

Skalierbare Analyse von räumlichen Daten in großangelegten wissenschaftlichen Simulationen¹

Farhan Tauheed²

Abstract: Die Wissenschaft befindet sich heutzutage in einem radikalen Umbruch. Wissenschaftler müssen sich heute nicht mehr nur auf traditionelle wissenschaftliche Methoden des Experimentierens, dem entwickeln von Theorien und dem Testen von Hypothesen verlassen. Zusätzlich können sie heute auch massive Mengen von Daten analysieren um neue wissenschaftliche Erkenntnisse zu erlangen. Immer schnellere Hardware sowie immer genauere Sensoren ermöglichen ihnen das Sammeln von Daten, das Bauen von Modellen und das Simulieren von Naturphänomenen im großen Maßstab. Wissenschaftliche Entdeckungen in diesem Zusammenhang bedingen allerdings Algorithmen für die effiziente Analyse von massiven Datenmengen. Die heute enormen Datenmengen machen die Ausführung der notwendigen Analysen zu einer beispiellosen Herausforderung. Die Effizienz heutiger Algorithmen reicht nicht aus um die Datenmengen schnell genug zu analysieren und das Problem wird immer schlimmer da die Datenmengen rasant wachsen.

Diese Doktorarbeit konzentriert sich auf die Trends des Datenwachstums in den Wissenschaften und wie diese Trends die Leistung von Analysealgorithmen beeinflussen. Eine interessante Entwicklung bezüglich wissenschaftlicher Daten ist, dass wenn Wissenschaftler die Datenmenge erhöhen, die Daten gleichzeitig auch komplexer werden, so dass bekannte Algorithmen ineffizient werden. Die "Komplexität" der Daten wird durch die Veränderung der Datencharakteristiken, wie beispielsweise Verteilung, Dichte und Auflösung verursacht. Die Zunahme der Komplexität verschlechtert die Effizienz bestehender Algorithmen wesentlich und, noch wichtiger, hemmt auch die Skalierbarkeit der Algorithmen. Diese Arbeit schlägt eine Methodik zur Entwicklung neuer Analysealgorithmen vor welche effizient sind und besser skalieren als existierende Methoden. Mit wissenschaftlichen Daten demonstrieren wir, dass Algorithmen die mit unserer Methodik entwickelt wurden nicht nur schneller als heutige Algorithmen sind, sondern auch, dass sie wesentlich besser mit Komplexität und Größe zukünftiger Datensätze skalieren.

1 EINLEITUNG

Meteorologische Vorhersagen, die Erprobung neuer Medikamente durch Computersimulationen und die Entdeckung neuer Phänomene haben Simulationen allgegenwärtig in den Naturwissenschaften gemacht und sind heute ein wichtiges Instrument im Arsenal der Wissenschaftler. Obwohl in Simulationen in verschiedenen wissenschaftlichen Disziplinen unterschiedliche numerische Methoden und Lösungstechniken verwenden, haben sie auch einige wichtige Gemeinsamkeiten. So erfordert die Simulation eines Naturphänomens grundsätzlich den Bau eines Modelles der Struktur des Phänomens. Dieses Modell wird während der Simulation ständig verändert, um das Verhalten des realen Phänomens zu imitieren. Die strukturelle dreidimensionale räumliche Darstellung des Phänomens liegt damit im Kern der Simulationsworkflows und Modellierung, Simulation, Analyse und Visualisierung hängen stark von der effizienten Analyse von räumlichen Daten ab.

Die Erfassung von riesigen Mengen von räumlichen Daten hat den Aufwand für deren Speicherung und Verarbeitung zu einem beispiellosen Unterfangen gemacht. Hinzu kommt,

¹ Scalable Exploration of Spatial Data in Large-Scale Scientific Simulations

² École Polytechnique Fédérale de Lausanne, Data-Intensive Applications and Systems Lab, Switzerland, farhan.tauheed1@gmail.com

dass die Erhöhung der Datensatzgröße in wissenschaftlichen Anwendungen als Nebeneffekt auch die Datenkomplexität erhöht. Die Datenkomplexität lässt bekannte Datenanalysealgorithmen ineffizient werden. Das bedeutet, dass auch wenn ein Algorithmus effizient ist um einen bestimmten Datensatz zu analysieren, der Algorithmus nicht zwingend mit Größe und Komplexität der Datensätze skalieren wird und dass er daher zu einem Leistungsengpass für größere und komplexere Datensätze werden kann. Das Beispiel der Neurowissenschaftler welche im Human Brain Project ¹ das Hirn simulieren zeigt dies deutlich: um biologisch realistische Hirnmodelle zu simulieren, begannen die Neurowissenschaftler mit dem Bau eines Modells einer kleinen Hirnregion und vergrößern dieses nun sukzessive bis das Modell das gesamte Hirn umfassen. Dies erhöht die Datensatzgröße für die Speicherung der Modelle, es erhöht aber auch die räumliche Dichte (die Anzahl der räumlichen Objekten die zur Darstellung von Nervenzellen pro Volumeneinheit verwendet werden) der Datensätze. Die Erhöhung der Dichte ist ein exzellentes Beispiel der Steigerung der Datenkomplexität die sich direkt auf die Leistung und Skalierbarkeit der bekannten Analysealgorithmen auswirkt. Verteilung, räumliche Auflösung und Unregelmäßigkeiten in geometrischen Formen sind andere Beispiele von sich erhöhender Datenkomplexität die wir in Daten aus wissenschaftlichen Simulationen beobachtet haben. Für die effiziente Analyse von wissenschaftlichen Daten benötigen wir daher Algorithmen, welche nicht nur mit Größe der Datensätze skalieren sondern auch mit der Erhöhung der Datenkomplexität. Wir formulieren das Problem wie folgt:

***Problemstellung:** Das Ermöglichen einer effizienten Analyse von räumlichen Daten in großangelegten wissenschaftlichen Simulationen durch die Entwicklung neuartiger Algorithmen, die nicht nur heute effizient sind, sondern die auch mit der zukünftigen Zunahme der Komplexität der räumlichen Daten skalieren.*

2 SCHLUESSELBEITRAEGE

Unsere Zusammenarbeit mit dem EU Flagship Project (Human Brain Project) erlaubte es uns die Kernprobleme in Bezug auf Skalierbarkeit in modernen wissenschaftlichen Simulationen zu analysieren. Die wichtigste Erkenntnis unserer Untersuchungen war das Problem der Datenkomplexität zu identifizieren und zu zeigen, wie stark sie die Skalierbarkeit wissenschaftlicher Simulationen limitiert. Diese Schlüsselerkenntnis hat uns inspiriert eine Methodik für die Entwicklung von Analysealgorithmen für räumlichen Daten zu definieren die nicht nur effizient sind um neurowissenschaftliche Daten zu analysieren aber, wie wir in dieser Dissertation mit Experimenten zeigen, auch für räumliche Daten anderer Anwendungen.

Algorithmen für dreidimensionale räumliche Daten bilden den Kern vieler Datenanalysen im Kontext von Simulationen. Existierende räumliche Analysemethoden verwenden auf konzeptioneller Ebene die Idee der räumlichen Nähe. Das heißt, dass existierende Methoden Objekte die im euklidischen Raum nahe zusammen liegen auch auf dem Speichermedium zusammen speichern um den Zugriff darauf zu beschleunigen. Dies ermöglicht es Algorithmen den Suchraum effizient zu verkleinern und nur Objekte in der Nähe der Abfrageregion vom Speichermedium zu lesen und damit letztendlich die Leistung massiv zu verbessern. Allerdings skalieren solche Techniken nicht wenn die Komplexität der Daten

¹ EU FET Flagship Project - The Human Brain Project, <https://www.humanbrainproject.eu>

auch erhöht wird wie wir zeigen werden. In dieser Arbeit beschreiben wir, wie wir das Konzept der räumlichen Konnektivität verwenden, um Analysealgorithmen für räumliche Daten zu entwickeln, die nicht nur effizient sind sondern die auch zu höherer Datenkomplexität skalieren. Räumliche Konnektivität stellt die Informationen dar wie Objekte miteinander verbunden sind (physikalisch oder anderweitig). So zum Beispiel verbinden Synapsen Neuronen zu einem neuronalen Netzwerk. Durch die Verwendung der Konnektivität erhalten wir eine kostengünstige Möglichkeit für den Zugriff auf Objekte welche räumlich nahe beieinander liegen und können so den unnötigen Aufwand traditioneller räumlicher Analysealgorithmen wesentlich reduzieren.

Analysetechniken die auf Konnektivität beruhen helfen nicht nur in der Analyse von wissenschaftlichen Daten, wie zum Beispiel in neurowissenschaftlichen Daten, sondern können auch in der Analyse anderer räumlicher Datensätze hilfreich sein in denen Konnektivitätsinformation vorhanden ist (oder durch einen Bearbeitungsschritt a priori hinzugefügt wird [Ta12a, Ta12b]). Auch einschließlich des potentiell benötigten Bearbeitungsschrittes sind die vorgeschlagenen Techniken deutlich schneller als der Stand der Technik und skalieren vor allem auch besser. Die vorgeschlagenen räumlichen Analysealgorithmen sind daher generisch und auch für nicht-wissenschaftliche Datensätze anwendbar.

Im Folgenden diskutieren wir basierend unserer Methodik drei Analysealgorithmen entwickelt haben um die drei häufigsten Analysen von räumlichen Daten in wissenschaftlichen Simulationen effizienter auszuführen. In dieser Übersicht diskutieren wir wie sie Konnektivität verwenden um Analysen effizient auszuführen und um zu skalieren.

3 RÄUMLICHE AD-HOC ANALYSE

Das Abfragen einer Teilmenge des Datensatzes um eine Hypothese zu testen, um die Modelle statistisch zu validieren oder um bestimmte Teile des Datensatzes zu visualisieren ist eine häufige Aufgabe, die Wissenschaftler in ihrem normalen Tagesablauf ausführen müssen. Ad-hoc-Untersuchungen ermöglichen den Wissenschaftlern die Eigenschaften der Daten durch Beobachtung aus unterschiedlichen Perspektiven zu untersuchen und zu verstehen. Aus Sicht des Datenmanagements ist die häufigste Art der Durchführung dieser Analyse eine **räumliche Bereichsanfrage** auf dreidimensionalen räumlichen Datensätzen um die Daten in der Region von Interesse abzurufen. Die Daten können Sensorrohdaten, Modelldaten, Simulationsresultate oder auch andere Daten sein. Dreidimensionale Bereichsanfragen zum Beispiel werden verwendet um Teile des Hirnmodells abzurufen so dass Neurowissenschaftler die statistischen Eigenschaften des Modells mit realem Nervengewebe vergleichen können wie es in Abbildung 1 dargestellt wird.

Räumliche Indizes sind Datenstrukturen mit denen effizient räumliche Bereichsanfragen ausgeführt werden können. Der heutige Stand der Technik ist die Familie der R-Tree Algorithmen [Gu84, LLE97, Ar04, KF84] welche in der Mehrheit der kommerziellen und quelloffenen Datenbanklösungen implementiert sind. Die verschiedenen Varianten des Al-



Abb. 1: Dreidimensionale räumliche Bereichsanfragen für die Ad-hoc Erkundung von Nervengewebe.

gorithmus bauen auf derselben konzeptionellen Idee auf, das heißt, auf einer Hierarchie von dreidimensionalen minimalen umgebenden Rechtecken (die durch Gruppieren Objekte die nahe beieinander liegen gebildet werden). Die Hierarchie oder Baumstruktur ermöglicht eine effiziente Durchführung von dreidimensionalen Bereichsabfragen durch das Durchlaufen des Baumes von oben nach unten und der damit einhergehenden Verkleinerung des Suchraumes. Das Problem der Verwendung einer hierarchischen Datenstruktur ist, dass jedes minimale umgebende Rechteck (Knoten in der Baumstruktur) mit anderen Knoten auf gleicher Ebene im Baum überlappen kann. Dadurch entsteht eine Mehrdeutigkeit in der Baumstruktur welche zu vermehrtem (und unnötigem) Lesen und Analysieren von Baumknoten führt. Jedes Lesen eines Knotens bedeutet dabei einen kostspieligen Zugriff auf die Festplatte. Das Überlappungsproblem ist von grundlegender Bedeutung für die Familie der R-Tree Algorithmen und wird wesentlich durch steigende (räumliche) Datenkomplexität verschärft, wie es zum Beispiel bei Neurowissenschaftlern der Fall ist die immer detailliertere (und damit komplexere) Hirnmodelle verwenden. Der Aufwand von Bereichsabfragen steigt deshalb überproportional an wenn die Komplexität steigt.

Die Datenkomplexität in diesem speziellen Anwendungsfall ergibt sich aus der räumlichen Dichte des verwendeten Datensatz. Um eine Lösung zu entwickeln die mit der Komplexität skaliert, entwerfen wir den FLAT-Algorithmus [Ta12a, St13b, St13a, Ta13] der eine grundsätzlich andere Strategie verfolgt um räumliche Bereichsabfragen auszuführen. Der Algorithmus fügt dem Datensatz während der Indexierung Konnektivitätsinformationen hinzu. Mit Hilfe dieser Information werden räumlicher Bereichsabfragen in zwei Phasen durchgeführt. Zunächst findet die Seeding-Phase ein beliebiges Objekt im Abfragebereich mit einem herkömmlichen räumlichen Index. In der zweiten Phase, der Crawling-Phase, wird mit Hilfe des initialen Objektes (welches in der Seeding-Phase gefunden wurde) und der Konnektivitätsinformation die räumliche Nachbarschaft traversiert um die restlichen Objekte in dem Abfragebereich zu finden. Beide Phasen sind so ausgelegt, dass sie unabhängig von der räumlichen Dichte effizient ausgeführt werden können und daher auch mit der Komplexität (**Raumdichte**) der Daten skalieren.

In dieser Übersicht über die Dissertation präsentieren wir die zentralen Resultate die wir mit FLAT und Daten aus dem Human Brain Project erzielt haben. Wir vergleichen FLAT mit drei verschiedenen Varianten von in Datenbanklösungen verwendeten R-Tree Algorithmen. Im ersten Experiment messen wir die Zeit welche benötigt wird um räumliche Bereichsabfragen auszuführen damit verschiedene Teile des räumlichen neuronalen Datensatzes analysiert werden können. Auf der x-Achse von Abbildung 2 erhöhen wir die Datensatzgröße genau so wie Neurowissenschaftler auch immer größere und umfassendere Modelle bauen um das Hirn realistischer nachzubilden. Indem Neurowissenschaftler immer detailliertere Modelle bauen, erhöhen sie auch die Anzahl neuronaler Strukturen im gleichen Volumen und erhöhen daher auch die Dichte des Datensatzes. Mit FLAT können wir eine Geschwindigkeitssteigerung von einem Faktor von drei bis acht im Vergleich zur

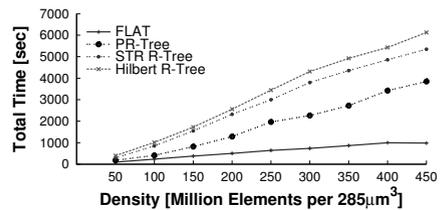


Abb. 2: Ausführungszeit für die Ausführung räumlicher Bereichsabfragen.

schnellsten R-Tree Variante erzielen. Der größte aktuell verwendete Datensatz umfasst rund 100000 Neuronen (was 450 Millionen Raumelementen/Zylindern und damit 27 GB an Daten entspricht). Das Endziel des Human Brain Projektes ist es Modelle des gesamten menschlichen Hirns zu bauen was ca. 86 Milliarden Neuronen entspricht. Modelle in dieser Größenordnung zu bauen bedeutet auch die Dichte (Anzahl Raumelemente pro Raumeinheit) und daher die Komplexität wesentlich zu erhöhen.

4 INTERAKTIVE RÄUMLICHE ANALYSE

In vielen Anwendungsszenarien bestehen Analysen nicht nur aus einzelnen isolierten Abfragen. Vielmehr führen Wissenschaftler eine Folge von Abfragen interaktiv aus, wobei jede Abfrage in einer Sequenz von den vorherigen Abfragen abhängt. Navigationsorientierter Zugriff auf die dreidimensionalen Daten und Walkthrough Visualisierung sind Beispiele für interaktive Analysen. Obwohl sich Indizes hervorragend eignen um die Leistung einzelner Abfragen in der Sequenz zu erhöhen, kann die Ausführung einer gesamten Sequenz wesentlich beschleunigt werden. Durch Prefetching können zum Beispiel Teile des nächsten Resultats bereits in den Speicher geladen während das Resultat der vorherigen Abfrage immer noch analysiert wird wie das in Abbildung 3 illustriert ist. Zentral um Prefetching effizient zu machen ist genau vorhersagen zu können wo die nächste Abfrage ausgeführt werden wird. Je genauer die Voraussage, desto mehr Daten können bereits vorgängig geladen werden und umso mehr kann die interaktive Analyse beschleunigt werden.

Bekannte Prefetchingtechniken verwenden vorherrschend Kurvenextrapolation [Ch98, SOR09, PK01] die auf den Positionen vorheriger Abfragen in der Sequenz basiert um die Position der nächsten Abfrage zu bestimmen. Dieser Ansatz ist effizient, wenn der Pfad der Abfragesequenz gut durch Kurvenextrapolationstechniken angenähert werden kann; ist der Pfad jedoch geometrisch unregelmäßig sinkt die Vorhersagegenauigkeit solcher Techniken erheblich. Dieses Problem entsteht besonders dann wenn Wissenschaftler ihre Modelle und damit die Datensätze vergrößern. Im Fall von neuronalen Modellen beispielsweise kann die Abfragesequenz beliebigen Pfaden im neuronalen Netzwerk folgen. Wenn Neurowissenschaftler die Anzahl der Neuronen erhöhen um größere Bereiche des Hirns zu simulieren wird die Anzahl der Nervenbahnen erhöht (da die Anzahl der neuronalen Verbindungen, d.h., SSynapsen“, zunimmt). Dies erhöht die Wahrscheinlichkeit, dass der Benutzer eine Abfragefolge mit unregelmäßiger Geometrie ausführt. In diesem Fall ist die Unregelmäßigkeit der Geometrie der Pfade im Datensatz die Komplexität der Daten, welche die Skalierbarkeit der Analysemethoden limitiert.

Zur Lösung dieses Problems entwickeln wir SCOUT [Ta12b], eine Prefetchingtechnik die nicht nur die Positionen vorheriger Abfragen berücksichtigt, sondern auch deren In-

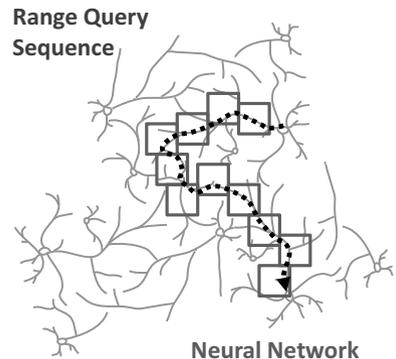


Abb. 3: Dreidimensionalen Raumbereich Abfragesequenz folgenden Nervenbahnen für interaktive Exploration.

halt um die Position der nächsten Abfrage vorherzusagen. SCOUT analysiert dazu wie Objekte innerhalb der vorangegangenen Abfrageregionen verteilt sind um damit wichtige Informationen zu erlangen wie (a) räumliche Objekte in der Abfrage zusammenhängen und Pfade bilden und (b) welchem Weg der Benutzer folgt.

Um Konnektivätsinformation zu den Objekten innerhalb einer Abfrage zu erhalten, berechnen wir eine Graphstruktur welche Objekte die sich nahe beieinander befinden miteinander verbindet. Diese Graphstruktur gibt uns eine gute Vorstellung davon wie viele geometrische Pfade sich im Abfragebereich befinden. Sobald wir alle Pfade innerhalb des vorherigen Abfragebereichs nachgebildet haben entfernen wir aus der Menge der potentiellen Pfade diejenigen welche in keiner der bisherigen Abfragen in derselben Sequenz vorkamen. Mit allen folgenden Abfragen wird diese Menge der potentiellen Kandidaten immer kleiner bis nach wenigen Abfragen nur noch eine Handvoll Pfade übrigbleibt. Je weniger Pfade übrigbleiben, desto besser kann vorausgesagt werden wo der Benutzer die nächste Abfrage ausführen wird und daher welche Daten bereits geladen werden sollen. Eine Prefetchingtechnik welche den Inhalt der vorangegangenen Abfragen in Betracht zieht liefert daher nicht nur bessere Vorhersagegenauigkeit, sondern skaliert auch besser weil sie unabhängig von der Geometrie der im Datensatz vorhandenen Wege ist.

In Abbildung 4 zeigen wir die wichtigsten Ergebnisse die wir mit neuronalen Simulationsdaten erzielt haben. Wir vergleichen drei bekannte Prefetchingtechniken welche die Positionen vergangener Abfragen mit einer dreidimensionalen Kurve interpolieren um dann die Position der nächsten Abfrage zu extrapolieren. Exponential Weighted Moving Average (EWMA) [Ch98] ist eine Extrapolationstechnik die Polynome (die zur Interpolation verwendet werden) verwendet und die mehr Gewicht auf neuere Abfragen in der Sequenz legt als auf frühere. Straight Line ist eine einfache lineare Extrapolationstechnik, während Hilbert Prefetching [PK01] eine raumfüllende Hilbertkurve verwendet, um Vorhersagen zu machen. Für dieses Experiment verwenden wir Daten der Neurowissenschaftler welche ein Hirnmodell von 100000 Neuronen umfassen. Wir testen die Prefetchingtechniken für fünf

Anwendungsfälle und messen die erreichte Beschleunigung. Die Anwendungsszenarien sind: (a) zwei Visualisierungsanwendungen in denen räumliche Bereichsabfragen ausgeführt werden um Daten abzufragen und auf dem Bildschirm zu visualisieren, (b) zwei Ad-hoc-Analyseanwendungen die kurze Abfragesequenzen ausführen um räumliche Daten entlang eines Pfades zu extrahieren, so dass Neurowissenschaftler Statistiken berechnen und mit dem eigentlichen Hirngewebe vergleichen können und (c) einen Modellbauanwendungsfall in dem die Abfragesequenzen verwendet wird um die Synapsen zwischen den Neuronen zu platzieren (innerhalb der Resultate der Abfragesequenz).

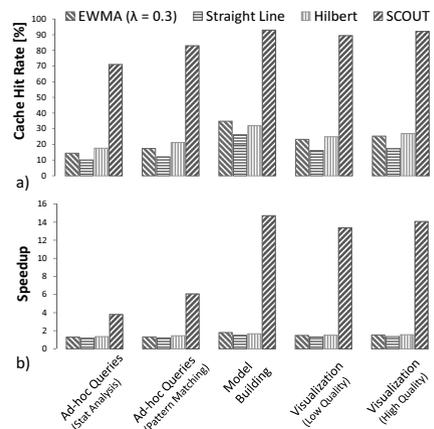


Abb. 4: (Oben) Vorhersagegenauigkeitsvergleich: der Anteil vorgängig korrekt geladenen Daten (Cachetrefferrate). (Unten) Die Beschleunigung der Ausführung von Bereichsabfragesequenzen im Vergleich zu keinem Prefetching.

SCOUT erreicht eine Leistungssteigerung von 4x-15x, die vor allem variiert weil verschiedene Anwendungsfälle unterschiedlich viele Anfragen einer Sequenz haben. Mehr Abfragen in einer Sequenz ermöglichen es die Anzahl der potentiellen Pfade rasch zu reduzieren und damit eine bessere Vorhersagegenauigkeit zu erreichen.

Extrapolationsverfahren die auf Kurven beruhen beschleunigen die Ausführung der Sequenz ebenfalls (verglichen mit dem Fall ohne Prefetching), allerdings sinkt die Beschleunigung falls unregelmäßige geometrische Wege angetroffen werden (die schlecht interpoliert werden können). In der Dissertation präsentieren wir auch Ergebnisse mit nicht-neuronalen Datensätzen wie Lungenmodellen, Straßennetzen und Modelle des Herzens, wo SCOUT auch eine erhebliche höhere Vorhersagegenauigkeit erreicht als bisherige Methoden.

5 DYNAMISCHE RÄUMLICHE ANALYSE

Wissenschaftliche Simulationen werden durch die Analyse der Unterschiede im Verhalten zwischen dem simulierten und dem tatsächlichen Phänomen iterativ verfeinert. Frühzeitiges Erkennen und Beseitigung von Fehlern in der Simulationskonfiguration bedeutet, dass die Wissenschaftler die Zeit für iterative Verfeinerung drastisch reduzieren können. Die Überwachung der Simulationen zur Laufzeit ermöglicht den Wissenschaftlern rechtzeitig reagieren zu können und die Ausführung der Simulation zu beeinflussen (indem Parameter anders gesetzt werden). Typischerweise wird eine dreidimensionale Gitterstruktur (Meshstruktur) verwendet, um die Phänomene zu modellieren. Abbildung 6 beispielsweise zeigt ein neuronales Gittermodell. Simulationen verändern und deformieren die Gittermodelle in diskreten Zeitintervallen um das Verhalten der Phänomene zu simulieren. Um Gittersimulationen zu überwachen müssen die Wissenschaftler den räumlichen Bereich von Interesse iterativ bei jedem Zeitschritt abfragen. Dieser Vorgang wird aus zwei Gründen zur Herausforderung. Erstens sind die Simulationen der Gittermodelle hochdynamisch, das heißt während jedes Simulationsschrittes wird die Position aller Objekte verändert. Zweitens, erhöhen Wissenschaftler ständig die Genauigkeit der Simulation und erhöhen dazu die Auflösung des Gittermodelles. Eine höhere Auflösung/Dichte bedeutet auch, dass die Effizienz bekannter Ansätze für die Ausführung dreidimensionaler Bereichsanfragen wesentlich degradiert wird.

Der einfachste Weg den sich verändernden Datensatz abzufragen, ist ein einfaches lineares Lesen des Datensatzes bei jedem Simulationsschritt. Lineares Lesen benötigt keine zusätzlichen Datenstrukturen die gewartet werden müssen und der Wartungsaufwand ist daher minimal. Um die Objekte (die Tetraeder des Gittermodells) in der Abfrageregion zu eruieren muss allerdings der gesamte Datensatz gelesen werden was die Skalierbarkeit beeinträchtigt (die Abfragekosten stehen in linearer Abhängigkeit zur Datensatzgröße). Das wird vor allem dann zum Problem, wenn die Wissenschaftler die Modellgenauigkeit erhöhen und die Anzahl der Tetraeder erhöhen. Mit Indexierungsansätzen andererseits kann die Zeit, um Objekte abzufragen erheblich reduziert werden. Gleichzeitig sind die Wartungskosten der Indizes allerdings beträchtlich weil alle Objekte in jedem Simulationsschritt Position wechseln. Auf Aktualisierungen spezialisierte Indexierungstechniken wie der LUR-Tree [KLL02] oder der QU-Trade können verwendet werden. Allerdings stellt auch in diesem Fall die Wartung ein ernstes Problem dar, weil diese Ansätze nicht

für den Fall ausgelegt sind in dem der gesamte Datensatz in jedem Simulationsschritt verändert wird.

Um die dynamische Analyse von Gitterdatensätzen zu ermöglichen entwickeln wir OCTOPUS [Ta14]. OCTOPUS nutzt die inhärenten Konnektivität der Gittermodelle um Abfrageergebnisse ohne Einbeziehung von wartungsintensiven Datenstrukturen zu bewerkstelligen. Der Algorithmus nutzt die Konnektivität inhärent in Gitterdatensätzen aus welche konstant bleibt obwohl sich die Position der Objekte ändert. Das bedeutet, dass OCTOPUS den Graph der durch die Gitterkonnektivität induziert wird traversiert um Objekte in räumlicher Nähe von einander zu bestimmen, ohne berücksichtigen zu müssen wie sich die

Objekte im letzten Zeitschritt bewegt haben. Um mit OCTOPUS das gesamte Resultat der Abfrage zu eruieren müssen wir mindestens ein Gitterobjekt (Tetraeder) in der Abfrageregion finden, um die Traversierung des Graphen zu initiieren welche das gesamte Ergebnis findet. Die Suche nach einem einzelnen Objekt in der Abfrageregion ist jedoch eine schwierige Aufgabe da wir nicht auf Datenstrukturen wie Indizes (welche hohe Wartungskosten haben) zurückgreifen wollen. OCTOPUS verwendet daher die Oberfläche des Gitters selbst um die Abfrage durchzuführen. Solange wir wissen, welche Objekte auf der Oberfläche des Gitters sind können wir diese linear durchlaufen und diejenigen Oberflächenobjekte (nur eine kleine Teilmenge des gesamten Datensatzes) zu identifizieren die sich im Abfragebereich befinden. Die Verwendung der Gitteroberfläche stellt zum einen die Korrektheit des Abfrageergebnisses sicher, zum andern verbessert sich dadurch auch die Leistung und Skalierbarkeit von OCTOPUS wesentlich. Erstens muss die Liste der Oberflächenobjekte nicht aufwändig gewartet werden da zwischen zwei Zeitschritten die Objekte auf der Oberfläche meist dieselben sind (obwohl sich die Lage & Position verändert). Zweitens wächst der Anteil der Objekte auf der Oberfläche quadratisch, während die Gesamtzahl der Objekte kubisch wächst wenn die Wissenschaftler die Gitterauflösung verbessern (das heißt die Anzahl Tetraeder erhöhen) um die Simulationsgenauigkeit zu erhöhen.

In dieser Zusammenfassung präsentieren wir die zentralen Resultate von Experimenten mit neuronalen Simulationsdaten vom Human Brain Projekt. In Abbildung 5 vergleichen wir OCTOPUS mit aktuellen Indexen sowie einer einfachen Lösung die linear alle Objekte liest. Wir verwenden zwei verschiedene Anwendungsfälle welche denselben neuronalen Gitterdatensatz der 100000 Neuronen mit 1,3 Milliarden Netzobjekten (Tetraeder) repräsentiert verwenden. Der erste Anwendungsfall führt 15 räumliche Bereichsabfragen pro Simulationszeitschritt aus um die Simulation visuell zu überwachen. Ein weiterer Anwendungsfall führt Anfragen aus um die Neurowissenschaftler zu benachrichtigen, so dass sie Probleme frühzeitig untersuchen und Simulationsparameter anpassen können sollten a priori definierte Statistiken nicht innerhalb definierter Grenzen bleiben. Dieser Anwendungsfall führt 9 räumliche Bereichsanfragen pro Zeitschritt aus.

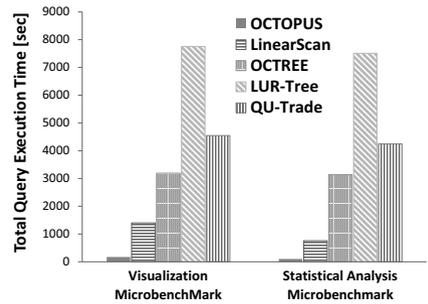


Abb. 5: Vergleich von Abfrageausführungszeit für zwei Simulationsanwendungsfälle.

6 BEDEUTUNG UND SCHLUSSFOLGERUNGEN

In dieser Arbeit machen wir die wichtige Feststellung, dass die Datenanalyse nicht nur wegen größerer Datensätze herausfordernder wird, sondern orthogonal dazu auch wegen der sich erhöhenden Datenkomplexität. Um eine skalierbare Lösung zu erreichen schlagen wir ein grundlegend neues Paradigma vor das auf der Konnektivität räumlicher Daten beruht um Analysealgorithmen für räumliche Daten zu entwickeln. Wir haben gezeigt wie Konnektivitätsinformation verwendet werden kann welche entweder inhärent im Datensatz vorhanden ist, wie zum Beispiel im Fall von

OCTOPUS [Ta14] das die Gitterkonnektivität nutzt, um Abfragen auszuführen oder die explizit dem Datensatz hinzugefügt wird, wie wir mit FLAT [Ta12a] und SCOUT [Ta12b] zeigen. Die vorgeschlagenen Algorithmen dienen nicht nur als neue Lösungen für spezifische räumliche Datenverwaltungsprobleme. Die Forschung die in dieser Arbeit beschrieben wird dient viel mehr als Methodik für die Entwicklung einer neuen Klasse von Analysealgorithmen für räumliche Daten die von der Nutzung von Konnektivitätsinformation profitieren können. Beispiele von Algorithmen die auf ähnlichen Prinzipien beruhen und die räumliche Konnektivität von Datensätzen verwenden sind TOUCH [No13] und GPSY [Pa13]. Bei allen vorgeschlagenen Lösungen haben wir keine applikationsspezifischen Annahmen gemacht, d.h. keiner der Algorithmen ist gezielt für eine bestimmte Anwendung optimiert. Um die generelle Anwendbarkeit zu demonstrieren haben wir umfassende Experimente mit nichtwissenschaftlichen Datensätzen durchgeführt und ähnliche Resultate erzielt.

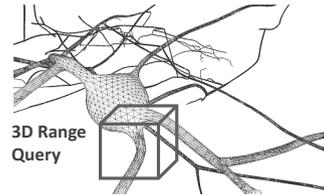


Abb. 6: Abfrage die über dynamischer Gitterstruktur ausgeführt wird um die Simulation zu überwachen.

Literatur

- [09] The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, Redmond, Washington, 2009.
- [Ar04] Arge, Lars; de Berg, Mark; Haverkort, Herman J.; Yi, Ke: The Priority R-tree: a practically efficient and worst-case optimal R-tree. In: SIGMOD. 2004.
- [Ch98] Chim, Jimmy HP; Green, Mark; Lau, Rynson WH; Va Leong, Hong; Si, Antonio: On caching and prefetching of virtual objects in distributed virtual environments. In: Proceedings of the sixth ACM international conference on Multimedia. ACM, S. 171–180, 1998.
- [Gu84] Guttman, Antonin: R-trees: A dynamic index structure for spatial searching, Jgg. 14. ACM, 1984.
- [HTA14] Heinis, Thomas; Tauheed, Farhan; Ailamaki, Anastasia: Spatial Data Management Challenges in the Simulation Sciences. In: Proceedings of the International Conference on Extending Database Technology. 2014.
- [KF84] Kamel, Ibrahim; Faloutsos, Christos: Hilbert R-Tree: An Improved R-Tree using Fractals. In: VLDB. 1984.

- [KLL02] Kwon, Dongseop; Lee, Sangjun; Lee, Sukho: Indexing the Current Positions of Moving Objects Using the Lazy Update R-tree. In: Proceedings of the Third International Conference on Mobile Data Management. MDM '02, IEEE Computer Society, Washington, DC, USA, S. 113–120, 2002.
- [LLE97] Leutenegger, Scott T; Lopez, Mario A; Edgington, Jeffrey: STR: A simple and efficient algorithm for R-tree packing. In: Data Engineering, 1997. Proceedings. 13th International Conference on. IEEE, S. 497–506, 1997.
- [No13] Nobari, Sadegh; Tauheed, Farhan; Heinis, Thomas; Karras, Panagiotis; Bressan, Stéphane; Ailamaki, Anastasia: TOUCH: In-Memory Spatial Join by Hierarchical Data-Oriented Partitioning. SIGMOD, 2013.
- [Pa13] Pavlovic, Mirjana; Tauheed, Farhan; Heinis, Thomas; Ailamakit, Anastasia: GIPSY: Joining Spatial Datasets with Contrasting Density. In: Proceedings of the 25th International Conference on Scientific and Statistical Database Management. SSDBM, ACM, New York, NY, USA, S. 11:1–11:12, 2013.
- [PK01] Park, Dong-Joo; Kim, Hyoung-Joo: Prefetch policies for large objects in a Web-enabled GIS application. *Data & Knowledge Engineering*, 37(1):65–84, 2001.
- [SOR09] Said, El Garouani; Omar, EL Beqqali; Robert, Laurini: Data prefetching algorithm in mobile environments. *European Journal of Scientific Research*, 28(3):478–491, 2009.
- [St13a] Stougiannis, Alexandros; Pavlovic, Mirjana; Tauheed, Farhan; Heinis, Thomas; Ailamaki, Anastasia: Data-driven neuroscience: enabling breakthroughs via innovative data management. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. S. 953–956, 2013.
- [St13b] Stougiannis, Alexandros; Tauheed, Farhan; Heinis, Thomas; Ailamaki, Anastasia: Accelerating spatial range queries. In: Proceedings of the 16th International Conference on Extending Database Technology. S. 713–716, 2013.
- [Ta12a] Tauheed, Farhan; Biveinis, Laurynas; Heinis, Thomas; Schürmann, Felix; Markram, Henry; Ailamaki, Anastasia: Accelerating Range Queries for Brain Simulations. In: Data Engineering (ICDE), 2012 IEEE 28th International Conference on. S. 941–952, 2012.
- [Ta12b] Tauheed, Farhan; Heinis, Thomas; Schürmann, Felix; Markram, Henry; Ailamaki, Anastasia: SCOUT: Prefetching for Latent Structure Following Queries. *Proc. VLDB Endow.*, 5(11):1531–1542, 2012.
- [Ta13] Tauheed, Farhan; Nobari, Sadegh; Biveinis, Laurynas; Heinis, Thomas; Ailamaki, Anastasia: Computational Neuroscience Breakthroughs through Innovative Data Management. In: *Advances in Databases and Information Systems*. Springer, S. 14–27, 2013.
- [Ta14] Tauheed, Farhan; Heinis, Thomas; Schürmann, Felix; Markram, Henry; Ailamaki, Anastasia: OCTOPUS: Efficient Query Execution on Dynamic Mesh Datasets. In: *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. 2014.



Die Forschungsinteressen von **Farhan Tauheed** umfassen das wissenschaftliche Datenmanagement. Im Rahmen des EU Projektes "Human Brain Project" arbeitete er mit Neurowissenschaftlern zusammen und entwickelte Lösungen welche derzeit in der Software zur Simulation von Hirnaktivität verwendet werden. Die Resultate wurden an den besten Datenbankkonferenzen veröffentlicht. Er arbeitet derzeit in der Forschungsabteilung von Oracle Labs. In seiner Freizeit reist er gerne und widmet sich der Fotografie.