Erzeugung von k-Ordnung Voronoi-Diagramm

Xinyu Tu,Xinchuan Wang

Rheinische Friedrich-Wilhelms-Universität Bonn

Zusammenfassung

1 Einleitung

Mit der Zunahme der Bevölkerung und der Zahl der Fahrzeuge, die die Menschen besitzen, steht der Verkehr in modernen Städten vor immer größeren Herausforderungen. Für die Stadtverwaltung wäre es sehr wichtig zu wissen, wo das größte Problem im derzeitigen Straßennetz liegt und ob der Bau oder die Verbesserung einer Straße das Problem lösen könnte. Da das Straßennetz in vielen modernen Städten einem rechteckigen Raster ähnelt, bei dem die Straßen entweder horizontal oder vertikal verlaufen,[1] können die Kapazität und der Zustand der Straßen als relative Geschwindigkeiten ausgedrückt werden, mit denen sich die Menschen auf ihnen bewegen können. Solche Gittersysteme können eine gute Annäherung für die Untersuchung des Verkehrsnetzes in Großstädten sein und ermöglichen es, die möglichen Auswirkungen neuer Straßen auf die Verkehrsbedingungen abzuschätzen.

Voronoi-Diagram ist eine Möglichkeit, ein Gebiet in kleine Regionen zu unterteilen, die sich auf mehrere bestimmte Orte auf dem Gebiet beziehen, so dass jede Region nur einen der Orte enthält [2]. In der Regel gewährleisten die unterteilten Regionen, dass für alle Punkte in einer Region der nächstgelegene Standort auf der Karte der einzige Standort innerhalb der Region ist. Das Voronoi-Diagramm findet breite Anwendung in der Geometrie, der Ökologie, dem Ingenieurwesen, der Informationstechnologie und so weiter. Die Verwendung von Voronoi-Diagrammen lässt sich bis ins Jahr 1644 zurückverfolgen. Nachdem es von Georgy F. Voronoy [3] definiert und untersucht wurde, ist es in vielen Bereichen intensiv erforscht und genutzt worden.

Das Voronoi-Diagramm erster Ordnung, das bei der Aufteilung von Regionen nur die nächstgelegene Seite berücksichtigt, ist gut untersucht und verstanden worden. Es hat eine strukturelle Komplexität von $o(n + c)$, wobei n die Anzahl der Scheitelpunkte und c die Anzahl der Standorte auf der gesamten Karte ist. Die Zeitkomplexität für die Konstruktion eines solchen Diagramms kann so klein wie $O((n + c)log(n + c))$ sein. Voronoi-Diagramme höherer Ordnung, bei denen mehr als ein nächstgelegener Standort bei der Konstruktion berücksichtigt werden muss, sind jedoch aufgrund ihrer Komplexität weniger untersucht worden. Für ein Voronoi-Diagramm k-ter Ordnung beträgt die strukturelle Komplexität $O(k(n-k))$. Die zeitliche Komplexität hängt jedoch von den verschiedenen Ansätzen zu seiner Erstellung ab. Bei einer einfachen iterativen Konstruktion beträgt die zeitliche Komplexität $O(k^2nlogn)$ [**?**]. Berücksichtigt man die

geometrische Dualität und die Anordnungen, so kann die Zeitkomplexität auf $O(n^2 + k(n-k)log^2 n)$ [**?**] reduziert werden.

In diesem Bericht wird Voronoi-Diagram der Ordnung k auf einem rechteckigen Gitter mit mehreren darauf verteilten Autobahnen erstellt. Die Implementierung ist in Java geschrieben, und das Diagramm wird auf der Grundlage des Dijkstra-Algorithmus durch iterative Berechnung des Abstands zwischen den Standorten und Scheitelunkten des Gitters erstellt. Da der Prozess der Diagrammerstellung sehr zeitaufwändig ist, wurden bei der Implementierung mehrere Threads verwendet, um den Berechnungsprozess zu beschleunigen. Die Zeitkomplexität dieser Implementierung wird ebenfalls erörtert, zusammen mit einigen Möglichkeiten, sie zu verringern.

2 Related Work

In diesem Abschnitt werden hauptsächlich die Schritte und Grundsätze der Lösung erläutert.
Wir beginnen mit dem Aufbau des grundlegenden Straßennetzes, indem wir die einzelnen Städte und Highway einrichten. Das Voronoi-Diagramm ist eine Möglichkeit, eine Region in kleinere Gebiete zu unterteilen. Als Nächstes müssen wir die Städte in Gebiete unterteilen, die entsprechend der Entfernung zwischen ihnen unterteilt werden können.

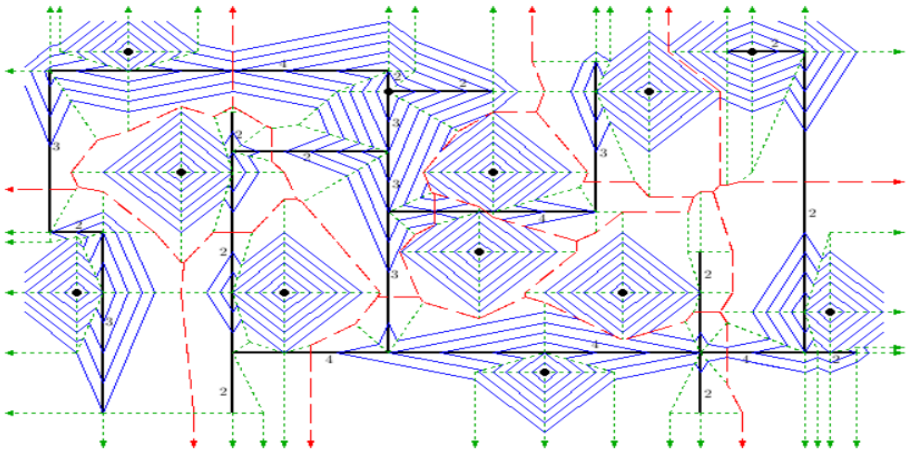

Abbildung 1. Wellen-Modell[6]

Wir wenden den Dijkstra-Algorithmus zur Berechnung des kürzesten Weges an, um die nächstgelegene Stadt zu finden, und verwenden ein Wellen-Modell

(Abb.1), um die nächstgelegene Stadt zur Abfahrtsstadt gleichzeitig visuell darzustellen. Wir wählen zunächst mehrere Städte auf dem Straßennetz aus, berechnen dann die Entfernungen von jeder der anderen Städte und sortieren sie in verschiedenen Farben, d. h. teilen sie in Regionen ein. Wenn wir zum Beispiel vier Städte a,b,c,d auswählen, wird die Stadt, die der Stadt a am nächsten liegt, in eine Farbe eingeteilt, die Stadt, die der Stadt b am nächsten liegt, in eine andere Farbe, ebenso wie c und d. So wird das gesamte Diagramm in vier verschiedenfarbige Regionen unterteilt. Dies ist nur der Fall, wenn k=1 ist, und wird auf der höheren Ordern näher erläutert. Dann kann die Komplexität des rechnerischen Urteils beginnen. Zum leichteren Verständnis werden einige wichtige Begriffe eingeführt.

2.1 Straßennetz

Wir nennen ein Voronoi-Diagramm eines gegebenen städtischen Straßennetzes C und einer gegebenen Menge S von Punkten in der Ebene ein städtisches Voronoi-Diagramm. Für jeden einzelnen Vertex gibt das Stadtvoronoi-Diagramm die Fläche aller Punkte an, die bei der Fahrt von diesem Vertex aus zuerst erreicht werden, d. h. $V_C(S)$.
Wir konstruieren äquidistante Gitter als Straßennetze C = (V, E) und können uns nur entlang von Straßen bewegen, d. h. auf horizontalen oder vertikalen Kanten. Dann setzen wir die Highway ein. Die Bewegungsgeschwindigkeit auf einer normalen Straße ist 1 und auf einer Highway ist $v(v > 1)$. Wir definieren $c := |V|$, und vom Vertex gibt es höchstens vier Punkte in der Nähe, also $|E| = \theta(c)$. Wir bezeichnen den Abstand zwischen zwei Punkten in der L_1-Metrik mit d_1 und den Abstand zwischen zwei Punkten in der Stadtmetrik mit d_c.

2.2 Dijkstra Algorithmus

Dijkstra-Algorithmus handelt sich um das Single-Source-Shortest-Path Problem. Wir verwenden das Wellen-Modell, um Karten des kürzesten Weges zu beschreiben, die für die Ableitung der Komplexität des k-Order City Voronoi Diagramms wichtig ist.

2.3 Wellen

Wellen hier sind die Wavefront Propagation[8].Die Wavefront Propagation ist ein perfektes Modell für die Definition von Voronoi-Diagrammen Wir können damit die Entstehung von $V_c(s)$ erklären und die Komplexität analysieren. Für einen Vertex $p \in S$ wenden wir das Konzept des Dijkstra-Algorithmus an, um die Nähepunkte $q \in R$ von p zu finden und sie zu verbinden, und finden dann die entsprechenden Nähepunkte $h \in R$ aus den Nähepunkten und verbinden sie wieder (wie in Abbildung 3). Wenn man auf eine Highway trifft, ändert sich die Geschwindigkeit und damit die Form (Abb. 3).

```
DIJKSTRA(G, w, s)
 1   for each (v ∈ V) { d(v) = ∞; π(v) = null; }
 2   d(s) = 0; S = ∅;
 3   while (S ≠ V) {
 4       Finde u ∈ V \ S, so dass d(u) minimal ist;
 5       S = S ∪ {u};
 6       for each ((u, v) ∈ E) {
 7           if (d(v) > d(u) + w(u, v)) {
 8               d(v) = d(u) + w(u, v);
 9               π(v) = u;
10           }
11       }
12   }
```

Abbildung 2. Dijkstra-Algorithmus[7]

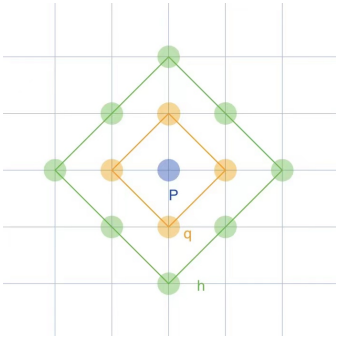

Abbildung 3.

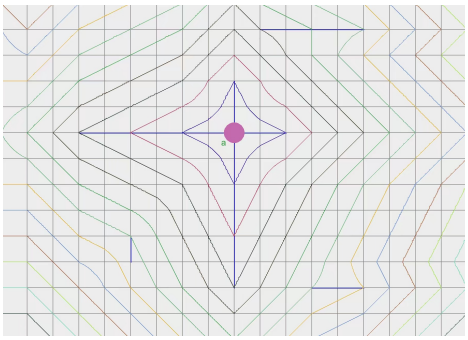

Abbildung 4.

2.4 Was ist K?

Wählen Sie eine beliebige Anzahl von Punkten in Gitter aus und berechnen Sie die Entfernung von den anderen Punkten zu diesen Punkten, indem Sie sie in der Reihenfolge der Entfernung vom nächsten zum weitesten Punkt sortieren. Wir haben 20 Punkte als Städte im Code ausgewählt, die als a,b,c,d,.... ,t zu bezeichnen. Unter den verbleibenden Punkten gibt es einen Punkt x, dessen Abstände zu diesen zwanzig Punkten wie folgt geordnet sind (c,d,g,j,s,a,e,....) . Der Rest der Stadt ist derselbe wie oben. Wir verwenden die sortierten Ergebnisse als Array, und wenn k=1 ist, wählen wir a[0],x die nächstgelegene Stadt zu c. An diesem Punkt wird die nächstgelegene Stadt zu c in der Entfernungssortierung aller Punkte in eine Region unterteilt. Die verschiedenen Regionen sind durch unterschiedliche Farben gekennzeichnet. Bei k=2 wird a[0],a[1] als nächstgelegene Stadt ausgewählt, und die ersten beiden Städte des Sortierergebnisses aller Punkte werden in eine Region eingeteilt. k=3 bedeutet, dass die ersten drei Städte entsprechend dem Sortierergebnis eingeteilt werden, und bei k=20 werden alle Städte in eine Region eingeteilt, und alle Punkte haben die gleiche Farbe.

3 Algorithmus

Der Prozess der Erstellung des Voronoi-Diagramms umfasst zwei Prozesse. Der erste Prozess berechnet die Entfernung zwischen jeder Stadt und den Eckpunkten auf der Karte, und der zweite Prozess gruppiert die Eckpunkte auf der Karte nach den k-ten nächstgelegenen Städten und malt die Karte entsprechend den Gruppen. Die beiden Prozesse werden in diesem Abschnitt im Detail besprochen. Darüber hinaus wird in diesem Abschnitt auch die Speicherung von Highways auf der Karte erörtert.

3.1 Lagerung von Highways

Highways sind spezielle Segmente zwischen den Eckpunkten auf der Karte. Im Gegensatz zu gewöhnlichen Straßen, deren Geschwindigkeit konstant bei 1,0 liegt, ermöglichen Highways den Menschen, sich schneller (mit einer Geschwindigkeit von $v > 1$) von einem Ort zum anderen zu bewegen. Das Vorhandensein von Highways erhöht die Komplexität unseres simulierten Gittersystems und macht es dem Verkehrsnetz in realen Städten sehr viel ähnlicher. Allerdings sind Highways keine unabhängigen Einheiten wie Städte. Stattdessen sind sie eng an die Karte gebunden und beschreiben die relative Entfernung zwischen den Eckpunkten. In der Implementierung existiert daher eine gerade Highway auf der Karte nicht als einzelnes Objekt. Sie werden als die Entfernungen jedes Segments auf der Karte gespeichert, als ein HashMap-Objekt, das Scheitelpunktpaare auf Entfernungswerte abbildet. Wenn zum Beispiel eine Highway vom Scheitelpunkt (1,2) zum Scheitelpunkt (1,5) führt, wird im Programm die Entfernung zwischen den Scheitelpunktpaaren ((1,2),(1,3)), ((1,3),(1,4)) und ((1,3),(1,5)) als

1/v angegeben. Der Einfachheit halber wird nur der Abstand für Segmente auf Highways explizit gespeichert. Für andere Segmente wird ein Standardwert von 1,0 zurückgegeben, da das Scheitelpunktpaar nicht in der HashMap gefunden wird.

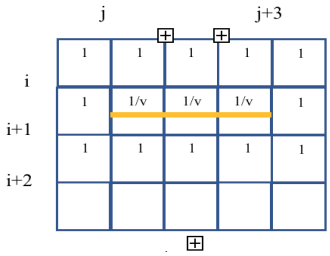

Abbildung 5.

3.2 Berechnung des kürzesten Abstands

Die Stufe der Entfernungsberechnung basiert auf dem grundlegenden Dijkstra-Algorithmus (Dijkstra, 1959). Es handelt sich um einen Algorithmus zur Ermittlung des kürzesten Weges zwischen zwei Punkten auf der Grundlage von Iterationen. Das Flussdiagramm für den in dieser Studie implementierten Algorithmus ist in Abbildung dargestellt. Die Berechnung der Entfernung der Eckpunkte auf der gesamten Karte für eine Stadt muss von der Stadt selbst ausgehen und schrittweise erweitert werden, um die gesamte Karte abzudecken. Zu beachten ist, dass in diesem Schritt auch der Unterschied zwischen normalen Straßen und Highways berücksichtigt wird. Da das Programm die Entfernung eines Eckpunktes (i,j) aus der Entfernung seines Nachbarn, z.B. (i+1,j), bestimmt, wird die Gesamtentfernung von (i,j) als Gesamtentfernung von (i+1,j) plus der Entfernung von (i,j) zu (i+1,j) berechnet. Die Entfernung zwischen den beiden Scheitelpunkten wird dann berechnet, indem berücksichtigt wird, ob das Segment zwischen ihnen auf einer Highway liegt:

Außerdem ist die Berechnung der Entfernung zu verschiedenen Städten nicht relevant, so dass sie parallel durchgeführt werden kann, um die Berechnungsgeschwindigkeit zu erhöhen. Bei der Implementierung wird der Dijkstra-Algorithmus für alle Städte gemeinsam in separaten Threads initialisiert und unabhängig voneinander gleichzeitig ausgeführt.

3.3 Gruppierung der Eckpunkte

Nachdem die Entfernung zwischen einer beliebigen Kombination aus Stadt und Knotenpunkt berechnet wurde, können die Knotenpunkte in verschiedene Grup-

pen unterteilt werden, um das Voronoi-Diagramm zu erstellen. Bei diesem Prozess wird jeder Punkt auf der Karte betrachtet, indem alle Entfernungen zwischen ihm und allen Städten ermittelt werden, und die k Städte mit den kürzesten Entfernungen werden ermittelt. Danach wird der Knoten in die entsprechende Gruppe eingeordnet, indem die k nächstgelegenen Städte berücksichtigt werden. Da alle Eckpunkte in verschiedene Gruppen eingeteilt werden, werden sie entsprechend der Gruppen eingefärbt. Die Farben für jede Gruppe werden aus einer vordefinierten Farbliste (Tatarize, 2012) ausgewählt, so dass die Farben zwischen verschiedenen Gruppen voneinander unterscheidbar sind. Auf diese Weise wird ein Voronoi-Diagramm k-ter Ordnung erstellt und visualisiert.

4 Complexität Analysis

5 Fazit

Literatur

1. Gemsa, A., Lee, D., Liu, C.-H., Wagner, D. (2012): *Higher order city Voronoi diagrams* SWAT'12: Proceedings of the 13th Scandinavian conference on Algorithm Theory, 59-70.
2. Cooperative authors. (2022, 3 11): Voronoi diagram. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Voronoi$_d$*iagram*
3. Voronoi, G. (1908) : Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. Journal für die Reine und Angewandte Mathematik, 97-178.
4. Lee, D. T. (1982). On k-nearest neighbor Voronoi diagrams in the plane, IEEE Trans. Comput., 478-487.
5. Chazelle, B., Edelsbrunner, H. (1987). An improved algorithm for constructing kth-order Voronoi diagrams, IEEE Transactions on Computers, 1349-1454.
6. https://i11www.iti.kit.edu/en/projects/geonet/cvd
7. Heiko Röglin,Tomas Kesselheim (2020). Algorithmen und Berechnungskomplexität I
8. Aichholzer, O., Aurenhammer, F. Palop, B. Quickest Paths, Straight Skeletons, and the City Voronoi Diagram. Discrete Comput Geom 31, 17–35 (2004). https://doi.org/10.1007/s00454-003-2947-0