

# ECG – Mathematische Grundlagen der Geometrischen Modellierung

Prof. Dr. Stefan Gumhold

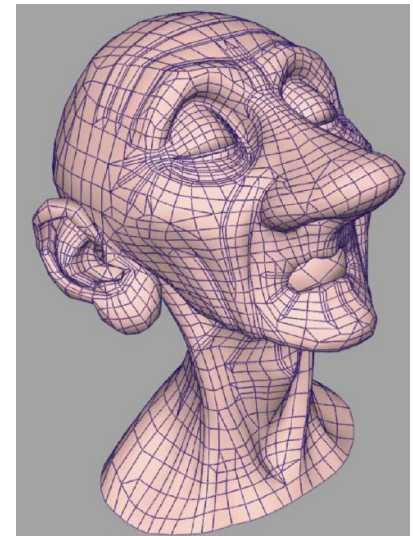




- ◆ Einführung
- ◆ Rechenoperationen mit Vektoren
- ◆ Repräsentation einer Ebene
- ◆ Baryzentrische Koordinaten
- ◆ Mathematische Modellierung

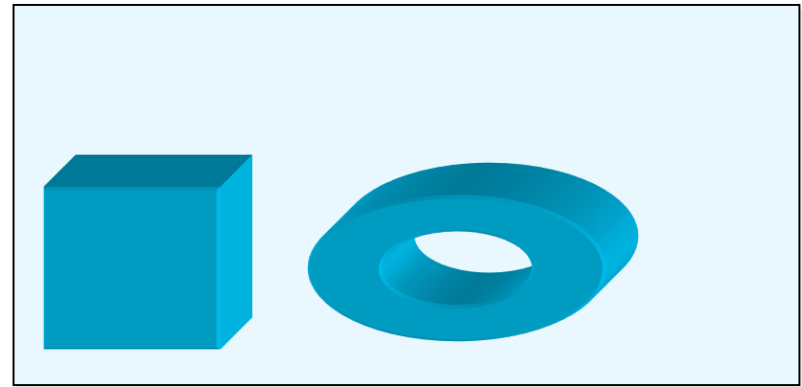
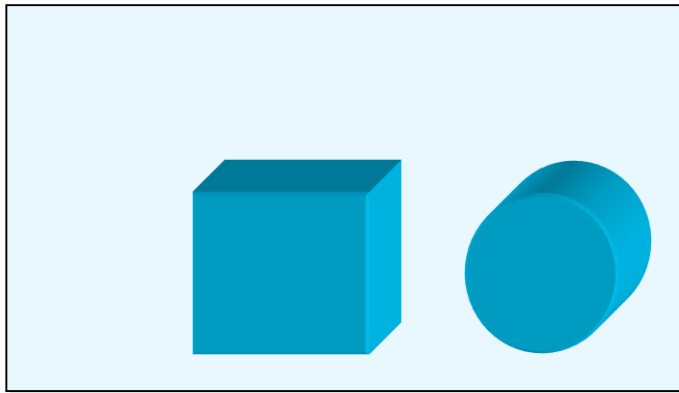
Geometrisches Modellieren ist:

- rechnergestütztes Entwerfen und Manipulieren von geometrischen Objekten und ihren Lagebeziehungen (CAD – *Computer Aided Design*)
- Alternativer Ansatz: 3D Scanning mit automatischer Nachverarbeitung der Rohdaten (meist Punktwolken)
- Grundlage für:
  - die Berechnung geometrischer Eigenschaften
  - die graphische Darstellung der Objekte
  - die Berechnung des physikalischen Verhaltens der Objekte (physikalischen Kenngrößen müssen modelliert / gemessen werden)



Structured Light 3D-Scanner  
(taken from David Laserscanner)

- ◆ Geometrische Objekte sind Objekte der realen oder theoretischen Welt, die durch ihre geometrischen und topologischen Eigenschaften beschrieben sind:
  - ◆ Geometrische Eigenschaften beschreiben die präzise Form eines Objekts und die konkrete Lage in einem Bezugssystem.
  - ◆ Topologische Eigenschaften definieren die fundamentale Objektstruktur, die bei kontinuierlichen Transformationen invariant ist.



verschiedene Körper derselben (links) bzw. unterschiedlichen Topologie (rechts)

# Einführung

## Topologie – Motivation

Typische Beispiele für topologische Fragestellungen sind:

- ◆ Gibt es eine Verbindung zwischen zwei Punkten auf einem Objekt?
- ◆ Umschließt ein Objekt einen geschlossenen Bereich?
- ◆ Kann ich ein Objekt stetig in ein anderes überführen?



# Einführung Topologie: ein Beispiel



Mein Finger kann  
nicht wieder heraus!

Hier gibt es einen  
Ausgang!

# Einführung

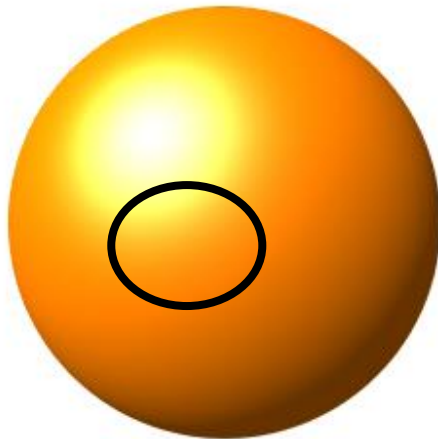
## Poincaré Vermutung



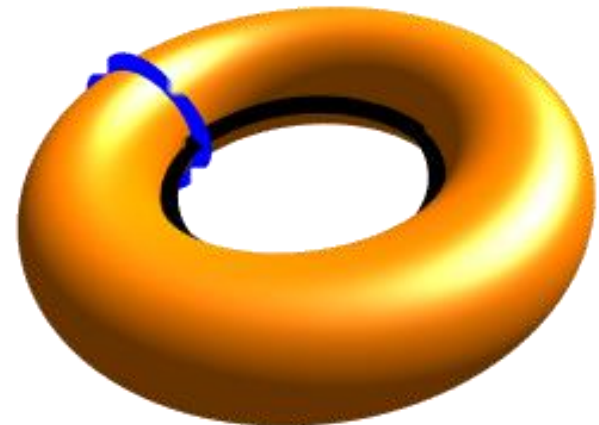
- ◆ Clay Mathematics Institute – The 7 Millennium Price Problems

- ◆ Poincaré Vermutung

*Wenn sich alle geschlossenen Kurven auf einer zusammenhängenden, kompakten und geschlossenen 3D-Fläche im 4D-Raum stetig zu einem Punkt schrumpfen lassen, kann man die Fläche stetig in eine 3-Kugel im 4D-Raum deformieren.*



Beispiel einer 2-Kugel im 3D mit einer geschlossenen Kurve (schwarz), die sich stetig zu einem Punkt schrumpfen lässt



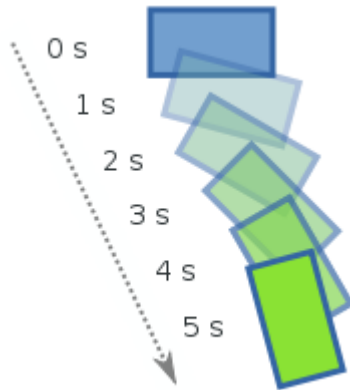
Beispiel für einen Torus im 3D mit zwei geschlossenen Kurven (blau und schwarz), die sich nicht stetig zu einem Punkt schrumpfen lassen

- ◆ 2002-3 bewiesen von Grigoriy Perelman

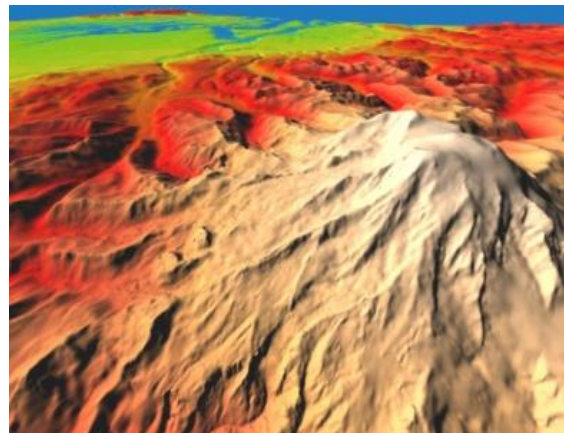


## ◆ Klassifikation von Geometrischen Modellen

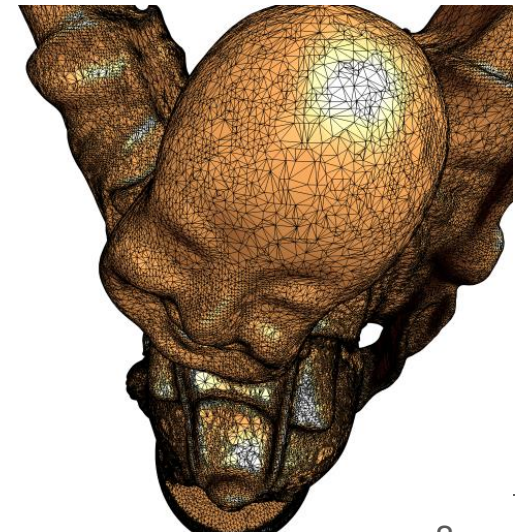
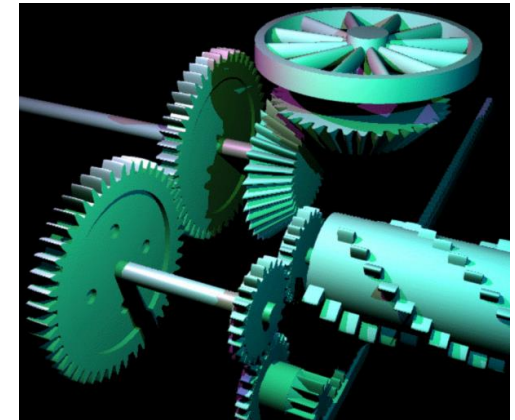
**2D**



**2½D**



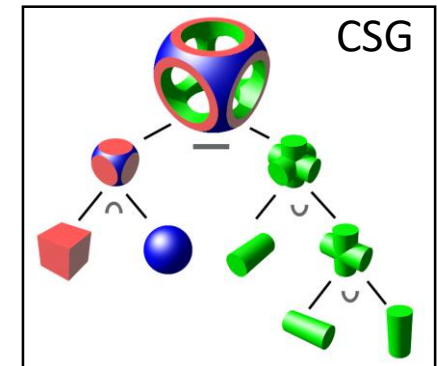
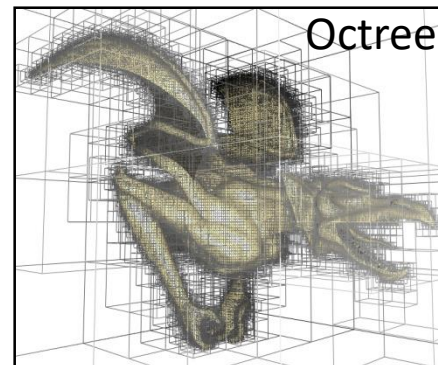
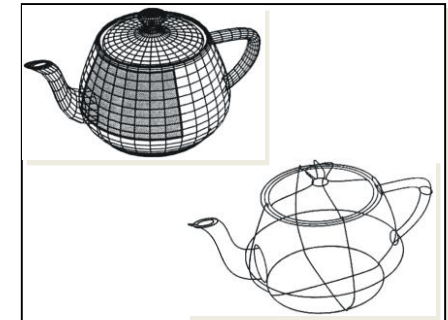
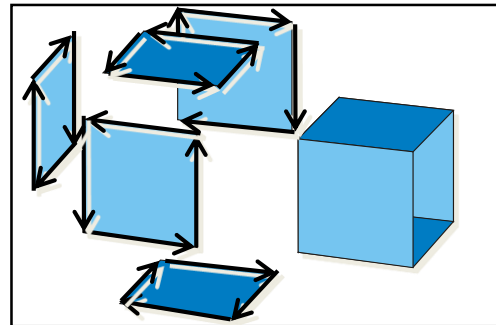
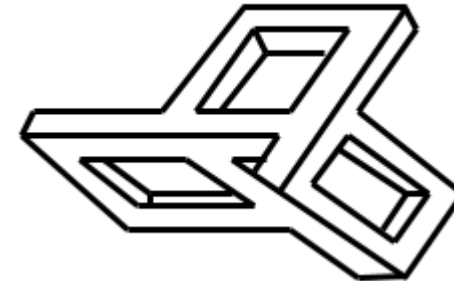
**3D**





## Klassifikation der Modellieransätze

- ◆ Drahtgitter (wireframe)
  - ◆ Graphen von Knoten und Kanten (auch Kurven möglich)
  - ◆ **Vorsicht:** Beim Entwurf im 2D können unmögliche Modell entstehen
- ◆ Flächenmodell (boundary representation)
  - ◆ Erweiterung um Flächen, die das Innere von Kantenschleifen füllen
- ◆ Körpermodell (solid rep.)
  - ◆ Beschreibt Körper durch
    - ◆ Mengenoperationen über Grundkörper
    - ◆ Markieren von Zellen in möglicherweise hierarchischer Gitterstruktur

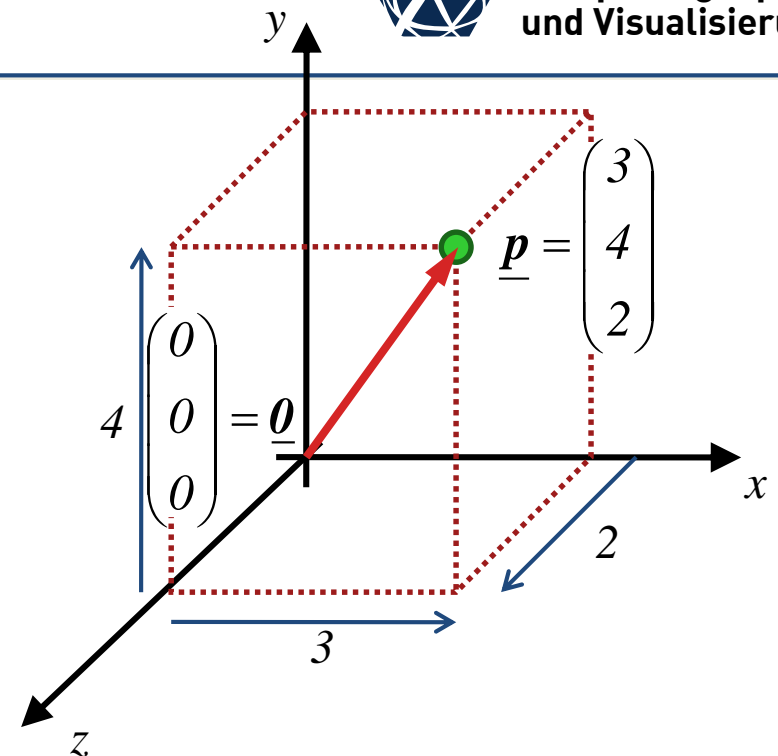


# Einführung

## Punkte und Vektoren



- Punkte, Normalen und Farben werden mathematisch durch Vektoren eines Vektorraumes repräsentiert
- In der Computergraphik finden sich 2D, 3D und 4D Vektoren
- Bei den 4D Vektoren hat die vierte Komponente die Funktion eines gemeinsamen Nenners für die drei ersten Komponenten
- Oft werden die Konzepte Punkt, Vektor, Richtung und Normale unterschieden, weil diese unterschiedliche Transformationseigenschaften haben.



Ortsvektor:  $\vec{p} = \underline{p} - \underline{0}$

von Punkt zu Punkt:  $\underline{q} = \underline{p} + \vec{v} = \begin{pmatrix} p_x + v_x \\ p_y + v_y \\ p_z + v_z \end{pmatrix}$



Differenzvektor:  $\vec{v} = \underline{q} - \underline{p}$



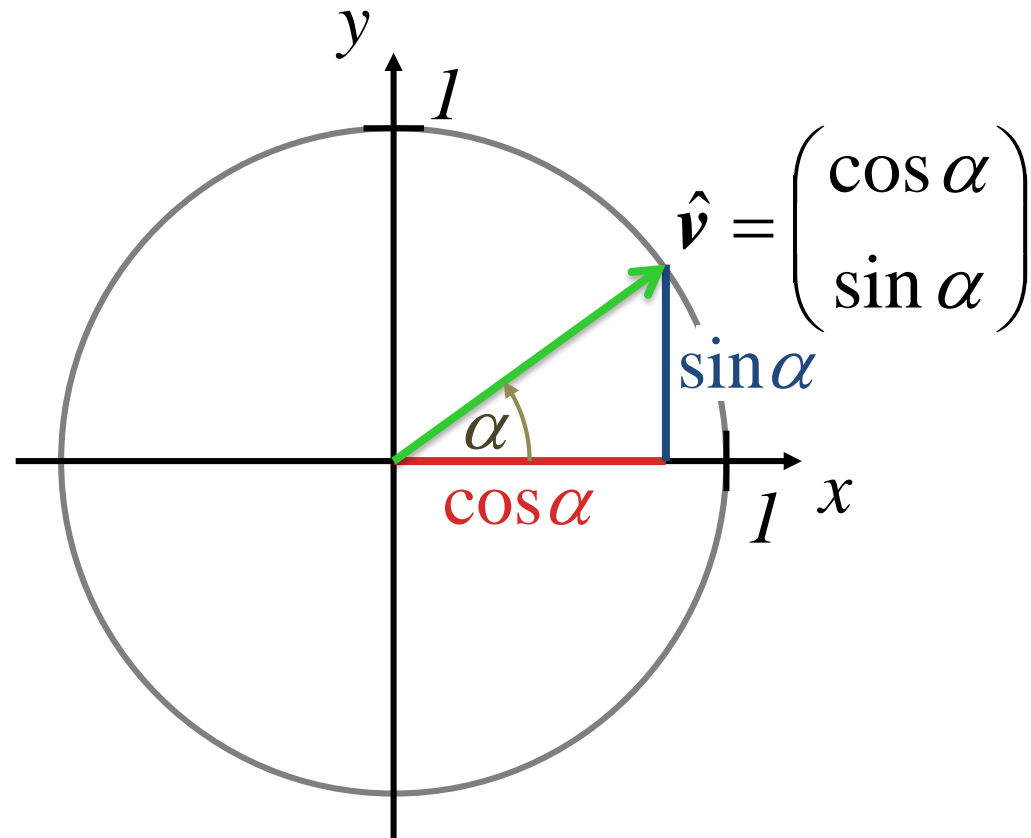
◆ Mengen	$\dots A, B, C, \mathbf{N}, \mathbf{R}, \mathbf{E}^2 \dots$
◆ skalare Größen	$\dots a, b, x, y, z, \alpha, \beta, \dots$
◆ Punkte	$\dots \underline{\mathbf{x}}, \underline{\mathbf{y}}, \underline{\mathbf{p}}, \dots$
◆ Vektoren	$\dots \vec{\mathbf{v}}, \vec{\mathbf{d}}, \dots$
◆ RGB[ $\alpha$ ]-Farben	$\dots \ddot{\mathbf{c}}, \ddot{\mathbf{L}}, \dots$
◆ normierte Vektoren	$\dots \hat{\mathbf{n}}, \hat{\mathbf{l}}, \dots$
◆ Matrizen	$\dots \mathbf{A}, \mathbf{M}, \mathbf{T}, \dots$
◆ homogene Vektoren	$\dots \tilde{\mathbf{p}}, \tilde{\mathbf{v}}, \dots$
◆ homogene Matrizen	$\dots \tilde{\mathbf{A}}, \tilde{\mathbf{H}}, \dots$

# Rechenoperationen mit Vektoren

## Wiederholung – Satz des Pythagoras



- Nach dem Satz des Pythagoras am Beispiel des Einheitskreises gilt:  
$$\sin^2 \alpha + \cos^2 \alpha = 1$$
- Mit Hilfe der im folgenden eingeführten Vektorprodukte kann man die Winkelfunktionen auch für 3D Vektoren berechnen



# Rechenoperationen mit Vektoren

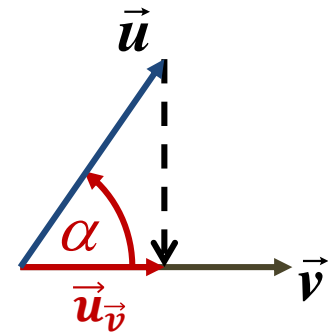
## Skalarprodukt / Punktprodukt



◆ Skalarprodukt:  $\vec{u} \cdot \vec{v} = \vec{u}^T \vec{v} = \langle \vec{u}, \vec{v} \rangle = u_x v_x + u_y v_y + u_z v_z$

$$= \begin{pmatrix} u_x & u_y & u_z \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \left\langle \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}, \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \right\rangle$$

- ◆ Symmetrie  $\langle \vec{u}, \vec{v} \rangle = \langle \vec{v}, \vec{u} \rangle$
- ◆ quadr. Länge  $\|\vec{v}\|^2 = v_x^2 + v_y^2 + v_z^2 = \vec{v} \cdot \vec{v}$
- ◆ Def. Kosinus  $\vec{u} \cdot \vec{v} = \cos \alpha \cdot \|\vec{u}\| \cdot \|\vec{v}\|$
- ◆ Normierung  $\hat{v} = \frac{1}{\|\vec{v}\|} \vec{v}$
- ◆ orthogonal  $\vec{u} \perp \vec{v} \Rightarrow \vec{u} \cdot \vec{v} = 0$
- ◆ Projektion  $\vec{u}_{\vec{v}} = (\vec{u} \cdot \hat{v}) \hat{v} = \frac{\vec{u} \cdot \vec{v}}{\vec{v} \cdot \vec{v}} \vec{v}$



### ◆ Interpretation Matrixmultiplikation:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} \vec{a}_{1.} \\ \vec{a}_{2.} \end{pmatrix} \begin{pmatrix} \vec{b}_{.1} & \vec{b}_{.2} \end{pmatrix} = \begin{pmatrix} \vec{a}_{1.} \cdot \vec{b}_{.1} & \vec{a}_{1.} \cdot \vec{b}_{.2} \\ \vec{a}_{2.} \cdot \vec{b}_{.1} & \vec{a}_{2.} \cdot \vec{b}_{.2} \end{pmatrix}$$

# Rechenoperationen mit Vektoren

## Kreuzprodukt / Vektorprodukt



### ◆ Kreuzprodukt:

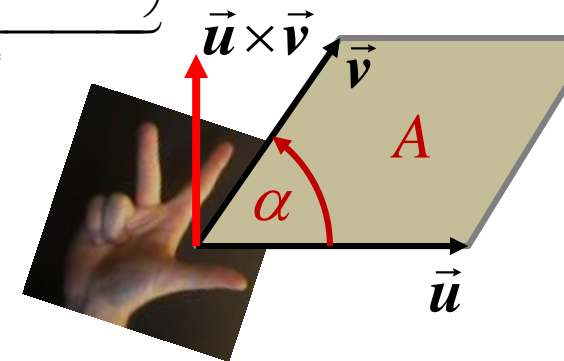
- ◆ Antisymmetrie
- ◆ rechte Handregel
- ◆ Sinusgesetz
- ◆ Flächeninhalt
- ◆ senkrecht

$$\vec{u} \times \vec{v} = \begin{pmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix}}_{\vec{u}^*} \vec{v} = -\vec{v} \times \vec{u}$$

$$\|\vec{u} \times \vec{v}\| = \sin \alpha \cdot \|\vec{u}\| \cdot \|\vec{v}\|$$

$$\|\vec{u} \times \vec{v}\| = A$$

$$\vec{u} \perp \vec{u} \times \vec{v} \perp \vec{v}$$



### ◆ Erweiterung zu orthonormalem Koordinatensystem

- ◆ geg.: ein normierter Vektor  $\hat{z}$
- ◆ ges.: Vektoren  $\hat{x}$  und  $\hat{y}$ , so dass  $(\hat{x}, \hat{y}, \hat{z})$  orthonormale Basis ist
- ◆ Algorithmus: wähle  $\vec{v}$ , das nicht parallel zu  $\hat{z}$  ist, aber wie?

$$\text{setze } \hat{x} \leftarrow \frac{\vec{v} \times \hat{z}}{\|\vec{v} \times \hat{z}\|} \text{ und } \hat{y} \leftarrow \hat{z} \times \hat{x}$$

$$\vec{v} \leftarrow \vec{0}$$

$$v_{\min \arg |z_k|} = 1$$

$k \in \{x, y, z\}$

### ◆ robuste Winkelberechnung: $\angle(\vec{u}, \vec{v}) = \text{atan2}(\langle \vec{u}, \vec{v} \rangle, \|\vec{u} \times \vec{v}\|)$

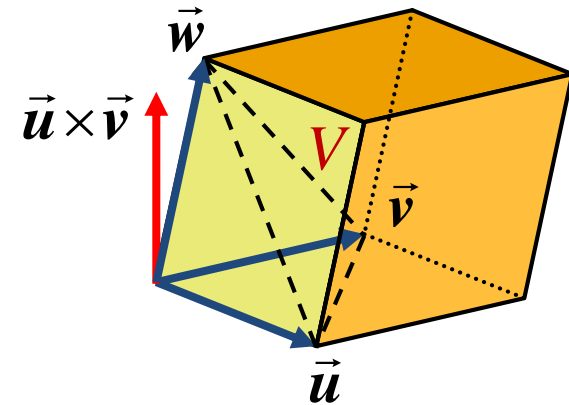
# Rechenoperationen mit Vektoren

## Spatprodukt

◆ Spatprodukt:  $[\vec{u}, \vec{v}, \vec{w}] = \langle \vec{u}, \vec{v} \times \vec{w} \rangle = \det(\vec{u} \quad \vec{v} \quad \vec{w})$

◆ Volumeninhalt  $V = |[\vec{u}, \vec{v}, \vec{w}]|$

◆ Tetraedervolumen  $V_{tet} = \frac{1}{6} V$



◆ Matrixinversion in 3D

◆ geg.:  $A = \begin{pmatrix} \vec{a}_1^T \\ \vec{a}_2^T \\ \vec{a}_3^T \end{pmatrix} \in \mathbf{R}^{3 \times 3}$

◆ ges.:  $B = \begin{pmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{pmatrix} \in \mathbf{R}^{3 \times 3} : AB = \mathbf{1}$ , d.h.  $\langle \vec{a}_i, \vec{b}_j \rangle = \delta_{ij} = \begin{cases} 1 \dots i=j \\ 0 \dots \text{sonst} \end{cases}$

◆ somit  $\vec{b}_i \perp \vec{a}_{j \neq i} \Rightarrow \vec{b}_i = \frac{\vec{a}_j \times \vec{a}_k}{[\vec{a}_i, \vec{a}_j, \vec{a}_k]}$  mit  $\{i, j, k\} \in \{\{1, 2, 3\}, \{2, 3, 1\}, \{3, 1, 2\}\}$

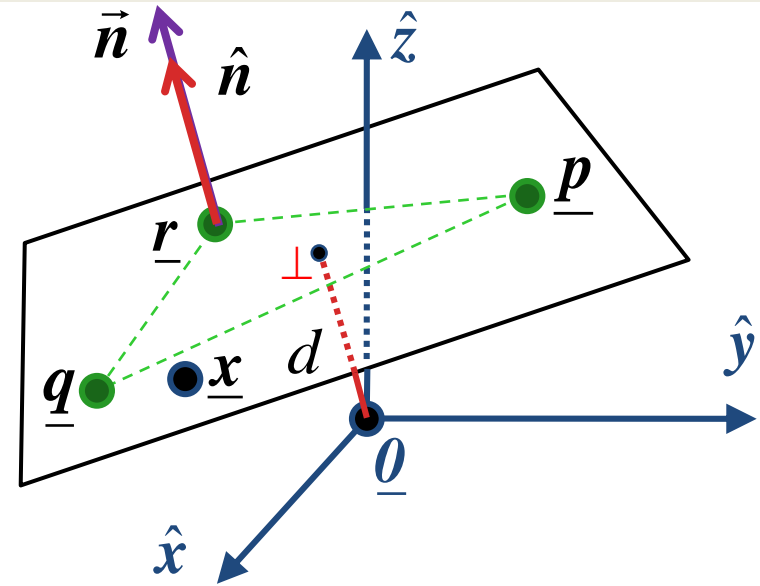


# Repräsentation einer Ebene

## Definitionen



- Ebenen werden in der Computergraphik sehr häufig eingesetzt, z.B. zur Definition eines Clipping Volumens, zur lokalen Approximation einer Oberfläche, ...
- Eine unendlich ausgedehnte Ebene kann unterschiedlich definiert werden:
  - aus drei Punkten im Raum durch alle Affinkombinationen
  - aus einem Fußpunkt  $\underline{r}$  und einer Normalenrichtung  $\vec{n}$
  - Hessesche Normalform: aus normierter Normalenrichtung  $\hat{n}$  und Abstand  $d$  zum Ursprung



$$\underline{x} = \sigma_1 \underline{r} + \sigma_2 \underline{p} + \sigma_3 \underline{q}, \quad \sum_i \sigma_i = 1$$

$$\langle \vec{n}, \underline{x} - \underline{r} \rangle = 0$$

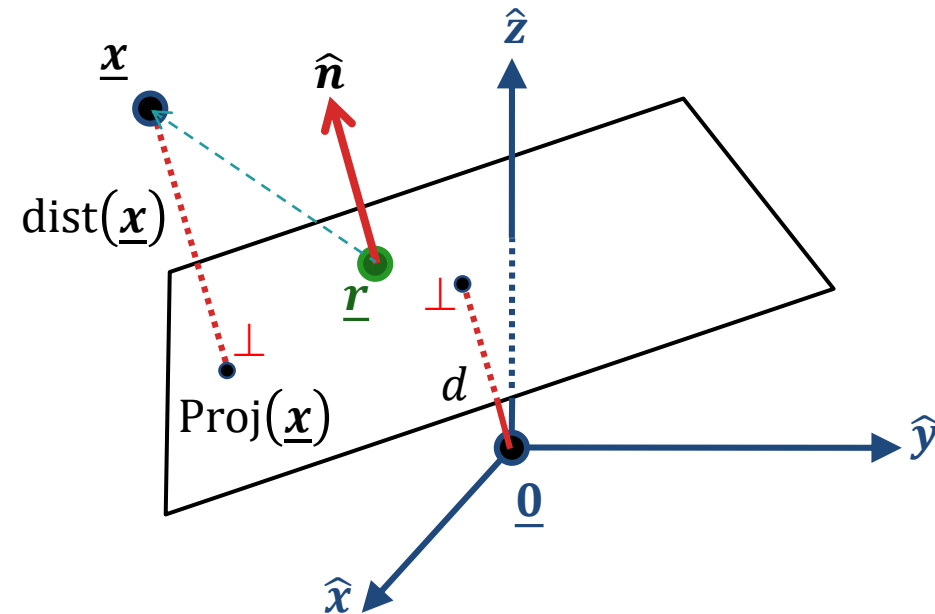
$$\langle \hat{n}, \underline{x} \rangle = d$$

# Repräsentation einer Ebene

## Abstandsberechnung

- Abstand eines Punktes zur Ebene berechnet sich aus
$$\text{dist}(\underline{x}) = \langle \hat{n}, \underline{x} \rangle - d$$
$$= \langle \hat{n}, \underline{x} - \underline{r} \rangle$$
- daraus ergibt sich
  - $\text{dist} = 0$  ... auf Ebene
  - $\text{dist} > 0$  ... auf Normale zugewandten Seite
  - $\text{dist} < 0$  ... auf Normale abgewandten Seite
- Projektion eines Punktes auf den nächsten Punkt der Ebene

$$\text{Proj}(\underline{x}) = \underline{x} - \text{dist}(\underline{x}) \cdot \hat{n}$$

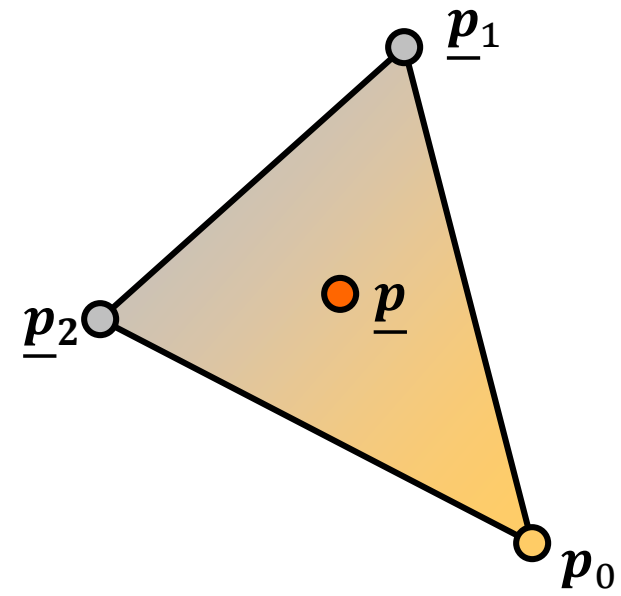


# Baryzentrische Koordinaten

## Motivation



- ❓ Wie testet man, ob ein Punkt im Innern eines Dreiecks liegt?
- ❓ Wie interpoliert man Farbe über ein Dreieck?
- 💡 Baryzentrische Koordinaten:



- ◆ Punkt  $\underline{p}$  im Dreieck  $(\underline{p}_0, \underline{p}_1, \underline{p}_2)$
- ◆  $\underline{p}$  hat jeweils eine Koordinate  $\sigma_0$ ,  $\sigma_1$  und  $\sigma_2$  pro Dreieckseckpunkt: mit  $\underline{p} = \sigma_0 \underline{p}_0 + \sigma_1 \underline{p}_1 + \sigma_2 \underline{p}_2$  und der Zusatzbedingung  $1 = \sigma_0 + \sigma_1 + \sigma_2$
- ◆ Für  $\underline{p} = \underline{p}_i$  gilt  $\sigma_i = 1$  und  $\sigma_{j \neq i} = 0$



### ◆ Berechnung aus Gleichungssystem

$$\begin{aligned}\underline{\mathbf{p}} &= \sigma_0 \underline{\mathbf{p}}_0 + \sigma_1 \underline{\mathbf{p}}_1 + \sigma_2 \underline{\mathbf{p}}_2 \\ l &= \sigma_0 + \sigma_1 + \sigma_2\end{aligned} \Rightarrow \begin{pmatrix} x \\ y \\ l \end{pmatrix} = \begin{pmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \sigma_0 \\ \sigma_1 \\ \sigma_2 \end{pmatrix}$$

### ◆ Interpolation von Attributen von den Ecken ins Dreiecksinnere

- ◆ berechne baryzentrische Koordinaten der Position
- ◆ übertrage Attribut durch lineare Interpolation
- ◆ Beispiel: Farbe  $\ddot{\mathbf{c}}_i$  an den Eckpunkten  $\underline{\mathbf{p}}_i$

$$\ddot{\mathbf{c}} = \sigma_0 \ddot{\mathbf{c}}_0 + \sigma_1 \ddot{\mathbf{c}}_1 + \sigma_2 \ddot{\mathbf{c}}_2$$

# Baryzentrische Koordinaten

## Weitere Interpretationen



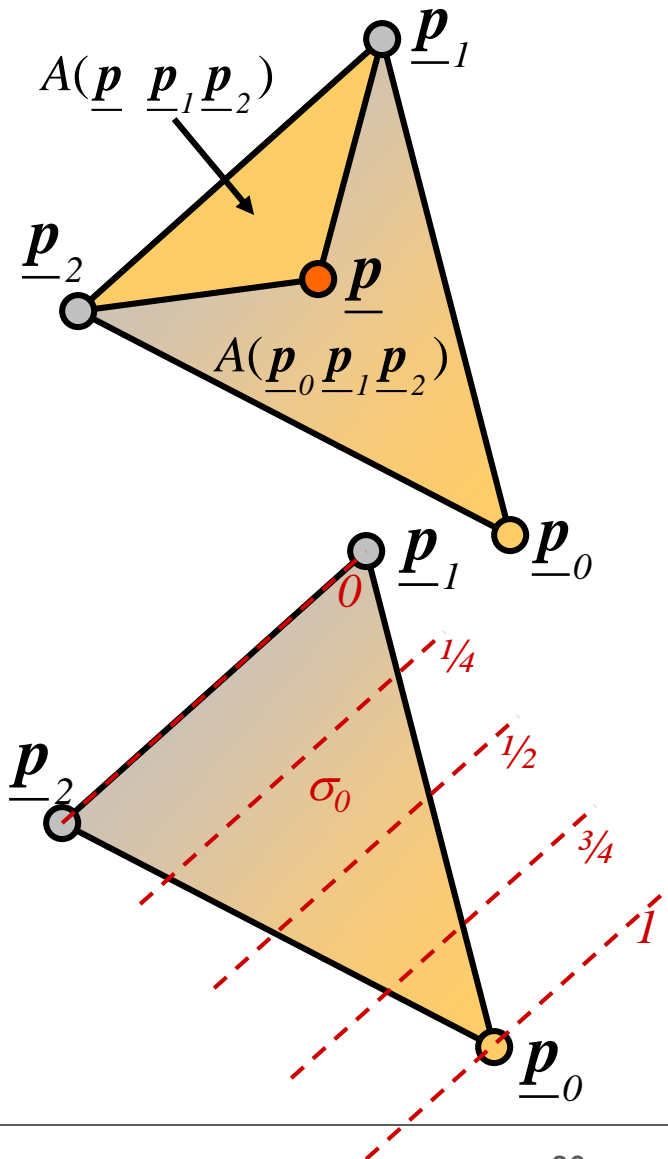
### Teilflächeninterpretation

- Sei  $A(\underline{p}_0 \underline{p}_1 \underline{p}_2)$  die Fläche des Dreiecks  $(\underline{p}_0 \underline{p}_1 \underline{p}_2)$
- die baryzentrischen Koordinaten verhalten sich wie die Flächen der abgewandten Dreiecke:

$$\sigma_0(\underline{p}) = \frac{A(\underline{p} \underline{p}_1 \underline{p}_2)}{A(\underline{p}_0 \underline{p}_1 \underline{p}_2)}$$

### Abstandsinterpretation

- die baryzentrischen Koordinaten geben den Abstand von der gegenüberliegenden Kante (normiert auf 1 bei Dreiecksecke)





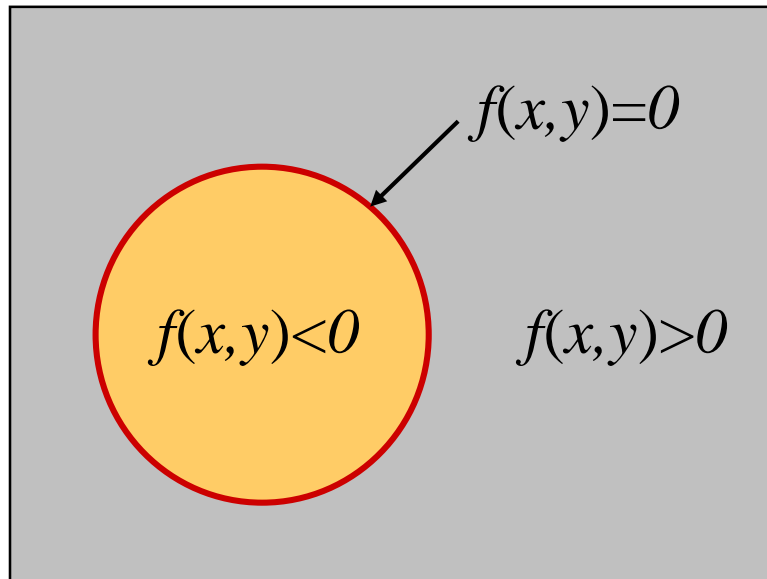
### ◆ Positionstests

- ◆ Punkt ist **Eckpunkt**, wenn ein  $\sigma_i$  eins ist und wenigstens ein anderes null
- ◆ Punkt ist **auf Kante**, wenn ein  $\sigma_i$  null ist und beide anderen größer ungleich null
- ◆ Punkt ist **im Dreieck**, wenn alle drei  $\sigma_i$  größer ungleich null sind
- ◆ Punkt ist **außen**, wenn ein  $\sigma_i$  kleiner ungleich null oder größer ungleich eins
- ◆ bei Rasterisierung ist inkrementelle Berechnung der baryzentrischen Koordinaten möglich

## ◆ mathematische Modellieransätze

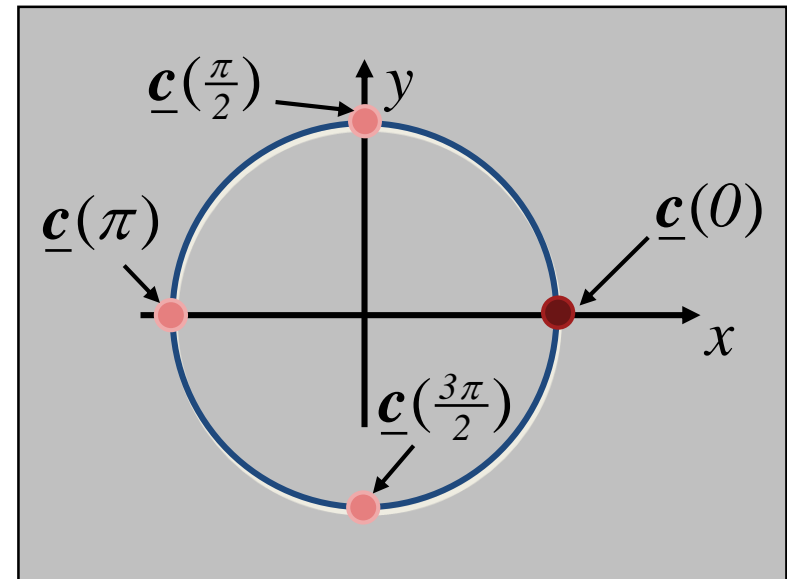
### Implizit

$$f(x,y)=x^2+y^2-r^2$$



### Parametrisch

$$\underline{c}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \cos t \\ \sin t \end{pmatrix}, t \in [0, 2\pi]$$



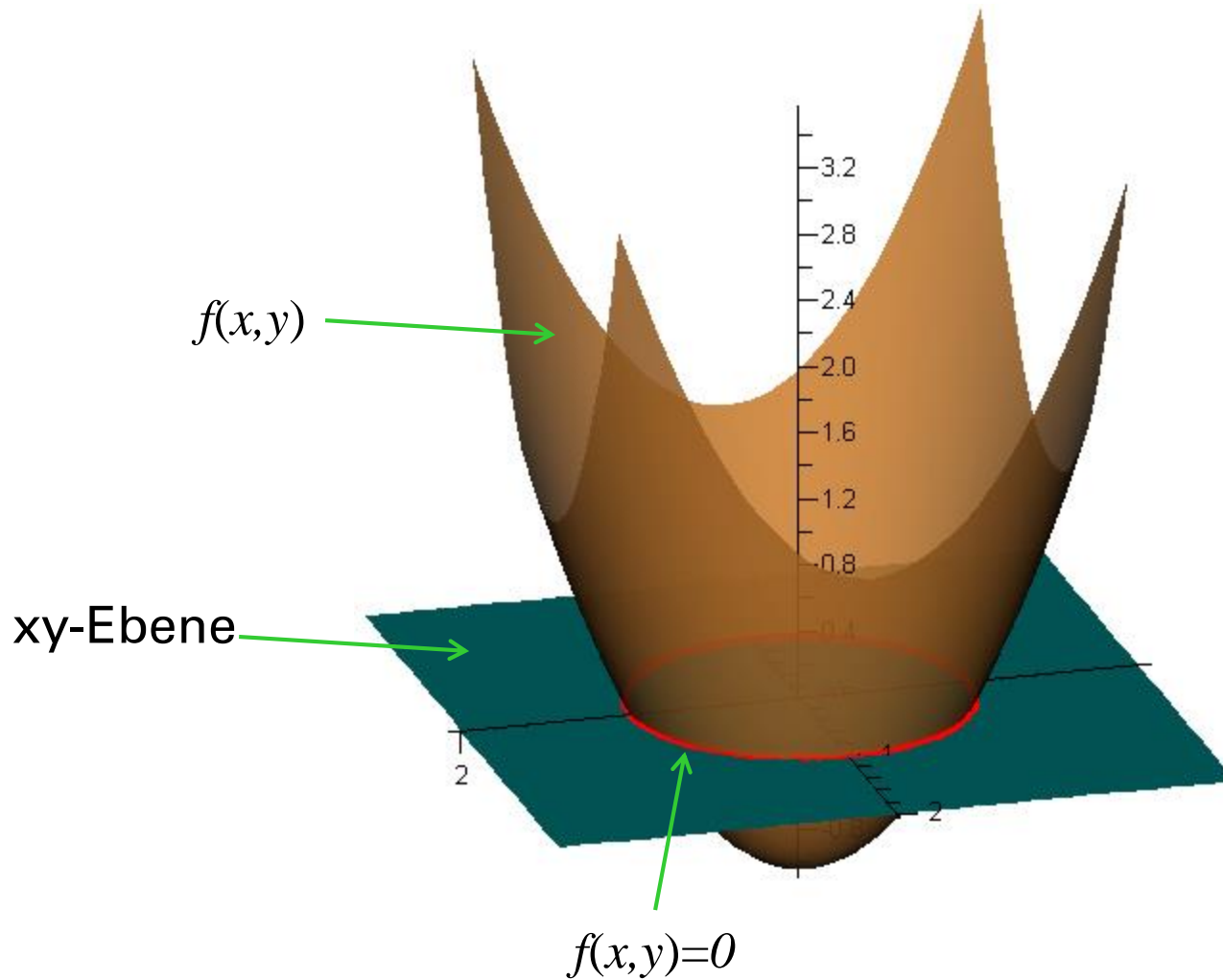
1D Parameterraum:





# Einführung

## Bsp: implizite Funktion und Kurve



# Einführung

## Überblick implizit vs parametrisch



### Implizite Modellierung

- Über dem 2D bzw 3D Modellerraum wird eine skalare Funktion  $f$  definiert.
- $f$  definiert Inneres und Äußeres eines geometrischen Objekts:
  - innen ... wenn  $f$  negativ ist
  - außen ... wenn  $f$  positiv ist
- Für die Computergraphik ist die Kurve (in 2D) bzw. Fläche (in 3D) interessant, die das Innere vom Äußeren trennt. Diese nennt man implizite Kurve bzw. implizite Fläche

### Parametrische Modellierung

- Neben dem Modellerraum gibt es bei Kurven einen 1D bzw. bei Flächen einen 2D Parameterraum und einen Parameterbereich z.B. ein Intervall bzw. ein Rechteck
- Eine vektorwertige Funktion ordnet jedem Parameterwert einen Punkt im Modellerraum zu. Diese Funktion wird Parametrisierung genannt.
- Die Umkehrfunktion heißt Einbettung und weist jedem Kurvenpunkt einen Parameterwert bzw. -punkt zu.

# Einführung

## Diskussion implizit vs parametrisch



### Implizite Modellierung

- Vorteile
  - Es können sehr einfach viele Schleifen bzw. [Köperteile](#) modelliert werden, ohne diese explizit identifizieren zu müssen.
  - Das [Abspalten](#) und Vereinen von Schleifen bzw. Körperteilen in einer Animation erfordert keine besondere Betrachtung
- Nachteile
  - Offene Schleifen oder Verästelungen bzw. offene Flächen oder Flächen mit Löchern können nicht modelliert werden

### Parametrische Modellierung

- Vorteile
  - Intuitive lokale Änderung von Kurve / Fläche möglich über ein Kontrollpunktparadigma
  - Durch Zusammensetzen und Ausblenden von Teilbereichen aus dem Parameterraum können beliebige Verästelungen und Löcher modelliert werden
- Nachteile
  - Beim Zusammensetzen kann es zu kleinen Schlitzten oder zu ungewollten scharfen Kanten und Ecken kommen