2D-Punkt-Transformationen

Zur Erinnerung – Drehung eines beliebigen Punktes B‘ um den Winkel θ um den Koordinaten-Ursprung zum Punkt B“:

$x_{B'} = r \cdot \cos\alpha$

$y_{B'} = r \cdot \sin\alpha$ [r, α: Hilfsgrößen]

$x_{B''} = r \cdot \cos(\alpha+\theta)$

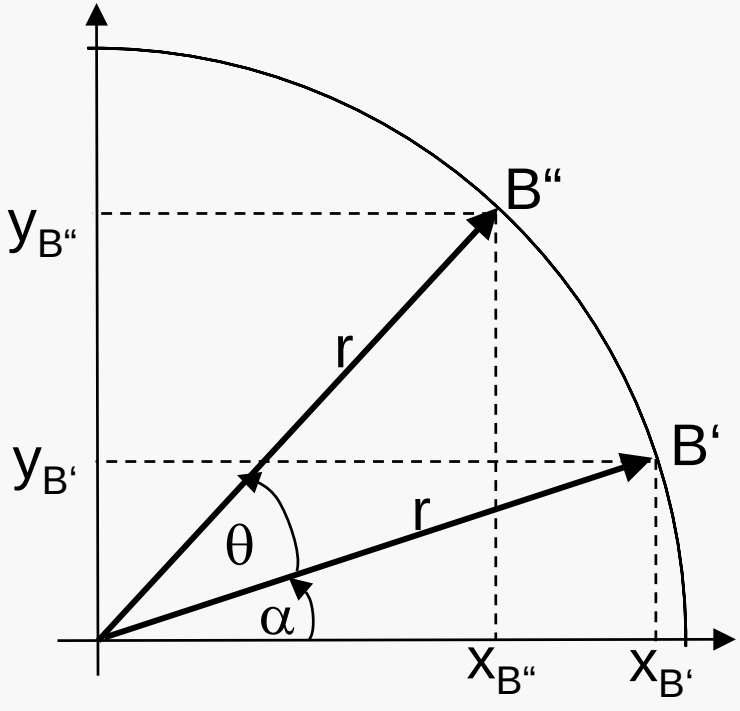
$= r \cdot (\cos\alpha \cos\theta - \sin\alpha \sin\theta)$

$= x_{B'} \cdot \cos\theta - y_{B'} \cdot \sin\theta$

$y_{B''} = r \cdot \sin(\alpha+\theta)$

$= r \cdot (\sin\alpha \cos\theta + \cos\alpha \sin\theta)$

$= x_{B'} \cdot \sin\theta + y_{B'} \cdot \cos\theta$



In Matrizen-Schreibweise:

$$\begin{pmatrix} x_{B''} \\ y_{B''} \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_{B'} \\ y_{B'} \end{pmatrix}$$

$\sin(\alpha+\beta) = \sin\alpha \cos\beta + \cos\alpha \sin\beta$

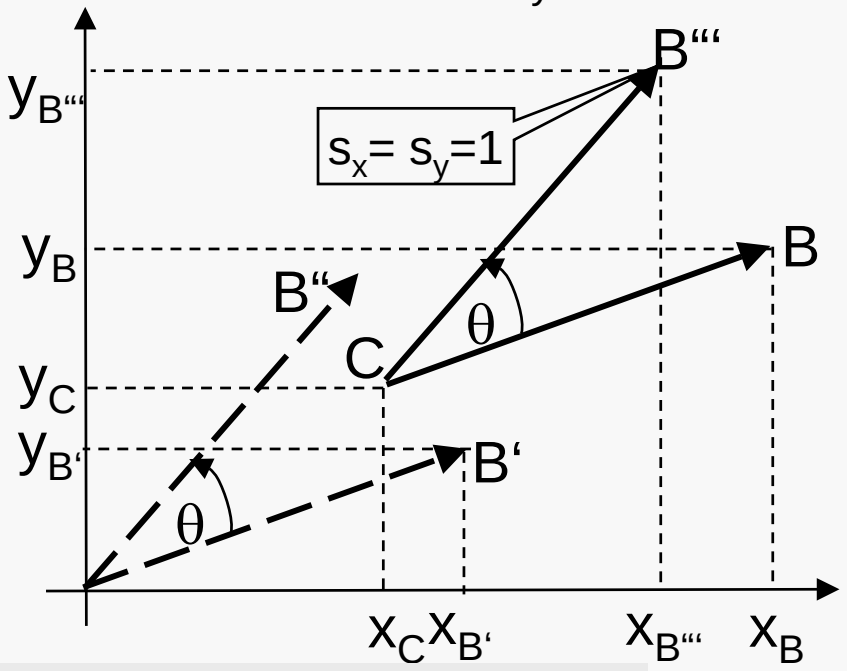
$\cos(\alpha+\beta) = \cos\alpha \cos\beta - \sin\alpha \sin\beta$

Verallgemeinerung: Rotation eines beliebigen Punktes B um den Winkel θ um einen beliebigen Punkt C zum Punkt B‘‘‘. Skalierung des Ergebnisses mit den Faktoren s_x und s_y.

$$x_{B'} = x_B - x_C$$
$$y_{B'} = y_B - y_C$$

$$x_{B''} = x_{B'} \cdot \cos\theta - y_{B'} \cdot \sin\theta$$
$$y_{B''} = x_{B'} \cdot \sin\theta + y_{B'} \cdot \cos\theta$$

$$x_{B'''} = s_x \cdot (x_{B''} + x_C)$$
$$y_{B'''} = s_y \cdot (y_{B''} + y_C)$$



In Matrizen-Schreibweise:

$$\begin{pmatrix} x_{B'''} \\ y_{B'''} \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \cdot \left\{ \begin{pmatrix} x_C \\ y_C \end{pmatrix} + \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \cdot \left[\begin{pmatrix} x_B \\ y_B \end{pmatrix} - \begin{pmatrix} x_C \\ y_C \end{pmatrix} \right] \right\}$$

2D-Punkt-Transformationen

Erweiterung um eine Dimension ermöglicht die Darstellung der Translation (Verschiebung) als Matrizen-Produkt:

$$x_{B'} = x_B - x_C$$
$$y_{B'} = y_B - y_C$$

„homogene Koordinaten“,
„homogene Punkt-Darstellung“

$$\begin{pmatrix} x_{B'} \\ y_{B'} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -x_C \\ 0 & 1 & -y_C \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix}$$

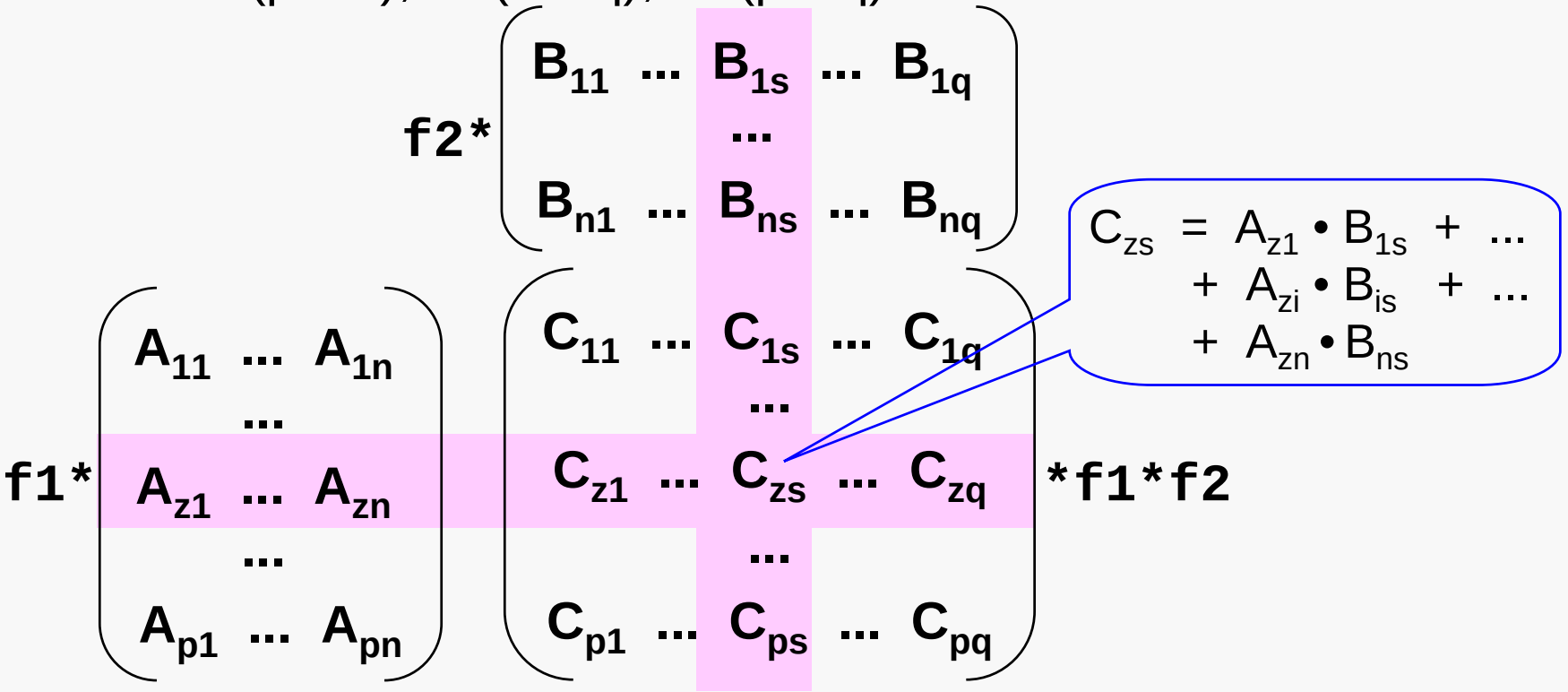
Rotation eines Punktes B um θ um bel. Punkt C nach B‘‘‘ und Skalierung mit s_x , s_y in homogenen Koordinaten:

$$\begin{pmatrix} x_{B'''} \\ y_{B'''} \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & x_C \\ 0 & 1 & y_C \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -x_C \\ 0 & 1 & -y_C \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix}$$

„Grafik-Pipeline“: Verarbeitungsschritte von den Objekt-Daten bis zur Bild-Erzeugung

2D-Punkt-Transformationen

- Implementierungshinweis zum Matrizenprodukt C = A • B:
  A: (p x n); B: (n x q); C: (p x q)




```
                    C[z][s] = 0.;//Initialisierg
for (i=1;i<=n;i++) C[z][s] += A[z][i]*B[i][s];
```

2D-Punkt-Transformationen

- Codierungshinweis zum Matrizenprodukt A • B in C/C++:
  A: (p x n); B: (n x q)); C: (p x q)
  Matrizen-Deklaration darf max. 1 Dimension offen lassen:

```
#define COL 3
float matrix[][COL];//1.Index(v.li.)offen
```

  Größere Flexibilität mit eindimensionalen Feldern und Umschreibung:

C[z][s] ⇨ C[z*q+s]

2D-Punkt-Transformationen

- Zur Erinnerung – einige Rechenregeln d. Matrizenrechnung*:

  $\underline{A} + \underline{B} = \underline{B} + \underline{A}$ [Kommutativgesetz d. Matrizenaddition]

  $\underline{A} \bullet (\underline{B} + \underline{C}) = \underline{A} \bullet \underline{B} + \underline{A} \bullet \underline{C}$ [Distributivgesetz]

  $(\underline{A} \bullet \underline{B}) \bullet \underline{C} = \underline{A} \bullet (\underline{B} \bullet \underline{C})$ [Assoziativgesetz]

  $\underline{A} \bullet \underline{B} \neq \underline{B} \bullet \underline{A}$ [Matrizenprodukt ist nicht kommutativ!]

  $\underline{A} \bullet \underline{B} = \underline{0} \iff \underline{B} = \underline{0} \vee \underline{A} = \underline{0}$!

  $\underline{A} \bullet \mathbf{\underline{I}} = \mathbf{\underline{I}} \bullet \underline{A} = \underline{A}$ [$\mathbf{\underline{I}}$:„neutrales Element“ d.M.-Multiplikation]

  $\underline{A} \bullet \underline{A}^{-1} = \underline{A}^{-1} \bullet \underline{A} = \mathbf{\underline{I}}$ (falls $\underline{A}$ invertierbar!)

  $\underline{A} \bullet \underline{x} = \underline{b} \Rightarrow \underline{A}^{-1} \bullet \underline{b} = \underline{A}^{-1} \bullet \underline{A} \bullet \underline{x} = \underline{x}$ (falls $\underline{A}$ invertierbar!)

  $(\underline{A} \bullet \underline{B})^{-1} = \underline{B}^{-1} \bullet \underline{A}^{-1}$ (falls $\underline{A}$, $\underline{B}$ invertierbar!)

  $(\underline{A} \bullet \underline{B})^{T} = \underline{B}^{T} \bullet \underline{A}^{T}$

  Wichtiger Spezialfall: Rotationsmatrizen sind orthogonal, d.h.

  $\underline{O}^{-1} = \underline{O}^{T}$

(*) unter d. Voraussetzung geeigneter Dimensionierung beteiligter Matrizen

Verkettung von Punkt-Transformationen und ihren Inversen:

- Translation (Verschiebung):

$$\begin{pmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_c - x_c \\ 0 & 1 & y_c - y_c \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Skalierung:

$$\begin{pmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Spiegelung ist ein Spezialfall der Skalierung!

- Rotation (Drehung):

$$\begin{pmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos(-\theta)\cos\theta - \sin(-\theta)\sin\theta & -\cos(-\theta)\sin\theta - \sin(-\theta)\cos\theta & 0 \\ \sin(-\theta)\cos\theta + \cos(-\theta)\sin\theta & -\sin(-\theta)\sin\theta + \cos(-\theta)\cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2D-Punkt-Transformationen

Beobachtungen an der Rotationsmatrix:
(vor allem an der o./li. 2x2-Untermatrix)

$$\underline{\mathbf{R}} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Jede Zeile u. jede Spalte ist ein Einheitsvektor (Länge=1)
- Sowohl die Zeilen- als auch die Spaltenvektoren stehen jeweils senkrecht aufeinander (Skalarprodukte =0)

➡ $\underline{\mathbf{R}}$ ist orthogonal: $\underline{\mathbf{R}} \cdot \underline{\mathbf{R}}^T = \underline{\mathbf{R}}^T \cdot \underline{\mathbf{R}} = \underline{\mathbf{I}}$

(^{T}:Transposition; $\underline{\mathbf{I}}$: Einheitsmatrix)

- Werden die Einheitsvektoren der Hauptachsen (x,y) mit $\underline{\mathbf{R}}$ transformiert, so ergeben sie die Spaltenvektoren von $\underline{\mathbf{R}}$.
- Werden die Zeilenvektoren von $\underline{\mathbf{R}}$ mit $\underline{\mathbf{R}}$ transformiert, so ergeben sie die Einheitsvektoren der Hauptachsen (x,y).

➡ Möglichkeit, aus dem Rotationsergebnis auf die Rotationsmatrix zu schließen!

2D-Punkt-Transformationen

Weitere 2D-Transformation: **Scherung** (engl. *shear*)

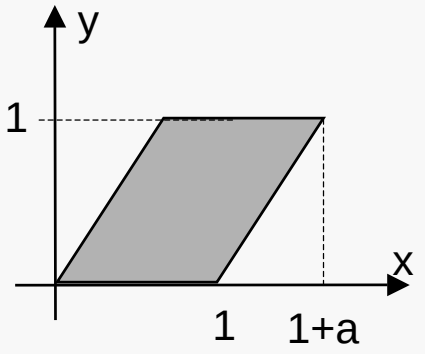
Scherung entlang einer Achse verändert dazugehörige Punkt-Koordinaten proportional zur jeweils anderen Koordinate:

- Scherung eines Einheitsquadrats entlang der x-Achse verändert jeden Punkt $[x, y, 1]^T$ zu $[x+ay, y, 1]^T$ mit:

$$\begin{pmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Inverse:

$$\begin{pmatrix} 1 & -a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

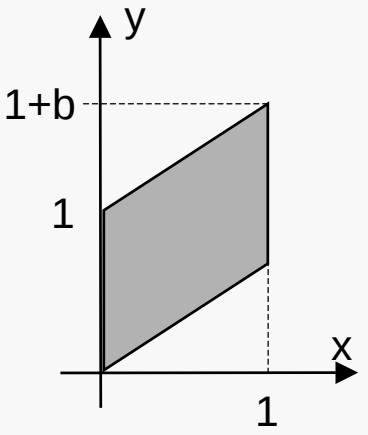


- Scherung des Einheitsquadrats entlang der y-Achse verändert jeden Punkt $[x, y, 1]^T$ zu $[x, bx+y, 1]^T$ mit:

$$\begin{pmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Inverse:

$$\begin{pmatrix} 1 & 0 & 0 \\ -b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



x-y-Kombination als Verkettung!

Window-Viewport-Transformation

Alte Begriffe der Computergrafik:

- (World-Coordinate-) **Window** heißt ein rechteckiger (Szene-, Welt-) Ausschnitt, der abgebildet werden soll; darf nicht mit dem (Window-Manager-) Window des Sw-Fenstersystems verwechselt werden.
- **Viewport** nennt man den rechteckigen Ausschnitt der Geräte-Ausgabefläche (Bildschirm, Plotter), in dem die Ausgabe erfolgen soll.

Im allgemeinen Fall besteht die Window-Viewport-Transform. aus

- einer Translation des Window an den Ursprung des Welt-Koordinatensystems,
- einer Skalierung auf die Größe des Viewport und
- einer Translation an die Lage des Viewport.

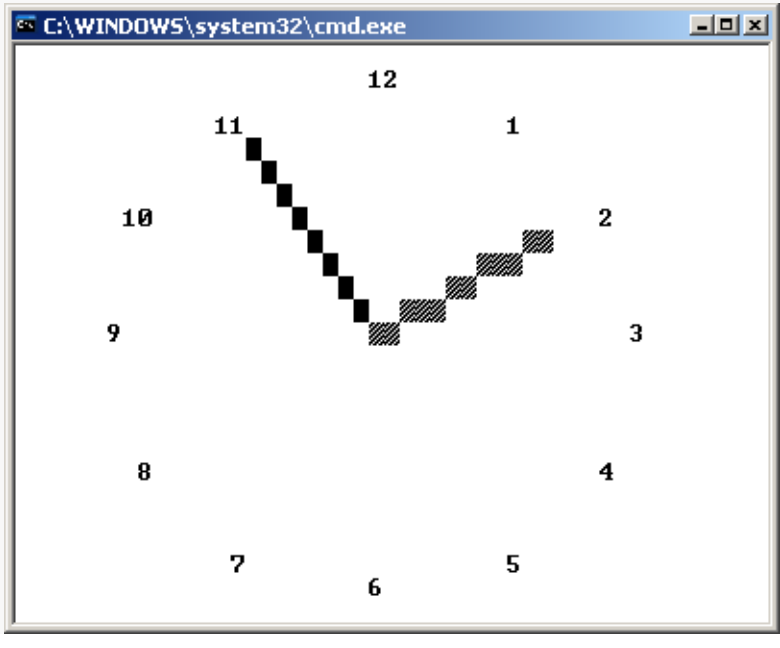
Übung

Übung: Erstellung einer Analog-Uhr aus ASCII-Zeichen der Größe 8x12:

(i) Zeichnung und Positionierung der Zeiger als Linien;

(ii) Realitätsnahe Animation.

Durch entsprechende Skalierung ist die Uhr auf eine kreisrunde Form zu bringen.



`BresenClock.exe`