

Computergrafik

Übungsblatt 1

Aufgabe 1 Index Berechnung von Pixel

- (a) Gegeben sei eine Bildkoordinate $[x, y] \in [-1..1] \times [-1..1]$. Geben Sie eine Transformation an, welche die Bildkoordinaten auf Pixelkoordinaten $[x_p, y_p] \in [0, 1, \dots, 1919] \times [0, 1, \dots, 1079]$ eines FullHD-Monitors abbildet.

Das Bild ist in einem 1D Array abgelegt.

- (b) Berechnen Sie aus den Pixelkoordinaten $[x_p, y_p] \in [0, 1, \dots, 1919] \times [0, 1, \dots, 1079]$ nun einen eindeutigen Index in dieses 1D-Array! Dabei sollen benachbarte Pixel einer Zeile in dem 1D Array benachbart sein.
- (c) Invertieren Sie die Funktion aus (b), d.h. berechnen Sie anhand eines Index in das 1D Array eine zugehörige Pixelkoordinate $[x_p, y_p] \in [0, 1, \dots, 1919] \times [0, 1, \dots, 1079]$.

Aufgabe 2 Speicherverbrauch von Bildern

Gegeben sei ein 4K UHD Bild mit der Auflösung $3840p \times 2160p$. Jeder Pixel besitzt drei Farbkanäle mit je 8 Bit.

- (a) Wieviel Speicherplatz benötigt ein Bild?
- (b) Wieviel Speicher benötigt eine Filmsekunde, wenn diese 40 Bilder beinhaltet?
- (c) Wie groß muss das Kompressionsverhältnis (unkomprimierter zu komprimierter Filmsekunde) mindestens sein, damit eine 50 Mbit/s-Leitung ausreicht, um ein Video in Echtzeit zu übertragen.

Aufgabe 3 Mathematik-Auffrischung: Kreis

Gegeben sei ein Kreis (Radius $r \in \mathbb{R}$, Mittelpunkt $\vec{c} = [c_x, c_y]^T \in \mathbb{R}^2$).

- (a) Geben Sie eine Formel für den Euklidischen Abstand $d(\vec{x}, \vec{c})$ des Mittelpunktes \vec{c} zu einem beliebigen anderen Punkt $\vec{x} = [x, y]^T \in \mathbb{R}^2$!
- (b) Was gilt für einen Punkt $\vec{x} = [x, y]^T$ wenn dieser sich
- (i) innerhalb,
 - (ii) auf oder
 - (iii) außerhalb
- des Kreises befindet? Stellen Sie dazu Gleichungen bzw. Ungleichungen auf!

Aufgabe 4 Programmieren: Kreise

Laden Sie die Datei `Circle.html` in einen Browser. Laden Sie die Datei `Circle.js` und `Circle.html` in einen Texteditor.

- (a) Vervollständigen Sie die function `drawCircle`. Diese soll einen ausgefüllten Kreis mit Mittelpunkt $[128, 128]^T$ und Radius 96 zeichnen! Die Füllung des Kreises soll `[128, 128, 128, 255]` sein.

- (b) Vervollständigen Sie die function `drawCircleBoundary`. Diese Funktion soll zusätzlich noch einen schwarzen Rand im inneren des Kreises mit der Breite 10 Pixel zeichnen.
- (c) Zeichnen Sie nun den Kreis aus (b) mittels der in JavaScript durch die `CanvasRenderingContext2D` API bereitgestellten Methoden (siehe <https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D>). Vervollständigen Sie dazu die function `drawCircleCanvasContext`!
- (d) Zeichnen Sie nun den gleichen Kreis direkt in `Circle.html` mittels SVG.
- (e) Bewerten und vergleichen Sie die Kreise hinsichtlich ihrer Qualität!

Aufgabe 5 Plasma

Laden Sie die Datei `Plasma.html` in einen Browser. Laden Sie die Datei `Plasma.js` und `Circle.html` in einen Texteditor.

- (a) Vervollständigen Sie die function `drawPlasma`! Berechnen Sie dazu für die roten, grünen und blauen Farbkanälen jeden Pixels $[x, y]$ einen Intensitätswert der proportional zu

$$\sin(\alpha \cdot x + \beta) \cdot \cos(\gamma \cdot y + \delta)$$

ist. Dabei sind $[x, y]^T \in [0..2\pi) \times [0..2\pi)$ Bildkoordinaten. Setzen Sie den Alpha-Kanal auf 255! Wählen Sie Werte für die Perioden α, γ sowie Phasen β, δ so nach Belieben!

- (b) Variieren Sie die Phasen β, δ mit über die Zeit-Variable t nach Belieben.