

## 10 Grundlegende mathematische Sachverhalte

In der Computer Graphik ist es üblich, zweidimensionale Objekte durch ihren Konturpolygonzug, d.h. den Streckenzug durch die Eckpunkte des Objektes, zu beschreiben. Bei dreidimensionalen Objekten wird die Objektfläche häufig in ebene Facetten (z.B. Dreiecke oder Vierecke) zerlegt, wobei die Facetten wiederum über die Eckpunkte ihres Konturpolygonzuges beschrieben sind. Grundlegende geometrische Objekte sind also Punkte und die durch diese Punkte definierten Verbindungsstrecken.

In den folgenden Teilkapiteln werden diese und weitere geometrische Grundobjekte definiert. Weiterhin werden einfache geometrische Transformationen eingeführt, mit deren Hilfe Objekte geometrisch transformiert (z.B. verschoben, gedreht und vergrößert bzw. verkleinert) werden können, indem man die Eckpunkte der Objekte transformiert.

### 10.1 Einfache geometrische Grundobjekte

Alle weiteren Betrachtungen dieses Teilkapitels beschränken sich auf die zweidimensionale Zeichenebene. In dieser werden Punkte als nulldimensionale Objekte, und Strecken, Strahlen und Geraden als eindimensionale Objekte betrachtet.

#### 10.1.1 Punkt

Die **Punkte** der Zeichenebene werden durch ihre x- und ihre y-Koordinaten festgelegt. Punkte werden im weiteren als Spaltenvektoren geschrieben:

$$\vec{P} = \begin{bmatrix} x_P \\ y_P \end{bmatrix} = [x_P, y_P]^t \quad (\text{mit } ^t \text{ für transponiert}) \tag{10-1}$$

#### 10.1.2 Gerade, Strahl, Strecke

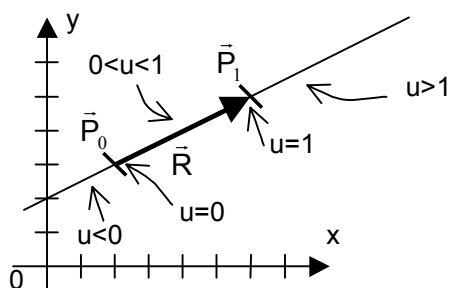
Strecken, Strahlen und Geraden werden im folgenden in parametrischer Form eingeführt. Dies hat unter anderem den Vorteil, dass man mit Hilfe des Parameters jeden Punkt der Objekte gezielt ansteuern kann.

##### 10.1.2.1 Parameterform (vektorielle Schreibweise)

Die (gerichtete) **Strecke** zwischen zwei Punkten  $\vec{P}_0$  und  $\vec{P}_1$  (siehe Abbildung 10.1) ist die Menge aller Punkte  $\vec{X}$ , für die gilt:

$$\vec{X}(u) = \vec{P}_0 + u * (\vec{P}_1 - \vec{P}_0) = \vec{P}_0 + u * \vec{R} \quad \text{mit } u \in [0,1]. \quad (\vec{R} \text{ ist der Richtungsvektor von } \vec{P}_0 \text{ nach } \vec{P}_1). \tag{10-2}$$

Diese Form wird als **vektorielle Schreibweise der Parameterform** bezeichnet.



**Abb. 10.1:** Beispiel für die parametrische Form einer Strecke bzw. eines Strahls oder einer Geraden.

Für  $u=0$  bzw.  $u=1$  erhält man die Punkte  $\vec{P}_0$  und  $\vec{P}_1$ , und für  $0 < u < 1$  erhält man alle Zwischenpunkte auf der Strecke zwischen  $\vec{P}_0$  und  $\vec{P}_1$ .

Der **Strahl**, der von  $\vec{P}_0$  ausgeht, und über  $\vec{P}_1$  hinaus nach Unendlich verläuft (siehe Abbildung 10.1), wird durch die Menge aller Punkte  $\vec{X}$  gebildet, für die gilt:

$$\vec{X}(u) = \vec{P}_0 + u * (\vec{P}_1 - \vec{P}_0) = \vec{P}_0 + u * \vec{R} \quad \text{mit } u \geq 0. \tag{10-3}$$

Die (gerichtete) **Gerade** durch die Punkte  $\vec{P}_0$  und  $\vec{P}_1$  besteht aus allen Punkten  $\vec{X}$  (siehe Abbildung 10.1), für die gilt:

$$\vec{X}(u) = \vec{P}_0 + u * (\vec{P}_1 - \vec{P}_0) = \vec{P}_0 + u * \vec{R} \quad \text{mit } u \in \mathbb{R} \text{ (reelle Zahl)}. \tag{10-4}$$

**10.1.2.2 Parameterform (Matrix-Schreibweise)**

Durch Umformung der vektoriellen Form erhält man die **Matrix-Schreibweise der Parameterform**:

$$\vec{X}(u) = \vec{P}_0 + u * (\vec{P}_1 - \vec{P}_0) = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + u * \begin{bmatrix} x_1 - x_0 \\ y_1 - y_0 \end{bmatrix} = \begin{bmatrix} x_0 - u * x_0 + u * x_1 \\ y_0 - u * y_0 + u * y_1 \end{bmatrix} = \begin{bmatrix} x_0 * (1-u) + u * x_1 \\ y_0 * (1-u) + u * y_1 \end{bmatrix}$$

und damit : (10-5)

$$\vec{X}(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = \begin{bmatrix} x_0 & x_1 \\ y_0 & y_1 \end{bmatrix} * \begin{bmatrix} 1-u \\ u \end{bmatrix}$$

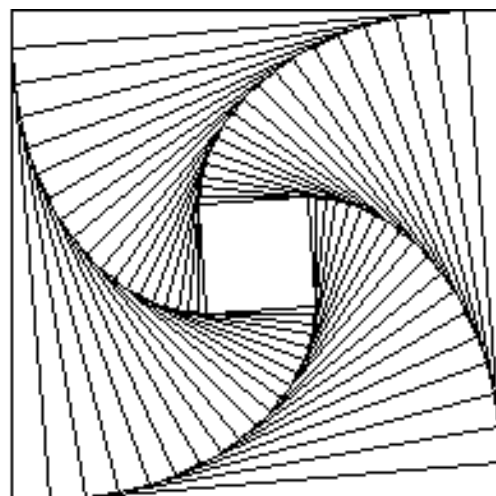
Mit diesem Produkt aus einer Matrix und einem Spaltenvektor erhält man entsprechend zu oben

- mit  $u \in [0,1]$  eine Strecke,
- mit  $u \geq 0$  einen Strahl und
- mit  $u \in \mathbb{R}$  (reelle Zahl) eine Gerade.

**Schlußbemerkungen:** Die Verwendung der Parameterform erlaubt auch die besonders einfache und effiziente Berechnung von Schnittpunkten. Dies ist z.B. im CAD-Bereich bei der Konstruktion von Teilen und in der Computer Graphik beispielsweise beim Raytracing (Strahlverfolgung) zur Erzeugung photorealistischer Darstellungen eine ständig wiederkehrende Aufgabe.

**Beispiel:**  $p[0]$ ,  $p[1]$ ,  $p[2]$  und  $p[3]$  seien vier im Quadrat angeordnete Punkte. Die Seiten des Quadrates sind Strecken, die in vektorieller Parameterform gegeben sind durch:  $x = p[0] + u*(p[1]-p[0])$  bis  $x = p[3] + u*(p[0]-p[3])$  mit  $u \in [0,1]$ . Für einen festen Wert  $u \in [0,1]$  erhält man auf jeder Seite des Quadrates einen Zwischenpunkt  $zp[i]$ , wobei alle Quadratseiten im selben Verhältnis unterteilt werden. Verbindet man diese Zwischenpunkte miteinander, so erhält man wieder ein Quadrat, auf dessen Seiten wiederum Zwischenpunkte ermittelt und verbunden werden, usw.. Das folgende Pseudocode-Programmstück liefert für  $u = 0.08$  das in Abbildung 10.2 gezeigte Bild.

```
// Eingaben
Parameter u und Punkte p[0] bis p[3] eingeben;
// 20 Quadrate zeichnen
for (int j=0; j<20; j++) {
    p[4]= p[0];
    // Ein Quadrat zeichnen
    for (int i=0; i<4; i++) {
        Zeichne Linie von p[i] nach p[i+1]);
        // neuen Zwischenpunkt berechnen
        zp[i] = p[i] + u*(p[i+1]-p[i]);
    }
    // Zwischenpunkte werden neue Eckpunkte
    for (i=0; i<4; i++)
        p[i] = zp[i];
}
```



**Abb. 10.2:** Konstruktion von 20 Quadraten unter Verwendung der Parameterform.

## 10.2 2D-Koordinatentransformationen

Mit Hilfe von geometrischen Transformationen können Objekte in der Zeichenebene positioniert, orientiert und skaliert werden. Auch Bewegungen, wie sie z.B. im Rahmen einer Animation auftreten, lassen sich mit geometrischen Transformationen durchführen. Weiterhin kann man bei der Modellierung komplexe Objekte aus mehreren geeignet positionierten, orientierten und skalierten Einzelteilen zusammensetzen.

### 10.2.1 2D-Elementar-Transformationen (konventionell)

Als Beispiele geometrischer **2D-Elementar-Transformationen** werden im weiteren die Translation, die Skalierung und die Rotation vorgestellt.

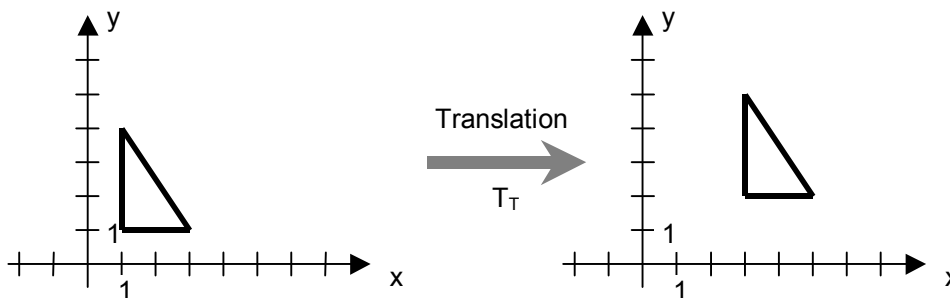
#### 10.2.1.1 Translation (Verschiebung)

Die **Translation (Verschiebung)**  $T_T$  eines Punktes  $\vec{P} = [x_P, y_P]^t$  um  $dx$  in  $x$ - und  $dy$  in  $y$ -Richtung liefert den Punkt  $\vec{P}' = [x_P', y_P']^t$  mit:

(10-6)

$$\begin{aligned} x_P' &= dx + x_P \\ y_P' &= dy + y_P \end{aligned} \quad \text{d.h.} \quad \begin{bmatrix} x_P' \\ y_P' \end{bmatrix} = \begin{bmatrix} dx \\ dy \end{bmatrix} + \begin{bmatrix} x_P \\ y_P \end{bmatrix} \quad \text{bzw.} \quad \vec{P}' = T_T + \vec{P}.$$

#### Beispiel:



**Abb. 10.3:** Beispiel für eine Translation um  $dx = 2$  und  $dy = 1$ .

#### 10.2.1.2 Skalierung (Größenänderung)

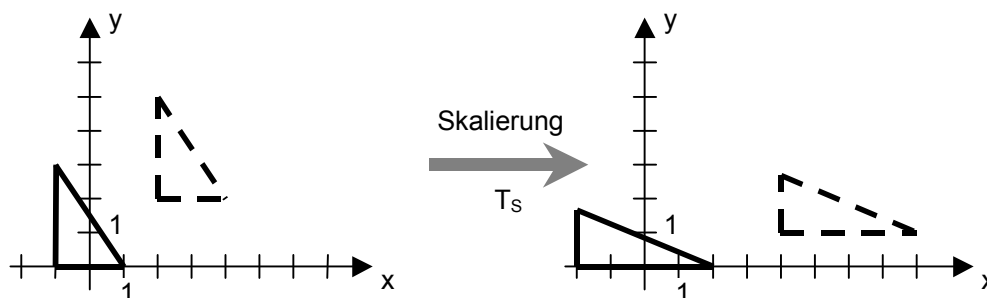
Die **Skalierung (Größenänderung)**  $T_S$  eines Punktes  $\vec{P} = [x_P, y_P]^t$  mit  $s_x$  in  $x$ - und  $s_y$  in  $y$ -Richtung bezüglich des Ursprungs liefert den Punkt  $\vec{P}' = [x_P', y_P']^t$  mit:

(10-7)

$$\begin{aligned} x_P' &= s_x * x_P \\ y_P' &= s_y * y_P \end{aligned} \quad \text{d.h.} \quad \begin{bmatrix} x_P' \\ y_P' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} * \begin{bmatrix} x_P \\ y_P \end{bmatrix} \quad \text{bzw.} \quad \vec{P}' = T_S * \vec{P}.$$

(Natürlich ändert sich bei einer Skalierung nicht die Größe, sondern nur die Lage eines Punktes.)

#### Beispiel:



**Abb. 10.4:** Beispiele für eine Skalierung mit  $s_x = 2$  und  $s_y = \frac{1}{2}$  bezüglich des Ursprungs.

Beispielsweise liefert die Skalierung mit  $s_x=2$  und  $s_y=1/2$  für die obere Ecke des linken Dreiecks:

$$\vec{P}' = \begin{bmatrix} x_{P'} \\ y_{P'} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix} * \begin{bmatrix} -1 \\ 3 \end{bmatrix} = \begin{bmatrix} -2 \\ 3/2 \end{bmatrix}$$

Das Beispiel zeigt, dass keineswegs gelten muss:  $s_x = s_y$ .

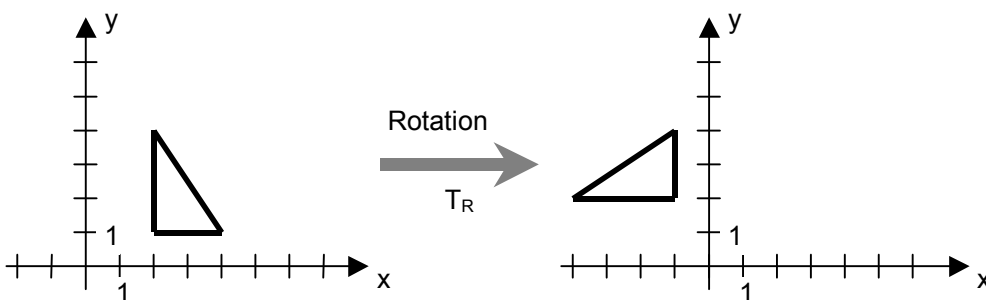
### 10.2.1.3 Rotation (Drehung)

Die **Rotation (Drehung)**  $T_R$  eines Punktes  $\vec{P} = [x_P, y_P]^t$  um den Winkel  $\varphi$  bezüglich des Ursprungs liefert den Punkt  $\vec{P}' = [x_{P'}, y_{P'}]^t$  mit:

(10-8)

$$\begin{aligned} x_{P'} &= \cos(\varphi) * x_P - \sin(\varphi) * y_P \\ y_{P'} &= \sin(\varphi) * x_P + \cos(\varphi) * y_P \end{aligned} \quad \text{d.h.} \quad \begin{bmatrix} x_{P'} \\ y_{P'} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} * \begin{bmatrix} x_P \\ y_P \end{bmatrix} \quad \text{bzw.} \quad \vec{P}' = T_R * \vec{P}.$$

#### Beispiel:



**Abb. 10.5:** Beispiel für eine Rotation um den Winkel  $\varphi = 90^\circ$  bezüglich des Ursprungs.

Beispielsweise liefert die Rotation um den Winkel  $\varphi = 90^\circ$  für den oberen Eckpunkt des Dreiecks:

$$\vec{P}' = \begin{bmatrix} x_{P'} \\ y_{P'} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} -4 \\ 2 \end{bmatrix}$$

### 10.2.1.4 Kombination von 2D-Elementar-Transformationen

Im allgemeinen besteht eine gewünschte Transformation nicht nur aus einer einzelnen Elementar-Transformation, sondern es müssen verschiedene Elementar-Transformationen miteinander kombiniert werden. Da die Skalierung (10-7) und die Rotation (10-8) als Matrixmultiplikationen, die Translation (10-6) aber als Vektoraddition eingeführt wurde, lassen sich Kombinationen von Elementar-Transformationen in der Regel nicht in einer einzigen Transformationsmatrix zusammenfassen. Das nachfolgende Kapitel zeigt, wie sich dieser Nachteil durch die Verwendung homogener Koordinaten vermeiden lässt.

### 10.2.2 2D-Elementar-Transformationen mit homogenen Koordinaten

**Homogene Koordinaten** stellen eine Erweiterung der „normalen“ Koordinaten um eine weitere, sogenannte **homogene Komponente** dar. Ein Punkt  $\vec{P} = [x_P, y_P]^t$  wird in homogenen Koordinaten geschrieben als  $\vec{P} = [x_P, y_P, 1]^t$ . Die homogene Komponente hat im weiteren bei Punkten immer den Wert 1.

Unter Verwendung von homogenen Koordinaten können alle drei eingeführten Transformationen als Matrix-Multiplikationen ausgedrückt werden.

#### 10.2.2.1 Translation (Verschiebung)

Die **Translation (Verschiebung)**  $T_T$  eines Punktes  $\vec{P} = [x_P, y_P, 1]^t$  um  $dx$  in  $x$ - und  $dy$  in  $y$ -Richtung liefert den Punkt  $\vec{P}' = [x_{P'}, y_{P'}, 1]^t$  mit:

(10-9)

$$\begin{bmatrix} x_{P'} \\ y_{P'} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_P \\ y_P \\ 1 \end{bmatrix} \quad \text{bzw. } \vec{P}' = T_T * \vec{P}.$$

Die inverse Transformation zu einer Translation um dx und dy ist die Translation um -dx und -dy.

Bei einer Translation mit dx≠0 oder dy≠0 werden alle Punkte der Zeichenebene verschoben, so dass eine Translation keinen **Fixpunkt** besitzt.

### 10.2.2.2 Skalierung (Größenänderung)

Die **Skalierung (Größenänderung)** T<sub>S</sub> eines Punktes  $\vec{P} = [x_P, y_P, 1]^t$  mit s<sub>x</sub> in x- und s<sub>y</sub> in y-Richtung bezüglich des Ursprungs liefert den Punkt  $\vec{P}' = [x_{P'}, y_{P'}, 1]^t$  mit:

(10-10)

$$\begin{bmatrix} x_{P'} \\ y_{P'} \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_P \\ y_P \\ 1 \end{bmatrix} \quad \text{bzw. } \vec{P}' = T_S * \vec{P}.$$

Die inverse Transformation zu einer Skalierung mit s<sub>x</sub> und s<sub>y</sub> ist die Skalierung 1/s<sub>x</sub> und 1/s<sub>y</sub>.

Da der Ursprung sich bei einer Skalierung (bezüglich des Ursprungs) nicht ändert, ist der Ursprung der Fixpunkt dieser Skalierung.

### Spezielle Skalierungen

1. Wenn genau einer der Skalierungsfaktoren s<sub>x</sub> oder s<sub>y</sub> gleich Null ist, so ergibt sich eine **Projektion**. Für s<sub>x</sub>=0 erhält man eine Projektion auf die y-Achse, und für s<sub>y</sub>=0 erhält man eine Projektion auf die x-Achse. Wenn der von Null verschiedene Faktor ungleich Eins ist, so ergibt sich zusätzlich eine Größenänderung.
2. Wenn einer oder beide Skalierungsfaktoren negativ sind, so erhält man eine **Spiegelung**:
  - mit s<sub>x</sub>=-1 und s<sub>y</sub>=1 erhält man eine Geraden-Spiegelung an der y-Achse.
  - mit s<sub>x</sub>=1 und s<sub>y</sub>=-1 erhält man eine Geraden-Spiegelung an der x-Achse.
  - mit s<sub>x</sub>=-1 und s<sub>y</sub>=-1 erhält man eine Punkt-Spiegelung am Ursprung.

(Wenn einer oder beide Faktoren betragsmäßig ungleich Eins sind, so ergibt sich zusätzlich eine Größenänderung).

### 10.2.2.3 Rotation (Drehung)

Die **Rotation (Drehung)** T<sub>R</sub> eines Punktes  $\vec{P} = [x_P, y_P, 1]^t$  um den Winkel φ bezüglich des Ursprungs liefert den Punkt  $\vec{P}' = [x_{P'}, y_{P'}, 1]^t$  mit:

(10-11)

$$\begin{bmatrix} x_{P'} \\ y_{P'} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_P \\ y_P \\ 1 \end{bmatrix} \quad \text{bzw. } \vec{P}' = T_R * \vec{P}.$$

Die inverse Transformation zu einer Rotation um den Winkel φ ist die Rotation um den Winkel -φ

Da der Ursprung sich bei einer Rotation (bezüglich des Ursprungs) nicht ändert, ist der Ursprung der Fixpunkt dieser Rotation.

### 10.2.3 Zusammengesetzte 2D-Transformationen

Mit Hilfe von homogenen Koordinaten lassen sich die Elementar-Transformationen (Translation, Skalierung und Rotation) eines Punktes als Multiplikation einer 3\*3-Matrix mit dem Spaltenvektor des zu transformierenden Punktes darstellen. Dies hat zur Folge, dass komplexe Transformationen, die sich als **Hintereinanderausführung (Verkettung)** mehrerer Elementar-Transforma-

tionen schreiben lassen, zu einer einzigen **Gesamttransformation**  $T_G$  zusammengefasst werden können.

Eine Gesamttransformation  $T_G$ , die sich aus

- einer Skalierung  $T_S$  mit  $s_x$  und  $s_y$  bezüglich eines Punktes  $\vec{P}_0 = [x_0, y_0, 1]^t$ ,
- einer Rotation  $T_R$  um den Winkel  $\varphi$  bezüglich desselben Punktes und
- einer Translation  $T_T$  um  $dx$  und  $dy$

zusammensetzt, wird vereinbarungsgemäß wie folgt zerlegt:

$$T_G = T_{T''} * T_R * T_S * T_T \tag{10-12}$$

Dabei ist  $T_{T''}$  die Hilfstranslation um  $-x_0$  und  $-y_0$ , die den Punkt  $\vec{P}_0 = [x_0, y_0, 1]^t$  in den Ursprung verschiebt. Die Skalierung  $T_S$  und die Rotation  $T_R$  können nun wie oben beschrieben bezüglich des Ursprungs durchgeführt werden. Die abschließende Hilfstranslation  $T_{T''}$  macht die erste Hilfstranslation  $T_T$  rückgängig und führt zusätzlich die gewünschte Translation um  $dx$  und  $dy$  durch.

Ausmultiplizieren von (10-12) liefert:

$$T_G = \begin{bmatrix} 1 & 0 & x_0 + dx \\ 0 & 1 & y_0 + dy \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} =$$

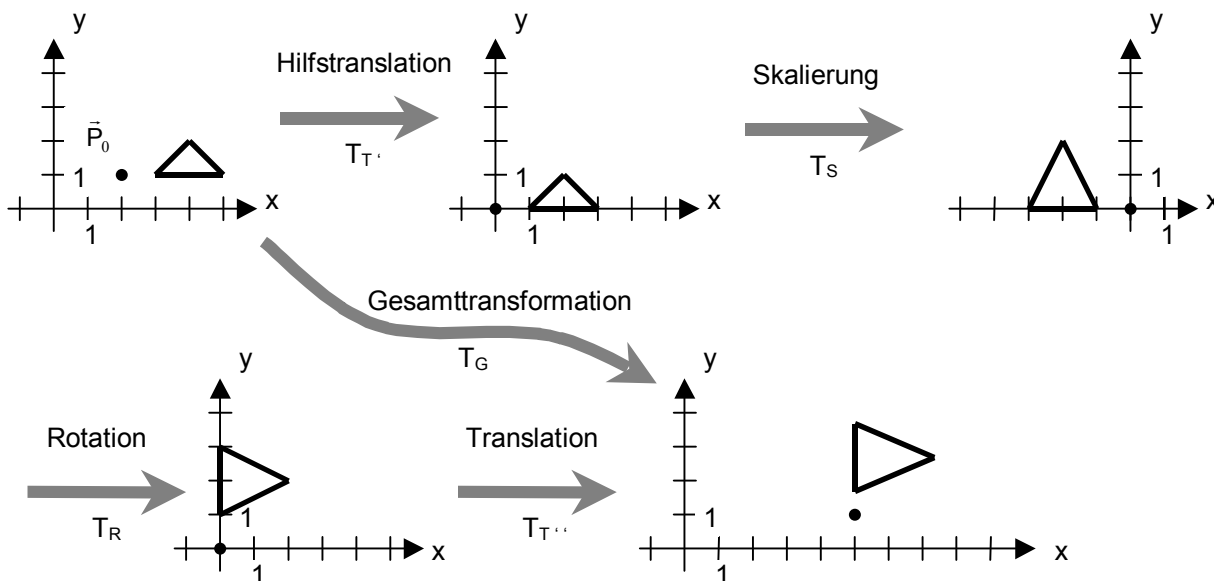
$$= \begin{bmatrix} s_x * \cos(\varphi) & -s_y * \sin(\varphi) & -x_0 * s_x * \cos(\varphi) + y_0 * s_y * \sin(\varphi) + x_0 + dx \\ s_x * \sin(\varphi) & s_y * \cos(\varphi) & -x_0 * s_x * \sin(\varphi) - y_0 * s_y * \cos(\varphi) + y_0 + dy \\ 0 & 0 & 1 \end{bmatrix} \tag{10-13}$$

Die Transformation eines allgemeinen Punktes  $\vec{P} = [x_P, y_P, 1]^t$  ergibt sich also zu:

$$\vec{P} = T_{T''} * T_R * T_S * T_T * \vec{P} = T_G * \vec{P} \tag{10-14}$$

**Beispiel:** Eine zusammengesetzte Transformation soll bezüglich eines Punktes  $\vec{P}_0 = [2, 1, 1]^t$  eine Skalierung mit  $s_x=-1$  und  $s_y=2$ , sowie eine Rotation um  $270^\circ$  und außerdem eine Translation um

$dx=3$  und  $dy=0$  durchführen. Die Gesamttransformation ergibt sich zu:  $T_G = \begin{bmatrix} 0 & 2 & 3 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix}$ .



**Abb. 10.6:** Beispiel für eine zusammengesetzte Transformation.

Mit Hilfe der **Fixpunkt-Bedingung**

(10-15)

$$\vec{P}_F = T * \vec{P}_F$$

kann man feststellen, ob eine Transformation T einen Fixpunkt  $\vec{P}_F$  hat, also einen Punkt, der sich bei der Transformation nicht ändert.

**Beispiel:** Mit Hilfe der Fixpunkt-Bedingung erhält man für die Gesamttransformation  $T_G$  aus dem obigen Beispiel:

$$\vec{P}_F = \begin{bmatrix} x_F \\ y_F \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 3 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_F \\ y_F \\ 1 \end{bmatrix} = \begin{bmatrix} 2 * y_F + 3 \\ x_F - 1 \\ 1 \end{bmatrix} \quad \text{also: } x_F = 2 * y_F + 3 \quad \text{und: } y_F = x_F - 1 \quad \text{und damit } \vec{P}_F = [-1, -2, 1]^t.$$

### 10.3 3D-Koordinatentransformationen mit homogenen Koordinaten

Ein 3D-Punkt wird in **homogenen Koordinaten** geschrieben als  $\vec{P} = [x_P, y_P, z_P, 1]^t$ . Unter Verwendung von homogenen Koordinaten können unter anderem die **3D-Elementar-Transformationen** Translation, Skalierung und Rotation als Matrix-Multiplikationen ausgedrückt werden. (Alternative Möglichkeiten, wie z.B. die perspektivische Projektion werden hier nicht betrachtet).

#### 10.3.1 Translation (Verschiebung)

Die **Translation (Verschiebung)**  $T_T$  eines Punktes  $\vec{P} = [x_P, y_P, z_P, 1]^t$  um dx in x-Richtung, dy in y-Richtung und dz in z-Richtung liefert den Punkt  $\vec{P}' = [x_P', y_P', z_P', 1]^t$  mit:

(10-16)

$$\begin{bmatrix} x_{P'} \\ y_{P'} \\ z_{P'} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} \quad \text{bzw. } \vec{P}' = T_T * \vec{P}.$$

#### 10.3.2 Skalierung (Größenänderung)

Die **Skalierung (Größenänderung)**  $T_S$  eines Punktes  $\vec{P} = [x_P, y_P, z_P, 1]^t$  mit  $s_x$  in x-Richtung,  $s_y$  in y-Richtung und  $s_z$  in z-Richtung bezüglich des Ursprungs liefert den Punkt  $\vec{P}' = [x_P', y_P', z_P', 1]^t$  mit:

(10-17)

$$\begin{bmatrix} x_{P'} \\ y_{P'} \\ z_{P'} \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} \quad \text{bzw. } \vec{P}' = T_S * \vec{P}.$$

#### 10.3.3 Rotationen (Drehungen)

Die **Rotation (Drehung)**  $T_{Rz}$  eines Punktes  $\vec{P} = [x_P, y_P, z_P, 1]^t$  um den Winkel  $\varphi$  um die z-Achse liefert den Punkt  $\vec{P}' = [x_P', y_P', z_P', 1]^t$  mit:

(10-18)

$$\begin{bmatrix} x_{P'} \\ y_{P'} \\ z_{P'} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} \quad \text{bzw. } \vec{P}' = T_{Rz} * \vec{P}.$$

Die **Rotation (Drehung)**  $T_{Rx}$  eines Punktes  $\vec{P} = [x_P, y_P, z_P, 1]^t$  um den Winkel  $\varphi$  um die x-Achse liefert den Punkt  $\vec{P}' = [x_P', y_P', z_P', 1]^t$  mit:

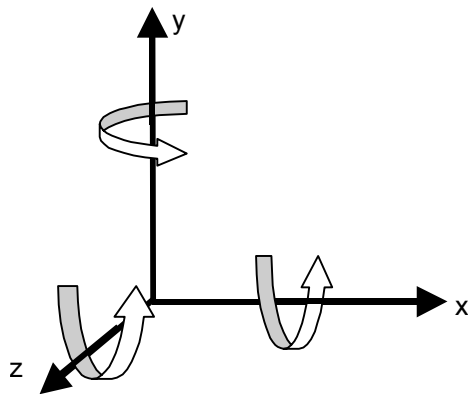
(10-19)

$$\begin{bmatrix} x_{P'} \\ y_{P'} \\ z_{P'} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) & 0 \\ 0 & \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} \quad \text{bzw. } \vec{P}' = T_{R_x} * \vec{P}.$$

Die **Rotation (Drehung)**  $T_{R_y}$  eines Punktes  $\vec{P} = [x_P, y_P, z_P, 1]^t$  um den Winkel  $\varphi$  um die y-Achse liefert den Punkt  $\vec{P}' = [x_{P'}, y_{P'}, z_{P'}, 1]^t$  mit:

(10-20)

$$\begin{bmatrix} x_{P'} \\ y_{P'} \\ z_{P'} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} \quad \text{bzw. } \vec{P}' = T_{R_y} * \vec{P}.$$



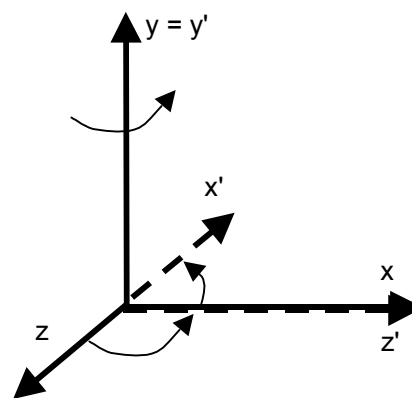
**Abb. 10.8:** Demonstration des mathematisch positiven Drehsinns.

**Bemerkungen zu Rotationen:**

- 1.) Wenn der Daumen der rechten Hand in die Richtung der betreffenden Achse zeigt, geben die angewinkelten Finger den mathematisch positiven Drehsinn an; siehe Abbildung 10.8.
- 2.) Die vordersten drei Spalten einer Rotationsmatrix sind die Transformaten der Einheits-Richtungs-Vektoren der x-Achse, y-Achse und z-Achse. Die Spalten geben also an, in welche Richtung die transformierten Koordinaten-Achsen verlaufen; siehe Abbildung 10.9. (Bei Richtungs-Vektoren in homogener Schreibweise hat die homogene Komponente den Wert 0).

$$T_{R_y}(90^\circ) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

transformierte z-Achse z' zeigt in Richtung der positiven x-Achse  
 y-Achse bleibt unverändert  
 transformierte x-Achse x' zeigt in Richtung der negativen z-Achse



**Abb. 10.9:** Beispiel für eine Rotation um 90 Grad um die y-Achse; die Spalten der Rotations-Matrix (links) geben die Richtung der transformierten Achsen (rechts) an.