

Mathematik der Ästhetik

Prof. Dr. Batu Güneysu

Technische Universität Chemnitz

Kulturgut Mathematik und Kunst
Schlossbergmuseum Chemnitz, 25.10.2023

- Teilweise motiviert von einer Arbeit von Rigau / Feixas / Sbert (2008)
- Programmierung von Dr. Sebastian Boldt

Die erste Versuch den Begriff der **Ästhetik** für Bilder zu formalisieren stammt vom amerikanischen Mathematiker **George David Birkhoff** aus dem Jahre 1933:



Das **ästhetische Maß** M eines Bildes ist als eine Funktion $f(O, C)$ der **Ordnung** O des Bildes und der **Komplexität** C des Bildes gegeben:

$$M = f(O, C).$$

Der Begriff **Ordnung** ist hier als **Maß der Geordnetheit** oder als **Maß dafür, wie gut sich das Bild in Komponenten zerlegen lässt** zu verstehen. Diese Definition macht für beliebige Objekte Sinn.

Wie können wir diese Daten nun konkret bestimmen?

Um die funktionale Abhängigkeit dieser Größen vom gegebenen Bild B zu verdeutlichen, schreiben wir auch

$$M = M(B), \quad O = O(B), \quad C = C(B).$$

Zunächst zur funktionellen Abhängigkeit $f(O, C)$: Wir schlagen

$$f(O, C) := O \cdot C$$

vor, also

$$M = O \cdot C.$$

Diese Definition gewährleistet, dass bei zwei Bildern der gleichen Komplexität das Bild mit größerer Ordnung als ästhetischer bewertet wird. Analog gewinnt bei gleicher Ordnung die größere Komplexität: Das Bild ist dann sozusagen weniger langweilig.

Eine sinnvolle Wahl der Funktion $f(O, C)$ hängt vom Kontext der Betrachtung ab. Wir stellen uns einen Museums- oder Galeriebesuch vor, so dass ein gewisser Unterhaltungseffekt bei der Betrachtung eine Rolle spielen sollte. Birkhoff selbst schlägt O/C statt $O \cdot C$ vor...

Um mathematische Definitionen der Größen $O(B)$ und $C(B)$ angeben zu können, werden wir nun einen kleinen Ausflug in die Stochastik bzw. Informationstheorie wagen:

- Eine **Zufallsvariable** X mit Werten in der endlichen Menge $\{x_1, \dots, x_m\}$ ordnet per Definition jedem Zufallsexperiment genau einen der Werte x_1, \dots, x_m zu.
- Die **Wahrscheinlichkeit** dafür, dass X den Wert x_i annimmt wird mit $P(X = x_i)$ bezeichnet. Dies ist eine Zahl, die jeden Wert zwischen 0 und 1 annehmen kann.
- Sind all diese Wahrscheinlichkeiten gleich, also $P(X = x_i) = P(X = x_j)$ für alle $i, j = 1, \dots, m$, so nennt man X **gleichmäßig verteilt**.
- In der Praxis bekommt man diese Werte durch abzählen....

Beispiel 1: Ist X das Würfeln mit einem normalen Würfel, so nimmt X Werte in $\{1, 2, 3, 4, 5, 6\}$ an, und es gilt $P(X = i) = 1/6$ für alle $i = 1, \dots, 6$, d.h. X ist gleichmäßig verteilt.

Beispiel 2: Sei X das Würfeln mit einem Gaunerwürfel, welcher drei Flächen mit der Augenzahl 6, sowie jeweils eine Fläche mit der Augenzahl 1, 2, 3 hat. Dann nimmt X Werte in $\{1, 2, 3, 6\}$ an, und es gilt

$$P(X = 6) = \frac{\text{Anzahl der Flächen mit Augenzahl 6}}{\text{Anzahl aller Flächen}} = \frac{3}{6} = \frac{1}{2},$$

und für $i = 1, 2, 3$,

$$P(X = i) = \frac{\text{Anzahl der Flächen mit Augenzahl } i}{\text{Anzahl aller Flächen}} = \frac{1}{6}.$$

Ist X eine Zufallsvariable mit Werten in $\{x_1, \dots, x_m\}$, so definiert man die **Entropie** $H(X)$ von X als

$$H(X) := - \sum_{i=1}^m P(X = x_i) \log P(X = x_i).$$

Es gilt $H(X) \geq 0$ und

- $H(X)$ wird maximal genau dann, wenn X gleichmäßig verteilt ist.
- $H(X)$ wird minimal genau dann wenn, X deterministisch ist (also nicht vom Zufall abhängt).

In diesem Sinne ist die Entropie ein Maß für die **Unordnung von X** .

Ist Y eine weitere Zufallsvariable die nur endlich viele Werte $\{y_1, \dots, y_n\}$ annimmt, so heißt die Zahl

$$H(X|Y) := - \sum_{i=1}^m \sum_{j=1}^n P(X = x_i, Y = y_j) \log P(X = x_i | Y = y_j)$$

die **durch Y bedingte Entropie von X** . Es gilt stets

$0 \leq H(X|Y) \leq H(X)$ und

- $H(X|Y)$ wird maximal, wenn X und Y unabhängig sind.
- $H(X|Y)$ wird minimal, wenn X vollständig durch Y bestimmt ist.

In diesem Sinne misst $H(X|Y)$ die Unsicherheit von X , falls Y kennen.

Die Zahl

$$I(X, Y) := H(X) - H(X|Y) \geq 0$$

heißt die **Transinformation von X und Y** .

Es gilt $I(Y, X) = I(X, Y)$ und es gilt $I(X, Y) = 0$ genau dann, wenn X und Y unabhängig sind.

In diesem Sinne misst $I(X, Y)$, wieviel wir über X sagen können, wenn wir Y kennen (und umgekehrt).

Wir stellen Bilder als Matrizen von Pixeln dar. Jedes Pixel kann einen der Farbtöne $\{0, 1, \dots, m\}$ annehmen.

Um eine Definition für die Ordnung eines Bildes B zu erhalten, haben wir einen Algorithmus entwickelt, der B sukzessive immer weiter aufteilt, mit dem Ziel B in seine 'Farbzusammenhangskomponenten' aufzuteilen. Je aufwendiger dieses Verfahren ist, desto unordentlicher soll das Bild per Definitionem sein.

Vorbereitung: Für ein Bild B definiere eine Zufallsvariable X_B mit

$$P(X_B = f_i) = \frac{\text{Anzahl der Pixel in } B \text{ mit Farbton } f_i}{\text{Anzahl aller Pixel in } B}, \quad i = 0, \dots, m.$$

Ist \mathcal{L} eine Zerlegung von B in n disjunkte Teilbilder B_1, \dots, B_n von B , so sei die Zufallsvariable $Y_{\mathcal{L}}$ definiert durch

$$P(Y_{\mathcal{L}} = B_j) = \frac{\text{Anzahl aller Pixel in } B_j}{\text{Anzahl aller Pixel in } B}, \quad j = 1, \dots, n.$$

Der Algorithmus läuft nun wie folgt ab:

- Im ersten Schritt erzeugen wir zwei Teilbilder B_1 und B_2 von B , indem wir eine Zerlegung von B in zwei disjunkte Teilbilder B_1, B_2 von B so wählen, dass für die induzierte Zerlegung \mathcal{L} von B die Information $I(X_B, Y_{\mathcal{L}})$ maximal wird.
- Im zweiten Schritt wiederholen wir die Prozedur mit B_1 statt B und mit B_2 statt B und bekommen zwei Teilbilder von B_1 , sowie zwei Teilbilder von B_2 , also vier Teilbilder von B .
- Im n -ten Schritt haben wir somit eine Zerlegung \mathcal{L}_n von B in 2^n Teilbilder von B produziert.
- Wir brechen ab, wenn der Informationszuwachs vom n -ten zum $(n + 1)$ -ten Schritt unter einem fixierten Wert liegt:

$$I(X_B, Y_{\mathcal{L}_{n+1}}) / I(X_B, Y_{\mathcal{L}_n}) < 1 + \epsilon \quad \text{mit } \epsilon > 0 \text{ sehr klein.}$$

Sei n_B genau diese Anzahl von Schritten.

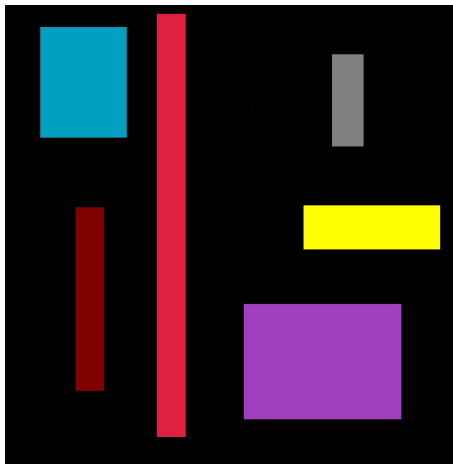
Die **Ordnung eines Bildes** B definieren wir als die Zahl

$$O(B) := I(X_B, Y_{\mathcal{L}_{n_B}}) / H(X_B).$$

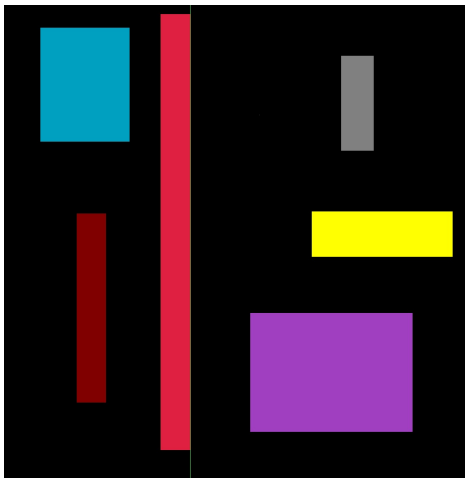
Diese Definition ist durch folgende beiden Tatsachen begründet:

- Es gilt $0 \leq O(B) \leq 1$, die Ordnung ist also normiert.
- $O(B_1) > O(B_2)$ bedeutet gerade, dass sich das Bild B_1 mit weniger Aufwand als das Bild B_2 in seine Farbzusammenhangskomponenten zerlegen lässt und in diesem Sinne eine höhere Ordnung hat.

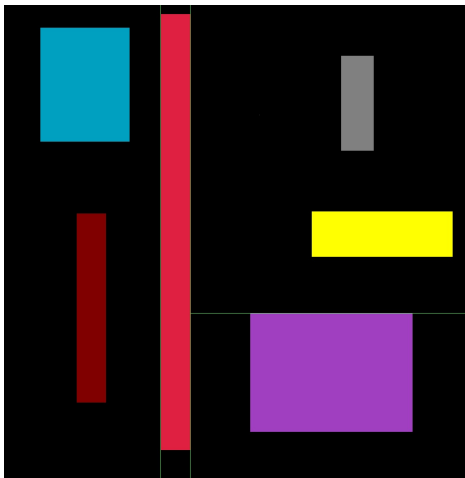
Betrachten wir das Bild **Rechteck**:



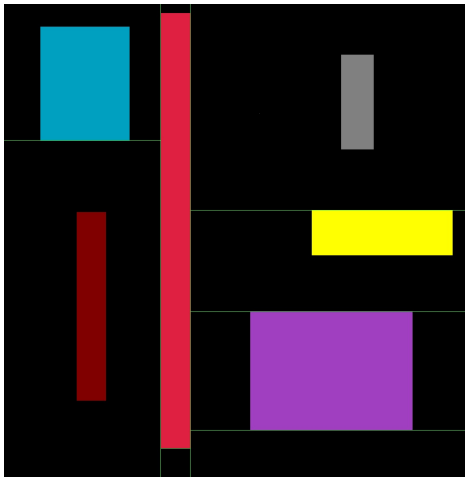
Der erste Schritt des Algorithmus liefert folgende Unterteilung:



Der zweite Schritt des Algorithmus liefert folgende Unterteilung:

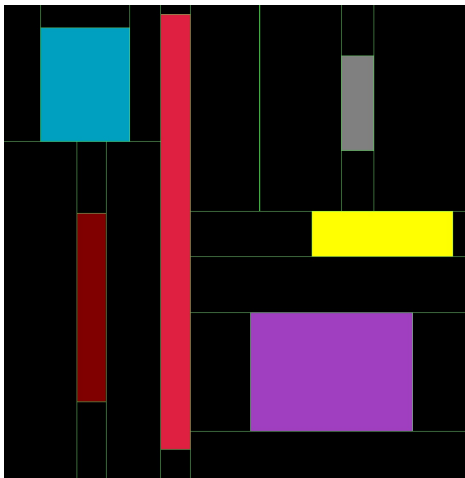


Der dritte Schritt des Algorithmus liefert folgende Unterteilung:



Und so weiter.....

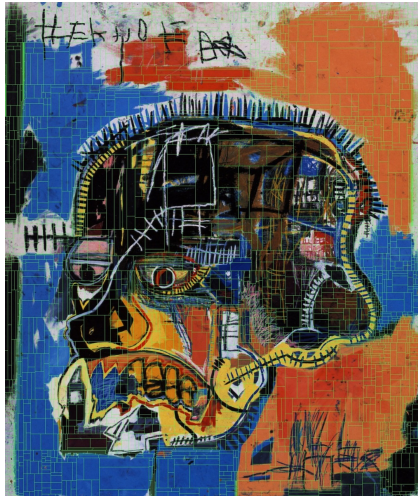
Unser Algorithmus benötigt insgesamt 7 Schritte und liefert eine Ordnung von $O(\mathbf{Rechteck}) = 0,9999$:



Betrachten wir das Bild **Skull** von J.-M. Basquiat (1981):



Unser Algorithmus benötigt 12 Schritte und liefert eine Ordnung von $O(\mathbf{Skull}) = 0,4560$:



Kommen wir nun zur Komplexität. Der folgende abstrakte Komplexitätsbegriff stammt vom russischen Mathematiker Andrey Kolmogorov aus dem Jahre 1963:



Die **Kolmogorov-Komplexität** des Worts $w = w_1, \dots, w_s$ mit Buchstaben w_i aus dem Alphabet $\{0, 1, 2, \dots, m\}$ ist definiert als die Länge des kürzesten Programms, das w beschreiben kann.

Beispiel zur Illustration der Definition: Das Wort

$w = 0, 0, 3, 4, 0$

mit Buchstaben aus $\{0, 1, 2, 3, 4\}$ kann mit dem Programm

Schreibe '0, 0, 3, 4, 0'

beschrieben werden, oder mit dem kürzeren Programm

Schreibe '0, 0, 3, 4' dann 24 mal ', 0'

beschrieben werden. Aber leider kann man Folgendes beweisen:

Es gibt keinen Algorithmus, der die Kolmogorov Komplexität ausrechnen kann.

Wir suchen stattdessen einen (verlustfreien)
Kompressionsalgorithmus, der die Kolomogorov-Komplexität
approximiert: LZW-Algorithmus, 1978/1983:



(a)



(b)

Abbildung: (a) Jacob Ziv, Abraham Lempel (b) Terry Welch

Der Algorithmus bekommt ein Wort $w = (w_1, \dots, w_s)$ mit
Buchstaben aus dem Alphabet $\{0, \dots, m\}$ als Eingabe und erzeugt
ein kürzeres Wort \tilde{w} als Ausgabe:

- Setze das Wörterbuch zu $\{(0) \leftrightarrow 0, (1) \leftrightarrow 1, \dots, (m) \leftrightarrow m\}$; hier steht jeweils links das Wort und rechts das zugehörige Codewort;
- setze das Hilfswort zu w ;
- setze die virtuelle Alphabet zu $\{0, 1, \dots, m\}$;
- setze das Ausgabewort als $()$;
- solange das Hilfswort nicht leer ist:
 - suche im Hilfswort von links nach rechts das längste Teilwort (w_1, \dots, w_q) , das im Wörterbuch steht;
 - erweitere das Ausgabewort nach rechts mit dem (w_1, \dots, w_q) zugeordneten Code;
 - erweitere das Wörterbuch mit dem Wort (w_1, \dots, w_{q+1}) mit zugehörigem Code gegeben als das kleinste $n \geq m$, das noch nicht im virtuellen Alphabet ist;
 - erweitere das virtuelle Alphabet mit n ;
 - kürze im Hilfswort (w_1, \dots, w_q) von links;

Das komprimierte Wort \tilde{w} ist dann gerade das finale Ausgabewort.

Beispiel:

Alphabet: $\{0, \dots, 7\}$

$w = (1, 2, 4, 1, 2, 7, 0, 1, 2, 7, 1, 2)$

Wörterbuch: $\{(0) \leftrightarrow 0, (1) \leftrightarrow 1, \dots, (7) \leftrightarrow 7\}$

Hilfswort: $(1, 2, 4, 1, 2, 7, 0, 1, 2, 7, 1, 2)$

Virtuelles Alphabet: $\{0, \dots, 7\}$

Hilfswort	längstes Teilwort im Wörterbuch	Erweiterung des Ausgabeworts um...	Erweiterung des Wörterbuchs um...
$(1, 2, 4, 1, 2, 7, 0, 1, 2, 7, 1, 2)$	(1)	1	$(1, 2) \leftrightarrow 8$
$(2, 4, 1, 2, 7, 0, 1, 2, 7, 1, 2)$	(2)	2	$(2, 4) \leftrightarrow 9$
$(4, 1, 2, 7, 0, 1, 2, 7, 1, 2)$	(4)	4	$(4, 1) \leftrightarrow 10$
$(1, 2, 7, 0, 1, 2, 7, 1, 2)$	$(1, 2)$	8	$(1, 2, 7) \leftrightarrow 11$
$(0, 1, 2, 7, 1, 2)$	(0)	0	$(0, 1) \leftrightarrow 12$
$(1, 2, 7, 1, 2)$	$(1, 2, 7)$	11	$(1, 2, 7, 1) \leftrightarrow 13$
$(1, 2)$	$(1, 2)$	8	-

Also wird das Wort $w = (1, 2, 4, 1, 2, 7, 0, 1, 2, 7, 1, 2)$ komprimiert zum kürzeren Wort $\tilde{w} = (1, 2, 4, 8, 0, 11, 8)$.

Es macht keinerlei Probleme den LZW-Algorithmus zeilenweise auf Matrizen statt auf Wörter (= Vektoren) anzuwenden. Dies führt zu:

Die **Komplexität eines Bildes** B definieren wir als die Kompressionsrate

$$C(B) := \frac{\text{Größe von } \tilde{B} \text{ in Bytes}}{\text{Größe von } B \text{ in Bytes}}.$$

- Es gilt wieder die Normierung $0 \leq C(B) \leq 1$.
- $C(B_1) > C(B_2)$ bedeutet gerade, dass sich B_1 schlechter als B_2 komprimieren lässt und in diesem Sinne komplexer ist.

Für das Bild **Skull** erhalten wir eine Komplexität von $C(\mathbf{Skull}) = 0,1601$ und somit das ästhetische Maß zu

$$\begin{aligned}M(\mathbf{Skull}) &= O(\mathbf{Skull}) \cdot C(\mathbf{Skull}) \\ &= 0,4560 \cdot 0,1601 = 0,0730.\end{aligned}$$

Für das Bild **Rechteck** erhalten wir eine Komplexität von $C(\mathbf{Rechteck}) = 0,0602$ und somit das ästhetische Maß zu

$$\begin{aligned}M(\mathbf{Rechteck}) &= O(\mathbf{Rechteck}) \cdot C(\mathbf{Rechteck}) \\ &= 0,9999 \cdot 0,0602 = 0,0601.\end{aligned}$$

- Wir haben das ästhetische Maß $M(B)$ eines Bildes B definiert als das Produkt $O(B) \cdot C(B)$, mit $O(B)$ der Ordnung und $C(B)$ der Komplexität.
- Die Ordnung ermitteln wir, indem wir algorithmisch das Bild in seine Farbzusammenhangskomponenten zerlegen. Je aufwendiger dies ist, desto weniger Ordnung hat das Bild.
- Die Komplexität definieren wir als eine (verlustfreie) Komprimierbarkeit.
- Der Mangel an Komplexität kann durch eine erhöhte Ordnung ausgeglichen werden (und umgekehrt).
- Ausblick...

Vielen Dank für Ihre Aufmerksamkeit!