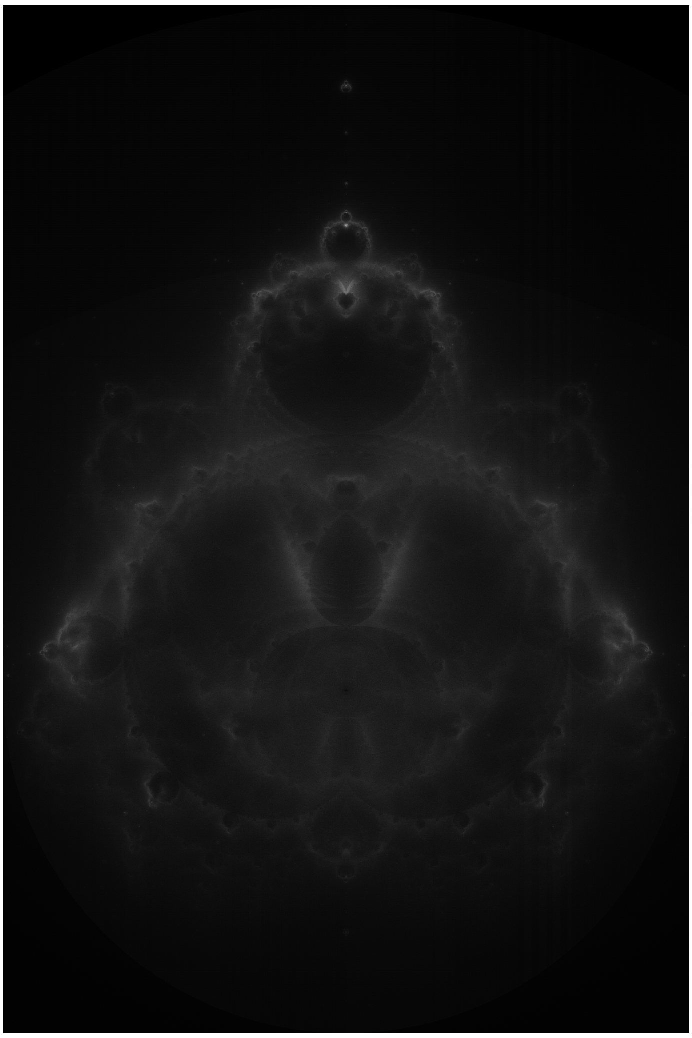
Zoom in das Buddhabrot

Ambros D. Anrig
2. Dezember 2021



Maturaarbeit
Kantonsschule Glarus
Betreuer: Fabio Thöny
Referent: Beat Temperli

Vorwort

Das Unendliche zieht die Menschen schon eh und je an, so ebenfalls mich. Es ist unbeschreiblich, doch spielen & rechnen wir damit gerne. Dazu kommt, dass unsere Technik immer besser und leistungsfähiger wird. Dies spielt sich zu, denn desto mehr Leistung man hat, umso schneller und näher kann man sich der Unendlichkeit approximieren. Ich bin schon seit ich klein bin, von komplexeren mathematischen Vorgängen beeindruckt, deshalb schaue ich in meiner Freizeit gerne YouTube-Videos über solche. Eines Tages sah ich das Buddhabrot auf dem Titelbild eines Videofilms. Als ich es mir dann anschaute, verstand ich nichts davon, denn es war auf Englisch. So beschäftigte ich mich vorerst nur mit der Mandelbrot-Menge. Erst als ich in der Schule die Fraktalen kennenlernte, verstand ich es besser. So entschloss ich mich nun, es nochmals zu versuchen, mich mit dem Buddhabrot auseinander zu setzen. So schreibe ich unter anderem deshalb eine Arbeit darüber.

Ich möchte mich noch am Anfang bei einigen Leuten bedanken. Zuerst bei meiner Betreuungsperson Fabio Thöny, der mir bei jeglichen Fragen zur Verfügung stand und mich unterstützte. Ebenfalls ist meiner Mutter Bernadett Rita Bejczy zu danken, die meine Arbeit entgegenlas. Auch Julian Steiner danke ich, der mir beim Programmieren mental zur Seite stand und mir bei allfälligen Problemen versuchte und schaffte zu helfen. Ebenfalls geht mein Dank an Linus Romer, der mir bei der CUDA-Implementierung half.

Inhaltsverzeichnis

1 Einleitung

An der Kantonsschule Glarus werden im Schwerpunktfach Anwendung der Mathematik und Physik unter anderem auch die Julia-Mengen und die Mandelbrotmenge angeschaut. Kurz wird ebenfalls das Buddhabrot erwähnt. Man schaut auch einen Zoom an, welcher in die Mandelbrotmenge hineinfokusiert. Es sind jedoch wenige Zooms ins Buddhabrot zu finden, geschweige denn solche, welche gleich tief in das Buddhabrot gehen, wie die beim Mandelbrot.
Ziel dieser Arbeit ist es, einen Zoom vom Buddhabrot zu berechnen. Es wird nach einer möglichst effizienten Methode für die Berechnung gesucht. Es soll mit rein mathematischen Algorithmen bewerkstelligt werden, jedoch werden schon zum Anfang leichte Hilfen von Programmiertricks benutzt. Diese Arbeit wird mit der Programmiersprache Julia erstellt, einer schnellen und verständlichen Sprache. Es wird jedoch kein Video erstellt, sondern ein Bild. Ein Video ist eine rasche Abfolge von Bildern. Hat man also eine Methode für das Bild, so ist der Schritt zum Video schon erleichtert. Jedoch würde dieser zusätzliche Schritt den Rahmen dieser Arbeit strapazieren.

2 Präliminarien

2.1 Komplexe Zahlen

Wenn man mit den reellen Zahlen arbeitet, bekommt man Probleme, wenn man die Wurzel aus einer negativen Zahl zieht. Jedoch haben Mathematiker im 17. Jahrhundert eine Lösung für dieses Problem entdeckt, indem dieses Zahlensystem mit den imaginären Zahlen, die die imaginären Einheit i haben, erweitert wird (Geschichte des Zahlenbegriffs 1970, S.66). Addieren und Subtrahieren zweier imaginären Zahlen funktioniert genau gleich, wie wenn man mit einer reellen Zahl rechnet. Dies heisst, dass das i wie die 'herkömmlichen' Variablen behandelt werden kann. Jedoch muss man beim Multiplizieren, Dividieren und beim somit entstehenden Rechnen mit Potenzen aufpassen, denn es gilt für $n \in \mathbb{Z}$:

$$\begin{aligned} \mathrm{i}^{4n} &= 1 \\ \mathrm{i}^{4n+1} &= \mathrm{i} \\ \mathrm{i}^{4n+2} &= -1 \\ \mathrm{i}^{4n+3} &= -\mathrm{i} \end{aligned}$$

Man sollte Potenzen mit der Basis i nach den oben genannten Regeln vereinfachen. Hier sieht man gut, dass eine Verknüpfung zwischen den imaginären und reellen Zahlen in die andere Direktion ebenfalls existiert. Wenn man nun eine imaginäre Zahl ib mit einer reellen Zahl a zusammenaddiert, bekommt man eine komplexe Zahl $c = a + ib$ mit dem Realteil a und dem Imaginärteil b. a und b sind hier reele Zahlen. Beim Addieren von komplexen Zahlen $z = a + \mathrm{i}b$ und $w = e + \mathrm{i}f$ folgt man diesem Beispiel:

$$z + w$$

$$a + \mathrm{i}b + e + \mathrm{i}f$$

$$a + e + (b + f)\mathrm{i}$$

Nun merkt man, dass eine komplexe Zahl mit einem Vektor vergleichbar ist, denn um die Zahl darstellen zu können, benutzt man die 2-dimensionale komplexe Ebene (Mathematik - Die faszinierende Welt der Zahlen 2015, S. 144). Daraus schliesst sich, dass komplexe Zahlen 2-dimensional sind. Multipliziert man eine komlexe Zahl und beaobachtet dies auf der komplexen Ebene, fängt der Punkt an, scheinbar unkontrolliert herumspringen. Der Punkt folgt jedoch weiterhin logischen Regeln. Beim Quadrieren verschiebt sich der Punkt in die positive Drehrichtung (Gegenuhrzeigersinn).
Ebenfalls kann, da die Zahl vergleichbar mit einem Vektor ist, der absolute Betrag der komplexen Zahl c bestimmt werden, welcher mit dem Pythagoras berechnet wird (Ein Weg zur fraktalen Geometrie 1989, S. 22):

$$|c| = |a + \mathrm{i}b| = \sqrt[2]{a^2 + b^2}$$

2.2 Fraktale Geometrie

2.2.1 Allgemein

Um zum Buddhabrot zu kommen, müssen wir noch einen weiteren Begriff klären: das Fraktal. Würde bei einem $3n$ grossem Strich der mittlere Drittel fehlen, an dieser die anderen zwei Seiten eines gleichseitigen Dreiecks mit der Seitenlänge n stehen, hätte man die erste Iteration einer Kochkurve. Würde man nun in die einzelnen n grosse Striche die vorige Iteration der Kochkurve einfügen, ensteht die zweite Iteration. Man kann dies ab nun immer wieder machen, sodass ein immer detailierteres und komplizierteres Bild entsteht.

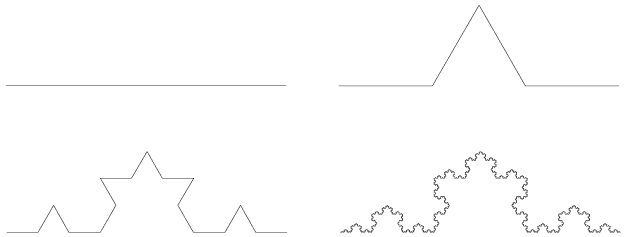


Abbildung 1: Die Enstehung der Kochkurve

Wenn man in die Kochkurve hineinzoomt, findet man die Kochkurve immer wieder: ein rekursives Bild oder eben ein Fraktal (Lexikon der Mathemathik - 3 2001, S. 128).
Man definiert nun das Fraktal als eine Figur, bei der sehr oft Selbstähnlichkeit auffindbar ist, das heisst, dass das gesamte Fraktal oder Teile davon mehrfach im Fraktal vorkommen und das Fraktal selbst eine gebrochene und somit keine ganzzahlige Dimension besitzt (Mathematik - Die faszinierende Welt der Zahlen 2015, S. 258).

2.2.2 Mandelbrotmenge

Die nach dem Mathematiker Benoît B. Mandelbrot (*20.11.1924; †14.10.2010) bennante Menge ($\mathbb{M}$) beinhaltet jede Zahl c die nicht gegen ∞ divergiert für folgende Folge (Ein Weg zur fraktalen Geometrie 1989, S. 54):

$$z_0 = 0$$
$$z_{n+1} = z_n^2 + c$$

Man fand heraus, dass wenn $|z_n| > 2$ gilt, wird die Folge gegen ∞ divergieren (Ein Weg zur fraktalen Geometrie 1989, S. 74).
Die erstellte Abbildung, jenachdem mit ein bisschen Farbe dazu, ergibt ein sehr schönes Gebilde, dass die deutschprachigen Leute aufgrund der Form an ein 'Apfelmännchen' errinerte, weshalb es von ihnen auch so genannt wird (Ein Weg zur fraktalen Geometrie 1989, S. 54).
Die $\mathbb{M}$ ist ein Fraktal (Mathematik - Die faszinierende Welt der Zahlen 2015, S. 258). Man findet das erst gesehene Bild der Menge beim hineinzoomen immer wieder, somit ist es ebenfalls selbstähnlich. Immer wieder findet man verschiedene Julia-Mengen mit dem Zugehörigem c (Ein Weg zur fraktalen Geometrie 1989, S. 54). Diese sind ebenfalls Fraktale und sehen dazu auch noch für die Norm der Menschheit schön aus. $\mathbb{M}$ besitzt durch die vorgegebene Formel ein chaotisches System (Ein Weg zur fraktalen Geometrie 1989, S. 32).
All dies führt sicher dazu, dass es einige YouTube-Viedeos gibt, die einen Zoom in die Menge zeigen, welche auch viel Rechenleistung brauchen.

2.2.3 Buddhabrot

Schaut man den Namen als Erstes an, merkt man, dass das Wort 'Buddha' vom meditierenden Buddha kommt, denn dieser ist in der Abbildung ersichtlich. Ebenfalls fällt das Wort 'Brot' auf. Dies ist eine Andeutung, dass diese Abbildung etwas mit dem Mandelbrot zu tun hat, denn

es stellt eine andere Variante dar, die $\mathbb{M}$ abzubilden.
Das Bild entseht, indem das Mandelbrot nochmals berechnet wird, jedoch nur die Punkte, die bei der Mandelbrotberechnung gegen das ∞ divergieren. Nun wird auch nicht mehr geschaut, nach wie vielen Schritten der Punkt c ins ∞ abdriftet, sondern bei welchen Punkten c nach jeder Iteration landet.
Ebenfalls wird ein Zoom in das Buddhabrot durch das chaotische System von $\mathbb{M}$ und durch die fraktalen und selbstähnlichen Eigenschaften von $\mathbb{M}$ interessant.

3 Methode

3.1 Allgemeines Rechnen

Um Bilder vom Buddhabrot zu generieren wurde mit der Programmiersprache Julia eine 2-Dimensionale Matrix erstellt, bei dem jeder Wert der Matrix zu einem Pixel zugehört. Zuerst muss man jedoch wissen, welche Punkte man iterieren muss, so wird zuerst das Mandelbrot ausgerechnet. Um Speicherplatz zu sparen, ordnet man ihnen die Werte 0 und 1 zu: 1 divergiert gegen ∞, 0 gehört der Mandelbrotmenge an. Alle Punkte, die gegen ∞ divergieren, werden nochmals iteriert, nun wird geschaut wo die Punkte durchgehen. Am Ende wird geschaut, welcher Punkt die meisten Treffer bekam, den dieser Wert wird gebraucht, um die Grauabstufung zu machen. Wie man sicher schon merkt, muss mal 3 Mal die riesige Matrix durchgehen und auch 2-Mal iterieren. Dies ist somit im Vergleich zur Mandelbrotmenge sehr rechenaufwändig, denn bei der wäre nur ein Durchlauf nötig. Als Deffinitionsbereich wird $\{z \in \mathbb{C} \mid -2 < \Re(z) < 1 \& -1 < \Im(z) < 1\}$ gewählt.

3.2 Erstellen eines ersten Zoom

Beim ersten Ansatz wurde für den Zoom eine riesige Matrix, welche der Zoom im Quadrat so gross ist, wie das zu erwartene Bild, erstellt mit all den Werten drin, die man braucht. Dann wurde der Teil ausgewählt, den man haben möchte und lässt den Zeichnen. Dies mithilfe der Berechnung vom Offset im Array, welches Bewertstelligt wird, indessen man angibt welches der zum Bezoomende Punkt ist.

3.3 CUDA optimierung

Um lange Wartezeiten zu vermeiden, wurde der Code mit CUDA formuliert. Es wurde CUDA.jl hinzugefügt, um so mehrere Punkte gleichzeitig rechnen zu lassen. Dadurch, dass nun auf der Grafikkarte gerechnet wird, hat man weniger Speicherplatz zur Verfügung. Für dies müssen die Funktionen umgeschrieben werden, damit CUDA dies kennt. Zu bemerken ist, dass CUDA eine Plattform um auf der Grafikkarte zu Rechnen ist, welche durch CUDA.jl mit Julia genutz werden kann. CUDA ist von Nvidia, was dazu führt, dass das Programm nurnoch auf PC laufen kann, die eine Grafikkarte von Nvidia haben.

3.4 Analyse

Da das Buddhabrot als chaotisches System gilt, versucht man im Chaos ein Muster zu finden. Als erstes wurde geschaut, ob ein gegebener Bereich einen überwiegender Einfluss hat auf einen Bereich, als die anderen Bereichen oder auch markant keinen Einfluss. So könnte man in einem Zoom, nur noch diesen Bereich anschauen oder eben nicht. Dies wurde einfach bewertstelligt, indessen man Bilder erstellte, die zeigten, wie sich Punkte aus diesen Bereichen iterierten. Zuerst wurden die 4 Quadranten als Startbereiche gewählt.
Als zweites wurde noch zusätzlich die Überlegung gemacht, dass der absolute Wert von c ebenfalls einen Einfluss haben könnte, wie wenn der kleiner als 1 wäre. Dies wurde getestet indessen man eine Variable dem vorigem Aufbau mitgab, welches mit einem XOR dafür sorgte, dass entweder der Bereich, bei dem $|c| >= 1$ gilt, oder der andere ausgewertet wird.
Die daraus Resultierende Bilder wurden Analysiert, indessen man die Flächen bestimmte, welche Pixelwerte haben von RGB{Float64}(0,0,0) mit sehr kleiner Tolleranz.

3.5 Implementierung der Ergebnisse

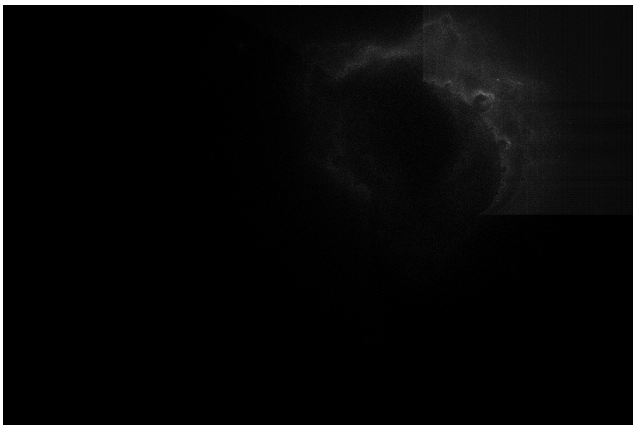
Der zu verwerfende Bereich wurde schon in der Mandelbrotberechnung verworfen, um Zeit zu sparen. Die anderen nützlichen Ergebnissen (In Kapitel Ergebnisse schauen) wurde geschaut in welchen Bereichen auf den Analysenbilder es Schwarz ist oder nur ein Treffer gezeigt wird. Danach wurde diese Fläche zu einer Funktion umgewandelt. Mit dieser wird geschaut, ob der massgebende Eckpunkt des Zoomsbereich in der Fläche ist. Falls es so ist, wird dessen nicht Quadranten miteinbezogen. So muss man jenachdem nurnoch 2 Quadranten von einem Bereich berechnen.

4 Ergebnisse

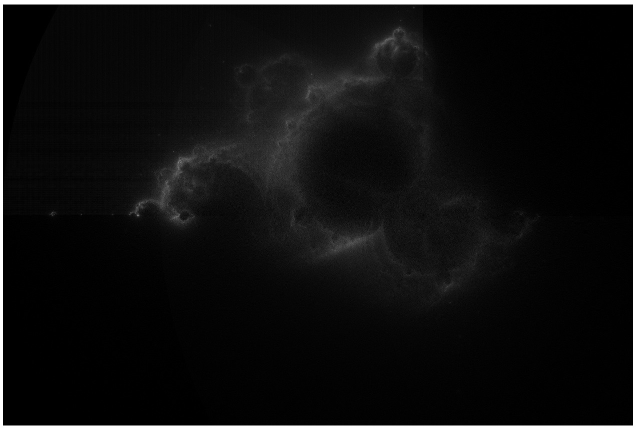
4.1 Zahlen ohne Optimierung

Beim ersten Zoom war ein 16-Facher Zoom nicht mehr möglich, da die Matrix so gross wird, dass es nicht genügend RAM freiräumen kann. Ebenfalls dauert es dann schnell mal viel Zeit, ein 12-facher Zoom, mit der Auflösung 4'001 auf 2'667 Pixel, zum Punk -1.25 und mit 150 Iterationen dauerte etwa 45.6h.
Durch die CUDA optimierung kann man nur noch einen 6.25-fachen Zoom machen, bei gleichen Einstellung, bis auf die Zoomtiefe auf 6.25, konnte das Programm innerhalb von 38 Sekunden Fertig gerechnet sein. Das einen Zoom nicht gleich Tief möglich ist, ist eigentlich egal, denn es geht ja darum, die Rechenleistung zu verringern und einen Allfälliger Allgorythmus zu finden. Das gleiche Programm, ein 1-facher Zoom, dauert nun anstelle von 3 Stunden nurnoch 2 Minuten.

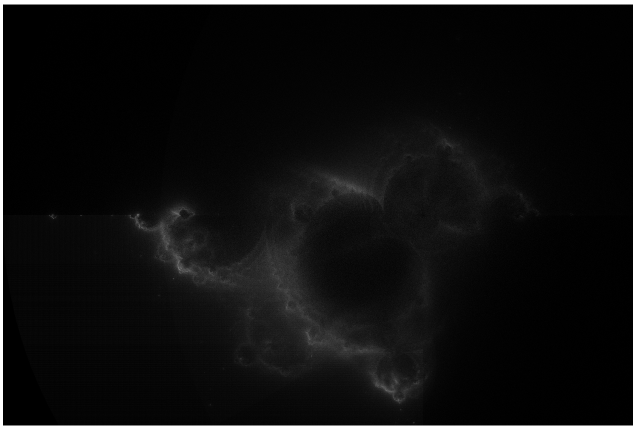
4.2 Analyse Ergebnisse



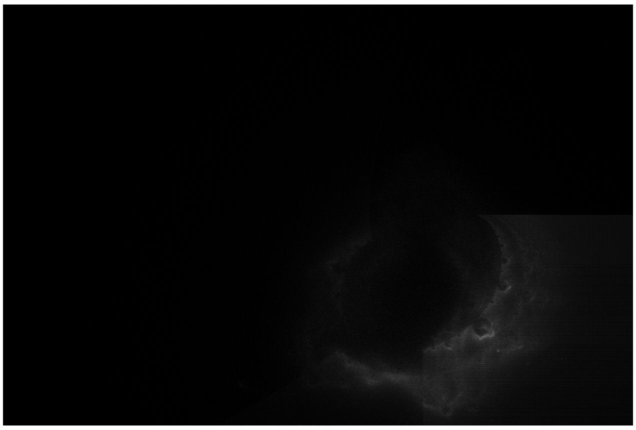
(a) 1. Quadrant



(b) 2. Quadrant



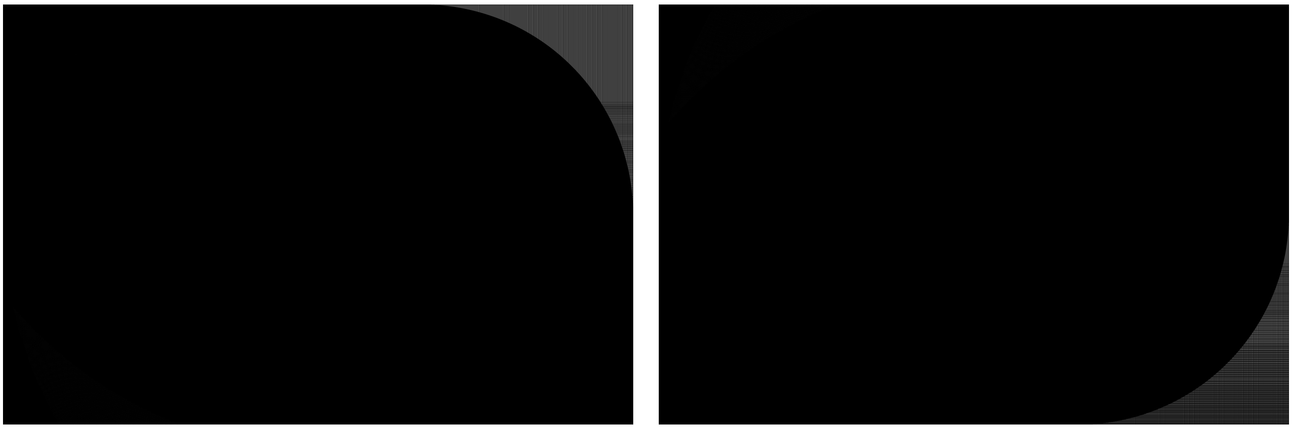
(c) 3. Quadrant



(d) 4. Quadrant

Abbildung 2: Das Buddhabrot der einzelnen Quadranten

Bei Abbildung 3 ist der Maximal erreichte Treffer auf einem Pixel 4.



(a) 1. Quadrant

(b) 4. Quadrant

Abbildung 3: Das Buddhabrot für $\{z \in \mathbb{C} | |z| > 1\}$ in den Quadranten 1 & 4

4.3 Implementierungszahlen

Durch die gefunde Lösung ist zumal nurnoch einen 6.48-Facher Zoom möglich, dies wird daran liegen das es zwei Variabeln hat, welch ein kleiner einfluss ist, und dass nun grosse Matrixen in einer Liste zu finden sind, welches viel Speicherplatz nimmt. Es ist jedoch ein minimaler Verlust und somit nicht schlimm. Ein 6.48-Facher Zoom zum Punkt -1.25, mit der Auflösung 4'002 auf 2'668 Pixel und mit 100 Iteration dauert es nur noch 2 Minuten und 10 Sekunden. Bei einem 6.25-fachem Zoom braucht es 2 Minuten 1 Sekunde.

5 Disskussion

Bei den Analysen der Quadranten hat sich gezeigt, dass die Quadranten auf Bereichen kein Einfluss geben und auf gewissen Starken. Die Quadranten hatten auf Folgende Bereiche keinen Einfluss im Definitionsbereich:

1. Quadrant

$$\{z \in \mathbb{C} \mid \Im(z) > 57.4\frac{\Re(z)+2}{11.4} - \frac{29.8}{3.8}\}$$

2. Quadrant

$$\{z \in \mathbb{C} \mid ((\Re(z) - \frac{3'504}{667})^2 + \Im(z)^2 > 6.25 \;\&\; \Im(z) < -\frac{25}{667}) \vee (\Im(z) > (\Re(z) - 1)^2)\}$$

3. Quadrant

$$\{z \in \mathbb{C} \mid ((\Re(z) - \frac{3'504}{667})^2 + \Im(z)^2 > 6.25 \;\&\; \Im(z) > \frac{25}{667}) \vee (\Im(z) < -(\Re(z) - 1)^2)\}$$

4. Quadrant

$$\{z \in \mathbb{C} \mid \Im(z) < -57.4\frac{\Re(z)+2}{12} + \frac{24.5}{4}\}$$

Bei den durchgeführten Analysen mit den Radien hat sich gezeigt, dass vom Bereich $\{z \in \mathbb{C} \mid |z| > 1 \;\&\; 1 >= \Re(z) >= 0 \;\&\; -1 <= \Im(z) <= 1\}$ nur ein maximaler Treffer von 4 erreicht wird. Somit kann dieser Bereich vernachlässigt werden in der Berechnung, da 4 von in der Norm 69 Maximum sehr wenig ist und vor allem der Beinflusste Bereich nicht wirklich erkenntlich ist beim Anblick des Buddhabrotes.

6 Fazit

Eine Steigerung ist klar ersichtlich, zwar wurde einen Punkt gewählt bei dem es die Variante gelohnt hat, jedoch sind auch die Punkte in dieser Umgebung sehr Spannend. Hätte man einen Anderen Punkt ausgewählt, wäre eine klare Verbesserung nicht ersichlich, wenn den Sogar vorhanden. Ebenfalls ist das Programm minimal nicht gleich effizient, da es nun mehr if-Konditionen hat, welche so das Programm verlangsamen.
Es gibt einiges, dass man probieren hätte gekonnt, welches einen Zoom tiefer gemacht hätte, eines währe eine Datenbank, welches währen dem Rechnen erstellt währe, sodass die Threads bei alten Resultaten hätten weiter Rechnen gekonnt. Das Programm währe zwar wiederum langsamer, da nun mehr auffrufe auserhalb der CUDA geschehen. Eine andere Methode währe den Metropolis-Hashtings Algorithmus von Alexander Boswell nutzen. Dies wurde nicht gemacht, da dort die Warscheinlichkeit, dass ein Punkt divergiert berechnet wurde und so nicht unbedingt alle Punkte miteinbezogen werden. Dies ist durch den Verwerfungsbereich, zwar bei der Lösung dieser Arbeit ebenfalls nicht gegeben, jedoch, war der Maxtreffer 4 bei verschiedenen grössen der Variabel Iterationen.

7 Selbständigkeitserklärung

8 Quellenverzeichnis

8.1 Literaturverzeichniss

[1] Reinhart Behr, *Ein Weg zur fraktalen Geometrie*, Ernst Klett Schulbuchverlag, Stuttgart, 1989.

[2] Bertram Maurer, *Mathematik - Die faszinierende Welt der Zahlen*, Fackelträger Verlag GmbH, Köln, Emil-Hoffmann-Strasse 1, D-50996 Köln, 2015.

[3] Liste, *Lexikon der Mathemathik*, Spektrum Akademischer Verlag GmbH Heidelberg, Berlin & Heidelberg, 2001.

[4] Helmuth Gericke, *Geschichte des Zahlenbegriffs*, Bibliographisches Institut, Mannheim, 1970.

8.2 Bilderverzeichnis