Fraktale Bildcodierung

Jens Ochsenmeier

Die Menge aller echten Bilder

Ein **echtes Bild** ist nicht direkt mathematisch greifbar — es ist etwas, dass man irgendwo sehen oder sich zumindest zu sehen vorstellen könnte. Wir nennen

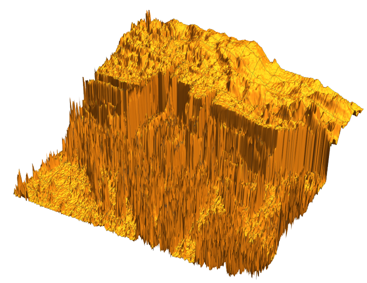
- $\mathcal{R}$ die Menge der echten Bilder und
- $\mathcal{I} \in \mathcal{R}$ ein echtes Bild.

Ein solches $\mathcal{I}$ muss folgende Eigenschaften erfüllen:

1. Es besitzt einen **Träger** $\square = [a, b] \times [c, d] \subset \mathbb{R}^2$ und die **Maße** $(b - a)$ und $(d - c)$.
2. Es besitzt gewisse **chromatische Werte** $c_i : \square \to \mathbb{R}$, welche jedem Punkt des Bildes einen reellwertigen Wert zuordnet.
3. Einem Bild $\mathcal{I} \in \mathcal{R}$ ist *keine* Auflösung zugeordnet — verschiedene Auflösungen beschreiben dasselbe Bild. Rasterung ist nicht Teil dieses Vortrags.
4. $\mathcal{R}$ ist unter Zuschneiden von Bildern abgeschlossen.
5. $\mathcal{R}$ ist unter invertierbaren affinen Transformationen abgeschlossen.

Ein "echtes" Bild. Offensichtlich erfüllt es die erste Eigenschaft, da es gedruckt ist und somit einen endlichen Träger (dieses Blatt) besitzt.

In diesem Vortrag werden wir uns auf Bilder mit Graustufen beschränken. Die erklärten Prinzipien funktionieren uneingeschränkt für farbige Bilder, sind aber etwas komplexer zu notieren.



Plot der chromatischen Funktion c für das gegebene Bild. Man erkennt, dass die Funktionswerte in der oberen Hälfte des Bildes größer sind; dort ist das Bild heller.

Collage-Theorem

- Komplexe Bilder können durch einfache IFS beschrieben werden
- **Problem**: Wie bekommt man zu einem gegebenen Bild ein passendes IFS?

Das **Collage-Theorem** erlaubt die Konstruktion von IFS, deren Attraktoren einem gegebenen Bild möglichst genau entsprechen:

$$h\left(L, \bigcup_{i=0}^{n} \Phi_i(L)\right) \leq \varepsilon \Rightarrow h(L, A) \leq \frac{\varepsilon}{1 - s}$$

Um ein gegebenes Bild $L \in \mathcal{H}(X)$ durch den Attraktor eines IFS anzunähern, muss die Vereinigung ("Collage") der Kontraktionen des IFS, angewandt auf das gegebene Bild L, dem gegebenen Bild ähneln (im Sinne der Hausdorff-Metrik).

Zutaten für das Collage-Theorem:

- vollständiger metrischer Raum (X, d)
- $L \in \mathcal{H}(X)$
- $\varepsilon \geq 0$
- IFS $\{\Phi_1, \ldots, \Phi_n\}$
  - Kontraktionskoeffizient $0 \leq s < 1$
  - Attraktor A

Stetige Abhängigkeit von Parametern

Ein weiterer wichtiger Satz zur Komprimierung von Bildern ist der Satz über die stetige Abhängigkeit des Attraktors eines IFS von den Parametern des IFS.

Die stetige Abhängigkeit ist deswegen so wichtig, weil sie uns erlauben wird, durch kleine Änderungen der Parameter kleine Änderungen am Attraktor eines IFS provozieren zu können. Eine praktische Folgerung davon ist, dass zwischen verschiedenen Attraktoren ein stetiger Übergang durch Parameteränderungen konstruiert werden kann — diese Eigenschaft ist besonders praktisch zur Konstruktion und Komprimierung von animierten Bildern und Filmmaterial.

Fixpunkt

Für jedes $x \in X$ sei $\Phi(\cdot, x)$ stetig. Dann ist der Fixpunkt $\tilde{x}(p)$ von Φ stetig abhängig von p.

Diese Erkenntnis lässt sich eingeschränkt auf Attraktoren übertragen.

Zutaten stetige Abhängigkeit:

- (P, d_P) metrischer Raum
- (X, d) vollständiger metrischer Raum
- $\Phi : P \times X \to X$ Familie von Kontraktionen auf X
- $0 \leq s < 1$ Kontraktionskoeffizient von $\Phi(p, \cdot)$

Attraktor

Jedes Φ_i hänge von $p \in (P, d_P)$ im Sinne der Einschränkung $d(\Phi_{i,p}(x), \Phi_{i,q}(x)) \leq k \cdot d_P(p, q)$ für beliebiges x ab.

Dann ist der Attraktor $A_p \in \mathcal{H}(X)$ stetig von p mittels der Hausdorff-Metrik $h(d)$ abhängig.

Fraktale Bildkomprimierung

- **Idee**: Durch Angabe eines IFS statt Pixelwerten kann massiv Platz gespart werden
- **Problem 1**: Wie für gegebenes Bild passendes IFS bestimmen?
- **Problem 2**: Bilder in der Regel nicht wirklich selbstähnlich

Beispiel zur Platzersparnis:

- Farn kann durch 4 Transformationen erzeugt werden

  → es müssen nur 18 Zahlen gespeichert werden ⇒ ~ 18 Byte
- Konventionelle Speicherung: 1 Byte für 8 Pixel ⇒ 500 mal mehr (10kB)!

  (200*400 Pixel, je höher desto mehr Bytes werden gebraucht)

Partitionierte IFS

Affine Transformationen eines IFS werden aufgerüstet:

1. Eine neue Vorschrift adaptiert pro Transformation die Helligkeit
2. Eine Maske legt pro Transformation fest, welcher Teil des Urbilds abgebildet wird

$$\Phi_i : \square \supset D_i \to w_i(D_i) =: R_i \subset \square, \quad w_i(x, y, z) = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & g_i \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ h_i \end{pmatrix}$$

Trick ist hier also, dass Teilbereiche des Bilds separat transformiert werden.

Neu sind in der formalen Darstellung die Parameter f_i und h_i, welche für den Kontrast bzw. die Helligkeit zuständig sind.

Zusätzlich müssen folgende Eigenschaften gefordert werden, damit das gewünschte Bild $\mathcal{I} \in \mathcal{H}(X)$ Attraktor des PIFS werden kann:

1. $\bigcup_{i=1}^{n} R_i = \mathcal{I}$
2. $i \neq j \Rightarrow R_i \cap R_j = \emptyset$

PIFS bestimmen

Man kann die Parameter eines PIFS beispielsweise folgendermaßen bestimmen:

1. Unterteile das gegebene Bild in nicht-überlappende Quadrate (8*8 Pixel) $R := \{R_1, \ldots, R_n\}$.
2. Unterteile das gegebene Bild in überlappende Quadrate (16*16 Pixel) $D := \{D_1, \ldots, D_m\}$.
3. Durchsuche für jede der 8 möglichen Transformationen eines R_i alle D nach einem D_j, sodass sich R_i und D_j unter der gegebenen Transformation so sehr ähneln wie möglich.
4. Passe für jede Transformation Belichtung und Kontrast an.

Resultat dieses Algorithmus ist ein PIFS, bestehend aus Transformationen für jedes $R_i \in R$.

Quellen:

- Barnsley — **Fractals Everywhere** (1993)
- Barnsley, Hurd — **Fractal Image Compression** (1993)
- Fisher — **Fractal Image Compression** (1992)