronoi Diagram with Applications. Networks, 2000, 36(3): 156–163.

[6] *Okabe, A.; Okunuki, K.; Funamoto, S.; Ishitomi, T.:* A Toolbox for Spatial Analysis on a Network and its Software. In: Zavala, G. (Hrsg.), GIScience 2002. University of California Regents, Boulder, CO, 2002, Seiten 124–126.

[7] *Okabe, A.; Okunuki, K.; Funamoto, S.:* SANET: A Toolbox for Spatial Analysis on a Network, Center for Spatial Information Science, University of Tokyo, 2002, http://okabe.t.u-tokyo.ac.jp/okabelab/atsu/sanet/sanet-index.html.

[8] *Dijkstra, E.W.:* A note on two problems in connexion with graphs. Numerische Mathematik, 1, 1959: 269–271.

Anschrift der Autoren:

Margot Graf: Große Sperlgasse 41/28, 1020 Wien, email: margot.graf@aon.at

Stephan Winter, Institut für Geoinformation, Technische Universität Wien, Gusshausstr. 27–29, 1040 Wien, email: winter@geoinfo.tuwien.ac.at