SLT-Dateiformat

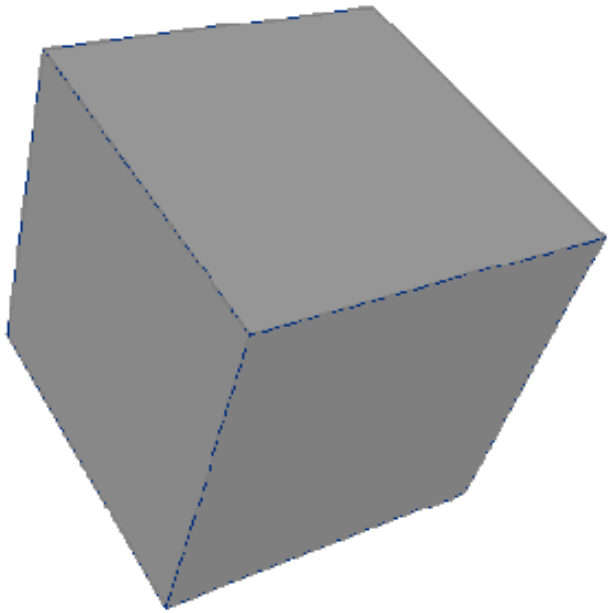
André Schledermann

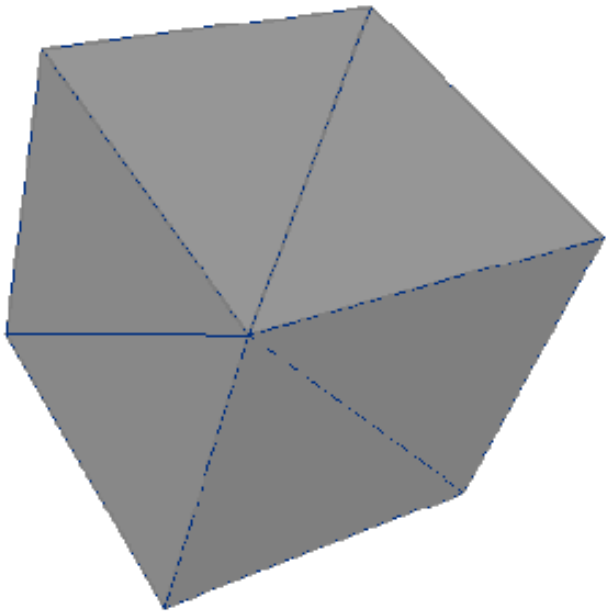
- STL‘s Ursprung – geometrische Oberfläche
- Speicherarten von STL.
- ASCII-STL-Format
- Binary-STL-Format
- Farbe im Binary-STL
- Vertex-to-vertex rule
- Normaler Vector / The Facet Normal
- Redundanz, Ineffizienz
- New Compressed Binary STL File under Lossless Data Compression
- New Compressed Binary STL File under Loss Implicit Information Data Compression

Bedeutung STL

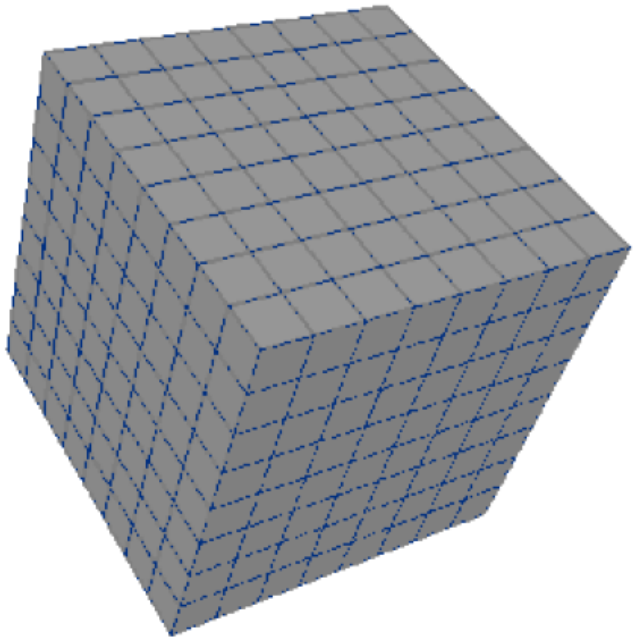
- ***S**urface **T**essellation **L**anguage* (Beschreibung der Oberfläche durch Dreiecke)
- ***S**tandard **T**riangulation **L**anguage*
- ***S**tandard **T**essellation **L**anguage*

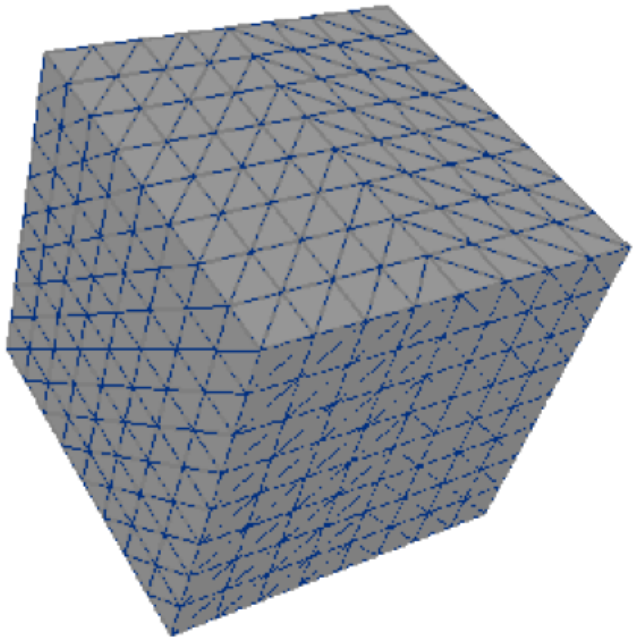
Polygon Count



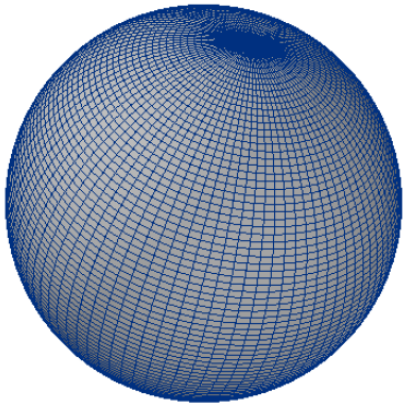


Würfel links mit 384 und rechts mit 768 Polygonen

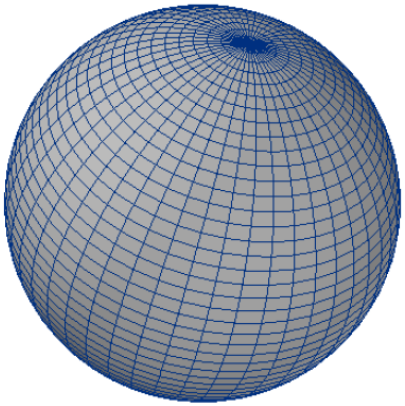




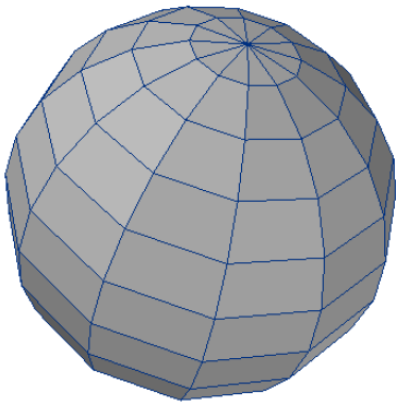
Sphäre mit unterschiedlich vielen Polygonen



10,000 polygons



2,500 polygons



144 polygons

- Speicherarten der STL-Formate

- ASCII STL-Format // Sehr großer Speicherbedarf

- Binary-STL-Format // Deutlich geringerer Speicherbedarf

ASCII STL

- Das ASCII und Binary folgen dem gleichen Prinzip

- Modifikationsmöglichkeiten-Texteditor. (Programmiersprache)

- ASCII Aufbau
- 1. solid name
- 2. facet normal ni nj nk
- 3. outer loop
- 4. vertex v1x v1y v1z
- 5. vertex v2x v2y v2z
- 6. vertex v3x v3y v3z
- 7. endloop
- 8. endfacet
- * * * * * * * *
- 9. endsolid name
- ni-nk sowie v1x-v3z sind Gleitkommazahlen

Binary STL

- Informationen werden nicht mehr in Quelltextform abgespeichert.
- Unterschied Dateikopf (80 Zeichen lang, wird meist ignoriert)
- Ende nach letztem gelesenen Dreieck

- Binary Aufbau
- 1. UINT8[80] //Dateikopf
- 2. UINT 32 //Anzahl der Dreiecke
- 3. for each triangle
- 4. REAL32[3] //Normale
- 5. REAL32[3]
- 6. REAL32[3]
- 7. REAL32[3]
- 8. UINT 16 //Attribute byte count
- 9. end
- REAL32[3] auch jeweils 3 Gleitkommazahlen

Speicher unterschied

ASCII Hex Format eines Float:

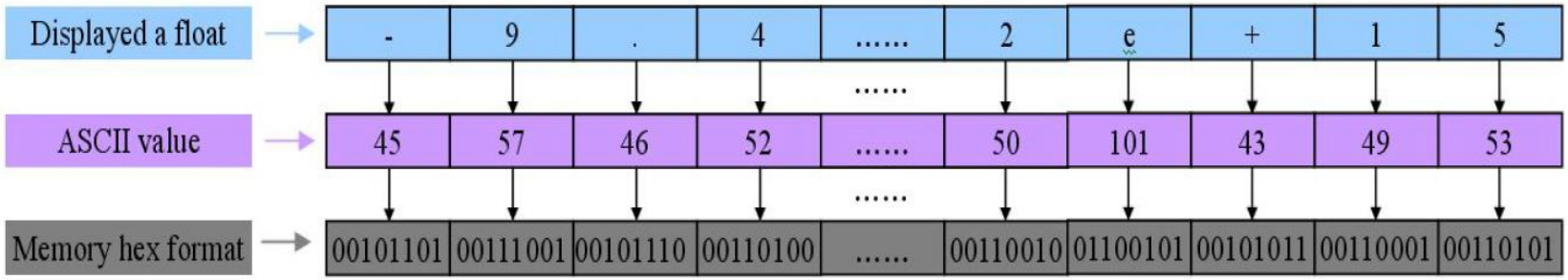


Fig. 5 ASCII Value and Memory Hex Format of a Float with Sign-mantissa-exponent Format

Binary Value und Memory Binary Format:

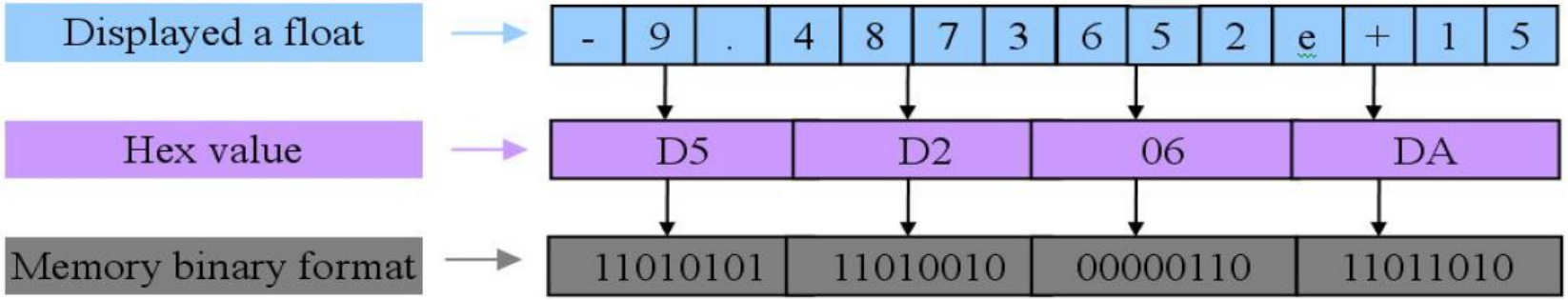


Fig. 6 Binary Value and Memory Binary Format of a Float with Sign-mantissa-exponent Format

Farbe im Binary-STL

- Mindestens zwei Arten im Binary STL-Format
- 1. VisCam/SolidView
- 2. Magics //Materialise Magics software

1. VisCam/SolidView

- 8. UINT16 //Attribute byte count
- Attribute byte count um Farbinformationen abzuspeichern
- Farbe für jedes einzelne Dreieck
- 15 Bit (0-14) RGB
- Jede Farbe hat 5 Bits (31 Farbstufen)
- Bit Nr. 15 als Boolscher Ausdruck

2. Magics

- Farbvergabe des ganzen Objekts durch Header
- ACII String „Color=“
- Farbweite wie VisCam/SolidView von 0-255
- Farbe der Elemente individuell überschreibbar
- 4tes Element Alpha Channal (transparenz)
- Bit 15 hier, ob Farbe Individuell ist oder die „Default /Header“ Farbe bekommt

Vertex-to-vertex rule

- Jedes Dreieck muss zwei Eckpunkte mit dem anliegenden Dreieck haben.
- Ein Dreieck darf keinen (vierten) Eckpunkt von anderen Dreiecken auf einer Seite haben.
- Zwei aneinander liegende Dreiecke müssen sich eine Seite teilen.

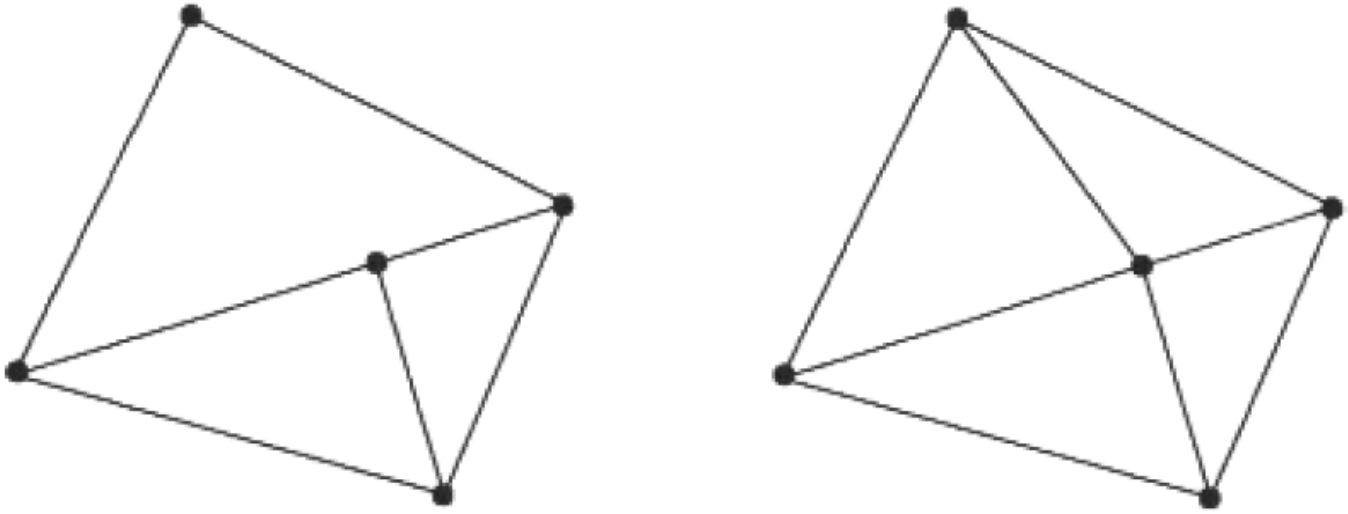
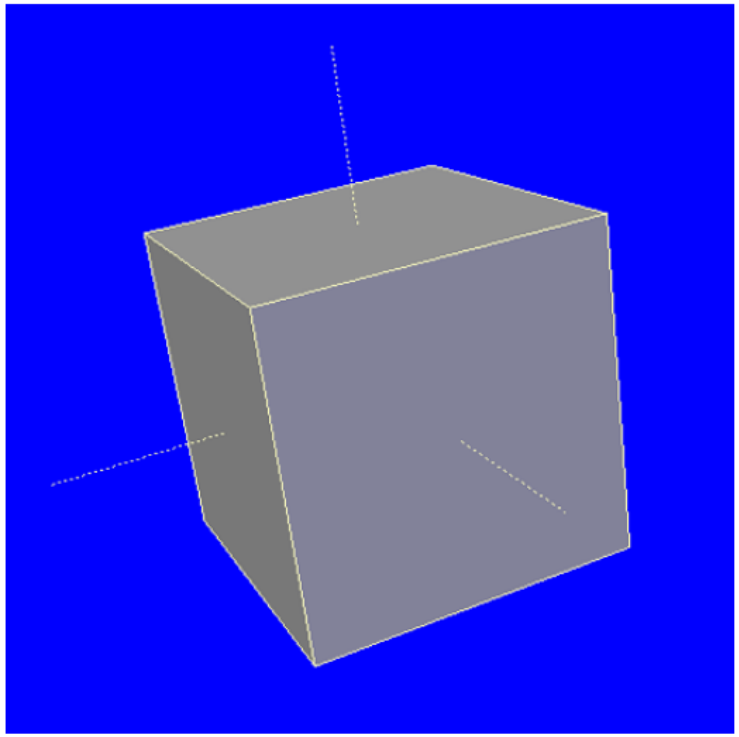


Figure 7. The vertex-to-vertex rule.
The left figure shows a violation of the rule.
A correct configuration is shown on the right.

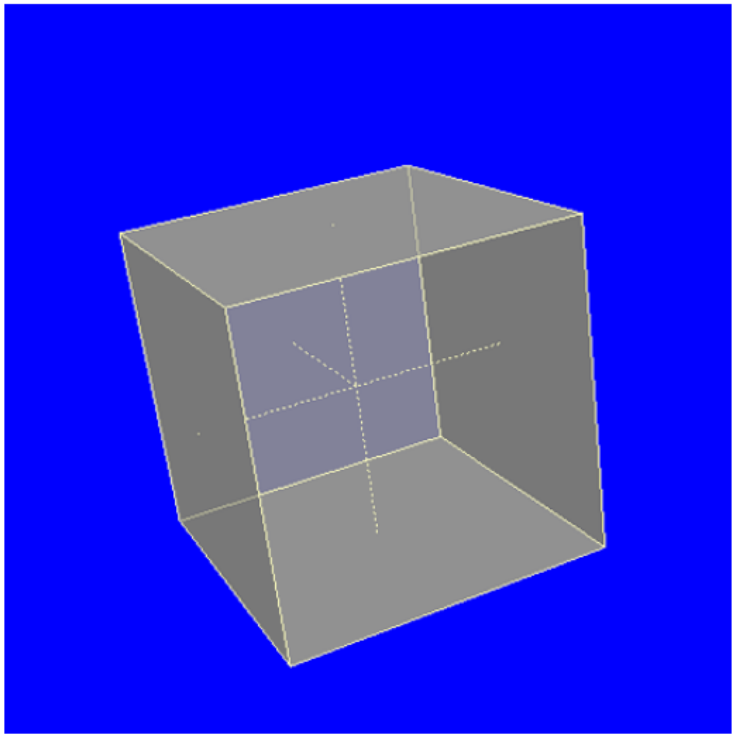
Normaler Vector / The Facet Normal

- Beide STL Formate haben einen Normalen Vector
- Normalerweise ist dieser (0,0,0)
- Die Normale kann mittels der „right hand rule“ berechnet werden
- Zeigt danach weg vom festen Objekt

Normals direction



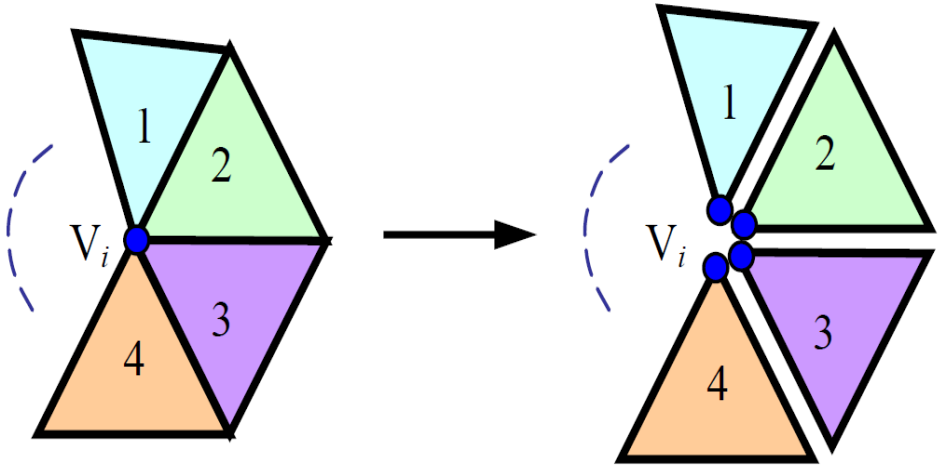
CORRECT NORMALS DIRECTION
Normals facing outward



INCORRECT NORMALS DIRECTION
Normals facing inward

Redundanz, Ineffizienz

- Unnötig hoher Speicherbedarf
- Jeder Eckpunkt mindestens 3 mal



New Compressed Binary STL File under Lossless Data Compression ca. 60%

Bytes	Data type	Description		
80	ASCII	Header No data significance		Second part of a file
4	unsigned long integer	Number of vertexes in the file		
4	float	x	Three coordinates of the vertex for a triangular mesh	Second part (vertexes part) of a triangular mesh
4	float	y		
4	float	z		
	.....		Three coordinates of other vertex	
The same information as the first vertex			Three coordinates of the last vertex	
4	unsigned long integer	Number of facets in the file		
4	float	x	Three coordinates of the normal for the first facet	
4	float	y		
4	float	z		
4	unsigned long integer	V1	Three vertexes IDs of the first facet for the triangular mesh	Third part (triangles part) of a triangular mesh
4	unsigned long integer	V2		
4	unsigned long integer	V3		
	.....		The information of other facet	
The same information as the first facet			The information of the last facet	

New Compressed Binary STL File under Loss Implicit Information Data Compression ca. 36%

Bytes	Data type	Description		
80	ASCII	Header No data significance		Second part of a file
4	unsigned long integer	Number of vertexes in the file		
4	float	x	Three coordinates of the vertex for a triangular mesh	Second part (vertexes part) of a triangular mesh
4	float	y		
4	float	z		
	.....		Three coordinates of other vertex	
The same information as the first vertex			Three coordinates of the last vertex	
4	unsigned long integer	Number of facets in the file		
4	unsigned long integer	V1	Three vertexes IDs of the first facet for the triangular mesh	Third part (triangles part) of a triangular mesh
4	unsigned long integer	V2		
4	unsigned long integer	V3		
	.....		Three vertexes IDs of other facet	
The same information as the first facet			Three vertexes IDs of the last facet	

Normalen Berechnung

$$\vec{n} = \vec{p}_1 \times \vec{p}_2$$

$$\vec{p}_1 = V_2 - V_1, \vec{p}_2 = V_3 - V_1$$

or

$$\vec{p}_1 = V_3 - V_2, \vec{p}_2 = V_1 - V_2$$

or

$$\vec{p}_1 = V_1 - V_3, \vec{p}_2 = V_2 - V_3$$

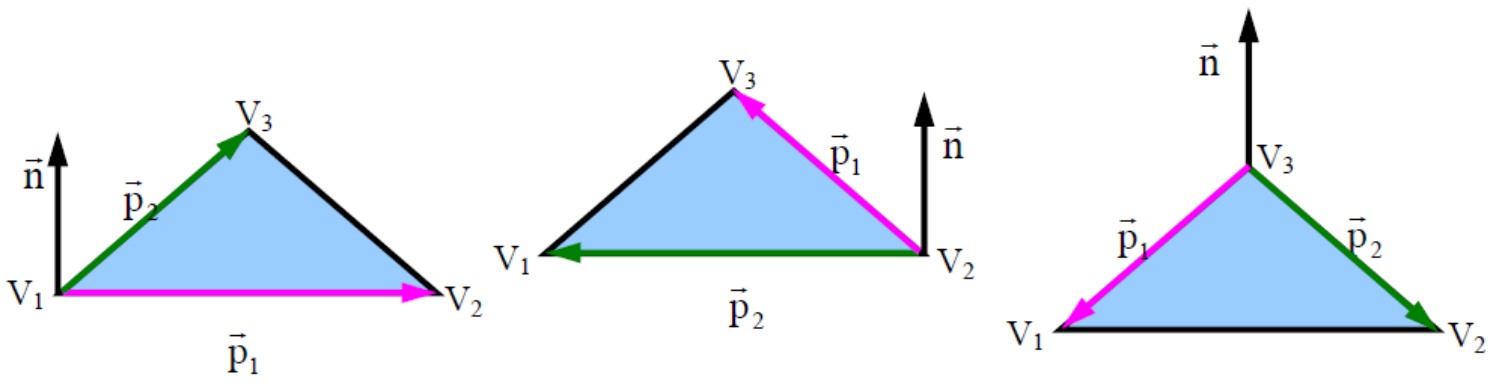


Fig. 9 Facet Normal Obtained by Vector Cross-multiply

Beispiel

$$\vec{a} \times \vec{b} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}.$$

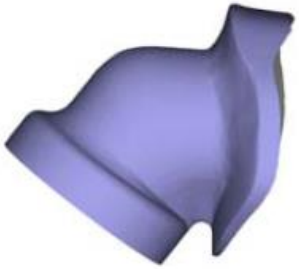
Ein Zahlenbeispiel:

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \times \begin{pmatrix} -7 \\ 8 \\ 9 \end{pmatrix} = \begin{pmatrix} 2 \cdot 9 - 3 \cdot 8 \\ 3 \cdot (-7) - 1 \cdot 9 \\ 1 \cdot 8 - 2 \cdot (-7) \end{pmatrix} = \begin{pmatrix} -6 \\ -30 \\ 22 \end{pmatrix}.$$

Auf Beispiele angewandt

(a) hat-hook

(b) knee



(c) elbow

(d) vane

Table 1 Data Size Comparison of the Original Binary STL File and Two New Format Files

Model Name	Original Size (KBytes)	Lossless Compressed Data Size (KBytes)	Loss Compressed Data Size (KBytes)
Hat-hook	3,363	2,022 (60.1%)	1,215 (36.1%)
knee	791	478 (60.4%)	288 (36.4%)
elbow	746	459 (61.5%)	280 (37.5%)
vane	6,250	3,750 (60.0%)	2,250 (36.0%)