

Fotogrammetrian seminaari

Kolmiulotteisen pinnanmuodostuksen asettamat vaatimukset 3D-digitoinnin suorittamiseen

Johdanto	3
Alkueoletukset pisteaineistolle	3
Kolmiopinnan muodostaminen	5
2D-Marching cubes.....	5
3D-Marching cubes.....	6
Marching cubes sovellettuna käytäntöön	7
Tangenttipinnan estimointi.....	8
Verkon optimointi	9
Valinta tarkkuuden ja kompaktiuden välillä	9
Energiafunktion määrittely.....	10
Tulokset.....	13
Päätelmät.....	13
viitteet.....	15

Johdanto

Pinnanmuodostukseen pisteaineistosta on kolme eri algoritmia:

- Delanykolmiointi, joka toimii vain 2D+1D pinnoille.
- Pisteitten yhdistys mittausjärjestyksessä eli peräkkäin mitatut pisteet ovat aina rinnakkaisia.
- Marching cubes, jossa pisteaineistosta muodostetaan kuutiohila.

Delany ei sovellu aidolle 3D-kappaleelle ja jos pisteaineisto on mitattu järjestyksessä ei tarvita muuta kuin alkeellinen yhdistysalgoritmi. Jos pisteaineisto on mitattu useasta eri suunnasta ja mahdollisesti eri mittausmenetelmillä, on marching cubes ainut mahdollinen algoritmi pinnan muodostukseen. Sen päälle on tehty useita pinnan muotoa optimoivia ratkaisuja. mutta kaikki perustuvat samaan kuutiohilaan.

Alkuoletukset pisteaineistolle

Jos aluksi oletamme pisteaineiston järjestämättömäksi, emme voi sanoa varmuudella, mikä piste on kunkin pisteen naapuri tai pinnan piste. Näin on, jos mittaukset on tehty useasta suunnasta ja pisteaineistot on yhdistetty. Peräkkäisten ja järjestettyjen pisteiden muuntaminen pinnaksi on yksinkertainen tehtävä, jossa vierekkäiset pisteet voidaan helposti liittää toisiinsa miettimättä pinnan muotojen muutoksia.

Kun pisteaineisto on koottu useasta eri suunnasta, täytyy pisteaineisto ensin koota yhdeksi ja järjestää 3D-avaruuteen. Samalla voidaan karsia myös häiriöpisteet mallilta. Mittausta tehtäessä on suunniteltava, kuinka tarkka malli pinnasta halutaan eli mitkä ovat pienimmät näkyvät muodot. Pinnan muodostuksessa määritellään r joka on pisteiden etäisyys toisistaan ja δ joka on etäisyyksien vaihteluväli.

Parametria r δ δ käytetään useaan eri tarkoitukseen. Parametri määrittää pisteitten tiheyden sekä tiheyden vaihtelun seuraavasti:

- Se määrittelee pisteiden naapuruuden $Nbhd(x_i)$ estimoitaessa tangenttitasoja pinnalle.

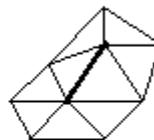
$$\{i\} \in K \quad \cdot$$

$$nbhd(\{i\}; K)$$



$$\{i, j\} \in K \quad /$$

$$nbhd(\{i, j\}; K)$$



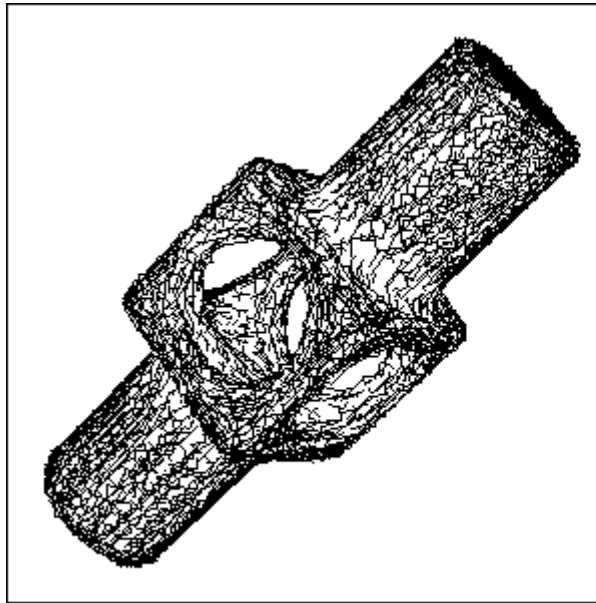
$$\{i, j, k\} \in K \quad \triangle$$

$$nbhd(\{i, j, k\}; K)$$

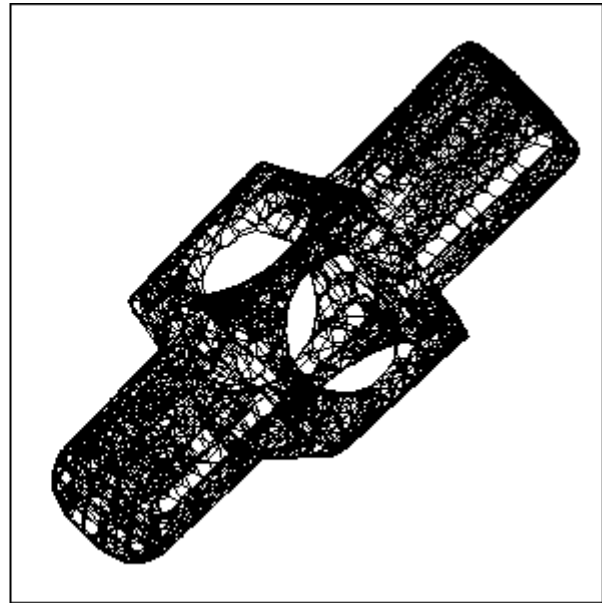


- Sitä käytetään tangenttipintojen yhdistämiseen toisiinsa Riemanin kuvaajassa.

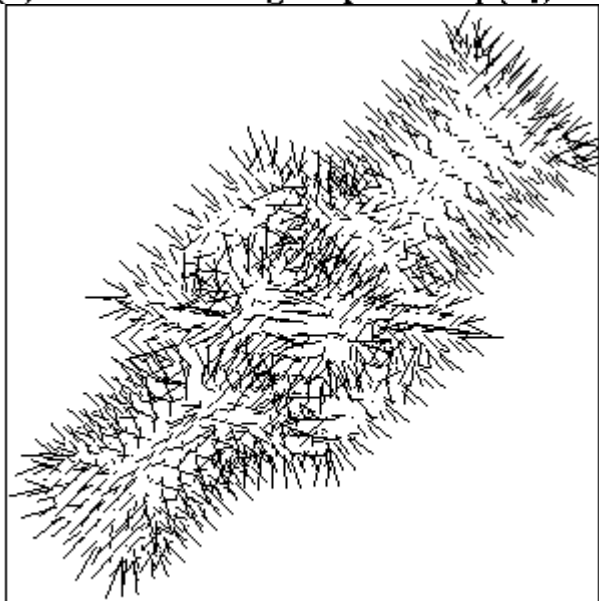
- Se määrittää onko, arvioitaessa määrättyä etäisyysfunktioita $\tilde{d}_V(p)$, p :n projektiio lähimmälle tangenttipinnalle rajan sisällä.
- Sitä käytetään määrittämään pienimpiä kappaleessa esiintyviä muotoja.



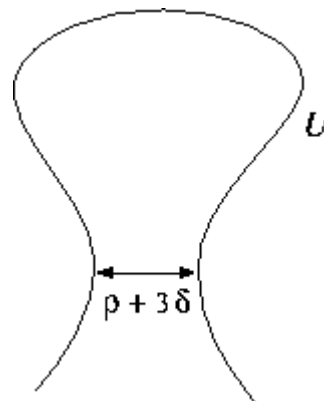
(a) Estimated tangent planes $Tp(x_i)$



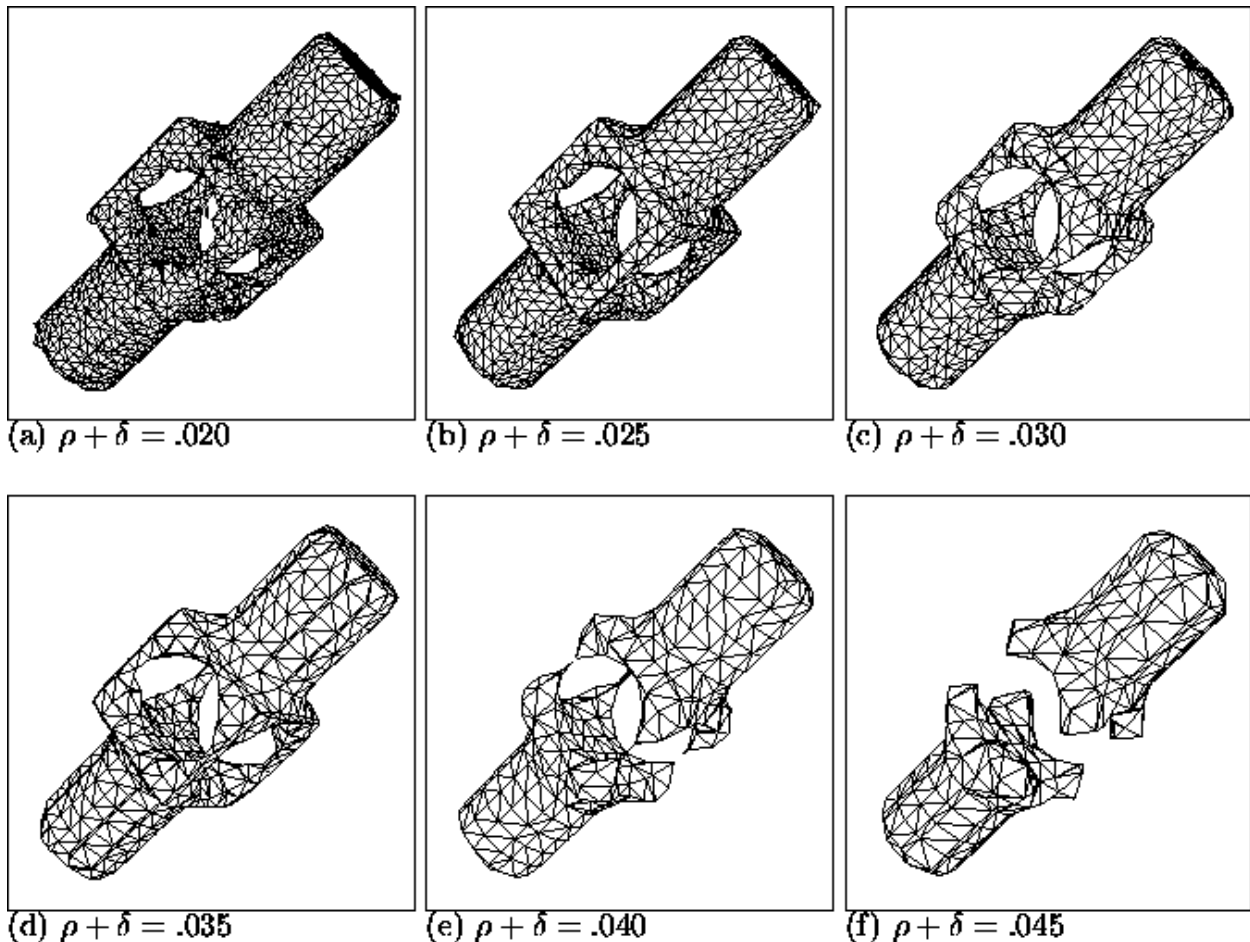
(b) Riemannian Graph on α_i



(b) Estimated signed distance (shown as $p \leftrightarrow z$)



Parametrin ? ? ? vaikutus ensimmäisessä vaiheessa muodostettuun verkkoon



Kolmiopinnan muodostaminen

Kolmiopinnan muodostamiseen tasopinnalle on useita eri algoritmeja, joista yleisimmin tunnettu on Delany-kolmiointi. Kirjallisuudessa sanotaan sen soveltuvan kolmiulotteiseen pintaan vaihtamalla ympyrät palloiksi, mutta todellisuudessa sitä voidaan käyttää vain 2D+1D-tapauksissa, koska Delany-kolmiointissa pinnat kasvavat yhteen. Ainoa laajasti tunnettu tapa pinnan muodostukseen 3D-mallista on Marching cubes tai sen optimoidut toteutukset. Marching cubes on William E. Lorenzen ja Harvey E. Clinen kehittämä pinnansovitusalgoritmi 3D-mallien muodostukseen. Edellisessä vaiheessa muodostettua tangenttipintaa voidaan käyttää kappaleen pinnan rajaukseen.

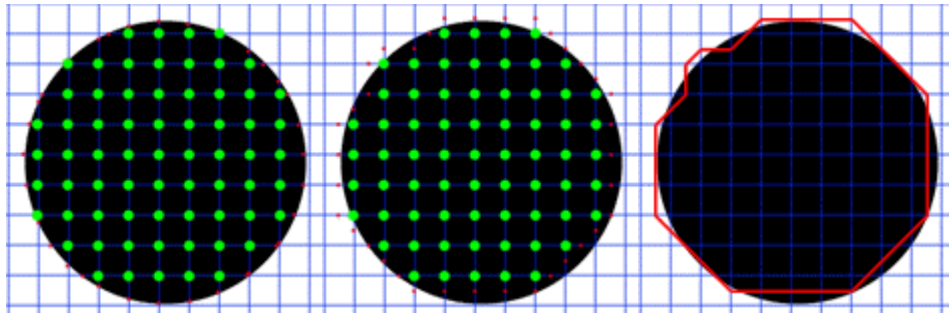
Toteutus on kuvattu 2D- ja 3D-muodossa, koska asian hahmottaminen on paljon helpompaa, jos se esitetään ensin 2D-tilassa.

2D-Marching cubes

Jos pisteellä, joka on pienempi kuin raja-arvo, on naapuripiste, joka on suurempi kuin raja arvo, on pinnan on oltava pisteitten välissä. Kun käytetään ensimmäisen kuvan data, niin algoritmi määrittää ensin, mitkä pisteet ovat sisäpuolella. Seuraavaksi määritetään, mitkä neliön sivut ovat sisällä osittain tai kokonaan. Nyt voidaan valita sivut, jotka leikkaavat reunaviivaa.

Yksinkertaistuksena oletetaan, että sivu leikkaa reunaviivaa aina sivun keskikohdasta. Tarkempi

ja parempia määrittämissä menetelmillä olisi määrittää sivulle 3 tai 5 leikkaus kohtaa reunaviivalle. Menetelmällä saadaan pisteet, joiden kautta voidaan piirtää reunaviivan approksimaatio.



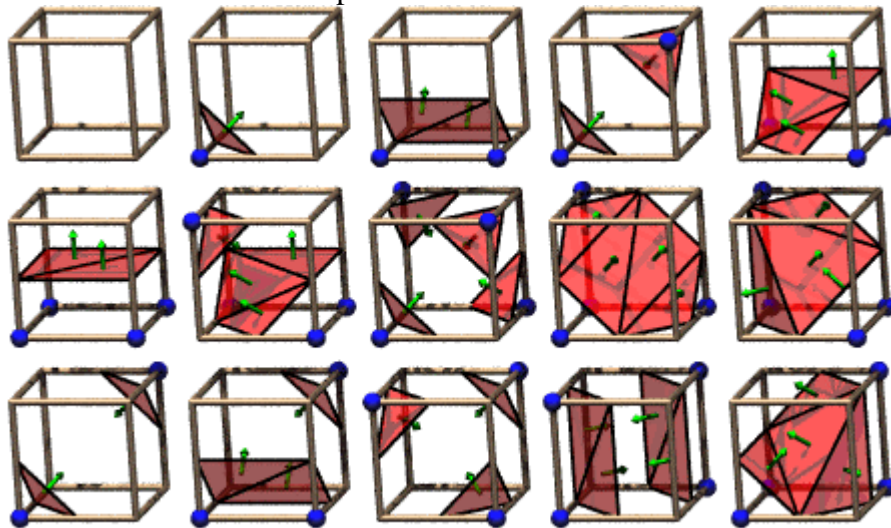
3D-Marching cubes

Sama algoritmi toimii myös 3D-tilassa. Tekniikka on sama kuin 2D-mallissa, mutta

- neliöt on muutettu kuutioiksi
- tuloksena ei saada viivoja vaan kolmiopintoja.

Kuutiossa on 8 kulmaa, jokainen kulma voi olla joko sisä- tai ulkopuolella pinnasta eli saadaan 256 vaihtoehtoista tapausta pisteitten sijainnille pinnan suhteen. Koska sivu voi kohdata pinnan vain kerran, saadaan 256 mahdollista pinnanmuodostustapausta kuutiosta. Tapausten määrä voidaan pelkistää 256 tapauksesta 15 tai 14 tapaukseen, jos jätetään pois tapaukset, joissa yksikään piste ei ole pinnalla. Yksinkertaistus voidaan suorittaa käsittelemällä samana tapauksena tilat, joissa tapaus voidaan muodostaa toisesta tapauksesta:

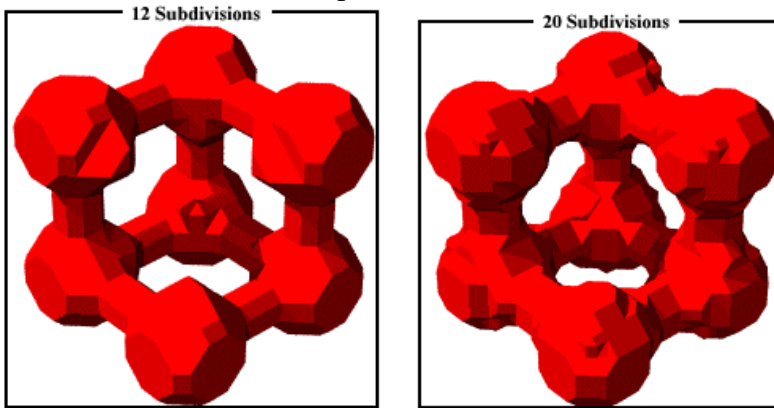
- kiertämällä minkä tahansa x,y,z-akselin ympäri,
- pelaamalla minkä tahansa x,y,z-akselin suhteen tai
- kiertämällä kaikkia pisteitä.



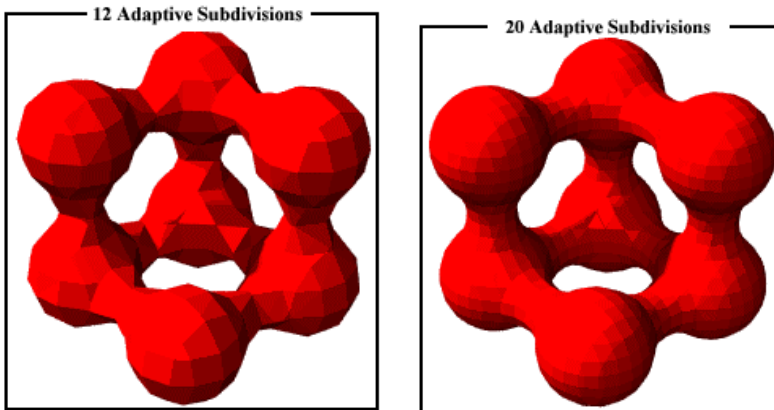
The 15 Cube Combinations

Tapausten yhdistäminen vähentää merkittävästi algoritmin monimutkaisuutta, koska kaikki tapaukset voidaan muodostaa 15 perustapauksesta yksinkertaisilla operaatioilla. Kuvassa sininen piste kuvaa pistettä, joka on pinnan sisällä ja vihreä nuoli pinnan normaalin suuntaa. Kolmioitten laskentaan voidaan yksinkertaisimmillaan olettaa sivujen leikkaavan aina keskikohdassa, mutta se antaa epätarkan tuloksen, koska kolmioitten asennoille on vain

rajallinen määrä vaihtoehtoja. Parempaan tulokseen päästään pienentämällä kuutioita tai jakamalla kuution sivu useampaan osaan.



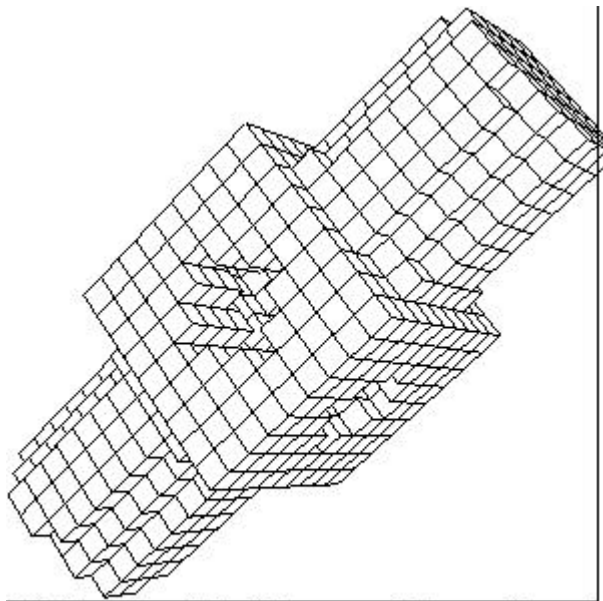
Huonon approksimaation vaikutukset voidaan korjata vain paremmalla algoritmilla määrittämällä kolmion pinnat kuution sisällä. Yksi nopeimmista ja käytetyimmistä menetelmistä on *linear interpolation*. Jokaiselle pinnalle annetaan interpolaatioarvo, minkä jälkeen pinnan asento on helppo määrittellä samaksi kuin raja-arvopinnan.



Kolmion nurkkapisteitten gradientit voidaan laskea interpoloimalla kuution nurkkapisteitten gradienteista.

Marching cubes sovellettuna käytäntöön

Tarkan approksimaation muodostamiseksi pinnasta ja sen muodosta määritellään pisteistölle kuutiohila, jossa kuution sivun pituuden on oltava h . Jokaisen kuution sisällä on vähintään yksi piste ja algoritmin nopeuttamiseksi pinnan etsinnässä tutkitaan vain naapurikuutioiden pisteitä. Pinnan määrittämiseksi lasketaan määrätty etäisyys lähimmästä hilapisteestä mitatun pisteen tangenttipinnalle. Jos etäisyys hilapisteestä pinnalle on suurempi kuin h , niin piste ei voi olla pinnan piste. Pisteitten etäisyyksistä approksimoidaan pienimmän neliösumman menetelmällä pinta, jonka muoto jäljittelee alkuperäistä kappaletta.



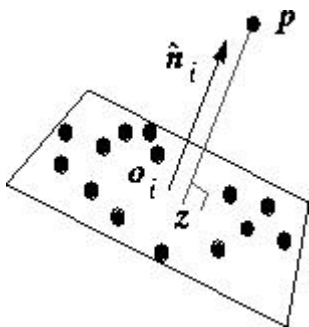
(a) Cubes visited by marching cubes

Ensimmäisen vaiheen tulos on approksimoitu verkko $Z(\tilde{d}_U)$. Verkko on tiheä ja sillä on sama topologia kuin pinnalla U , mutta se ei ole tarkka malli alkuperäisestä kappaleesta.

Tangenttipinnan estimointi

Verkon optimointiin tarvitaan malli pinnan tangenttipinnan muutoksista, joilla voidaan määrittää pinnan muotoa. Ensimmäiseksi lasketaan orientoitu tangenttipinta jokaiselle pisteelle. Valitaan pisteen läheisyydestä k pistettä joukosta X ja pienimmän neliösumman menetelmällä etsitään parhaiten sopiva tangenttipinta pisteelle. Samalla tehdään myös läheisten pintojen keskinäinen

orientointi. Orientointia varten jokaiselle pinnalle määrätään painoarvo $1 - |\hat{n}_i \cdot \hat{n}_j|$, jossa n on pinnan tangenttivektori. Jos $n_i \approx n_j$, ovat tangenttipinnat lähes toistensa suuntaisia. Pintojen keskinäiseen orientointiin muodostetaan kuvaajan optimointitehtävä, joka ratkaistaan käyttämällä "ahne"(greedy) -algoritmia. Algoritmi pyrkii levittämään orientoinnin aina siihen suuntaan, jossa pinnat ovat lähes toistensa suuntaisia ja välttämään alueita, joilla orientointi on vaikeampaa.



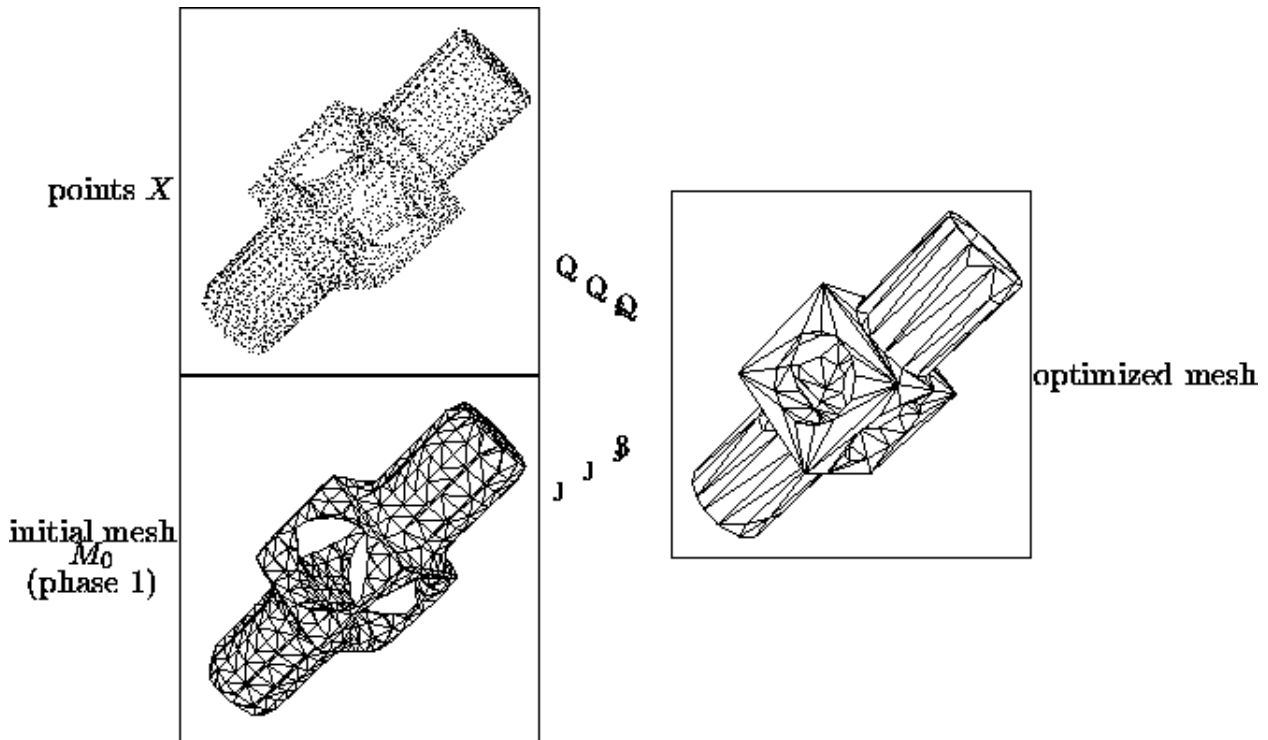
Määrätty etäisyysfunktio

Määrätty etäisyysfunktio on mielivaltaisen pisteen ja pinnan välinen lyhin etäisyys kerrottuna $\frac{1}{|n|}$:llä. Jos piste on kauempana pinnasta kuin $\frac{1}{|n|}$, se ei voi olla pinnan piste, koska pinnalla pisteiden etäisyys toisistaan on $\frac{1}{|n|}$ ja virhemittauksessa $\frac{1}{|n|}$.

Verkon optimointi

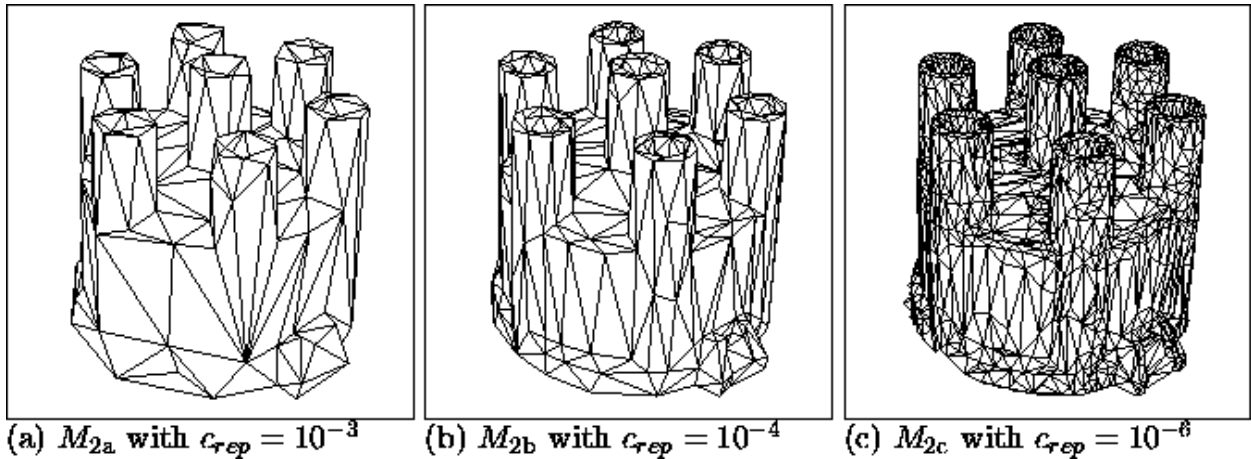
Marching cubes antaa pinnasta hyvän ja tarkan pintamallin, joka jäljittelee pinnan topologiaa, mutta se ei ole optimoinut pintamallia ja verkon optimointi joudutaan tekemään erikseen. Optimoinnilla poistetaan tarpeettomia kolmio pintoja ja tihennetään verkkoa sopivilta alueilta.

Verkon optimoinnin ratkaisemiseksi minimoidaan *energiafunktio* (*energy function*), jossa on vastakkain kompakti verkko ja tarkka malli pinnasta. Käyttämällä vaiheessa 1 saatua verkkoa M_0 lähtökohdaksi minimoidaan epälineaarinen *energiafunktio* vaihtelemalla sivujen määrää, pisteiden sijainteja ja niiden välisiä kytkentöjä (ehtona on että alkuperäinen topologia säilyy). Oletuksena on että optimointi käy läpi kaikki verkot, joilla on sama topologia kuin verkolla M_0 . Parhaan mahdollisen verkon löytymistä ei voida taata, mutta tulos on kuitenkin riittävän hyvä.



Valinta tarkkuuden ja kompaktiuden välillä

Verkon optimoinnissa valinta tarkkuuden ja kompaktin välillä on käyttäjän määrittelemällä parametrilla c_{rep} . Suurella c_{rep} arvolla tulee verkosta karkea ja pienellä kompakti. Arvo on suhteutettava pisteitten mittaustiheyteen eikä verkon tiheyttä voi kasvattaa yli mittaustarkkuuden rajan.



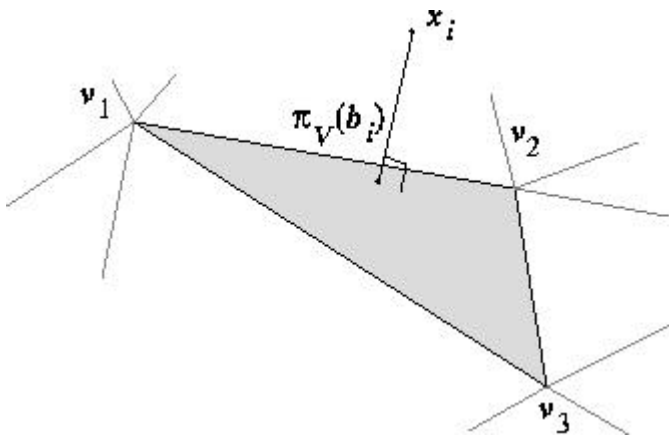
Energiafunktion määrittely

Verkon optimoinnin tarkoituksena on tehdä verkko, joka on hyvä sovitus mitatuille pisteille X ja jonka sivujen määrä on mahdollisimman pieni. Etsitään ryhmä K ja joukko sivuja V määrittämään verkko $M=(K,V)$, joka minimoi *energiafunktion* niin, että se vastaa asetettuja vaatimuksia.

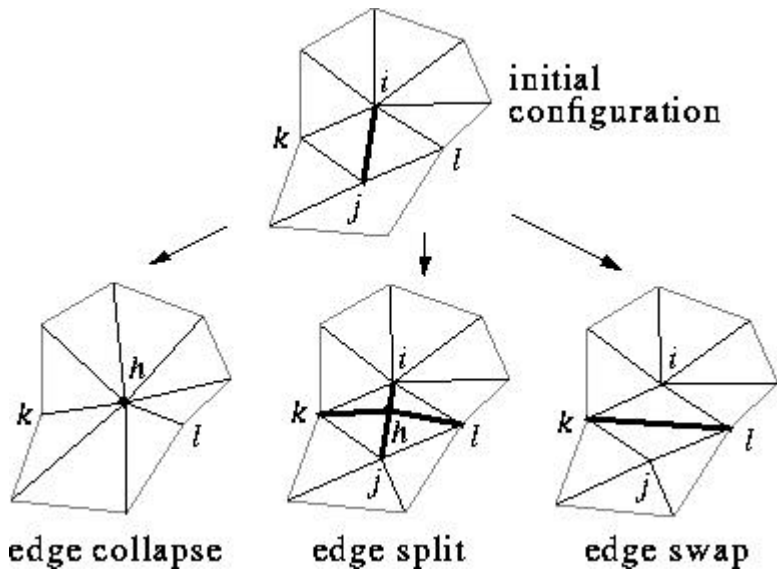
$$E(K,V) = E_{\text{dist}}(K,V) + E_{\text{rep}}(K,V) + E_{\text{spring}}(K,V).$$

i

E_{dist} on pisteiden etäisyyksien neliösumma verkosta M_0 , joka rankaisee, kun pisteiden topologia poikkeaa verkon topologiasta.



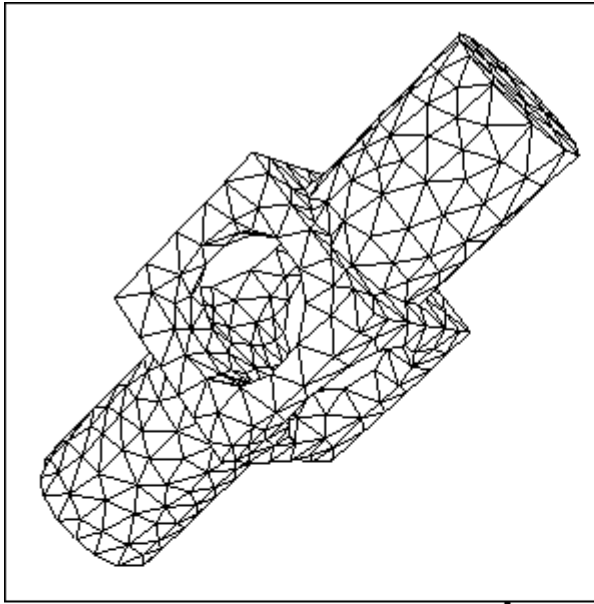
E_{rep} rankaisee sivujen määrästä ja mahdollistaa sivujen poistamisen.



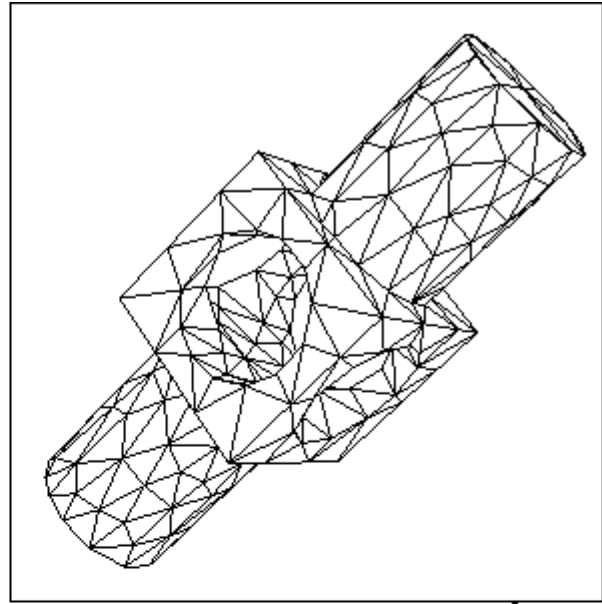
Optimointi mahdollistaa sivujen poiston ja lisäyksen verkkoon. Kun sivu lisätään, E_{dist} pienenee. Samalla E_{rep} rankaisee operaatiosta niin, että sivuja ei lisätä loputtomiin.

Vastaavasti jos sivu poistetaan, kasvaa E_{dist} -arvo ja E_{rep} -arvo pienenee. Käyttäjä määrittelee arvon, jolla voidaan säädellä verkon tiheyttä ja mallin tarkkuutta.

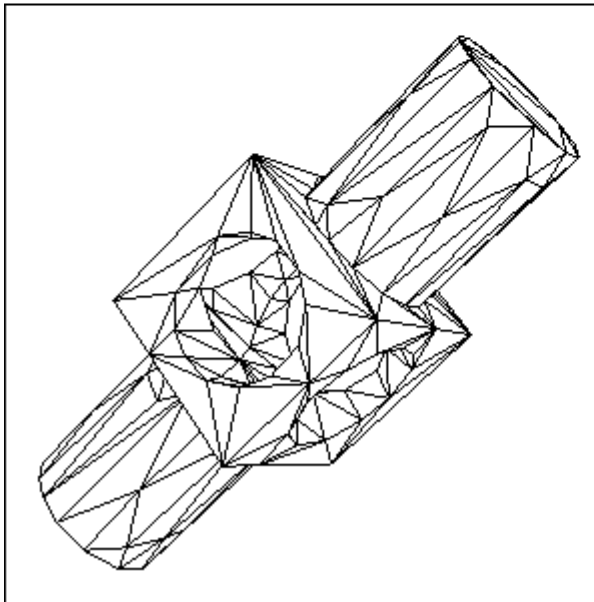
Pelkästään summan $E_{\text{dist}} + E_{\text{rep}}$ minimointi ei tuota haluttua tulosta. Kun minimoidaan vain E_{dist} , estimoidussa pinnassa on monia lisäyksiä, joita ei ole alkuperäisessä kappaleessa, koska minimiä $E_{\text{dist}} + E_{\text{rep}}$ ei välttämättä ole olemassa.



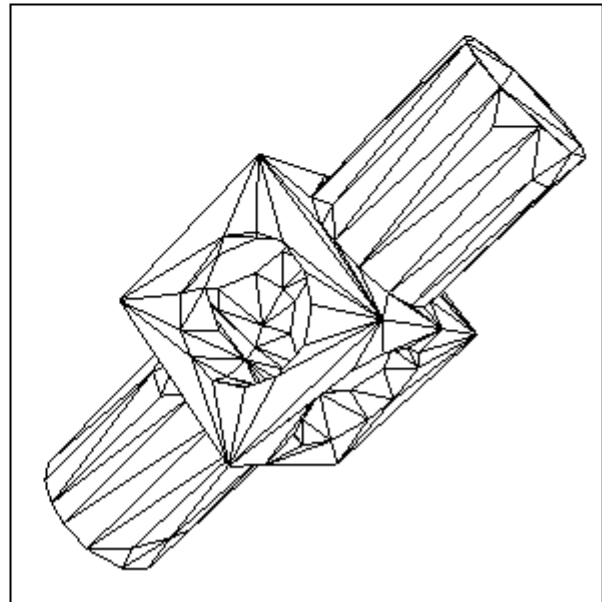
(a) Minimum of E with $\kappa = 10^{-2}$



(b) Minimum of E with $\kappa = 10^{-3}$

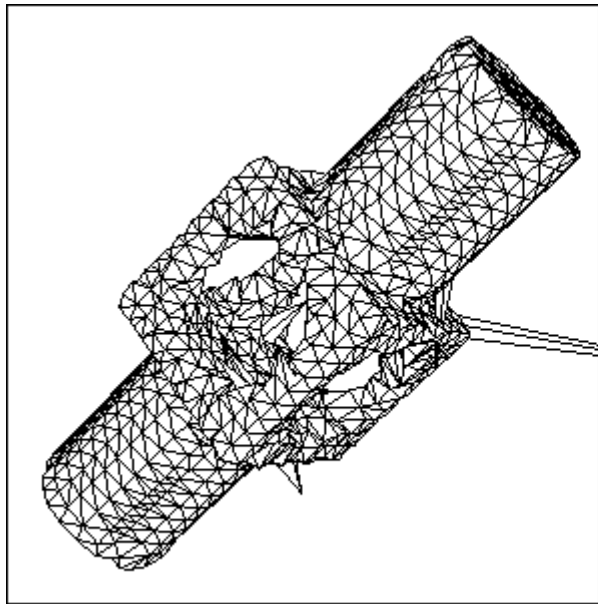


(c) Minimum of E with $\kappa = 10^{-4}$

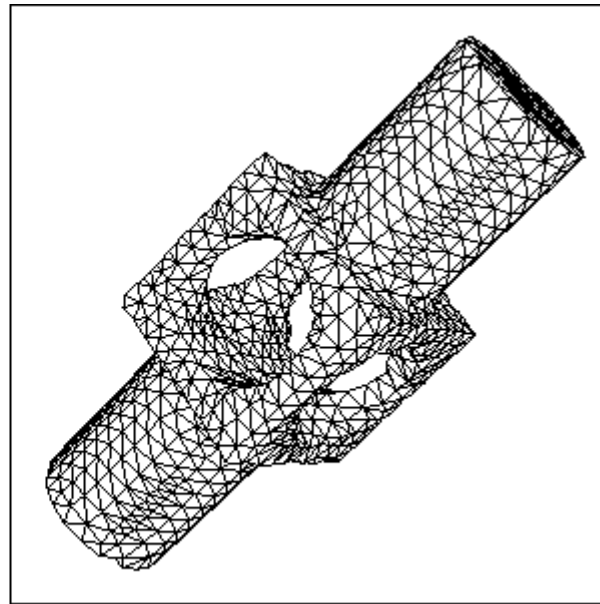


(d) Minimum of E with $\kappa = 10^{-8}$

Minimin olemassaolon takaamiseksi lisätään kolmas termi E_{spring} . Se asettaa jokaiselle sivulle verkossa 0-pituisen jousen ja jousivakion ? .



(a) Minimizing E without E_{spring}



(b) Minimizing E with E_{spring}

Minimoimalla E käyttäen lisäksi E_{spring} saadaan paljon stabiilimpi lopputulos.

Jousivoima ei ole tasainen voima. Sen tehtävänä ei ole poistaa teräviä reunoja, joita voi esiintyä kappaleessa. Jousivoiman tehtävänä on tasata verkkoa ja helpottaa optimointia.

Tulokset

Jos kolmiulotteisesta mallista halutaan muodostaa pintamalli, joka on tarkka, ja kolmion pintamalli on optimoitu, niin on lähtöaineiston oltava mahdollisimman tasalaatuinen. Mittaus-tarkkuuden on pysyttävänä samana joka puolella kappaletta ja pinnan muodot eivät saa olla pienempiä kuin $\approx 3\lambda$ tai pinnat kasvavat yhteen. Tiheä pisteytys jää optimoinnin jälkeen vain kohtiin, joissa pinnan muodot muuttuvat voimakkaasti ja tasaisilla alueilla kolmiointi on harva.

Pieniä puutteita mittauksissa voidaan paikata:

- Aukot voidaan peittää, jos voidaan varmasti olettaa, että aukon kohdalla pinnan normaali on saman tai lähes saman suuntainen kuin ympäröivien pintojen normaalit.
- Häiriöpisteet, jotka poikkeavat selvästi verkon muusta topologiasta, on helppo suodattaa pois.
- Pintojen päällekkäiset pisteet eivät haittaa pinnan muodostusta, kunhan pisteaineisto on riittävän tiheä.
- Jos pisteistön tiheys vaihtelee suuresti, on mallin muodostus jaettava osiin, joissa tarkemmat alueet mallinnetaan erikseen ja liitetään myöhemmin epätarkkoihin alueisiin.
- Verkon optimoinnilla voidaan poistaa huomattava määrä tarpeettomia kolmiopintoja, mutta alkuperäisen pisteistön on oltava tarkka.

Päätelmät

Marching cubes soveltuu lähes kaikkiin 3D-mallinnus kohteisiin, jos pinta on mitattu riittävän tiheällä verkolla ja verkko on tasalaatuinen. Algoritmi pystyy käsittelemään pinnan haarautumat, reiät ja tangenttipinnan voimakkaat muutokset edellyttäen pienimmän muodon, jonka halutaan näkyvän, olevan mittaustiheys + $3 \times$ mittaustiheyden hajonta. Jos kappaleessa on pinnan muotoja, joiden koko on pienempi kuin edellä mainittu miniarvo, niitä ei pystytä kappaleesta

mallintamaan. Pistetiheyttä ei voida muuttaa eri puolilla kappaletta, vaikka se pinnan muotojen kannalta olisikin järkevää. Käytännöllisin menetelmä on mitata kappale useassa osassa ja mallintaa osat erikseen ja yhdistellä ne toisiinsa valmiina kappaleina.

Viitteet

<http://research.microsoft.com/~hoppe/>

<http://www.exaflop.org/docs/marchcubes/ind.html>

<http://www.siggraph.org/education/materials/HyperVis/vistech/volume/surface4.htm>

<http://www.cosc.brocku.ca/Project/info/grossi/detmarch.html>

http://www.wheatchex.com/projects/viz/02b_mcubes/

<http://astronomy.swin.edu.au/~pbourke/modelling/polygonise/>

http://www.it.uu.se/edu/course/homepage/grafik2/public_html_2002/DiscretizedMarchingCubes.pdf

http://www.cs.technion.ac.il/~u_shani/cs236807-S2/lectures-students/Marching%20Cubes.pdf
