Interaktive Computergrafik

Michael Gabler

18. Juli 2019

Inhaltsverzeichnis

1 Grundlagen

Computergrafik beschreibt das Erstellen von 2D-Bildern aufgrund von 3D-Daten.
Anwendungsgebiete

- Human-Computer-Interaction
- CAD & (wissenschaftliche) Visualisierung
- Filme
- Computer Spiele

3D-Repräsentation Wie können Objekte als 3D-Modell abgebildet werden?

- Implizite Parameter (z.B. als Funktion)
- Oberfläche annähernd beschrieben durch Dreiecke oder Polygone (manuell, Laser Scanner, Fotos von allen Seiten)
- Volume Solids (z.B. durch Sensoren wie MRT oder CT)

Animation z.B. über Referenzpunkte, die mit echter Welt gemappt werden
Rendering Abbilden von 3D-Daten auf 2D-Repräsentation z.B. durch Raytracing oder Rasterization
Immersion Maß in wie weit eine virtuelle Darbietung äußere, reale Wahrnehmungen ausgrenzt und diese durch virtuelle ersetzt.
Präsenz/Presence In wie weit fühlt sich ein Subjekt in einer Umgebung angekommen/eingebungen auch wenn es sich in einer anderen befindet.
Digitalisierung analoger Signale

Digitization

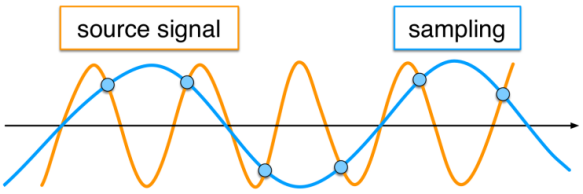


Figure: Sampling an analog wave.

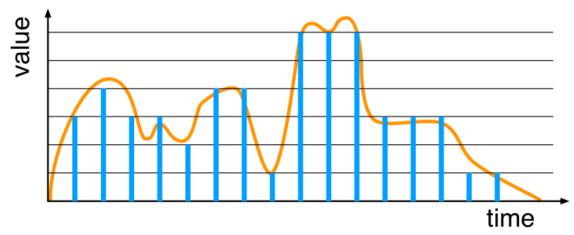


Figure: Quantization of values.

Conversion process of information into a digital (i.e., computer-readable) format, in which the information is organized into bits.

1. Discretization

- Reading (sampling) of an analog signal at regular intervals (frequency).
- Each reading (sample) may be considered to have infinite precision at this stage.

2. Quantization

- Approximating/rounding samples to a fixed set of numbers (such as integers).

Rastergrafik Grafik wird als Pixel beschrieben, die jeweils eine Farbe haben → Skalierung schwierig. Beispiel: JPG, PNG, GIF, TIFF, PBM

Vektorgrafik Inhalt der Grafik wird durch geometrische Formen beschrieben. Kann gerastert und beliebig skaliert werden. Beispiel: SVG, PS (Postscript), CGM, IGES, DWF/DXF

1.1 Menschliche Wahrnehmung von Licht

zwischen 380nm (violet/blau) und 780nm (rot)
Zapfen/Cones Farbliche Wahrnehmung (ca. 6 Millionen) je für einen Farbkanal zuständig (64 % rot, 32 % grün, 4 % blau)
Stäbchen/Rods Helligkeitswahrnehmung (ca. 120 Millionen)
Farbsysteme Repräsentation durch unterschiedliche Modelle, wie:

- biologisch orientiert: CIE XYZ
- Hardware-orientiert: RGB, CMY, CMYK (mit Schwarz, um Tinte zu sparen)
- Anwender-orientiert: HSV, HSB

Steven's Power Law physikalische Intensität (Helligkeit) ist nicht proportional zur wahrnehmbaren Helligkeit.

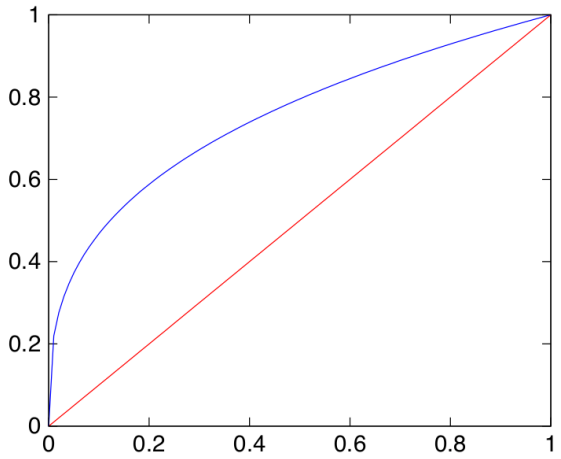


Figure: Example plot of $y = x$ (red) vs $y = x^{\frac{1}{3}}$ (blue)

Perceived Intensity by Eye

$$0.3 \leq a \leq 0.5$$

e.g.:

$$a = \frac{1}{3} \Rightarrow \psi(I) = kI^{\frac{1}{3}}$$

Observation

- Sensitivity is intensity-dependent
- **High** in dark areas
- **Lower** in bright areas

Question

- What consequences does this have on the digital representation of values?

Gamma Korrektur korrigiert physikalische Intensität, um kontinuierlichen wahrnehmbaren Intensitätszuwachs zu bekommen.

Before

After

Pros

- Finer intensity resolution in dark areas.
- Reduced perceived discontinuities.

Cons

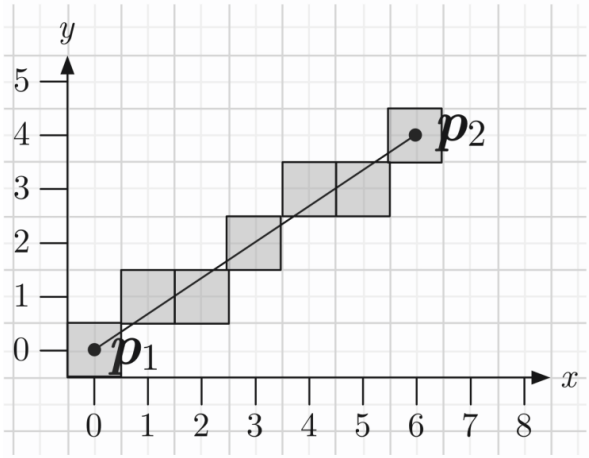
- Image appears overall to bright.
- Specifically relevant in dark areas.

$$n = \lfloor I^{\frac{1}{\gamma}} 2^N \rfloor$$

mit $I \in [0,1]$, $n \in [0, 2^N]$: Abbildung der physikalischen Intensität auf wahrnehmungskorrigierte mit N Bit Genauigkeit.

1.2 Rasterisierung von Vektorgrafiken

Digital Differential Analyzer (DDA) Rastern von Linien zwischen zwei beliebigen Punkten p_1 und p_2. Linie kann als Funktion $y = mx + b$ repräsentiert werden.



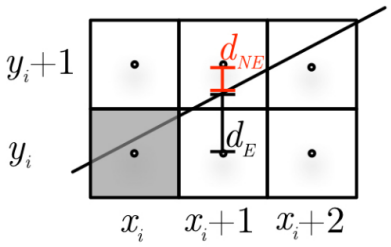
Follows

```
p1x = round(p1x); p1y =
  round(p1y);
p2x = round(p2x); p2y =
  round(p2y);
double m = (p2y-p1y)/(p2x-p1x);

pixelSet(p1x, p1y);
pixelSet(p2x, p2y);

for (int x = p1x+1; x < p2x; x++)
  {
    pixelSet(x, round(m * x + b));
}
```

Bresenham's Algorithmus Verfeinerung von DDA



Assume

- Pixel (x_i, y_i) is set.

Follows choice for next pixel as

- $(x_i + 1, y_i)$ or $(x_i + 1, y_i + 1)$.

Observation

$$y_{i+1} = \begin{cases} y_i & \text{if } d_E < d_{NE} \\ y_i + 1 & \text{otherwise.} \end{cases}$$

Given coords (x', y') follows

1. $d_E = y' - y_i$
2. $d_{NE} = y_i + 1 - y'$
3. $\Rightarrow \epsilon = d_E - d_{NE} = 2y' - 2y_i - 1$

Sign of decision variable ϵ determines pixel:

$$y_{i+1} = \begin{cases} y_i & \text{if } \epsilon \leq 0 \\ y_i + 1 & \text{otherwise.} \end{cases}$$

Aliasing Da Pixel entweder an oder aus sind (haben Farbe oder nicht), bildet sich eine Treppe beim Rastern von Linien. Kann beim Samplen auftreten
→ **Nyquist-Shannon Sampling Theorem** Sample Frequenz >= Doppelte Signal Frequenz
⇒ **Antialiasing** hat keine harten Übergänge sondern bildet "Farbverlaufäm Kantenrand (Pixel sind an, aus oder abgeschwächt farbig) durch Supersampling (mehr Punkte als Raster berechnen) und Durchschnittsbildung
Supersampling feinere Auflösung als Zielbild wählen und regelmäßig samplen (beste Ergebnisse), zufällige Punkte wählen im gesamten Bild, zufällige Punkte in definierten Räumen

1.3 Vektoren

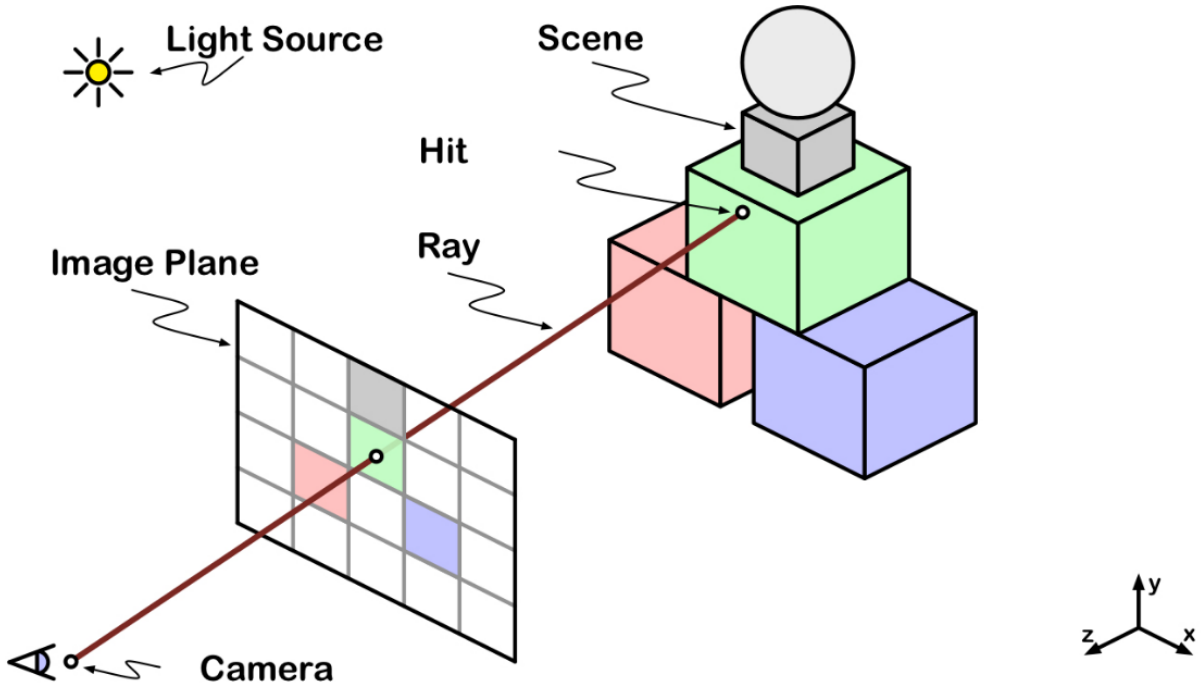
Skalarprodukt/Dot-Produkt $u \cdot v = (u_0, u_1, u_2)^T \cdot (v_0, v_1, v_2)^T = u_0v_0 + u_1v_1 + u_2v_2$
Vektorlänge $||v|| = \sqrt{v \cdot v}$
Kreuxprodukt $w = u \times v = \begin{bmatrix} u_1v_2 - u_2v_1 \\ u_2v_0 - u_0v_2 \\ u_0v_1 - u_1v_0 \end{bmatrix}$
w ist orthogonal zu u und v. Sie bilden ein rechthand-(Koordinaten)-System

2 Raytracing

Rendern von 3D-Objekten als 2D-Repräsentation. Orientiert an Physik, dadurch sehr realistische Darstellungen möglich. Strahlen werden von Kamera in die Szene gesendet. Schneiden diese ein 3D-Objekt, wird der Strahl von dort zu einer Lichtquelle verfolgt und der aussendende Pixel entsprechend gefärbt.

```
raytrace(scene, camera, image):
   # For all pixels in image
   for (x, y) in image:
      # 1. Generate ray through pixel
      ray = camera.generateRay(x, y)
```

```
# 2. Find closest intersection with scene
hit = scene.intersect(ray)
# 3. Calculate light intensity
color = shade(hit, scene)
# 4. Set pixel color
image.set(x, y, color)
```



Photon Lichtstrahl mit bestimmter Energie bzw. Farbe

$$E = h \cdot f$$

mit E: Energie, h: Plancksche Konstante, f: Frequenz

$$1 Lumen = 4 \cdot 10^{15} Photonen/sec$$

Absorption Photon verschwindet, wird von Gegenstand geschluckt
Reflektion Photon prallt an Oberfläche ab
Refraktion Photon geht durch eine Oberfläche hindurch (z.B. Glas)
Ray/Strahl Dargestellt als Startpunkt mit Richtungsvektor: $x(t) = x_0 + t\vec{d}$ **Licht-Material Interaktion** Mögliche Modelle zur realistischen Farbermittlung:

- Quantum Theorie (Emission, Absorption)
- Spezielle Relativität (aberration, blueshift, redshift, time dilatation)
- Wellenoptik (diffraction, dispersion, Interferenz)
- Geometische Optik (Reflektion, Refraktion)

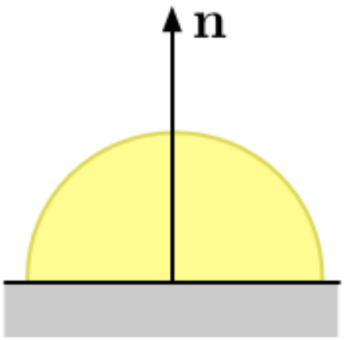
2.1 Shading

Verwende Objektfarbe bei Rayintersection. Naiver Ansatz, da physikalische Gesetze wie Reflektion oder Schatten und Objektstruktur ignoriert werden. Gleichmäßig gefärbte Objekte.

2.2 Phong Relektionsmodell

Modell zur realistischen Farbberechnung. Basiert auf geometischer Optik. Setzt sich zusammen aus der Farbe (Ambient) + Oberflächenbeschaffenheit und Schatten (Diffuse) + Reflektion (Specular).

Ambient



Rational

- Model for 'background lighting'
- Simulates a global contribution

Note

- No explicit light sources
- Indirect, constant illumination for the entire scene

Ambient reflection term

$$L_r = k_a L^a$$

With

L_r: Reflected (ambient) energy

k_a: Material's ambient reflectivity coefficient

L^a: (fictitious) ambient light energy

Diffuse

Specular