

FORSCHUNGS- UND ENTWICKLUNGSPROJEKT 17/18

Beschreibung von Wachstums- und Verdrängungsprozessen in der Systembiologie mit Zellulären Automaten

Markus Noack

Betreut von

Prof. Dr. Anja Voß-Böhme

21. August 2017

Inhaltsverzeichnis

1	Zielstellung	1
2	Zelluläre Automaten	2
3	Vorarbeit	4
4	Methodik	5
5	Konzeptionelle Vorgaben der Simulation	6
5.1	Eingabeparameter	7
5.2	Vergleichsparameter	7
5.3	Umrechnung biologischer Kenngrößen in Wahrscheinlichkeiten	8
6	Programmaufbau	9
6.1	Zellaktionen	10
6.1.1	Teilung	10
6.1.2	Bewegung	10
6.1.3	Wachstum	10
6.2	Datengenerierung und Auswertung	10
7	Ergebnisse und Ausblick	12

1 Zielstellung

Zellen bilden die kleinste Einheit allen Lebens. Seit ihrer Entdeckung vor über 350 Jahren wird ihr Verhalten eingehend studiert und obwohl mittlerweile ein gutes Verständnis für die Prozesse auf zellulärer Ebene besteht, ist die Regulierung und Steuerung dieser Prozesse noch nicht vollständig ergründet.

Im Rahmen des Forschungs- und Entwicklungsprojekts 2017/2018 des Masterstudiengangs Angewandte Informationstechnologien soll die Frage beantwortet werden, ob sich Zellverhalten mittels eines zellulären Automaten (nachfolgend CA abgekürzt) beschreiben und vor allem erklären lässt.

Ziel des Projektes ist es, aufbauend auf den Ergebnissen eines Vorprojektes einen CA zu entwickeln, der die Wachstums- und Verdrängungsprozesse in der Systembiologie möglichst realistisch abbildet und somit zum Verständnis dieser beitragen kann.

Mit Hilfe eines funktionsfähigen Automaten könnte man noch unbekannte Korrelationen erkennen, die unter Umständen mit derzeitigen technischen Mitteln nicht messbar sind. Des Weiteren könnte man ihn nutzen um Vorhersagen zu treffen, zum Beispiel wie eine Verletzung heilen wird oder wie ein Tumor sich ausbreitet.

Diese Arbeit schätzt den Forschungsprojektstand nach dem Ersten, der Zwei für die Bearbeitung vorgesehenen Semester, ein. Um einen verständlichen Einstieg in das Thema zu geben, werden zunächst CA im Allgemeinen und damit einhergehende Begrifflichkeiten beschrieben. Anschließend wird ein Überblick über die bereits im Vorfeld dieser Arbeit bestehende Vorarbeit gegeben. Im Kapitel Methodik wird die Herangehensweise an das Projekt beschrieben. Darauf folgt ein Kapitel über die konzeptionellen Vorgaben an den CA. Im nächsten Kapitel wird die aus dem Konzept resultierende Implementierung beschrieben. Abschließend werden die bei der bisherigen Implementierung gewonnenen Ergebnisse evaluiert und ein Ausblick für das kommende Semester gegeben.

2 Zelluläre Automaten

CA ordnen sich in die Kategorie der diskreten Simulationen ein. Grundlage stellt dabei ein diskreter Raum dar, der auch oft als Spielfeld oder Gitter bezeichnet wird. Dieser Raum kann beliebige Dimensionen haben, wobei ein zweidimensionaler Raum wie in Abbildung 2.1 gängig ist.

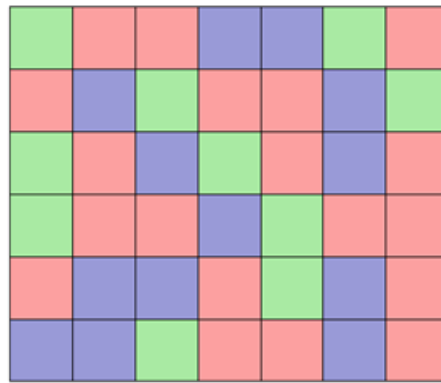


Abbildung 2.1: Zweidimensionaler CA mit 7x6 Gitterzellen [1]

Die einzelnen Gitterzellen des Automaten können durch festgelegte Regeln ihren Zustand wechseln. Die Zustandswechsel erfolgen in diskreten Zeitschritten. Das Besondere an zellulären Automaten ist, dass die Interaktion der Gitterzellen meist auf Nachbarschaften beschränkt sind, wobei diese für jede Zelle gleich definiert sind. Darum eignen sie sich gut um die Interaktionen auf biologischer Zellebene darzustellen, da Wachstum und Bewegung dort ebenfalls räumlich lokal beeinflusst werden. Es gibt mehrere Möglichkeiten zur Definition der Nachbarschaften. Abbildung 2.2(a) und Abbildung 2.2(b) stellen die bekanntesten Nachbarschaftsbeziehungen dar.

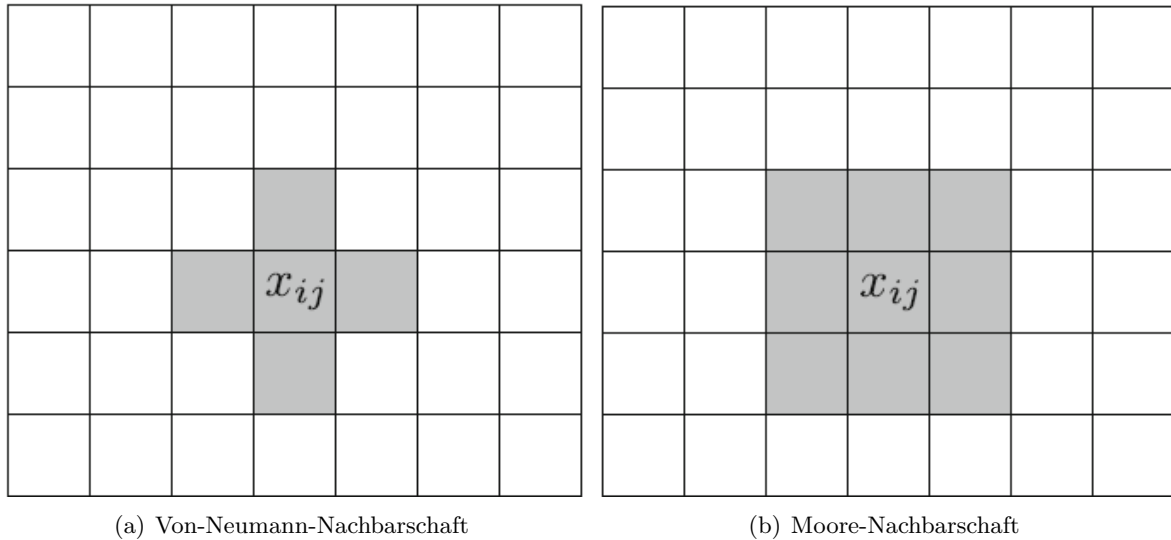


Abbildung 2.2: Nachbarschaftsbeziehungen [1]

Für den Fall, dass eine Gitterzelle am Rand des Spielfelds liegt und somit keine komplette Nachbarschaft innerhalb des Felds besitzt, müssen Randbedingungen definiert werden. Man unterscheidet zwischen abgeschlossenen und periodischen Randbedingungen. Bei einem abgeschlossenen Rand wird davon ausgegangen, dass Randzellen eingeschränkte Nachbarschaften haben und somit weniger Optionen für einen Zustandswechsel. Periodische Ränder hingegen ergänzen die fehlende Nachbarschaft durch Zellen vom gegenüberliegenden Rand.

Da wie bereits zu Beginn erwähnt, an die Vorarbeit von zwei Studenten angeknüpft wurde, werden im nächsten Abschnitt deren bisherige Ergebnisse vorgestellt bevor der derzeitige Projektfortschritt beleuchtet wird.

3 Vorarbeit

Hauptziel des bisherigen Projektverlaufs war es, den biologischen Sachverhalt durch unterschiedliche Modelle zu realisieren und diverse Darstellungsmöglichkeiten zu vergleichen. Damit einhergehend wurde eine Webanwendung entwickelt, die in zwei Darstellungsformen (Graphen-basiert: Abbildung 3.1(a) & Klassisch: Abbildung 3.1(b)) unterteilt ist.

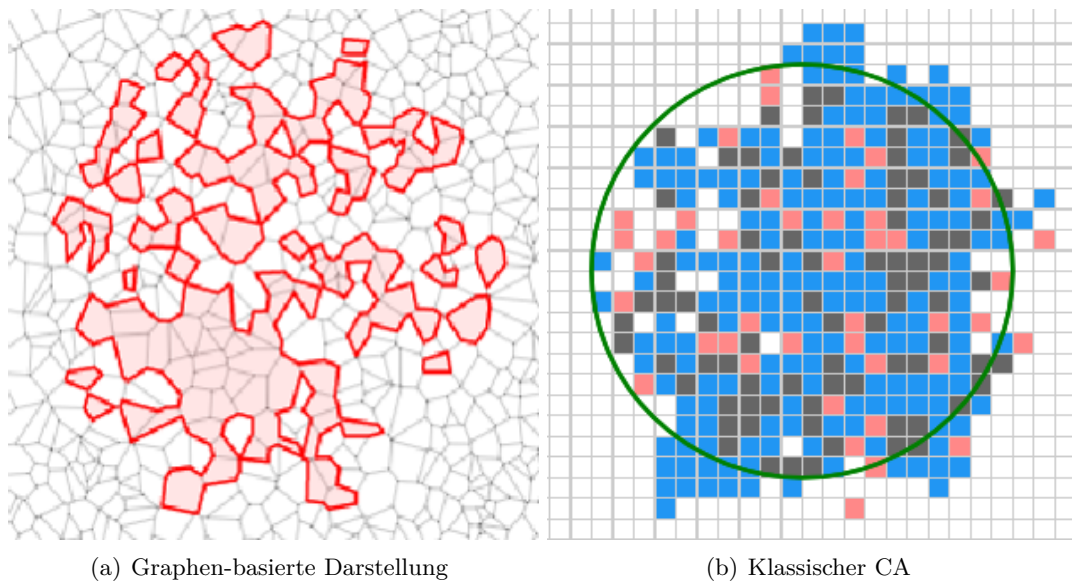


Abbildung 3.1: Darstellungsformen der Webanwendung [2]

Zudem wurden die Ergebnisse der beiden Simulationen mit den Ergebnissen der Simulationssoftware Morpheus verglichen. Morpheus basiert auf dem Cellular Potts Model und verfolgt einen deutlich komplexeren Simulationsansatz als zelluläre Automaten. Umso erstaunlicher ist es, dass mit allen drei Simulationen vergleichbare Resultate erzielt werden konnten. Der klassische CA-Ansatz hat sich dabei als am vielversprechendsten herausgestellt und sollte im weiteren Verlauf verfolgt werden. Da eine mit der Systembiologie vergleichbare Datengenerierung noch nicht im Vordergrund stand, wurden die Eingangsparameter des CA zum damaligen Zeitpunkt frei gewählt. Zudem beschränkte sich die Komplexität der einzelnen Simulationsschritte auf nur wenige einfache Regeln.

Die aktuelle Arbeit knüpft hier an, indem der CA an geeignete Kenngrößen der Systembiologie angepasst wird. Auf welcher Grundlage und mit welchen Mitteln dies erfolgt, wird im nächsten Kapitel beschrieben.

4 Methodik

Während der Einarbeitungsphase erregte ein Artikel von Alberto Puliafito (2010) [3] besondere Aufmerksamkeit. Dieser beschäftigt sich mit der Zellkontakthemmung, einer Eigenschaft von Zellen, das Zellwachstum und die Zellteilung ab einer bestimmten Zelldichte einzustellen. Der Artikel liefert dabei detaillierte Einblicke in das Wachstum von Nierenzellen. Da es sich bei den beobachteten Zellen um einlagige Gewebezellen handelt, eignen sie sich hervorragend für einen Vergleich mit einem zweidimensionalen zellulären Automaten. Eine genaue Auflistung, der dort entnommenen Kenngrößen, befindet sich in den Abschnitten über Eingabe- und Vergleichsparameter.

Mit dem Hinzukommen der zahlreichen Kenngrößen und der damit steigenden Komplexität, wurde der Grundaufbau des CA überdacht. Im Zuge dessen wurde beschlossen, die bisherige Implementierung in eine C#-Anwendung zu migrieren.

Die Webanwendung hat zwar eine sehr gute Visualisierung geboten, allerdings eignet sich JavaScript aufgrund von z.B. fehlenden Parallelisierungsmöglichkeiten nicht für umfangreichere Simulationen.

C# hingegen bietet durch die Nutzung von Threads eine deutlich performantere Grundlage für den zellulären Automaten. Zudem bietet sich mit Visual Studio eine mächtige Entwicklungsumgebung, die einfacheres Debuggen und Testen ermöglicht. Die konkrete Implementierung wird im Abschnitt Programmaufbau detaillierter beschrieben.

Die von der Anwendung generierten Daten werden zum Teil direkt im Programm analysiert, sowie in eine CSV-Datei gespeichert um dann weiterverarbeitet zu werden. Die Weiterverarbeitung erfolgt in MATLAB bzw. der alternativen Open Source Lösung Octave. Gleichzeitig kann an dieser Stelle eine Visualisierung der Daten durchgeführt werden. Details dazu finden sich im Abschnitt Datengenerierung und Auswertung.

5 Konzeptionelle Vorgaben der Simulation

Das Zellverhalten soll zunächst auf ein Zusammenspiel aus den drei Zellaktionen Bewegung, Wachstum und Teilung heruntergebrochen werden. Auf CA-Ebene spiegeln diese Aktionen Zustandsübergänge wieder. Da nicht bekannt ist welche Aktion eine Zelle in welchem Zeitschritt durchführt, wird jeder Aktion eine Wahrscheinlichkeit zugeordnet. Da ein Simulationsschritt somit eine Vielzahl von Zufallsexperimenten ist, kann man die Simulation auch als Monte-Carlo-Simulation betrachten. Ein Zeitschritt in der Simulation ist somit ein Monte-Carlo-Schritt (MCS).

Die Simulation besteht aus n hintereinander ausgeführten MCS, wobei n so gewählt wird, dass ein Simulationsdurchlauf einen Zeitraum von 15-20 Tagen simuliert. Damit die einzelnen Wahrscheinlichkeiten der Zellaktionen sich nicht gegenseitig ausschließen, also gewährleistet wird, dass eine Zelle pro MCS die Möglichkeit hatte sich zu bewegen, zu teilen oder zu wachsen, wird jeder MCS in x Einzelschritte unterteilt, wobei x die Anzahl der Zellen zu Beginn des MCS mal die möglichen Zellaktionen ist. Jeder dieser Schritte läuft nach folgendem Schema ab:

1. Eine zufällige Zelle wird ausgewählt.
2. Es wird gleichverteilt bestimmt welche Aktion die Zelle ausführt.
3. Nachdem alle externen Parameter eingeflossen sind, wird die Aktion zu einer bestimmten Wahrscheinlichkeit ausgeführt.

Somit ist gewährleistet, dass im Mittel jede Zelle die Möglichkeit hatte jede ihr möglichen Aktionen auszuführen.

Für die Nachbarschaftsbeziehungen wird die im Kapitel Zelluläre Automaten beschriebene Moore-Nachbarschaft verwendet, da Zellen keine statische Form besitzen und somit in Kontakt mit mehr als 4 Nachbarn treten können.

Für die Randbedingungen wird zunächst ein abgeschlossener Rand verwendet, wobei späteres Testen periodischer Randbedingungen denkbar ist.

Wie bereits erwähnt, konnten viele Kenngrößen aus dem Artikel von Alberto Puliafito (2010) [3] entnommen werden. Diese können entweder als Eingabeparameter für den CA dienen, oder als Vergleichswerte für die resultierenden Ergebnisse. Da mit vielen Wahrscheinlichkeiten gearbeitet wird, müssen einige der Kenngrößen umgerechnet werden. Dies wird am Ende des Kapitels am Beispiel der Bewegungsgeschwindigkeit verdeutlicht. Es gilt zu erwähnen, dass noch nicht alle hier aufgelisteten Parameter im jetzigen Stand des CA implementiert sind.

5.1 Eingabeparameter

Die Simulation hängt von vordefinierten Parametern ab, die entweder zwingend notwendig (Teilungswahrscheinlichkeit, Bewegungswahrscheinlichkeit, etc.) oder optional (z.B. Energielevel oder Teilungslevel) sein können.

Tabelle 5.1: Eingabeparameter

Parameter	Bemerkung
Knotenanzahl	Definiert die Zustandsraumgröße
Gitterabstand	Breite einer ausgewachsenen Zelle (<i>Gitterabstand</i> ² gibt die Größe einer ausgewachsenen Zelle an)
Zeitschrittlänge	Gibt an wie viele Stunden Echtzeit pro MCS vergehen
Anfangssituation	Immer gleich aber mit zwei Varianten. Entweder volles Rechteck im Zentrum, oder X% des gesamten Grids gefüllt (zufällige Verteilung)
Zellteilungswahrscheinlichkeit	Siehe Abschnitt Zellaktionen
Bewegungswahrscheinlichkeit	Siehe Abschnitt Zellaktionen
Zellwachstumsrate	Siehe Abschnitt Zellaktionen
Energielevel	Eingeengte Zellen schicken Signale an ihre Nachbarzellen und erhöhen damit deren Energielevel. Ein erhöhtes Energielevel sorgt für eine Priorisierung der Bewegung gegenüber anderer Zellaktionen. (Erster Ansatz für Verdrängungsmechanismen)
Teilungslevel	Ist maximal nachdem sich eine Zelle geteilt hat und wird in jedem Zeitschritt gesenkt. Ein erhöhtes Teilungslevel sorgt für eine Senkung der Teilungswahrscheinlichkeit.

5.2 Vergleichsparameter

Um die Ergebnisse verschiedener Simulationsdurchläufe miteinander vergleichen zu können, müssen Vergleichsparameter definiert werden.

Tabelle 5.2: Vergleichsparameter

Parameter	Bemerkung
Durchschnittliche Zellgeschwindigkeit	-
Koloniefläche	Summe der belegten Kapazitäten aller Knoten
Zelldichte	Koloniefläche im Verhältnis zur Zellanzahl

5.3 Umrechnung biologischer Kenngrößen in Wahrscheinlichkeiten

Zunächst muss eine Annahme darüber getroffen werden, wie viele Stunden realer Zeit ein MCS abdeckt. In der momentanen Implementierung wird davon ausgegangen, dass ein MCS 0,1 Stunden entspricht. Eine bekannte Kenngröße ist, dass sich Zellen ungefähr um eine Zellbreite einer ausgewachsenen Zelle pro Stunde bewegen. Da dies gleich der Kapazität eines Knotens im Gitter ist, kann ferner angenommen werden, dass durchschnittlich aller 10 MCS eine Zellbewegung durchgeführt werden muss. Somit ergibt sich eine Bewegungswahrscheinlichkeit von 10% je MCS.

Im nächsten Kapitel wird die C#-Anwendung beschrieben und dargestellt wie die in diesem Kapitel beschriebenen Schritte programmtechnisch umgesetzt wurden.

6 Programmaufbau

Kernstück des Programms ist eine Simulationsklasse, die der Übernahme der Simulationsparameter und der Ablaufsteuerung dient. Die Kenngrößen und Parameter sollen später über eine entsprechende Oberfläche flexibel auswählbar und einstellbar sein, um die Simulation ohne Anpassungsaufwand unter verschiedenen Bedingungen laufen lassen zu können. Dabei ist es möglich mehrere Instanzen der Simulationsklasse gleichzeitig zu erstellen, um parallele Simulationen zu ermöglichen.

Der Zustandsraum wird durch ein 2D-Array aus Knoten abgebildet. Ein Knoten beinhaltet neben seiner Position und seiner Kapazität eine Liste von Zellen. Die Kapazität ist für jeden Knoten gleich und sorgt dafür, dass nicht unendlich große und viele Zellen an einem Ort sein können. Der Ansatz den Raum aus Knoten, anstatt direkt aus Zellen bestehen zu lassen, wurde aus mehreren Gründen gewählt. Zum einen lassen sich viele kleine Zellen auf engem Raum darstellen. Ferner lässt sich die Zelldichte an einem Ort gut bestimmen, indem man Kapazität und Zellanzahl ins Verhältnis setzt. Des Weiteren lässt sich die Zellteilung realistischer darstellen, da sich Zellen an der Stelle teilen können und keinen neuen Gitterplatz beanspruchen.

Eine Zelle besitzt einen Zelltyp (Nierenzelle, Krebszelle, etc.) der Auskunft über folgende Eigenschaften gibt:

- Die Minimalgröße (Zellen können sich nicht mehr teilen, wenn die Minimalgröße erreicht ist. Dadurch wird gewährleistet, dass ein Knoten nicht unendlich viele Zellen beinhaltet.)
- Die Teilungswahrscheinlichkeit
- Die Bewegungswahrscheinlichkeit
- Die Wachstumsrate (Anteiliger Flächenzuwachs in %, wenn eine Zelle wächst)

Im nächsten Abschnitt sind die Anforderungen an die einzelnen Zellaktionen definiert.

6.1 Zellaktionen

6.1.1 Teilung

Eine Zelle kann sich teilen, solange durch die Teilung ihre Minimalgröße nicht unterschritten wird. Die Teilung erfolgt dabei an der Stelle, d.h. zur Zellliste des Knotens, in dem sich die Zelle befindet, wird eine neue Zelle hinzugefügt. Nach der Teilung entsprechen die Größen von Mutter- und Tochterzelle der Hälfte der ursprünglichen Größe der Mutterzelle.

6.1.2 Bewegung

Eine Zelle kann sich in einen benachbarten Knoten bewegen, vorausgesetzt der Zielknoten bietet genügend Platz. Die Zelle wird aus der Zellliste ihres aktuellen Knotens entfernt und der Zellliste des Zielknotens hinzugefügt.

6.1.3 Wachstum

Solange der umschließende Knoten über nicht belegte Kapazität verfügt, kann eine Zelle innerhalb des Knotens wachsen. Sie wächst dabei um $Y\%$ ihrer ursprünglichen Größe an. Sollte der vorhandene Platz dafür nicht ausreichen, beansprucht sie die ihr zur Verfügung stehende Kapazität.

Nachdem der Programmablauf erläutert wurde, wird im folgenden Abschnitt die Ausgabe des Programms beschrieben.

6.2 Datengenerierung und Auswertung

Für erste statistische Auswertungen wurde eine statische Klasse Statistics angelegt, in der zur Laufzeit Daten gesammelt werden. Momentan dient die Klasse lediglich zu Testzwecken. Unter anderem ermittelt sie die Anzahl der versuchten Teilungen, sowie die Anzahl der tatsächlich durchgeführten Teilungen. Damit kann nachvollzogen werden ob der Zufallsgenerator eine korrekte Gleichverteilung erzeugt und ob die angegebene Teilungswahrscheinlichkeit eingehalten wird.

Abhängig vom gewählten Darstellungsmodus wird der komplette Simulationszustand aller X Zeitschritte in eine CSV-Datei gespeichert.

Die Darstellung der Daten erfolgt über MATLAB/Octave. Je nach Datenmenge können dabei verschiedene Ergebnisse erzielt werden:

- Animation: Jeder Simulationsschritt wird abgebildet.
- Snapshot: Ein bestimmter Zeitschritt wird als Bild dargestellt.

- Overlap: Die Bilder von X Zeitschritten werden übereinandergelegt. (Analog zu Abbildung 6.1)

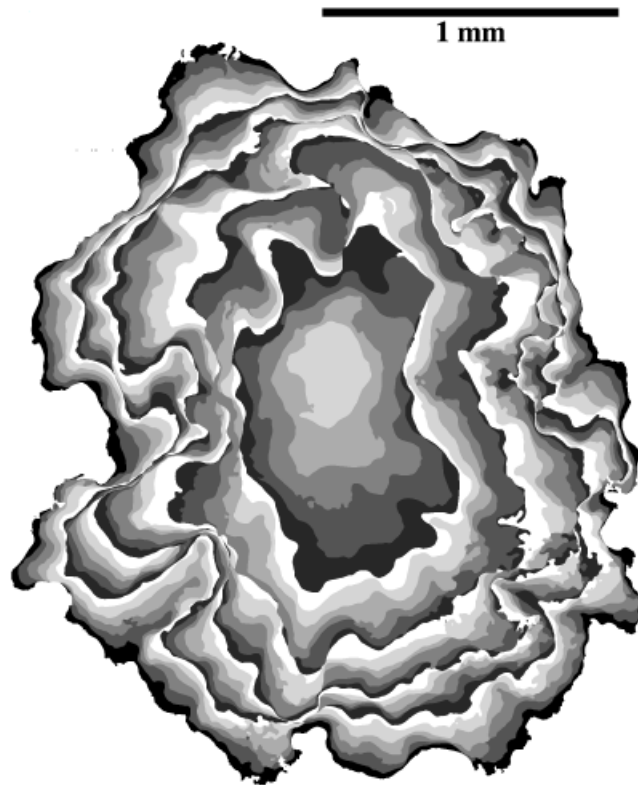


Abbildung 6.1: Overlap einer Nierenzellenkolonie [3]

7 Ergebnisse und Ausblick

Im Laufe des Semesters konnte die Zielstellung des Projekts eingegrenzt werden. Die Kenngrößen aus dem im Kapitel Methodik beschriebenen Artikel liefern ein konkretes Ziel für die Entwicklung des zellulären Automaten. In seiner momentan noch prototypischen Form kann der CA erste Daten generieren und für die Weiterverarbeitung speichern. Eine Visualisierung der Daten mittels MATLAB befindet sich bereits in der Entwicklung.

Im nächsten Schritt soll die Funktionsweise der prototypischen Implementierung mittels Sensibilitätstests auf ihre Richtigkeit geprüft werden. Konkret bedeutet dies, einzelne Simulationskomponenten zu isolieren und zu testen.

Anschließend soll die Laufzeit des CA mit einer für eine Zellkolonie realistischen Zellanzahl getestet werden. Eine Prüfung des Parallelisierungsgrad sollte damit einhergehen. Sollten zusätzliche Geschwindigkeitsoptimierungen nötig sein, wäre eine parallelisierte Aktualisierung der Knoten innerhalb eines Zeitschritts denkbar indem der Zustandsraum auf einzelne Prozessoren aufgeteilt wird.

Des Weiteren müssen Testroutinen entwickelt werden um die Ergebnisse unterschiedlicher Simulationsdurchläufe untereinander und mit den biologischen Kennzahlen zu vergleichen. Auf dieser Grundlage kann dann die Komplexität des CA nach und nach gesteigert werden.

Erste geplante Erweiterungen des CA umfassen die im Kapitel Parameter beschriebenen Energielevel und Teilungslevel. Diese spiegeln lokale Einflüsse auf die Zellaktionen wieder. Einen großen Bestandteil lokalen Zellverhaltens bilden Verdrängungsmechanismen, welche bisher noch nicht berücksichtigt wurden. Im Falle einer durch Platzprobleme unmöglichen Bewegung oder eines nicht möglichen Wachstums könnten Zellen aus dem Zielknoten „vertrieben“ werden, um die geplante Zellaktion zu ermöglichen.

Eine weitere denkbare Erweiterung wäre eine eigene Historie für jede Zelle. In dieser könnten z.B. Bewegungspfade oder Teilungszeitpunkte gespeichert werden um später daraus Informationen zu gewinnen.

Letztendlich müssen solange Anpassungen an den Parametern vorgenommen werden, bis der zelluläre Automat realistische Ergebnisse produzieren kann.

Literaturverzeichnis

- [1] D. Scholz, *Pixelspiele - Modellieren und Simulieren mit zellulären Automaten*. Berlin Heidelberg New York: Springer-Verlag, 2014.
- [2] M. Krockert and R. Erzgräber, “Cell simulator.” <http://krockema.github.io/>.
- [3] A. Puliafito, “Collective and single cell behavior in epithelial contact inhibition,” tech. rep., Kavli Institute for Theoretical Physics, Nov. 2011.