

# Abkürzungsverzeichnis

**GUI** Graphical User Interface

**ECA** Ear-Clipping-Algorithmus

**DT** Delaunay Triangulation

# 1 Einleitung

Die größte technische Wende nach der industriellen Revolution war die digitale Revolution gegen Ende des 20. Jahrhunderts.[1] Eingeleitet durch die Entwicklung des Mikrochips und der damit verbundenen Verbreitung des Computers in allen Lebensbereichen, führte sie zu einer dramatischen Veränderung. Nicht nur in der Industrie und Produktion findet diese Veränderung statt, welche sich in flexibler Automatisierung äußert, sondern auch in anderen Bereichen. So wird die Entwicklung des Internets durch vernetzte Rechner möglich. Als dann Computer nicht mehr nur in der Forschung und für die automatische Produktion genutzt werden, sondern auch für den täglichen Gebrauch im Büro und daheim genutzt werden können, braucht es grafische Benutzeroberflächen und Betriebssysteme. Doch dort macht die Entwicklung nicht halt. Auch die Unterhaltungsbranche erfährt mit Videospielen eine Revolution, welche ebenso auf Computergrafik angewiesen ist, wie ein einfaches Graphical User Interface (GUI).

Zu Beginn beschränkte sich die Darstellung auf sogenannte ASCII-Art, bei der übliche Zeichen aus dem ASCII-Alphabet benutzt wurden, um komplexe Bilder zu erzeugen. Da dies jedoch nicht genügt, um Flächen und Objekte lückenlos darzustellen, bedurfte es einer Innovation. Obwohl es für Menschen einfach ist, Flächen als Ganzes zu betrachten und Polygone in unterschiedlichster Komplexität zeichnerisch darzustellen, ist es für Computer als Ganzes nicht so einfach, diese zu speichern, geschweige denn darzustellen. Flächen und dreidimensionale Objekte kann man, so die Idee, über ihre Eckpunkte (Vertices) und die dazwischen liegenden Kanten (Edges) darstellen. Man erzeugt also ein Polygonnetz, welches den Körper abbildet. Dabei ist die Wahl des Polygons zunächst irrelevant. So könnte man beispielsweise einen Würfel aus Quadraten oder Dreiecken aufbauen, je nach dem wie man die Kanten definiert.[2] In der Praxis sind Polytope und Polygone jedoch meistens unregelmäßig, ergeben sie sich doch zum Beispiel aus Umgebungs-scans mit einem Laser-Scanner oder sind die Oberfläche einer Videospieldigur. Es bietet sich in solchen Fällen nicht an, regelmäßige Polygone, wie Quadrate oder Rechtecke, als Grundlage für das Polygonnetz zu nutzen.

Eine geeignete Methode, um diese komplexen Polygone für Computer effizient darzustellen, ist die Nutzung von Dreiecken als grundlegende Form für die Zerlegung. Dieses Vorgehen bezeichnet man als Triangulation. Diese ist formal die Zerlegung eines topologischen Raumes, hier also eines Polygons in Simplexe. Das Simplex der zweiten Dimension das Dreieck und damit ist die Triangulation ein Verfahren zur Zerlegung eines Polygons in Dreiecke. Es sei erwähnt, dass es Computern durchaus möglich ist, Flächen und Körper darzustellen, welche nicht aus Dreiecken bestehen. Dies ist jedoch wesentlich speicher- und rechenaufwendiger, als es bei Dreiecken der Fall ist. Für die Darstellung eines Objektes ließe sich alternativ auch eine sogenannte Punktwolke nutzen. Wie der Name bereits

andeutet, wird das Objekt dabei aus einer großen Menge von Einzelpunkten gebildet. Für einen hohen Detailgrad sind dafür allerdings auch sehr viele Punkte nötig, was den Speicheraufwand stark erhöht. Bei der Verwendung von Dreiecken handelt es sich, vorallem bei runden Objekte, eher um eine Approximation der Form. Eine Kugel wäre dann nicht vollständig rund, sondern würde als Polyeder repräsentiert werden. Dadurch spart man jedoch sehr viel Speicherplatz. Man kennt diese optische Vereinfachung aus alten Videospielen, in denen alle Objekte doch recht kantig und spitz aussehen. Man kann auch hier den Detailgrad steigern, indem man die Anzahl der Dreiecke erhöht und ihre Größe senkt. Da Dreiecke Flächen sind, benötigt man von ihnen jedoch eine geringere Anzahl, um ein Objekt darzustellen, als wenn dies mittels einer Punktwolke geschieht.

Um eine Triangulation per Computer durchzuführen, bedarf es eines Algorithmus, welche das Verfahren beschreibt. Von diesen gibt es viele verschiedene, welche unterschiedlichste Herangehensweisen nutzen. Hier seien der Ear-Clipping-Algorithmus (ECA), die Delaunay Triangulation (DT) und das Voronoi-Diagramm als Beispiele für solche Algorithmen angeführt. Diese unterscheiden sich erheblich in ihrer Komplexität und Effizienz. Mit einer Laufzeit von  $O(n^3)$ [10], ist der ECA bei weitem nicht so effizient wie beispielsweise die DT mit  $O(n \log n)$ . Für den ECA spricht jedoch seine relative Einfachheit im Vergleich zu anderen Algorithmen.

In dieser Arbeit soll jedoch nicht die Laufzeitoptimierung im Vordergrund stehen, sondern etwas anderes. Anschaulichkeit ist ein wichtiger Punkt, wenn es um Didaktik geht. Sie fördert das Verständnis ebenso wie Interaktivität. So hat diese Ausarbeitung zum Ziel eine interaktive Visualisierung für die Triangulation von Polygonen zu schaffen. Dafür ist der ECA aufgrund seiner relativen Einfachheit gut geeignet. Er lässt sich schrittweise durchlaufen und ist somit sehr anschaulich, da in jedem Schritt ein Dreieck der Triangulierung erzeugt wird. Es liegt in der Natur dieses Algorithmus, dass Uneindeutigkeiten auftreten, was die Auswahl des nächsten Dreiecks angeht. Diese führen zum zweiten wichtigen Punkt in dieser Arbeit - der Interaktivität. Der Nutzer der Visualisierungssoftware soll interaktiv entscheiden können, welches das nächste Dreieck ist, welches bearbeitet wird. Er beeinflusst somit direkt das endgültige Resultat. Neben dem direkten Eingriff des Nutzers, sollen auch Heuristiken genutzt werden können, um diese Auswahl zu treffen. Beispielsweise kann hierfür die Größe des Dreiecks im Bezug auf seinen Flächeninhalt genutzt werden oder auch die Innenwinkel. Es ist angestrebt, dass die Nutzerauswahlen ausgewertet und in eine Heuristik überführt werden. Um dies zu bewerkstelligen, soll die Qualität der Triangulation mittels verschiedener Metriken beurteilt werden. Hierfür kann ein Vergleich zum Voronoi-Diagramm ebenso wie zum Beispiel die Anzahl der sogenannten *Slivers*[12] betrachtet werden. Letztere führen in Anwendungen der Computergrafik oft zu Fehlern, welche vermieden

werden sollten.

Es soll somit nicht nur eine Visualisierung für Triangulationen geschaffen werden, sondern es sollen auch die Auswirkungen einfacher Heuristiken auf die Qualität dieser Zerlegungen betrachtet werden. Es steht dabei, wie bereits erwähnt, nicht die Laufzeit des Algorithmus im Vordergrund, welche üblicherweise Ziel der Optimierung ist.

## Literatur

- [1] *Digitale Revolution*  
(<https://www.staatslexikon-online.de>)
- [2] *Darstellung von Kurven und Flächen*, Christoph Dähne  
(<https://www.inf.tu-dresden.de>)
- [3] *Polygon Triangulation*, Subhash Suri  
(<https://sites.cs.ucsb.edu/~suri/cs235/Triangulation.pdf>)
- [4] *Polygon, Definition*  
(<https://mathepedia.de/Polygone.html>)
- [5] *Regular Polygons. In: Michiel Hazewinkel (Hrsg.): Encyclopedia of Mathematics. Springer-Verlag und EMS Press, Berlin 2002*
- [6] *Convex Polygon*  
(<https://www.mathopenref.com/polygonconvex.html>)
- [7] *Triangulation, Definition*  
(<https://encyclopediaofmath.org/wiki/Triangulation>)
- [8] *Simplex, Definition*  
(<https://encyclopediaofmath.org/wiki/Simplex>)
- [9] *Ear-clipping Based Algorithms of Generating High-quality Polygon Triangulation*, Gang Mei, John C. Tipper and Nengxiong Xu  
(<https://arxiv.org/ftp/arxiv/papers/1212/1212.6038.pdf>)
- [10] *Triangulation by Ear Clipping*, David Eberly  
(<https://www.geometrictools.com>)
- [11] *Ear-Clipping Triangulierung*  
([wiki.delphigl.com](http://wiki.delphigl.com))
- [12] *Geometrical Mesh Quality*  
(<https://www.iue.tuwien.ac.at>)