

§1 Hardwaregrundlagen

§2 Transformationen und Projektionen

2.1 Koordinatentransformationen

2.2 Transformationen in der Ebene

2.3 Transformationen im Raum

§3 Repräsentation und Modellierung von Objekten

§4 Rasterung

§5 Visibilität und Verdeckung

§6 Rendering

§7 Abbildungsverfahren (Texturen, etc.)

§8 Freiformmodellierung

Anhang: Graphiksprachen und Graphikstandards

Anhang: Einführung in OpenGL

Weitere Themen: Netze, Fraktale, Animation, ...

- Grundlage der Bildgestaltung auf dem Bildschirm oder dem Ausgabegerät sind Koordinatentransformationen im \mathbb{R}^2 oder \mathbb{R}^3 .
- Im Allgemeinen unterscheidet sich das Koordinatensystem eines Objektes von dem Koordinatensystem des Ausgabegerätes.
- Das Koordinatensystem des Objektes ist oft über geometrische Eigenschaften des Objektes festgelegt (z. B. ausgezeichnete Richtungen, Symmetrien).
- Das Koordinatensystem des Bildschirms oder die Größe der Bildfenster ist durch das Gerät selbst definiert (z. B. Nullpunkt in der linken, oberen Ecke, x- und y-Achsen parallel zu den Bildrändern).

- Mit Hilfe von Koordinatentransformationen (Translationen, Skalierungen, Rotationen) wird das Objektsystem in das Gerätesystem transformiert.
- Generell setzen wir hier orthonormierte (kartesische) Koordinatensysteme voraus.
- Allgemeine Vorgehensweise bei der Koordinatentransformation:
 - (1) Bildschirm oder Ausgabegerät mit einem Koordinatensystem versehen
 - (2) Objekt mit einem Koordinatensystem versehen.
 - (3) Objekt- und Objektkoordinatensystem mittels Parallel- oder Zentralprojektion in Bildebene abbilden (IR3-Transformation).

- (4) Das Koordinatensystem des Bildschirms stimmt nun in der Regel nicht mit dem Bild des Objektsystems überein. Die notwendige „Anpassung“ erfolgt über eine Koordinatentransformation (\mathbb{R}^2 -Transformation).
- Gegeben seien im Folgenden die beiden Koordinatensysteme S (z. B. Gerätesystem) und S' (z. B. Objektsystem) durch $S:(0; x_1, x_2)$ und $S':(0'; x_1', x_2')$.

Translation (Verschiebung)

- Die einfachste Transformation zwischen dem System S' und S ist eine translatorische. Dabei wird vorausgesetzt, dass die beiden (gerichteten) Koordinatenachsen jeweils zueinander parallel sind.

Translation (Fortsetzung)

- Es gilt für den Punkt P:

$$p_1 = t_1 + p_1'$$

$$p_2 = t_2 + p_2'$$

wobei (t_1, t_2) die Koordinaten des Ursprungs von S' im System S sind. Der Punkt P hat also in S' die Koordinaten (p_1', p_2') und in S die Koordinaten $(t_1 + p_1', t_2 + p_2')$.

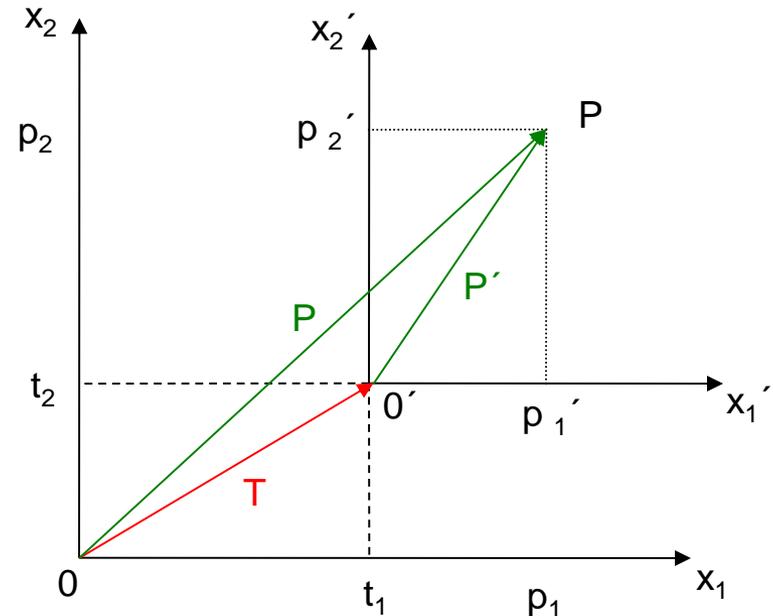
-

In Vektor-Schreibweise:

kurz:

$$\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} + \begin{pmatrix} p_1' \\ p_2' \end{pmatrix}$$

$$P = T + P'$$



Rotation (Drehung)

- Wir betrachten die Drehung eines Systems S' gegen das System S um den Winkel φ um den gemeinsamen Ursprung $0=0'$:

- Es ist $l/p_2' = \sin\varphi$ und $L/p_1' = \cos\varphi$

also $p_1 = L - l = p_1' \cdot \cos\varphi - p_2' \cdot \sin\varphi$

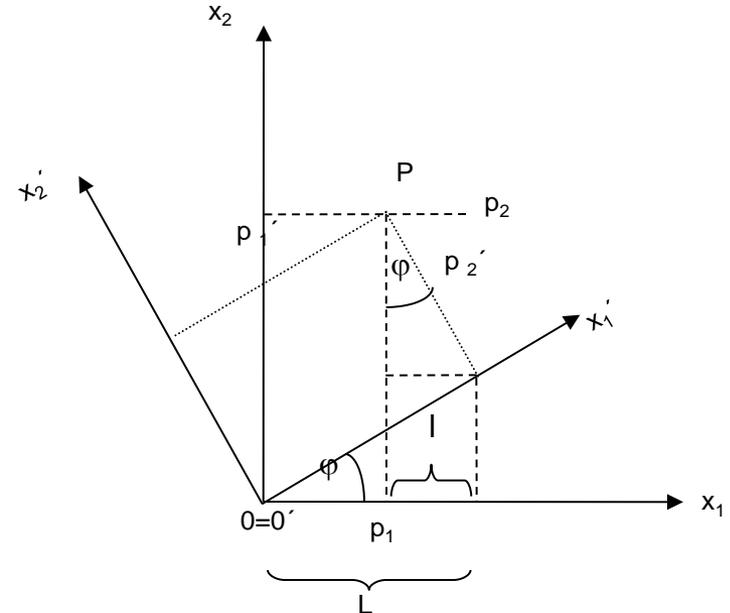
und (analog) $p_2 = p_1' \cdot \sin\varphi + p_2' \cdot \cos\varphi$

- In Vektor-Matrix-Schreibweise: $P = R \cdot P'$

- mit der (orthonormalen) Rotationsmatrix:

$$R = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$$

Bem.: R ist orthonormal gdw. $R^{-1} = R^T$



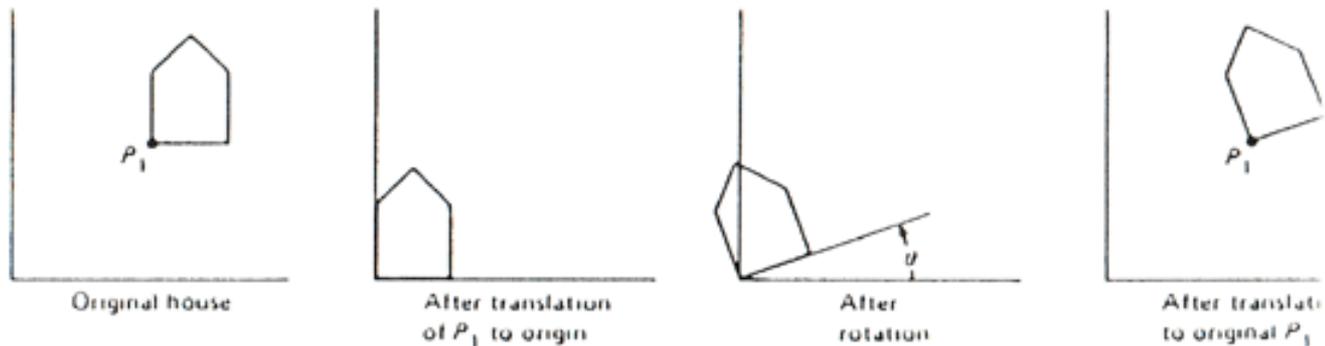
Rotation (Fortsetzung)

- Bemerkung: Sichtweisen von $P = R \cdot P'$
 - a) R transformiert die Koordinatendarstellung (p_1', p_2') des Punktes bezüglich des Systems S' in die Koordinatendarstellung (p_1, p_2) bezüglich des Systems S.
entspricht: (globales) Koordinatensystem S; „Punkt wird gedreht“
auf die Koordinaten (p_1', p_2') des Punktes wirkt die Matrix R.
 - b) (Lokales) Koordinatensystem S wird mit Matrix R (Rotation im mathematisch positiven Sinn um Winkel φ) transformiert; es resultiert ein (lokales) Koordinatensystem S' .
Der Punkt wird dann bezüglich S' (also mit den Koordinaten (p_1', p_2')) definiert. „Koordinatensystem wird gedreht“

Bem.: Gleiche Überlegungen gelten bei anderen Transformationen ebenso.

Rotation (Fortsetzung)

- Bei der Rotation um einen beliebigen Punkt P_1 müssen noch zwei Translationen hinzugenommen werden:
 - (1) Translation von P_1 in den Ursprung
 - (2) Rotation um den Ursprung
 - (3) „Rücktranslation“ von P_1 in die ursprüngliche Position.



Rotation (Fortsetzung)

Bemerkung:

- Die Matrizenmultiplikation ist i. a. nicht kommutativ, d. h. $A \cdot B \neq B \cdot A$
- Bei hintereinander geschalteten Rotationen muss also darauf geachtet werden, dass die Reihenfolge der Matrizen der Reihenfolge der Rotationen entspricht.

Skalierung (Scaling, Größenänderung)

- Soll das System S' „vergrößert“ oder „verkleinert“ werden, so muss eine Skalierung durchgeführt werden:

$$p_1 = \lambda_1 \cdot p'_1$$

$$p_2 = \lambda_2 \cdot p'_2$$

- In Vektor-Matrix-Schreibweise: $P = \tilde{S} \cdot P'$

- mit der Skalierungsmatrix:
$$\tilde{S} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Scherung (Shear)

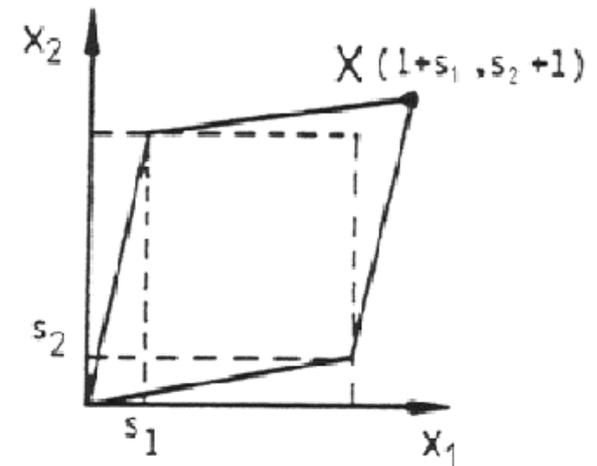
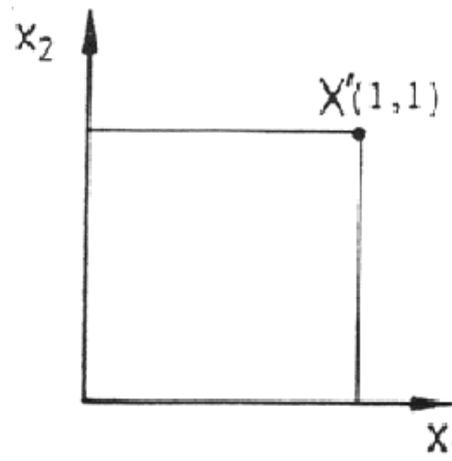
- Eine Scherung ergibt sich, wenn Abhängigkeiten folgender Form bestehen:

$$p_1 = p'_1 + s_1 \cdot p'_2$$

$$p_2 = s_2 \cdot p'_1 + p'_2$$

- In Vektor-Matrix-Schreibweise:

$$P = S \cdot P'$$



- mit der Scherungsmatrix:
$$S = \begin{pmatrix} 1 & s_1 \\ s_2 & 1 \end{pmatrix}$$

Einschub: Affine Transformationen

- Affine Transformationen lassen sich als Kombination einer linearen Abbildung und einer Translation schreiben:

$$P = AP' + T$$

- Die bisher genannten Transformationen (Translation, Rotation, Skalierung, Scherung) sind Beispiele affiner Transformationen.

Affine Invarianz von Teilungsverhältnissen:

- Für eine affine Transformation F und die Punkte P und Q gilt immer:

$$F(\lambda P + (1-\lambda)Q) = \lambda F(P) + (1-\lambda)F(Q) \quad \text{für } 0 \leq \lambda \leq 1$$

Einschub: Affine Transformationen (Frts)

- Diese Beziehung zeigt, dass das Bild einer Strecke (Strecke von Q nach P) unter einer affinen Abbildung F wieder eine Strecke ist, und dass Teilungsverhältnisse $\lambda : (1 - \lambda)$ unter F invariant bleiben.
- Es genügt also, die Endpunkte Q und P auf der Strecke abzubilden. Zwischenpunkte erhält man durch Interpolation von $F(Q)$ und $F(P)$.
- Man beachte, dass unter affinen Abbildungen parallele Linien parallel bleiben.

Einschub: Affine Transformationen (Frts)

weitere affine Transformationen:

- Reflektion an der Gerade $y=x$,

$$F = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

- Reflektion an der Gerade $y=-x$,

$$F = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

- Reflektion an der x-Achse,

$$F = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

- Reflektion an der y-Achse,

$$F = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- Reflektion am Ursprung,

$$F = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

Homogene Koordinaten

- Homogene Koordinaten entstammen der projektiven Geometrie, an dieser Stelle soll jedoch eine andere Motivation herhalten
- Die Hintereinanderschaltung von Rotation, Translation und Skalierung führt auf die Abbildungsgleichung

$$P = \tilde{S} \cdot (T + R \cdot P')$$

- Müssen jedoch mehrere solcher Transformationen hintereinander ausgeführt werden, so stört die Addition in der obigen Gleichung.
- Da heutige Computergraphikhardware insbesondere auch Matrixmultiplikationen unterstützt, ist es günstig, Transformationen ausschließlich mittels Matrixmultiplikationen auszuführen, also:

$$P = M_n \cdot \dots \cdot M_3 \cdot M_2 \cdot M_1 \cdot P'$$

Homogene Koordinaten (Fortsetzung)

- Dies erreicht man durch folgenden Übergang auf die nächst höhere Dimension:
- Das Tripel (x,y,w) ($w \neq 0$) stellt die homogenen Koordinaten des Punktes $(x/w, y/w) \in \mathbb{R}^2$ dar.
- Da es unendlich viele solcher Darstellungen desselben Punktes gibt, verwendet man die so genannte Standarddarstellung mit $w=1$.
- Also besitzt ein Punkt $P=(x,y) \in \mathbb{R}^2$ als homogene Koordinaten $(x,y,1)$.

Bem.: Für Punkte im \mathbb{R}^3 gilt eine analoge Konstruktion.

Homogene Koordinaten (Fortsetzung)

- Dies erlaubt nun eine „neue“ Darstellung unserer Transformationen:
- Translation eines Punktes (x', y') um den Vektor (t_1, t_2) :

$$\begin{pmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x' + t_1 \\ y' + t_2 \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- Rotation eines Punktes (x', y') um den Winkel φ :

$$\begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x' \cdot \cos \varphi - y' \cdot \sin \varphi \\ x' \cdot \sin \varphi + y' \cdot \cos \varphi \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Homogene Koordinaten (Fortsetzung)

- Skalierung eines Punktes (x', y') mit den Faktoren λ_1 und λ_2 :

$$\begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \lambda_1 \cdot x' \\ \lambda_2 \cdot y' \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- Rotation eines Punktes (x', y') um einen Punkt P_1 um den Winkel φ :

$$\begin{pmatrix} 1 & 0 & P_{1x} \\ 0 & 1 & P_{1y} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -P_{1x} \\ 0 & 1 & -P_{1y} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Translation

- Die Verschiebung eines Punktes (x', y', z') um den Translationsvektor (t_x, t_y, t_z) ergibt den Punkt (x, y, z) mit

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{T(t_x, t_y, t_z)} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} x' + t_x \\ y' + t_y \\ z' + t_z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

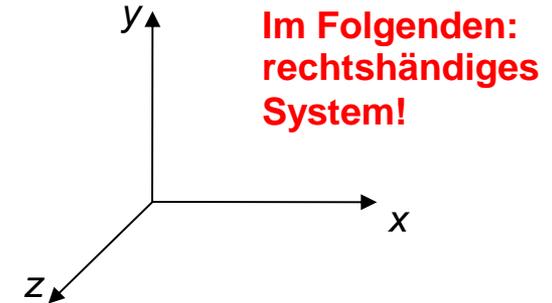
Skalierung

- Eine Skalierung mit den Faktoren s_1 , s_2 und s_3 für die drei Achsenrichtungen hat die Gestalt:

$$\begin{pmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} s_1 \cdot x' \\ s_2 \cdot y' \\ s_3 \cdot z' \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

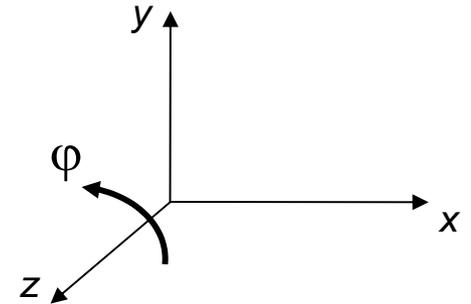
Rotation

- Alle Rotationen erfolgen im mathematisch positiven Sinn (d. h. gegen den Uhrzeigersinn).
- Der Betrachter „sitzt“ dabei auf der positiven Rotationsachse und schaut in Richtung Ursprung des Koordinatensystems.
- Wir betrachten zuerst die Rotationen um einzelne Koordinatenachsen um den Winkel φ .
⇒ Transformationsmatrizen $R_x(\varphi)$, $R_y(\varphi)$, $R_z(\varphi)$
- Wir verwenden die Sichtweise „**globales festes Koordinatensystem; Punkt wird transformiert (gedreht)**“.



Rotation um die z-Achse

- Die Rotation eines Punktes (x', y', z') um die z-Achse um den Winkel φ ergibt den Punkt (x, y, z) mit



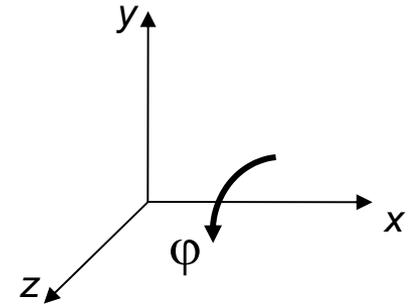
$$\underbrace{\begin{pmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{R_z(\varphi)} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} x' \cdot \cos\varphi - y' \cdot \sin\varphi \\ x' \cdot \sin\varphi + y' \cdot \cos\varphi \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- Man beachte, dass eine Drehung um den Winkel φ um die z-Achse dem 2D-Fall entspricht, wobei die z-Koordinate konstant bleibt.

Rotation um die x-Achse

- Die Rotation eines Punktes (x', y', z') um die x-Achse um den Winkel φ ergibt den Punkt (x, y, z) mit

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi & 0 \\ 0 & \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{R_x(\varphi)} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \cdot \cos \varphi - z' \cdot \sin \varphi \\ y' \cdot \sin \varphi + z' \cdot \cos \varphi \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

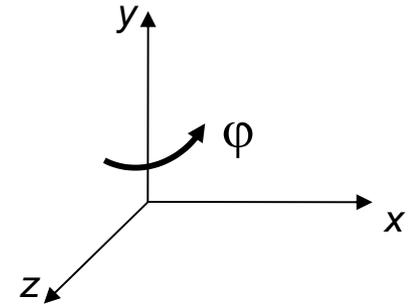


- Hier bleibt die x-Komponente der Koordinaten konstant, so dass die Einschränkung auf eine (y, z) -Ebene dem 2D-Fall entspricht.

Rotation um die y-Achse

- Die Rotation eines Punktes (x', y', z') um die y-Achse um den Winkel φ ergibt den Punkt (x, y, z) mit

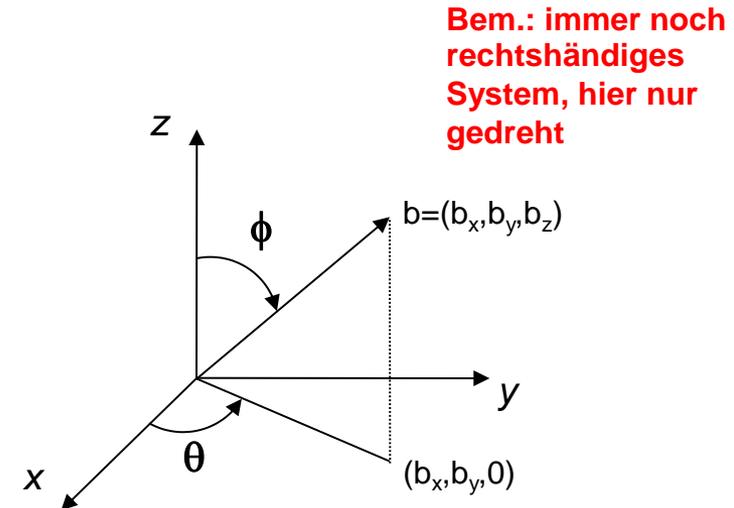
$$\underbrace{\begin{pmatrix} \cos \varphi & 0 & \sin \varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \varphi & 0 & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{R_y(\varphi)} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} x' \cdot \cos \varphi + z' \cdot \sin \varphi \\ y' \\ -x' \cdot \sin \varphi + z' \cdot \cos \varphi \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



- Hier bleibt die y-Komponente der Koordinaten konstant, so dass die Einschränkung auf eine (z, x) -Ebene dem 2D-Fall entspricht.

Rotation um eine beliebige Achse

- Jede Rotation um eine beliebige Achse kann aus Rotationen um die einzelnen Koordinatenachsen zusammengesetzt werden (\Rightarrow Euler).
- Wir entwickeln nun die Rotation $R_G(\alpha)$ für die Drehung eines Punktes P um eine beliebig orientierte Achse G im Raum um einen Winkel α .
- Zuerst betrachten wir den Sonderfall, bei dem die Drehachse G durch den Ursprung geht und von einem Vektor b ($b=(b_x, b_y, b_z)$ mit $\|b\|=1$) generiert wird, also $G : \lambda \cdot b \quad (\lambda \in \mathbb{R})$.



$$b_x = \sin \varphi \cos \theta$$

$$b_y = \sin \varphi \sin \theta$$

$$b_z = \cos \varphi$$

Rotation um eine beliebige Achse (Frts)

- Gesucht sind nun die Koordinaten eines Punktes P nach einer Drehung um die Achse G um den Winkel α .
- Vorgehensweise:
Der Punkt P wird so transformiert, dass die Drehachse mit der z -Achse zusammenfällt. Anschließend wird für die Drehung um α die Rotationsmatrix $R_z(\alpha)$ verwendet. Hinterher werden die „Hilfstransformationen“ wieder rückgängig gemacht.

(Bem.: Ist G mit der z -Achse identisch, so entfallen die Hilfstransformationen)

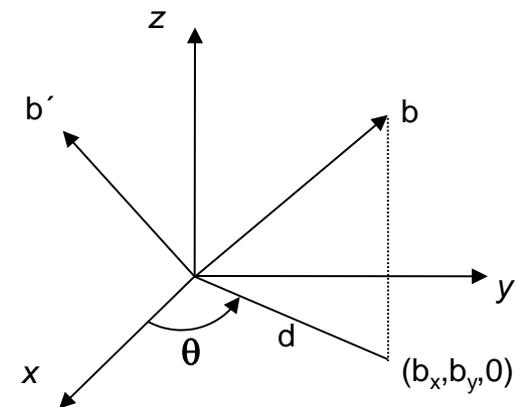
- Man geht in mehreren Teilschritten vor.

Rotation um eine beliebige Achse (Frts)

Schritt 1:

- Zunächst dreht man so, dass der Vektor b in die (z,x) -Ebene gedreht wird (b'). Aus P entsteht dabei P' mit $P' = R_z(-\theta) P$, wobei

$$R_z(-\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \frac{1}{d} \begin{pmatrix} b_x & b_y & 0 & 0 \\ -b_y & b_x & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 0 & d \end{pmatrix}$$



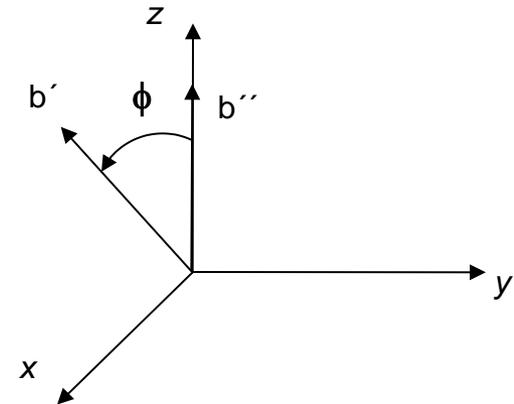
$$d^2 = b_x^2 + b_y^2$$

Rotation um eine beliebige Achse (Frts)

Schritt 2:

- Dann dreht man so, dass der Vektor b' mit der z-Achse zusammenfällt (b''). Aus P' entsteht dabei P'' mit $P'' = R_y(-\phi) P'$, wobei

$$R_y(-\phi) = \begin{pmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} b_z & 0 & -d & 0 \\ 0 & 1 & 0 & 0 \\ d & 0 & b_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

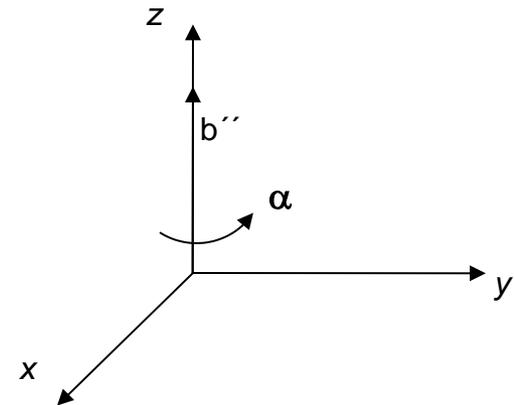


Rotation um eine beliebige Achse (Frts)

Schritt 3:

- Man dreht nun um den Winkel α um die z-Achse. Aus P'' entsteht dabei P''' mit $P''' = R_z(\alpha) P''$, wobei

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



Rotation um eine beliebige Achse (Frts)

Schritte 4 und 5:

- Zum Schluss werden die Rotationen aus den Schritten 1 und 2 in umgekehrter Reihenfolge rückgängig gemacht. Aus P''' entsteht dabei der gewünschte „gedrehte“ Punkt Q des ursprünglichen Punktes P mittels $Q = R_z(\theta) R_y(\phi) P'''$, wobei

$$R_y(\phi) = \begin{pmatrix} b_z & 0 & d & 0 \\ 0 & 1 & 0 & 0 \\ -d & 0 & b_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\theta) = \frac{1}{d} \begin{pmatrix} b_x & -b_y & 0 & 0 \\ b_y & b_x & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 0 & d \end{pmatrix}$$

Rotation um eine beliebige Achse (Frts)

Ergebnis:

- Die Gesamttransformation lässt sich in einem Schritt durch die Verknüpfung aller Transformationen realisieren:

$$M_b(\alpha) = R_z(\theta) R_y(\phi) R_z(\alpha) R_y(-\phi) R_z(-\theta)$$

Allgemeiner Fall:

- Ist die Drehachse eine allgemeine Gerade

$$G: a + \lambda \cdot b \quad (\lambda \in \mathbb{R}, \|b\|=1, a=(a_x, a_y, a_z))$$

- so ist vor Schritt 1 und nach Schritt 5 eine entsprechende Translation einzuschieben, also

$$M_G(\alpha) = T(a_x, a_y, a_z) R_z(\theta) R_y(\phi) R_z(\alpha) R_y(-\phi) R_z(-\theta) T(-a_x, -a_y, -a_z)$$