
Team 12

Frédéric Blondeel	h
Martijn Debeuf	h
Toon Sauvillers	h
Dirk Vanbeveren	h
Bert Van den Bosch	h
Seppe Van Steenberghe	h

1 Introductie

2 Delaunay triangulatie

Om verschillende punten mooi te verbinden, kan er gebruik worden gemaakt van een Delaunay triangulatie. Deze triangulatie wordt toegepast op de middenpunten van de schermen. Wanneer deze triangulatie op de schermen wordt geprojecteerd, zullen de liggingen van de schermen mooi worden weergegeven. Bij een Delaunay triangulatie worden alle punten zo verbonden dat ze verscheidene driehoeken vormen. De zijden van de driehoeken mogen niet overlappen en de kleinste hoek in de triangulatie moet gemaximaliseerd worden. [1]

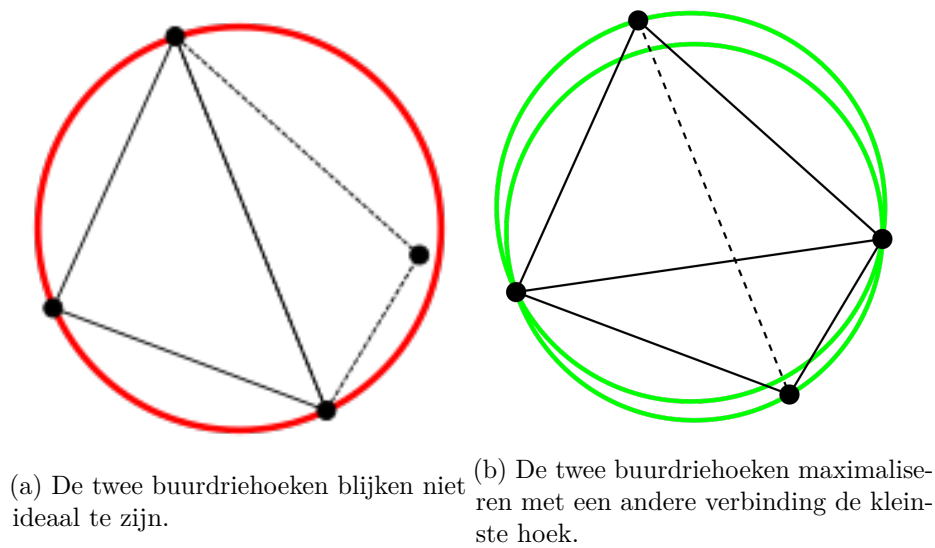
De triangulatie kan op vele manieren worden berekend, in dit verslag zijn er twee soorten algoritmen onderzocht. Het S-hull algoritme en het Boyler-Watson algoritme.

2.1 S-hull

Het eerste algoritme is het S-hull algoritme. Het is een $O(n \log(n))$ algoritme, gebruikmakend van een radiaal propagerende sweep hull. S-hull start met twee punten, één daarvan is willekeurig geselecteerd. Het tweede punt is het dichtsbijzijnde punt. Vervolgens zal er een derde punt worden gezocht dat samen met de twee voorgaande punten een zo klein mogelijke cirkel maakt. Het middenpunt van deze cirkel wordt gebruikt om alle punten te sorteren t.o.v. de afstand tussen het punt en het gevonden middenpunt. Deze gesorteerde punten zullen één voor één toegevoegd worden aan de triangulatie.

Nadat alle punten sequentieel zijn toegevoegd, is er een niet overlappende triangulatie bekomen. Om een Delaunay triangulatie te bekomen moet er echter ook gekeken worden naar de hoeken. Elke driehoek zal nu worden afgegaan en er wordt gekeken of er geen andere driehoek kan gevormd worden met de buurdriehoeken zodat de minimale hoek gemaximaliseerd wordt, zie figuur 1. [2]

Dit algoritme is echter niet gebruikt. Ook al is het algoritme $O(n \log(n))$ het vereist een zeer goede data management om deze snelheid te behalen. [3] Er is daarom gekozen voor een simpeler algoritme, namelijk het Boyler-Watson algoritme.



Figuur 1: Het maximaliseren van de kleinste hoek [1]

2.2 Boyler-Watson

Het Boyler-Watson algoritme is wel geïmplementeerd. Het heeft een tijdscomplexiteit van $O(n^2)$, dit is aanzienlijk trager dan het S-hull algoritme. [4] Het vereist echter een minder complex data management en is daarom ook makkelijker te implementeren. De tijdscomplexiteit is ook relatief te zien, er zijn namelijk maar maximaal 120 schermen. Meer schermen kunnen er door het identificatiemechanisme niet worden geïdentificeerd. Gezien het laag aantal punten, moet het algoritme niet nodeeloos complex worden en kan er gekeken worden naar ‘tragere’ algoritmen, zoals het Boyler-Watson algoritme.

3 Besluit

Referenties

- [1] Delaunay triangulation. https://en.wikipedia.org/wiki/Delaunay_triangulation.
- [2] David Sinclair. S-hull: a fast sweep-hull routine for delaunay triangulation. s-hull.org, 2010.
- [3] Erik Thune Lund. Implementing high-performance delaunay triangulation in java. mathesis, Department of Informatics, University of Oslo, 2014.
- [4] Bowyer-watson algorithm. https://en.wikipedia.org/wiki/Bowyer%E2%80%9993Watson_algorithm.