

Asteroidin kolmiointi

Anni Järvenpää

0143368836

21. toukokuuta 2016

Sisältö

1	Delaunay-kolmiointi	1
1.1	Algoritmit	1
1.1.1	Quickhull	2
2	Pisteiden generointi ellipsoidin pinnalle	6
2.1	Pisteiden luominen kerroksittain	6
2.2	Pisteiden arpominen	7
3	Lommel-Seeliger -heijastuslaki	9
4	Asteroidin muodon kolmiointi	10
5	Asteroidin sisäinen rakenne	10
6	Asteroidin integroitu kirkkaus	13

1 Delaunay-kolmiointi

Kolmioinnissa pistejoukon pisteet yhdistetään toisiinsa janoilla siten, että pisteiden välille muodostuu kolmioita. Delaunayn kolmioinnissa minkään kolmion ympäripiirretyn ympyrän sisäpuolella ei ole muita pisteitä. Mikäli pisteet eivät osu samoille ympyröille, on Delaunayn kolmiointi yksikäsitteinen. Kahdessa ulottuvuudessa tämä kolmiointimenetelmä maksimoi pienimmän kolmioista löytyvän kulman, jolloin kapeat kolmiot ovat harvinaisia. [4]

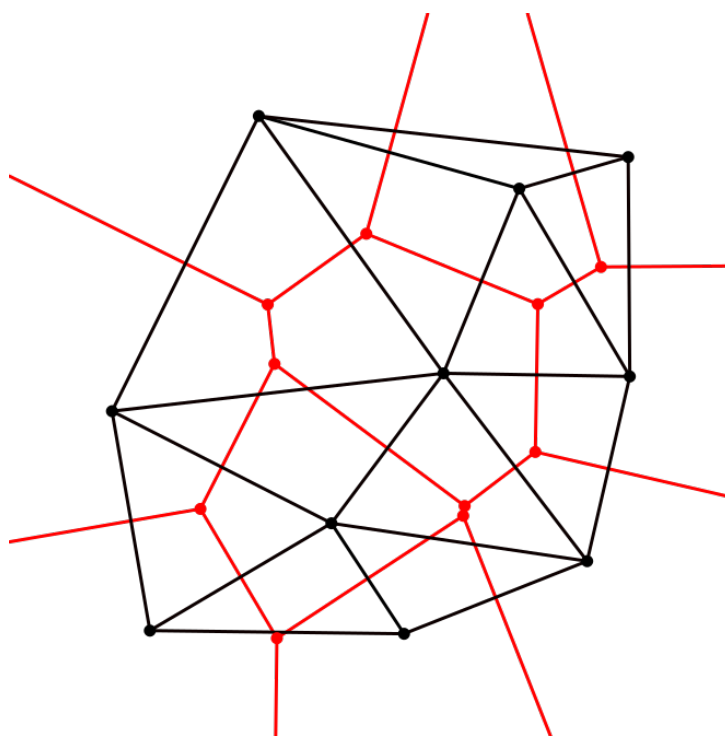
Delaunayn kolmiointi liittyy läheisesti moniin käytännön ongelmiin. Sitä voidaan käyttää esimerkiksi 3D-mallinnuksessa, jolloin kappaleen pinta voidaan mallintaa kolmioina. Kolmiointia voidaan käyttää myös esimerkiksi verkon pienimmän virittävän puun etsimiseen, sillä pienimmät virittävät puut ovat aina Delaunayn kolmioinnin osajoukkoja. Lisäksi yhdistämällä Delaunayn kolmioinnissa käytettyjen ympyröiden keskipisteet, saadaan niin kutsuttu Voronin tesselaatio (kuva 1), jossa lähin kolmioinnissa käytetty piste on sama. [4, 7]

Menetelmä on yleistettävissä myös korkeampiin ulottuvuuksiin, jolloin esimerkiksi kolmiulotteisessa tapauksessa syntyy ympyröiden sisällä olevien kolmioiden sijaan pallojen sisällä olevia tetraedrejä. [4]

1.1 Algoritmit

Delaunayn kolmioinnin luomiseen on kehitetty useita erilaisia algoritmeja. Ne voidaan kuitenkin jakaa muutamaan peruskategoriaan:[1]

- *Paikallinen parantaminen*, jossa aluksi luodaan mielivaltainen kolmiointi ja erilaisilla paikallisilla muunnoksilla muutetaan kolmiointi Delaunayn kolmioinniksi.
- *Vaiheittainen lisääminen*, jossa kolmiointi rakennetaan lisäämällä kolmiointiin kolmioita yksittäin niin, että uudet kolmiot täyttävät aina kolmioinnille asetetut ehdot
- *Vaiheittainen rakentaminen*, jossa lisätään kolmiointiin pisteet yksi kerrallaan, ja kolmio, joka sisältää lisätyn pisteen, jaetaan uusiin kolmioihin siten, että kaikki toteuttavat kolmioinnille asetetut ehdot.
- *Korkeampaan ulottuvuuteen upottaminen*, jossa pisteet siirretään korkeampaan ulottuvuuteen, jossa lasetaan pisteille konvekksi kuori, jonka projektiosta alkuperäiseen ulottuvuuteen saadaan Delaunayn kolmiointi



Kuva 1: Mustien pisteiden Delaunayn kolmiointi (mustat viivat) ja sitä vastaava Voronoi tesselaatio punaisella. Punaiset pisteet ovat Delaunayn kolmiointinnissa käytettyjen ympyröiden keskipisteitä. (Kuva: Wikimedia Commons -käyttäjä *Hferee*, CC BY-SA 3.0)

- *Hajota ja hallitse* -algoritmit, joissa pistejoukko jaetaan rekursiivisesti pienempiin pistejoukkoihin ja nämä pistejoukot kolmioidaan. Tämän jälkeen yhdistämisvaiheessa nämä pienten alueiden kolmiointit yhdistetään vaiheittain siten, että lopullinen kolmiointi täyttää Delaunayn kolmiointin ehdot.

Kahdessa ulottuvuudessa monet näistä ovat yksinkertaisia, mutta kolmessa ulottuvuudessa monen algoritmin toteuttaminen muodostuu huomattavasti vaikeammaksi. Esimerkiksi neljän pisteen välille piirrettyjä kolmioita, jotka voivat toteuttaa Delaunay-kolmiointin ehdot, on vain kaksi erilaista kun taas neljän pisteen välille voidaan kolmessa ulottuvuudessa piirtää viisi erilaista tetraedrikonfiguraatiota, jolloin ns. *edge-flipping* -algoritmien toteuttaminen on huomattavasti haastavampaa. [4]

1.1.1 Quickhull

Mikäli kolmioitava kappale on konvekksi ja halutaan kolmioida ainoastaan kappaleen pinta, voidaan hyödyntää suoraan jotakin konveksin kuoren etsimiseen käytettävää algoritmia[4]. Esimerkiksi ellipsoidin muotoisen asteroidin pinnan muotojen mallintamiseen riittää pinnan kolmiointi ilman pisteiden tetraedrintia ja ellipsoidi on konvekksi kappale, joten konveksin kuoren etsimiseen käytettävät algoritmit ovat suoraan sovellettavissa tähän on-

gelmaan, sillä pistejoukon kovenksi kuori on tällöin suoraan pistejoukon Delaunayn kolmiointi.

Eräs nopea ja periaatteeltaan yksinkertainen myös suurille pistejoukoille sopiva konveksin kuoren etsintäalgoritmi on quickhull. Algoritmin suorittaminen aloitetaan valitsemalla pistejoukosta ensimmäinen tetraedri. Tätä varten pistejoukosta ensin minimi- ja maksimipisteet kaikkien koordinaattiakselian suunnassa (6 kpl) ja valitaan näistä ensin kaksi kauimpana toisistaan sijaitsevaa pistettä. Seuraavaksi etsitään jäljelle jääneistä neljästä pisteestä se, jonka kohtisuora etäisyys näiden pisteiden virittämästä suorasta on suurin, jolloin saadaan ensimmäisen tetraedrin pohja. Viimeisenä valitaan piste, jonka etäisyys tästä kolmiosta on suurin, jolloin on saatu luotua ensimmäinen tetraedri. [2]

Algoritmin suorituksen kannalta on tärkeää, että jokainen pistejoukon piste, joka ei ole syntyneen kuoren sisällä (konveksin kappaleen tapauksessa tällaisia pisteitä ei löydy, joten nämä eivät ole tämän työn kannalta relevantteja) tai jo osana jotakin kolmiota, kuuluu jostakin olemassaolevasta kolmiosta näkyviin pisteisiin. Tästä syystä käydään alussa kukin piste läpi, ja käydään kutakin pistettä kohden läpi tahkoja, kunnes löytyy tahko, josta katsottuna kyseinen piste on horisontin yläpuolella. Kun tällainen tahko löytyy, lisätään piste tahkosta näkyvillä olevien pisteiden listaan. Todellisuudessa moni piste näkyy useista tahkoista, mutta nekin merkitään vain yhden tahkon listaan.[2]

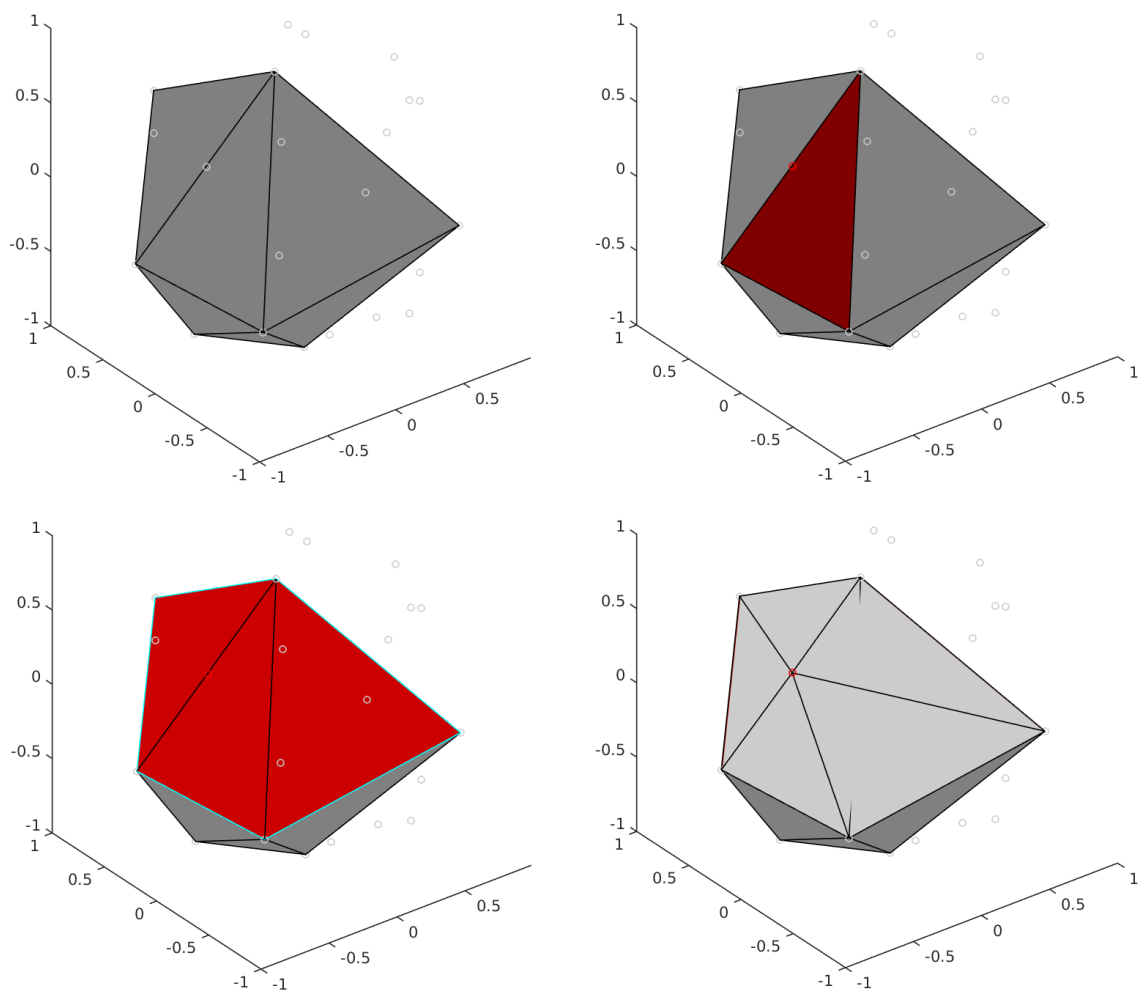
Pisteen näkyvyys tahkolta voidaan tarkastaa yksinkertaisesti tutkimalla, onko piste eri puolella tahkon määräämää tasoa kuin jokin jo luodun monitahokkaan sisällä oleva piste. Omassa toteutuksessani tämä toteutetaan määrittämällä ensin kolmion määräämän tason yhtälö muodossa $ax + by + cz + d = 0$, minkä jälkeen voidaan kahden pisteen koordinaatit sijoittaa yhtälön vasemmalle puolelle. Mikäli näin saadut luvut ovat samanmerkkiset, ovat pisteet samalla puolella tasoa. Merkki on yksinkertaista määrittää ottamalla lukujen tulo ja tarkastelemalla sen etumerkkiä.

Kun kaikille pisteille on löydetty tahko, josta ne näkyvät, työnnetään kaikki neljä luotua tahkoa käsiteltävien tahkojen pinoon. Tämän jälkeen aloitetaan varsinainen iteraatiovaihe, jossa monitahokasta laajennetaan, kunnes se kattaa kaikki pistejoukon pisteet.[2]

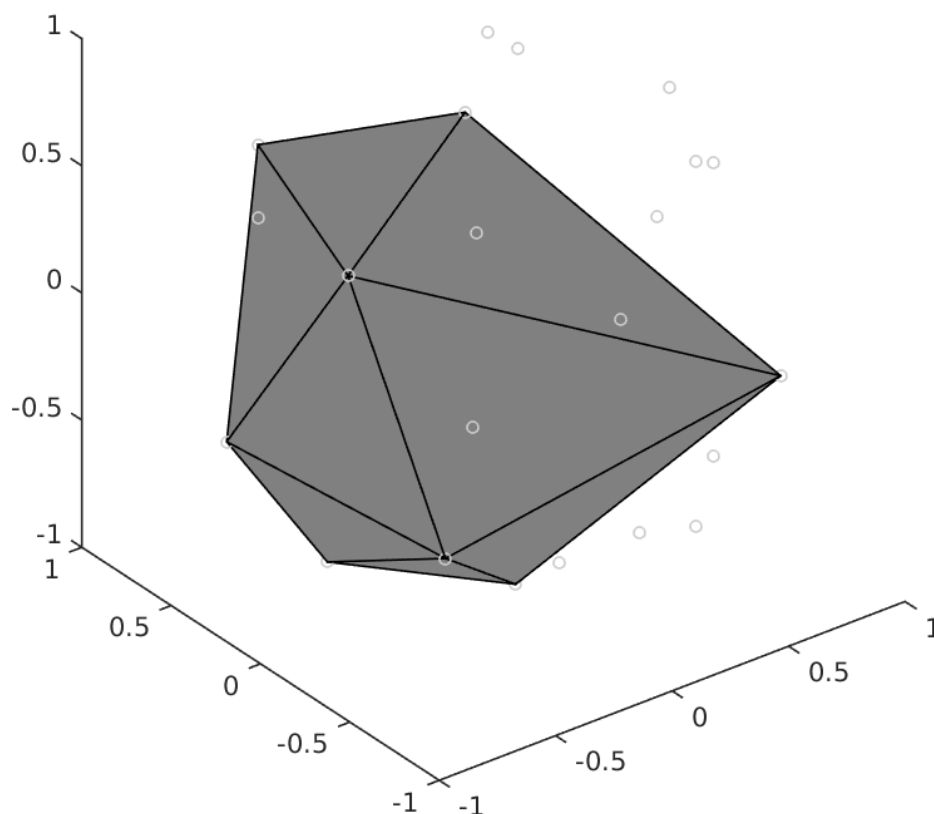
Iteraatiovaiheessa poimitaan pinosta päällimmäinen tahko, joka on tämän iteraation käsiteltävä. Kun käsiteltävien tahkojen pino on tyhjä, on kolmiointi valmis. Mikäli tällä tahkolla ei ole yhtään näkyvää pistettä siirrytään seuraavaan iteraatioon, muussa tapauksessa valitaan tahkon näkyvistä pisteistä kaukaisin. Tämän jälkeen etsitään edellä esitettyä näkyvydentarkastusmenetelmää käyttäen käsiteltävän kolmion naapureista (tahkoista, jotka jakavat vähintään yhden pisteen käsiteltävän tahkon kanssa) sellaiset, joista kaukaisin piste on näkyvissä. Näitä pisteistä näkyviä tahkoja ympäröi horisontti,

jonka pisteet yhdistämällä kaukaisimpaan pisteeseen, saadaan korvattua käsiteltävä ja näkyvät kolmiot uusilla, jolloin monitahokas kasvaa jonkin verran. Korvatuista kolmioista näkyvissä olevat pisteet jaetaan uusille kolmioille ja uudet kolmiot lisätään käsiteltävien kolmioiden pinoon. Korvatut kolmiot poistetaan käsiteltävien kolmioiden pinosta.[2]

Kuvassa 2 on nähtävillä yhden iteraation kulku Quickhull-algoritmissa. Kuvassa vasemmalla ylhäällä alkutila, jossa monitahokasta ympäröi vaaleanharmailla ympyröillä merkitty pistejoukko. Oikealla ylhäällä nähtävillä tummanpunaisella nyt käsittelyvuorossa oleva tahko. Tämän tahkon näkyvien pisteiden listassa on vain yksi piste (kirkkaanpunainen ympyrä), vaikka tahkosta todellisuudessa näkyy useita pisteitä. Ne kuuluvat kuitenkin jonkin muun tahkon näkyviin pisteisiin. Vasemmalla alhaalla on tästä pisteestä nähtävissä olevat käsiteltävän tahkon naapuritahkot merkitty punaisella ja niitä ympäröivä horisontti syaanilla. Oikealla alhaalla horisontti on yhdistetty käsiteltävään pisteeseen, jolloin syntyneet uudet kolmiot on piirretty vaaleanharmaalla. Vanhat kolmiot poistetaan, jolloin päästään kuvassa 3 esitettyyn tilaan, joka toimii seuraavan iteraation alkutilana.



Kuva 2: Yksi Quickhull-algoritmin iteraatio.



Kuva 3: Edellä esitetyn iteraation lopputila.

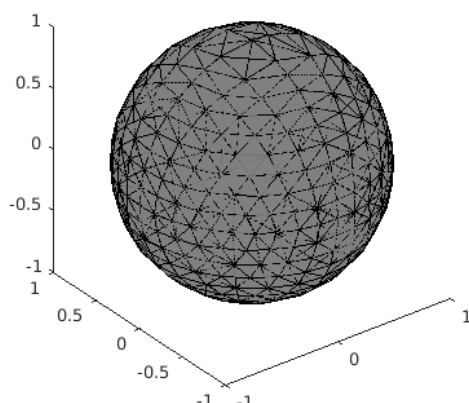
2 Pisteiden generointi ellipsoidin pinnalle

Jotkin ongelmat vaativat tutkittavan kappaleen pinnan mallintamista pistejoukkona. Tämä on helppo ensiaskeleksi esimerkiksi kappaleen muodon kolmioinnissa. Tässä työssä keskitytään generointimenetelmiin, joita voidaan hyödyntää ellipsoidien mallinnuksessa.

Pisteiden generointiin käytettävät menetelmät voidaan jakaa karkeasti kahteen luokkaan: systemaattisiin ja satunnaisiin menetelmiin. Kummassakin pyritään yleensä luomaan mahdollisimman tasainen jakauma kappaleen pinnalle. Seuraavissa aliluvuissa esitellään yksi systemaattinen ja yksi satunnainen menetelmä.

2.1 Pisteiden luominen kerroksittain

Eräs yksinkertainen tapa luoda ellipsin pinnalle pistejoukko, jossa lähimpien pisteiden etäisyydet pysyvät likimain vakioina, on generoida pisteet ellipsin leveyspiirien suuntaisina kehinä. Ellipsi jaetaan k kerrokseen, jolloin navalla olevaan ensimmäiseen ”kerrokseen” tulee yksi piste, seuraavaan kerrokseen 4 ja tästä kohti päiväntasaajaa kuhunkin kerrokseen aina 4 pistettä enemmän kuin edelliseen. Päiväntasaajan jälkeen pisteiden määrä



Kuva 4: Matlabin `delanayn`-komentoa käyttäen luotu kuva kerroksittain luodusta pistejoukosta ykikköpallon pinnalla.

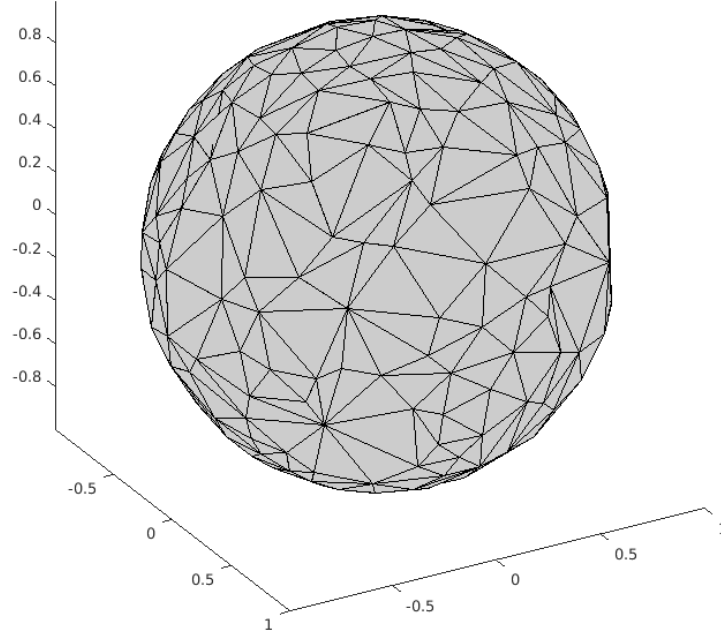
alkaa jälleen vähentyä neljällä pisteellä kerrosta kohti, kunnes vastakkaisella navalla on jälleen vain yksi piste.

Tämä on yksinkertaisinta tehdä (θ, ϕ, r) -koordinaatistossa, jossa θ mittaa kulmaa navalta kohti päiväntasaajaa ja ϕ kulmaa x-akselista positiiviseen kiertosuuntaan z-akselin ympäri. Tällä tavoin generoitaessa kunkin rivin kulmaetäisyydeksi θ -suunnassa muodostuu π/k ja kahden saman latitudin pisteen välinen etäisyys ϕ -suunnassa on $\frac{\pi}{2(k-1)}$.

Näistä pisteistä on yksinkertaista muodostaa ellipsin pinnan kattava kolmiointi, jossa napa toimii kärkenä neljälle kolmiolle ja seuraavan rivin pisteet kussakin oktantissa yhdistävät janat kolmioiden kantoina. Seuraavalla kolmiorivillä edellisen kolmiorivin kantoja vastaa aina kärki ja kunkin oktantin reunoihin muodostuu kaksi uutta kolmiota. Näin jatketaan jälleen päiväntasaajalle asti, minkä jälkeen kolmiot alkavat jälleen vähentyä napaa lähestyttäessä. Kuva 200 kerrosta sisältävästä näin luodusta pistejoukosta on nähtävillä kuvassa 4.

2.2 Pisteiden arpominen

Pisteiden generointi satunnaisesti on hyvin yksinkertainen tapa generoida vähällä vaivala halutunkokoinen pistejoukko. Pallopinnalle generointi on erittäin yksinkertaista. Kun tiedetään haluttu pallon säde, voidaan generoida satunnaisluvut $\theta \in [0, \pi]$ ja $\phi \in [0, 2\pi]$, jolloin saadaan suoraan piste (θ, ϕ, r) missä r on vakio. Esimerkki näin saadusta pistejoukosta on nähtävillä kuvassa 5.



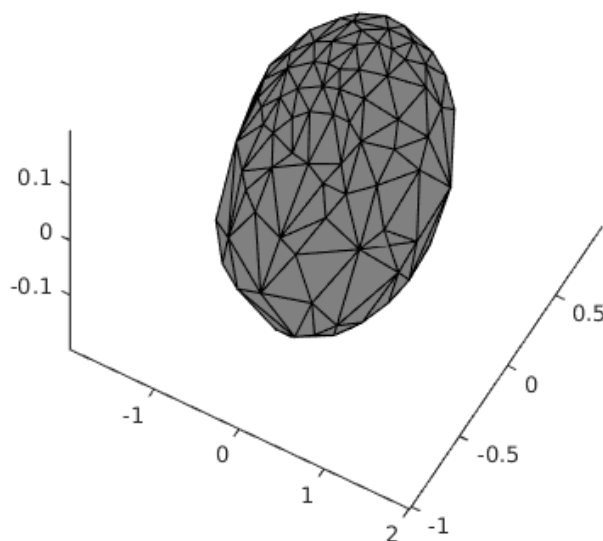
Kuva 5: Yksikköpallon pinnalle satunnaisgeneroitu pistejoukkoon tehty Delaunayn kolmiointi. Pallo koostuu 250 pisteestä ja 505 niiden välille piirretystä kolmiosta.

Ellipsoidin pinnalle generoiminen on kuitenkin monimutkaisempaa, sillä edellä kuvatulla tavalla pistejakaumasta tulee ellipsoidin eksentrisyydestä riippuen mahdollisesti hyvinkin epätasainen. Pohjimmiltaan voidaan kuitenkin käyttää samantyyppistä menetelmää kuin edellä, eli aloittaa generoimalla vastaavat kulmat θ ja ϕ ja määrittämällä näiden perusteella kyseisen pinnan kohdan etäisyys origosta kaavan 1 perusteella, missä a , b ja c ovat ellipsin muodon määräävät parametrit ellipsin yhtälöstä (yhtälö 2).

$$r(\theta, \phi) = \frac{abc}{\sqrt{b^2c^2 \sin^2(\phi) \cos^2(\theta) + a^2c^2 \sin^2(\phi) \sin^2(\theta) + a^2b^2 \cos^2(\phi)}} \quad (1)$$

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \quad (2)$$

Eräs tapa huolehtia jakauman tasaisuudesta halutulla tarkkuudella on asettaa tietty kynnyksarvo kunkin pisteen ympärillä olevalle säteelle, jonka sisälle ei generoida uusia pisteitä. Tämä voidaan toteuttaa esimerkiksi jakamalla ellipsin pinta-ala halutulla pistemäärällä



Kuva 6: Satunnaisgeneroidun pistejoukon Delaunayn kolmiointi ellipsin ($a=1,1$; $b=0,8$; $c=0,2$) pinnalla. Ellipsoidi koostuu 150 pisteestä ja 296 niiden välille piirretystä kolmiosta. Käytetty tasaisuuskerroin on 1,2.

ja määrittämällä tätä alaa vastaavan ympyrän säde, jolloin voidaan hylätä tätä sädettä lähempänä toista pistettä olevat pisteet. Uusia pisteitä arvotaan, kunnes hyväksytyjen pisteiden kokonaismäärä on haluttu pistemäärä.

Tätä menetelmää voidaan edelleen kehittää määrittämällä tasaisuuskerroin, jolla kerrotaan saatu ala kutakin pistettä kohden. Pienillä kertoimilla pisteet voivat olla lähempänä toisiaan, jolloin pisteitä hylätään vähemmän eli generointi on nopeampaa, mutta jakautuman tasaisuus kärsii. Liian suuren kertoimen valitseminen saattaa johtaa tilanteeseen, jossa uusien kaikki ehdot täyttävien pisteiden generointi on mahdotonta. Esimerkki ellipsoidin pinnalle generoiduista pisteistä on nähtävillä kuvassa 6.

3 Lommel-Seeliger -heijastuslaki

Lommel-Seeliger -heijastuslaki käsittelee yksinkertaista heijastusta, joka tapahtuu puoliäärettömässä väliaineessa, johon valo penetroituu heikentyen absorption ja sironnan vaikutuksesta. Sirottuminen kustakin väliainekerroksesta on isotrooppista ja pintaa kohti siroonnut valo heikkenee edelleen matkalla pintaan. [3]

Tällöin heijastuskertoimeksi saadaan

$$R_{Lommel-Seeliger} = \frac{1}{4} \tilde{\omega} P_{11}(\alpha) \frac{1}{\cos(\epsilon) + \cos(\iota)}, \quad (3)$$

missä ϵ on havaitsijan suunnan ja heijastavan pinnan normaalin välinen kulma ja ι on valonlähteen suunnan ja heijastavan pinnan normaalin välinen kulma. [6, 3].

4 Asteroidin muodon kolmiointi

Käytin asteroidin muodon kolmiointiin edellä esitettyä Quickhull-algoritmia. Toteutin pisteiden generointiin sekä mahdollisuudet generoida pisteet kerroksittain antamalla kerroksien määrä että mahdollisuuden generoida pisteet satunnaisesti antamalla pisteiden kokonaismäärä ja tasaisuuskerroin. Kummassakin tapauksessa annetaan lisäksi ellipsin muodon määräävät vakiot a , b ja c .

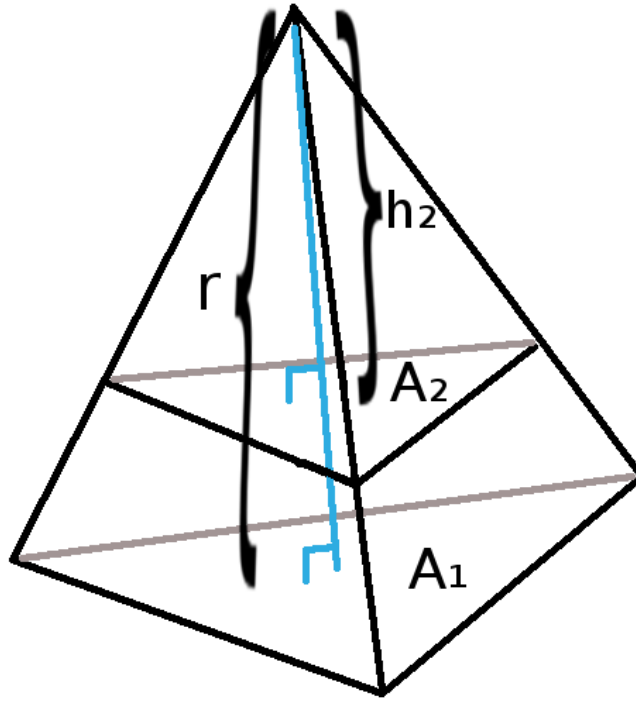
Luvussa 1.1.1 esitetystä poiketen ylläpidin lisäksi listaa valmiista kolmioista. Varsinaista tapaa tietää, koska kolmiota on käsitelty viimeisen kerran ei ole, sillä vaikka kolmiolla ei ole omia näkyviä pisteitä, se saattaa olla nähtävissä jonkin naapurinsa näkyvästä pisteestä, jolloin se tullaan vielä poistamaan. Tästä syystä lisäsin kaikki uudet kolmiot luotaessa valmiiden listaan ja poistin sieltä aina iteraation lopuksi työstettävän kolmion ja valoisat kolmiot. Näin käsiteltävien kolmioiden pinon tyhjentyessä, sisältää valmiiden kolmioiden lista kaikki pinnan Delaunay-kolmiointiin kuuluvat kolmiot.

Ohjelma kokonaisuudessaan on nähtävissä ja ladattavissa Githubissa (<https://github.com/aaajarven/aurfys/tree/master/Delaunay>). Projekti sisältää myös debuggauksessa ja visualisointien luonnissa käytettävää koodia, muun muassa tiedostojen kirjoittamiseen käytetyn IO-luokan. Animaatioita toteuttamani Quickhull-algoritmin toiminnasta on nähtävissä osoitteessa <https://www.cs.helsinki.fi/u/aaajarven/delaunay/>.

5 Asteroidin sisäinen rakenne

Kun asteroidin pinta on kolmioitu, voidaan asteroidin sisärakenne kattaa tetraedreillä yksinkertaisesti käyttämällä kutakin kolmiota yhden tetraedrin pohjana ja ellipsoidin keskipistettä tetraedrin neljäntenä kärkenä. Paikkaresoluution parantamiseksi voidaan nämä tetraedrit edelleen jakaa ellipsoidin säteen suunnassa katkaistuihin kartioihin.

Yksinkertaisinta on tehdä jako siten, että kunkin katkaistun kartion korkeus on sama.



Kuva 7: Katkaistujen kartioiden luomisessa käytetty notaatio.

Tällöin kuitenkin monitahokkaiden tilavuus vaihtelee säteen suunnassa: lähellä keskipistettä tilavuudet ovat pieniä ja lähellä pintaa hyvin suuria. Tästä syystä monissa sovelluskohteissa on hyödyllisempää suunnitella jako monitahokkaisiin siten, että niiden tilavuudet pysyvät likimain samoina.

Tällöin kunkin katkaistun kartion korkeuden laskeminen onnistuu, kun tunnetaan lausekkeet kartion tilavuudelle

$$V = \frac{1}{3}Ah \quad (4)$$

missä A on kartion pohjan ala ja h kartion korkeus.

Tarkastellaan tilannetta, jossa tehdään lähinnä ellipsoidin pintaa oleva leikkaus siten, että katkaistuun kartioon jää osuus $1/n$ koko kartion tilavuudesta. Laskut on kuitenkin helpompi suorittaa, jos tarkatellaan yhden katkaisemattoman ja yhden katkaistun kartion sijaan kahta katkaistua kartiota: kuvan 7 notaatiolla kartioita joiden pohjat ovat A_1 ja A_2 ja joilla on yhteinen neljäs kärki. Tällöin suuremman kartion tilavuudelle V_1 ja pienemmän kartion tilavuudelle V_2 pätee

$$V_2 = \frac{n-1}{n}V_1$$

eli sijoittamalla yhtälö 4

$$\frac{1}{3}A_2h_2 = \frac{n-1}{n}\frac{1}{3}A_1r. \quad (5)$$

Kartioiden korkeuksien suhde määräytyy toistaiseksi tuntemattomalla tavalla kartioiden lukumäärän perusteella. Merkitään tätä suhdetta k_2 siten, että $r = k_2 h_2$. Kartioiden pohjat ovat yhdenmuotoisia ja niiden pinta-ala on verrannollinen keskipisteestä mitatun etäisyyden neliöön, siis $A_2 = k_2^2 A_1$. Sijoitetaan nämä yhtälöön 7, jolloin saadaan:

$$\frac{1}{3} A_1 k_2^2 r k_2 = \frac{n-1}{n} \frac{1}{3} A_1 r. \quad (6)$$

Jaetaan puolittain pois molemmilla puolilla esiintyvät termit:

$$k_2^3 = \frac{n-1}{n} \quad (7)$$

Näin ollen ensimmäisen katkaistun kartio, jossa on osuus $1/n$ koko kartion tilavuudesta, pienempi pohja on etäisyydellä $\sqrt[3]{\frac{n-1}{n}} r$ ellipsoidin keskipisteestä. Seuraavan katkaistun kartion suurempana pohjana toimii edellisen kartion pienempi pohja ja pienemmän pohjan etäisyys keskipisteestä voidaan laskea vastaavalla menettelyllä kuin äsken. Seuraavan jaon tapauksessa kuitenkin ”kokonaisen” kartion tilavuus on $V_2 = \frac{n-1}{n} V_1$ ja siitä halutaan leikata pienempään kartioon osuus $\frac{n-2}{n-1}$, jolloin pohjien $A-2$ ja A_3 väliin jäävään katkaistuun kartioon jää jälleen tilavuus $\frac{1}{n} V_1$. Päättelyketju voidaan muilta osin toistaa jälleen samanlaisena.

Yleiselle m . kartiolle (pohjan ala A_m , korkeus $h_m = r k_m$) pätee

$$\frac{V_m}{V_1} = \frac{n-(m-1)}{n} = \frac{n-m+1}{n}.$$

Sijoittamalla tähän kartioin tilavuuden lausekkeet, saadaan

$$\frac{1}{3} A_m h_m = \frac{n-m+1}{n} \frac{1}{3} A_1 r.$$

Sijoitetaan tähän $h_m = k_m r$ ja $A_m = A_1 k_m^2$ ja supistetaan pois molemmilla puolilla esiintyvät termit:

$$k_m^3 = \frac{n-m+1}{n}. \quad (8)$$

Yhtälön 8 perusteella kartio saadaan jaettua n tilavuudeltaan yhtä suureen osaan, kun sitä leikataan pohjan kanssa yhdensuuntaisilla tasoilla siten, että leikkaavien tasojen etäisyys h_m kartion kärjestä on

$$h_m = \sqrt[3]{\frac{n-m+1}{n}} r, \quad (9)$$

missä m on kokonaisluku väliltä $[1, n]$.

6 Asteroidin integroitu kirkkaus

Kolmioidun asteroidin integroitu kirkkaus saadaan yksinkertaisesti laskemalla yhteen havaitsijan havaitsemat kirkkaudet eri kolmioista. Toteutin tämän siten, että aluksi laskeaan kolmion normaali, joka suuntautuu pois päin origosta. Kolmion kärkipisteiden perusteella mahdollisia normaaleja on kaksi, ja näistä valitaan se, jonka pistetulo origon (origo on väistämättä origokeskisen ellipsoidin sisällä) ja jonkin kolmion kärkipisteen välisen paikkavektorin kanssa on yli nolla eli vektorien välinen kulma ei ole yli 90° . Tämän normaalivektorin ja Auringon sekä havaitsijan paikkavektorien (asteroidikeskeisessä koordinaatistossa) välisistä kulmista saadaan kulmat ϵ ja ι . Mikäli kumpi tahansa kulmista ϵ ja ι on välin $[-\pi/2, \pi/2]$ ulkopuolella, voidaan kirkkaus asettaa suoraan nolnaan, sillä kolmio ei joko ole havaittavissa tai se ei ole valaistu. Muussa tapauksessa näistä voidaan edelleen laskea yhtälön 3 mukaisesti heijastuskerroin.

Kirkkauden laskemisen yksinkertaisuudesta huolimatta jäi toteutukseeni bugi, jonka takia sen antamat kirkkauksien arvot eivät ole fysikaalisia, eikä minulla valitettavasti ole aikaa etsiä enää tätä bugia. Rajoitutaan tästä syystä tarkastelemaan teoreettista mallia kirkkaudelle.

Analyttisen mallin perusteella Lommel-Seeliger -heijastuslain mukaan valoa heijastavan asteroidin kirkkaus määräytyy seuraavasti:

$$L_{Lommel-Seeliger} = \frac{1}{8} \pi F_0 \tilde{\omega} P(\alpha) abc \frac{S_\odot S_\oplus}{S} \{ \cos(\lambda' - \alpha') + \cos \lambda' + \sin \lambda' \sin(\lambda' - \alpha') \times \\ \times \ln \left[\cot\left(\frac{1}{2} \lambda'\right) \cot \frac{1}{2}(\alpha' - \lambda') \right] \}, \quad (10)$$

missä α on havaitsijan ja valonlähteen välinen kulma asteroidilta katsottuna ja muut vakiot on määritetty seuraavasti, kun \vec{e}_\odot on valonlähteen suunnan kertova yksikkövektori

ja \vec{e}_\oplus havaitsijan suunnan kertova yksikkövektori:

$$\begin{aligned}
S_\odot &= \sqrt{e_\odot^T C e_\oplus} \\
S_\oplus &= \sqrt{e_\oplus^T C e_\odot} \\
\cos(\alpha') &= \frac{e_\odot^T C e_\oplus}{S_\odot S_\oplus} \\
\sin(\alpha') &= \sqrt{1 - \cos^2(\alpha')} \\
S &= \sqrt{S_\odot^2 + S_\oplus^2 + 2S_\odot S_\oplus \cos(\alpha')} \\
\cos(\lambda') &= \frac{S_\odot + S_\oplus \cos(\alpha')}{S} \\
\sin(\lambda') &= \frac{S_\oplus \sin(\alpha')}{S}.
\end{aligned}$$

Näissä esiintyvä C on määrittää ellipsin muodon. Se on muutoin nollamatriisi, mutta sen diagonaalilla on alkiot $\frac{1}{a^2}$, $\frac{1}{b^2}$ ja $\frac{1}{c^2}$. [5]

Viitteet

- [1] Paolo Cignoni and Roberto Montani, Claudio ja Scopigno. Dwall: A fast divide and conquer delaunay triangulation algorithm in ed. *Computer-Aided Design*, 30(5):333–341, 1998.
- [2] Thomas Diewald. Convex hull 3D – quickhull algorithm. [online; luettu 20.5.2016].
- [3] M. B. Fairbairn. Planetary Photometry: The Lommel-Seeliger Law. *Journal of the Royal Astronomical Society of Canada*, 99:92, June 2005.
- [4] Pavel Maur. Delaunay triangulation in 3d. Technical report, Technical Report No. DCSE/TR-2002-02, University of West Bohemia in Pilsen, 2002.
- [5] K. Muinonen and K. Lumme. Disk-integrated brightness of a lommel-seeliger scattering ellipsoidal asteroid. *Astronomy & Astrophysics*, 584:A23, 2015.
- [6] Karri Muinonen. Fundamentals for theoretical photometry of asteroids, 2014.
- [7] Samuel Peterson. Computing constrained delaunay triangulations.