

Die Spiraldarstellung - ein experimentelles Visualisierungsverfahren

Nikita Braguinski

16.02.2015

Das hier beschriebene Verfahren stellt Klänge und andere Daten in Form eines Bildes dar, das in vielen Fällen erlaubt, zu erkennen, ob diese Daten ein Rauschen oder ein geordnetes Signal bilden. Gleichzeitig erlaubt es, störende Abweichungen in gleichbleibenden Schwingungen visuell zu erkennen (ich danke Frau Daniela Rings von Humboldt Innovation für diese zusätzliche Anwendungsidee). In seinem Kern besteht das Verfahren in der Verwendung der sogenannten Ulam-Spirale, einer mathematischen Darstellungsform, für die visuelle Datenanalyse. Die Art der Darstellung wird dabei auf die Besonderheiten der menschlichen Wahrnehmung abgestimmt (Unterstützung der sogenannten Gestaltbildung).

Bisherige Versuche, die Form der Spirale für die Datenanalyse zu verwenden, zielen nach meinem Wissensstand lediglich darauf, in den dargestellten Daten eine Periodizität aufzudecken, die sich bei solcher Darstellung als sich wiederholende Muster zeigt. Die dabei fast durchgängig verwendete runde ("archimedische") Spirale, teilweise mit Abständen zwischen den einzelnen Abschnitten der Kurve, stellt im Hinblick auf die Besonderheiten der menschlichen Wahrnehmung meiner Meinung nach kein effizientes Arbeitswerkzeug dar. Die oft unterschiedlich großen Bildelemente verzerren die Darstellung zusätzlich, denn die durch sie visualisierten Datenbereiche sind gleich groß. Die in meinem Verfahren eingesetzte viereckige Spirale dagegen, die durch gleich große pixelartige Bildelemente ein zusammenhängendes Bild ergibt, eignet sich hervorragend zur Herstellung von Überblicken und zur Visualisierung von Signalabweichungen. Die Bandbreite der Datenwerte wird dabei oft auf nur zwei diskrete, kontrastierende Zustände reduziert, um das unbewusste Bilden von visuellen Strukturen (sogenannten Gestalten) im Bild zu erleichtern. In manchen Anwendungsbereichen kann dagegen auf die kontinuierliche Abbildung des Spektrums der Datenwerte durch eine Palette an Farbwerten zurückgegriffen werden.

Die Idee, eine rechteckige, zusammenhängende kontrastreiche Spirale aus gleich großen Elementen für die Datenanalyse einzusetzen, entwickelte ich auf der Grundlage der sogenannten Ulam-Spirale, einer vom Mathematiker Stanislaw Ulam 1963 entwickelten Methode der grafischen Darstellung von Primzahlen mit Hilfe von Punkten auf einer rechteckigen Spirale. Eine der frühesten Darstellungen findet sich in: M. L. Stein, S. M. Ulam, M. B. Wells: A Visual Display of Some Properties of the Distribution of Primes. In: The American Mathematical Monthly, Vol. 71, No. 5 (May, 1964), S. 516-520. Während die Ulam-Spirale nach meinem Wissen bisher lediglich

für mathematisch-theoretische Belange eingesetzt wurde, hat die sich seitdem durchsetzende Digitalisierung von Daten und Signalen dazu geführt, dass nun die visuelle Datenanalyse mit Hilfe der rechteckigen Spirale (unter Anpassung der spezifischen Parameter) in vielen praktischen Bereichen eingesetzt werden kann. Abb. 1 zeigt beispielhaft die Musterbildung bei der Visualisierung eines regelmäßigen Signals nach dem von mir vorgeschlagenen Verfahren.

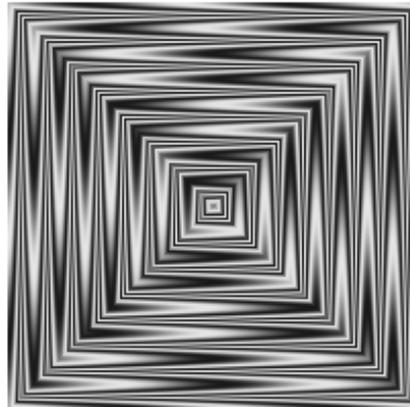


Abbildung 1: Die Musterbildung bei der Visualisierung eines gleichbleibenden Sinustons nach dem hier beschriebenen Verfahren (mit einer Palette an kontinuierlichen Farbwerten).

Ich habe die Spiraldarstellung als Arbeitswerkzeug für die Untersuchung elektronischer Klänge im Rahmen meiner Dissertation entwickelt und eingesetzt. Die Aufgabe, die ich mit diesem Verfahren gelöst habe, war die Darstellung eines Übergangs vom rauschhaften zum geregelten Signal in einer Klangaufnahme.

Die mathematische Konstruktion einer rechteckigen Spirale besteht darin, auf einer Fläche aus diskreten, gleich großen, quadratischen Elementen von einem Element aus schrittweise eine Spirale aufzubauen, und zwar so, dass jedes weitere Element an die bereits platzierten Elemente anschließt. Nummeriert man die ersten neun Elemente einer sich gegen den Uhrzeigersinn entwickelnden Spirale, ergibt sich die folgende Anordnung, die die Vorgehensweise illustriert:

5	4	3
6	1	2
7	8	9

Einzelne Elemente dieser Spirale werden für die Visualisierung einzelner Datenpunkte (etwa: Samplewerte einer Klangaufnahme oder einzelne Werte einer Zeitreihe aus einer Messung) bzw. Datenbereiche des analysierten Materials verwendet. Die Skala der Datenwerte wird dabei je nach Anwendungsbereich entweder auf nur zwei kontrastierende Farbwerte reduziert, oder aber in ihrer ganzen Bandbreite durch eine Skala an Farbwerten abgebildet. Abb. 2 zeigt die Visualisierung der binären Datenwerte eines Pseudozufallsgenerators (4-Bit-LFSR). Das erste Element der Spirale ist

durch einen Kreis markiert. Ein etwas größerer Ausschnitt aus der durch die Ulam-Spirale gebildeten Fläche erlaubt es bereits, sich wiederholende Strukturen in der Ausgabe dieses einfachen Pseudozufallsgenerators mit einer Periodenlänge von 15 Werten zu erkennen (siehe Abb. 3).

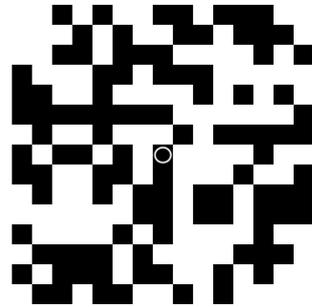


Abbildung 2: Visualisierung der Datenwerte eines 4-Bit-LFSR-Generators (vergrößert).

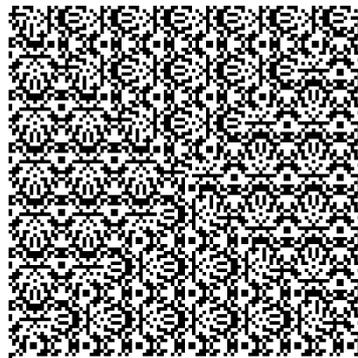


Abbildung 3: Datenwerte eines 4-Bit-LFSR-Generators (Überblick mit erkennbaren Strukturen).

Die Visualisierung erfolgt mit Hilfe eines Computerprogramms, das Datenwerte aus einer Quelle ausliest, diese bei Bedarf so mathematisch umformt, dass sie auf zwei Zustände reduziert werden, und anschließend auf einzelne Elemente der Ulam-Spirale als Farbwerte abbildet. Eine von mir erstellte Beispielrealisation in der Programmiersprache "Processing" findet sich in Anhang 2.

Beispiel: Digital abgespeicherte Klänge können als ein Datensatz mit einzelnen Werten im Bereich zwischen -1 und 1 betrachtet werden. Der graduellen Veränderung der Lautstärkewerte entspricht hier die Variation der Grauwerte auf einer Skala zwischen schwarz und weiß. Diese Darstellung ist kompakter, als eine bitweise Visualisierung und erlaubt bereits auf einer Bildfläche von 210 x 210 Pixeln die Repräsentation von einer Sekunde Klang (bei einer Abtastrate von 44100 Hz). Bei einer solchen Darstellung mit kontinuierlichen Farbwerten kann für die Umwandlung der Datenwerte in Farbwerte - neben anderen denkbaren Umformungen - die folgende Formel verwendet werden. Es hat sich dabei für natürliche Klänge oft als vorteilhaft erwiesen, den absoluten Betrag des Datenwerts zu verwenden:

$$\text{Farbwert} = (1 - \text{abs}(\text{Datenwert})) * \text{MaximalerFarbwert}$$

Für algorithmisch erzeugte Daten, wie etwa für die Ausgabe eines LFSR-Pseudozufallszahlengenerators, der eine Reihe aus 0 und 1-Werten bildet, ist es dagegen aus Gründen der visuellen Erkennbarkeit vorteilhafter, die einzelnen Datenwerte wie folgt umzuwandeln:

$$\text{Farbwert} = \text{Datenwert} * \text{MaximalerFarbwert}$$

Als eine Alternative zu dem von mir vorgeschlagenen Verfahren kann für Klänge und Daten die direkte Höranalyse (Audifikation bzw. Sonifikation) eingesetzt werden. Der sich erst in der Zeit entfaltende Charakter einer solchen Darstellung steht jedoch der Bildung von größeren Überblicken und Querbezügen entgegen. Die bisher allgemein eingesetzten Verfahren der Klangvisualisierung (Wellenform und Spektrogramm) sind primär auf andere Anwendungsbereiche hin konzipiert, als die Erkennung von strukturellen Regelmäßigkeiten bzw. Abweichungen auf Mikroebene. Die Unzulänglichkeiten dieser Verfahren im Hinblick auf die Aufdeckung von Regelmäßigkeit waren der Grund für die Entwicklung meines Verfahrens.

Wie oben erwähnt, stellt die Anwendung der Ulam-Spirale für die Signalanalyse nach meinem Kenntnisstand ein Novum dar. Die Anwendung der Ulam-Spirale für die Darstellung mathematischer Gesetzmäßigkeiten ist dagegen bereits erprobt worden. Ein aktuelles Beispiel ist die spiralenförmige Abbildung der Werte der Teilerzahlfunktion:

http://commons.wikimedia.org/wiki/File:Ulam_Spiral_Divisors_100000.png

*

Das hier beschriebene Verfahren kann unter anderem in folgenden Anwendungsbereichen eingesetzt werden:

- Klanganalyse im wissenschaftlichen und künstlerischen Bereich (Audiosoftware)
- Kontrolle des Gleichlaufs von Signalerzeugern, etwa in der Medizin (Hardware- und Softwarelösungen für klinische Anwendungen)
- Herstellung eines Überblicks über unbekannte Datensätze und Signale (ingenieurtechnische Anwendungen, Vorbereitung einer statistischen Analyse, eventuell sind auch Anwendungen im Bereich der Kryptoanalyse (Dechiffrierung) möglich)

Anhang 1. Beispielvisualisierung eines komplexen Datensatzes

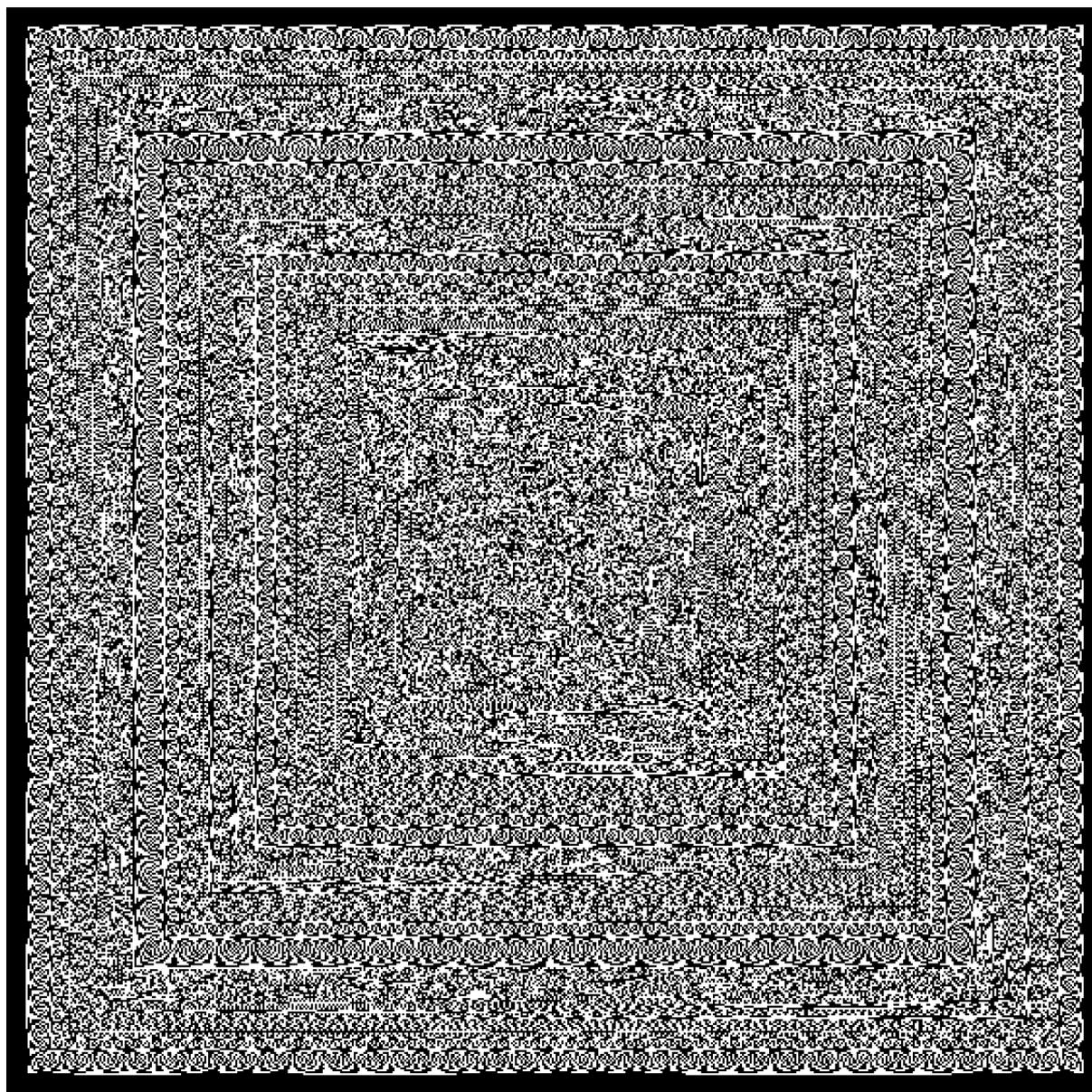


Abbildung 4: Rechenvorgänge des PASCAL-Computers.

Abb. 4 zeigt die Visualisierung der Rechenvorgänge des historischen Rechners PASCAL, mit dem am Anfang der 1960-er Jahre Primzahlensuche betrieben wurde. (Die von mir zum Zwecke der Visualisierung durchgeführte Rekonstruktion der Rechenvorgänge stützte sich auf die Beschreibungen in: Miyazaki, Shintaro: *Algorithmisiert. Eine Medienarchäologie digitaler Signale und (un)erhörter Zeiteffekte*. Kulturverlag Kadmos, 2013 sowie Nijenhuis, W. 1962: *Hörbares Rechnen der Pascal*. In: *Philips technische Rundschau*, 24. Jahrgang, 1962/1963, Nr. 4/5, S. 169-176).

Am Anfang der Suche, das heißt im Zentrum des Bildes, sind keine Muster erkennbar. Wenig später beginnen sich auffällige Bänder von wellenartigen Mustern zu

bilden, die jedoch zeitweise von Rauschteilen unterbrochen werden. Am leichtesten erkennbar werden diese wellenartigen Muster kurz von dem Auffinden der zweiten Primzahl, also am äußeren Rand des Bildes. Bemerkenswert ist, dass während die Darstellung des Daten in Form eines Sonogramms (hier nicht beigefügt) vollkommen regelmäßige Kurven erzeugt, sind die wellenartigen Muster in der von mir realisierten Visualisierung alle individuell und unterschiedlich, was einen Zuwachs an visueller Information bedeutet.

Anhang 2. Beispielrealisation des Visualisierungsverfahrens in der Programmiersprache "Processing"

```
int pointnr;
int maxwinglen;
int winglen;
int points;
int steps;
int x;
int y;
float strokecol;

BufferedReader reader;
String line;

void setup() {
  size(200, 200); // Festlegung der Größe des Bilddatei
  background(0);
  noLoop();
  reader = createReader("werte.txt"); // Werte werden aus dieser Datei gelesen
}

void draw() {
  pointnr = 1;
  maxwinglen = 200; // Größe des Bildes
  points = maxwinglen^2;
  x = width / 2;
  y = height / 2;
  // ersten Punkt setzen
  SetPoint();
  // Hauptschleife:
  // obere und linke Flanke haben die Länge winglen, untere und rechte: winglen-1
  while (winglen < maxwinglen+1) {
    // untere Flanke
    for (int steps = 1; steps < winglen-1; steps = steps+1) {
      x=x+1;
```

```

        SetPoint();
    }
    // rechte Flanke
    for (int steps = 1; steps < winglen-1; steps = steps+1) {
        y=y-1;
        SetPoint();
    }
    // obere Flanke
    for (int steps = 1; steps < winglen; steps = steps+1) {
        x=x-1;
        SetPoint();
    }
    // linke Flanke
    for (int steps = 1; steps < winglen; steps = steps+1) {
        y=y+1;
        // Punkt setzen
        SetPoint();
    }
    winglen = winglen+2;
}
// das vollständige generierte Bild speichern
save("bild.png");
}

void SetPoint() {
    // Lese-Routine, übernommen aus den Beispielen auf der Webseite der
    // Programmiersprache
    try {
        line = reader.readLine();
    }
    catch (IOException e) {
        e.printStackTrace();
        line = null;
    }
    if (line == null) {
        // Stop reading because of an error or file is empty
        noLoop();
    }
    else {

        // Umformung des Datenwerts in einen Farbwert
        strokecol = (1-abs(float(line)))*255;

    }

    // Festlegung der Farbe

```

```
stroke(strokecol);  
// Setzen des Punkts  
point(x, y);  
}
```