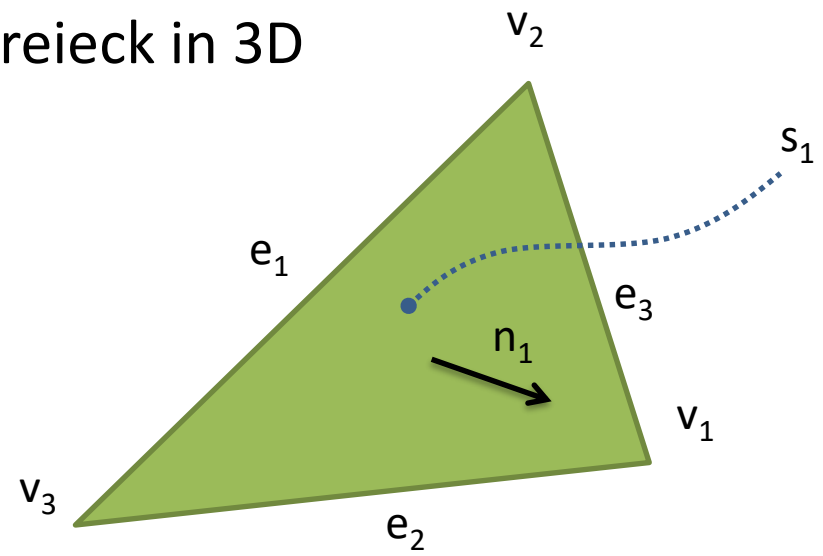
Beschreibung eines Körpers

- Die Geometrie eines Körpers wird in der CG beschrieben durch
  - Punkte (*Vertices*)
  - Kanten (*Edges*)
  - Flächen (*Surfaces*)
  - Normalen
- Beispiel: Dreieck in 3D

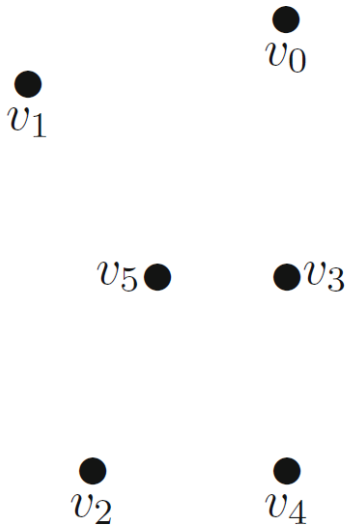


$$v_1 = (v_{1,x}, v_{1,y}, v_{1,z})$$

Definition von Grafik-Primitiven in OpenGL

- Verbindungsvorschriften für definierte Vertices:

GL_POINTS, GL_LINES, GL_LINE_STRIP, GL_LINE_LOOP, GL_POLYGON, GL_TRIANGLES, GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN, GL_QUADS, GL_QUAD_STRIP




```
glBegin(GL_POINTS);
   glVertex3fv(v0);
   glVertex3fv(v1);
   glVertex3fv(v2);
   glVertex3fv(v3);
   glVertex3fv(v4);
   glVertex3fv(v5);
glEnd();
```

Polygondefinition

- Polygone in OpenGL müssen konvex und planar sein
  - Test auf Konvexität: Abbiegerichtung
  - Test auf Planarität: Ebenengleichung, Skalarprodukt mit Normalenvektor
- Konsistente Orientierung wichtig für Oberflächen
  - Reihenfolge der Vertex-Definitionen bestimmt Vorder- oder Rückseite
  - Front-/Backface-Culling: Einstellen welche Flächen gezeichnet werden

Datenstrukturen bei Polygonnetzen

Knotenliste

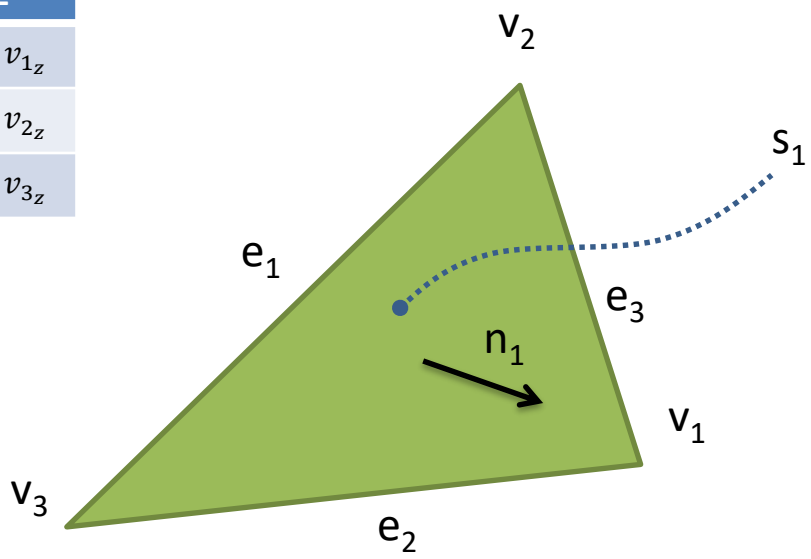
Polygone

ID	Knoten 1	Knoten 2	Knoten 3
1	1	2	3

Knoten

ID	X	Y	Z
1	v_{1_x}	v_{1_y}	v_{1_z}
2	v_{2_x}	v_{2_y}	v_{2_z}
3	v_{3_x}	v_{3_y}	v_{3_z}

- Polygon definiert als Liste von Zeigern auf ***Knoten***



Datenstrukturen bei Polygonnetzen

Kantenliste

Polygone

ID	Kante 1	Kante 2	Kante 3
1	1	2	3

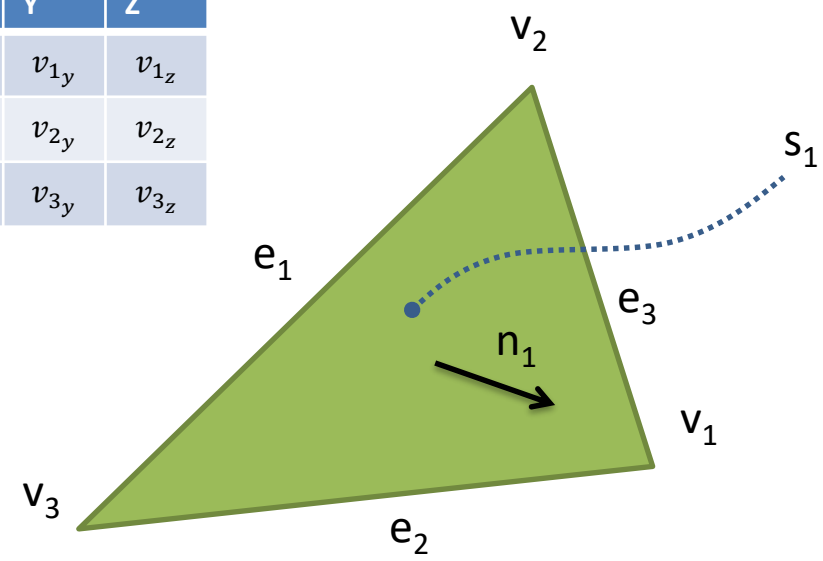
Kanten

ID	Knoten 1	Knoten 2
1	2	3
2	3	1
3	1	2

Knoten

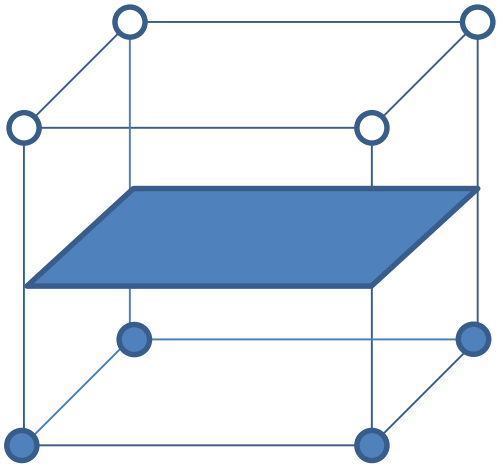
ID	X	Y	Z
1	v_{1_x}	v_{1_y}	v_{1_z}
2	v_{2_x}	v_{2_y}	v_{2_z}
3	v_{3_x}	v_{3_y}	v_{3_z}

- Polygon definiert als Liste von Zeigern auf ***Kanten***



Marching Cubes Algorithmus

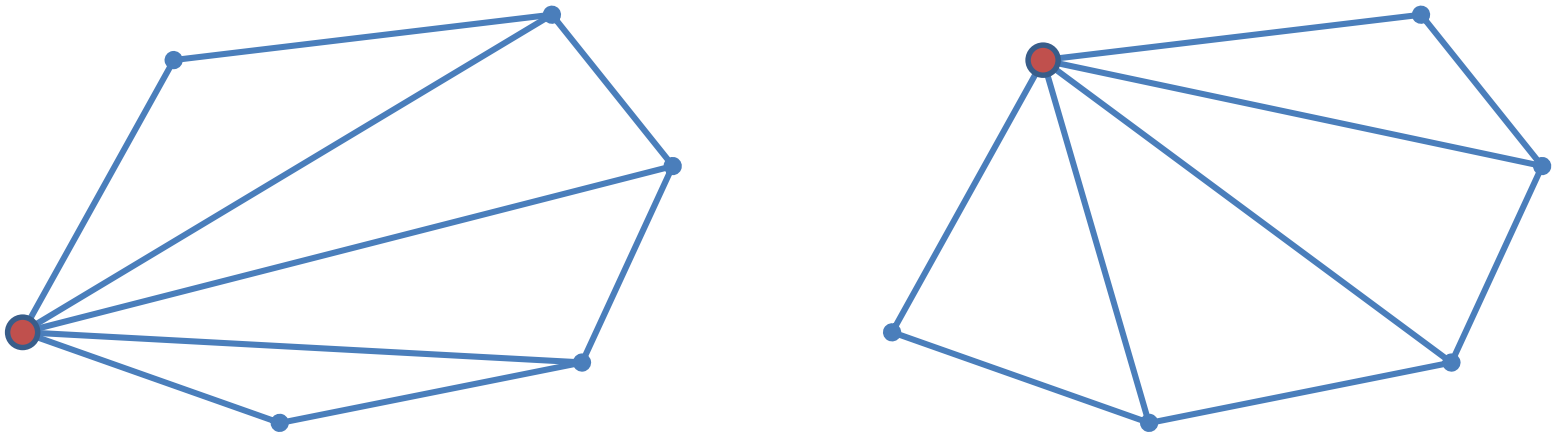
- Oft verwendet bei Erzeugung von Polygonnetzen aus Volumendaten
- Grundidee:
  - Unterteilung des Modells in kleine Würfel (*Cubes*)
  - Für jeden Würfel Schnitt des Objektes mit Würfel bestimmen (*lokales Meshing*)
- Umsetzung für Meshgenerierung aus Voxelgrafiken
  - Jeder Knoten *v* des Würfels liegt auf einem Voxel
  - Voxel-Grauwert *I* + Schwellwert *T* bestimmt ob Knoten innerhalb oder Außerhalb des Objektes liegt. Zuordnung eines Wertes an jedem Knoten:



$$\begin{matrix} 0 & \text{falls } I(v) < T \\ 1 & \text{falls } I(v) > T \end{matrix}$$

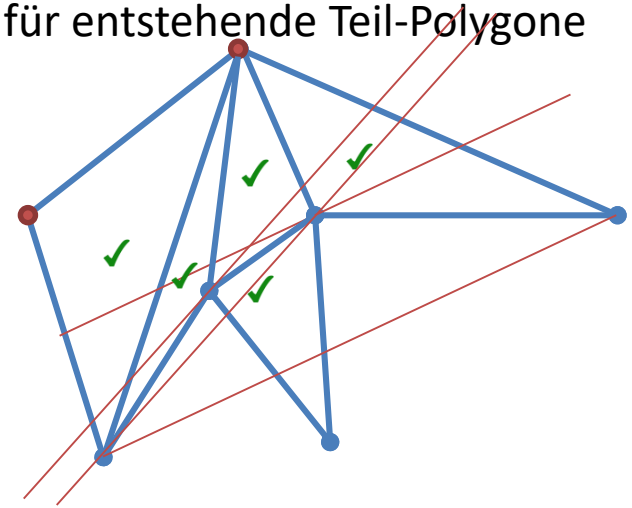
Polygontriangulation

- Polygon-Triangulation: konvexes Polygons
  - trivial, siehe **`GL_TRIANGLE_FAN`**
  - Verbinden eines Vertex mit allen anderen



Polygontriangulation

- Naiver Algorithmus für einfache Polygone: Bruteforce Diagonalen-Suche:
  - Durchlaufen aller Vertices des Polygons entlang der Polygonkante
  - Vorhergehenden und nachfolgende Vertex verbinden.
  - Rekursives Vorgehen für entstehende Teil-Polygone



- Sweepline-Algorithmus für monotone Polygone

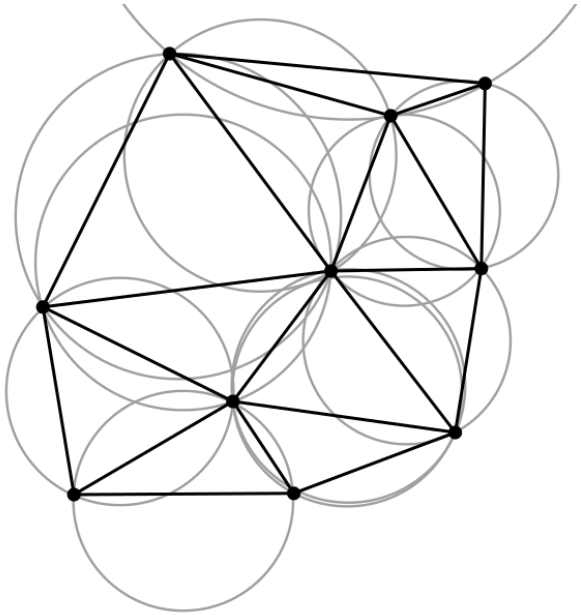
Sweep-Line-Algorithmus

```
FOR i=3 bis n
    IF v_i auf anderer Seite als S.top
        Diagonale von v_i zu Punkten in S bis auf Letzten
        Entferne alle Punkte aus S
        Lege v_i-1 und v_i auf S
    ELSE
        WHILE S.top-1 nicht für v_i von S.top verdeckt wird
            Erstelle Diagonale von v_i nach S.top-1 und entferne S.top
        Lege letzten Punkt und v_i auf S

Füge Diagonalen von v_n zu Punkten in S bis auf Ersten und Letzten ein
```

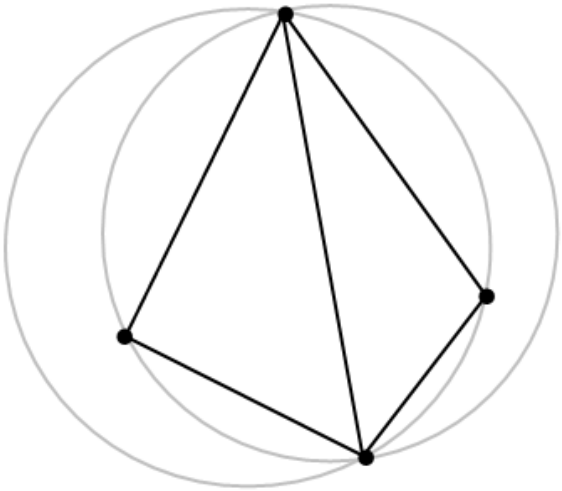
Delaunay Triangulation

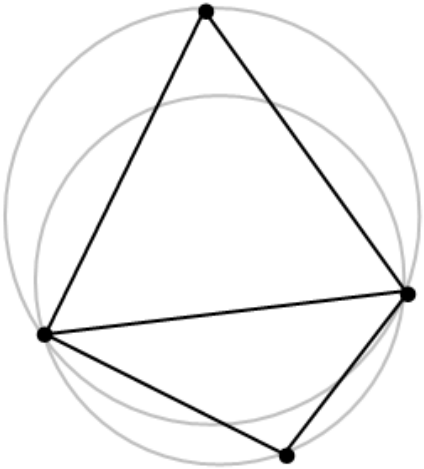
- Für Interpolationen - z.B. bei der Beleuchtung - sind Dreiecksnetze mit möglichst großen Innenwinkeln besser geeignet.
- Delaunay Triangulation maximiert den minimalen Winkel in einem Dreieck
- Grundprinzip: Der Umkreis eines Dreiecks des Netzes darf keine weiteren Punkte der vorgegebenen Punktmenge enthalten
  - In 3D: Umkugel-Bedingung für Tetraeder-Generierung



Delaunay Triangulation

- Edge Flipping
  - Erzeugen eines beliebigen Dreiecksnetzes
  - Für jedes Dreieck: prüfen ob der Umkreis einen weiteren Punkt einschließt, der Teil eines angrenzenden Dreiecks ist.
  - Ist dies der Fall wird ein Flip der gemeinsamen Kante durchgeführt.

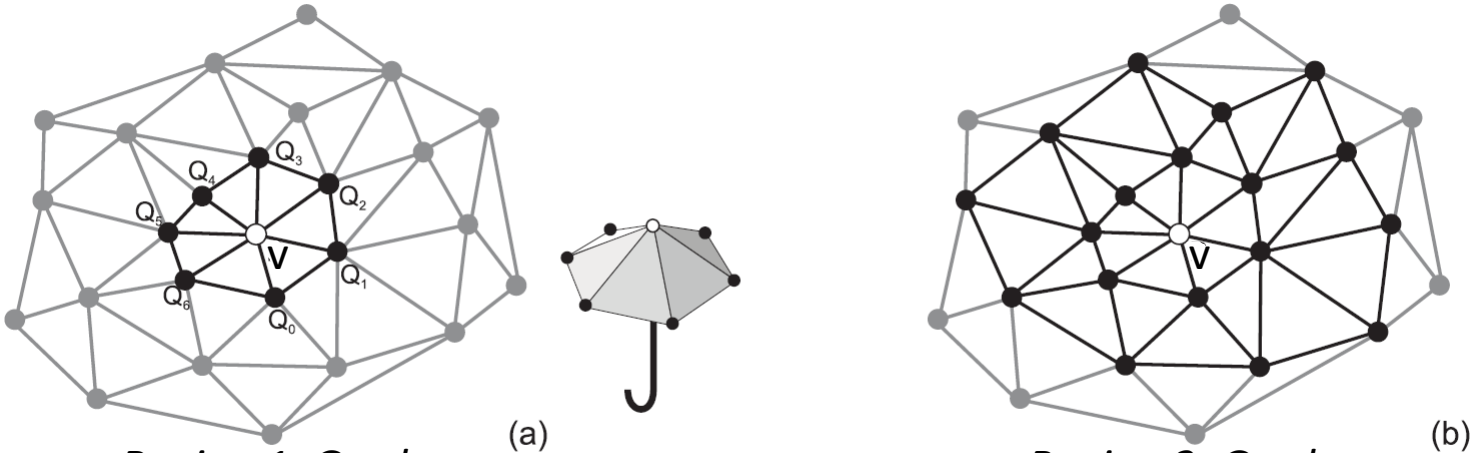




https://de.wikipedia.org/wiki/Delaunay-Triangulation

Mesh-Glättung

- Glättung im Allgemeinen durch Filter realisiert
  - Laplace Filter, Gauss Filter, Low Pass Filter
- Zu filternde Region über *Umbrella*-Operator definiert (Nachbarschafts-Operator)
  - Umbrella-Region 1. Grades: Alle Knoten q_i die mit v eine verbindende Kante haben
- Auswirkung der Glättung meist nur auf Knotenposition, Topologie bleibt erhalten (Ausnahme: Mesh-Subdivision)



Region 1. Grades *Region 2. Grades*

J. Haase: „Glättung von Polygonnetzen in medizinischen Visualisierungen“, Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, 2005