

BEZJĖ PAVIRŠIŲ APROKSIMAVIMO ALGORITMAS ATSKIRAI KVADRIANGULIARIŲ PAVIRŠIŲ KLASEI

Martynas Sabaliauskas

Vilniaus universiteto Matematikos ir informatikos institutas
Akademijos g. 4, Vilnius, Lietuva
martynas.sabaliauskas@mii.vu.lt

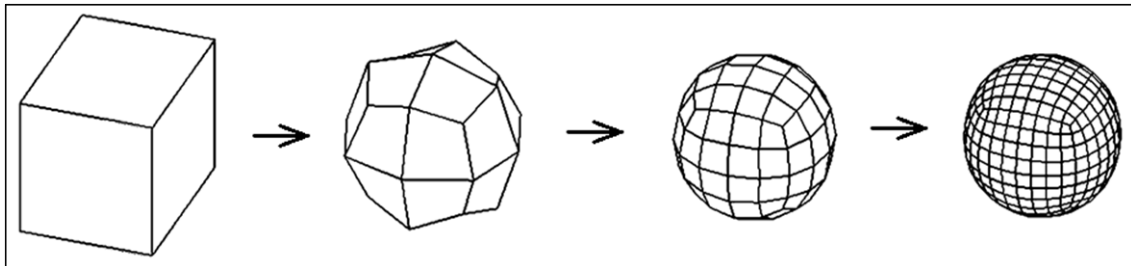
Anotacija. Darbe pasiūlytas $C2$ paviršiaus konstravimo algoritmas specialiai keturkampių paviršių klasei, kuriai priklausantys trimačiai objektai neturi gretimų ypatingųjų taškų. Šis algoritmas nuo standartinių $C2$ paviršių generavimo algoritmų skiriasi galimybe rezultato tikslumą keisti tiesiškai, t. y. aproksimuotus Bežjė paviršius atitinkančio tinklo kraštinėje esančių taškų skaičius gali įgyti natūralias reikšmes $2, 3, \dots, n$. Tuo tarpu Catmull-Clark algoritmo rezultato tikslumas keičiamas eksponentiškai, t. y. atitinkamos reikšmės lygios $2, 4, \dots, 2^n$. Galimybę gauti daugiau rezultatų sąlygoja begalinio Bežjė paviršių generavimo proceso ypatingųjų taškų aplinkoje pakeitimas baigtinių žingsnių algoritmu, pagal kurį apskaičiuojamos reikiamų taškų koordinatės. Šis $C2$ paviršių generavimo algoritmas realizuotas programiškai, atlikta rezultatų kokybės analizė.

Pagrindiniai žodžiai: Bežjė paviršius, $C2$ paviršius, keturkampis paviršius, aproksimavimas, algoritmas.

1. Įvadas

Sparčiai tobulėjant trimačių objektų skaitmeninio technologijoms reikalingi efektyvūs algoritmai, gebantys ne tik apdoroti didelės apimties duomenis, tačiau ir sisteminti, glaudinti bei tam tikru tikslumu atkurti informaciją. Vienas iš sprendimo būdų sisteminti tokio pobūdžio informacijai yra parametrinių paviršių lygčių konstravimas taip suvedant paviršių į parametrinį matematiškai apibrėžtą pavidalą bei paliekant galimybę šį paviršių norimu tikslumu atkurti ir atvaizduoti (Jiao, Bayyana, 2008). Čia susiduriama su daug sunkumų: iš pradžių reikia nagrinėjamą paviršių topologiškai įvertinti, tada jam pritaikyti trikampę, keturkampę ar kitokią konstrukciją, į kurią paviršius tilptų (Róth, Juhász, 2009). Tada seka įvairios optimizavimo problemos siekiant šios konstrukcijos viršūnes išdėstyti taip, kad šią konstrukciją atitinkantis Bežjė kreivėmis suformuotas paviršius (arba kitaip apibrėžtas) būtų kuo artimesnis savo forma pradiniam paviršiui, t. y. jį aproksimuotų. Jei turima konstrukcija yra kvadrianguliarus paviršius, praktiškai neišvengsime jame ypatingųjų taškų, t. y. tokių taškų, į kuriuos sueina 3, 5, 6, 7 ir t.t. keturkampių viršūnės. Tokiu atveju dažniausiai taikomas Catmull-Clark (Catmull, Clark, 1978) paviršių dalinimo algoritmas, kurio rezultatas konverguoja į glodų paviršių, vadinamą $C2$ paviršiumi. Šio paviršiaus ir bet kokios jį kertančios plokštumos sandūroje esančios kreivės kiekvienas vidinis taškas yra

dukart diferencijuojamas. Catmull-Clark algoritmas paprastas, efektyvus, įgyvendinamas rekursija, tačiau išlieka detalumo problema atvaizduojant rezultatą. Pagal rekursyvų Catmull-Clark algoritmą (kurį galima taikyti, beja, nebūtinai kvadrianguliariam paviršiui) nagrinėjamo paviršiaus n -kampė dalis yra dalinama į n dalių. Pavyzdžiui, turint kvadrianguliarių paviršių, kiekvienos iteracijos metu keturkampių skaičius padidėja keturis kartus, todėl norimas atvaizdavimo tikslumas tampa 2^n taškų rezultate grąžinamai keturkampio kraštinei, atitinkančiai pradinės konstrukcijos kraštinę (1 pav.).



1 pav. Catmull-Clark algoritmo 3 iteracijų vizualizacija.

Vienas Catmull-Clark algoritmo trūkumų yra per greitas konvergavimas į C2 paviršių, t. y. duomenų apimtis po kiekvienos iteracijos vidutiniškai padidėja 4 kartus, todėl dažnai trūksta tarpinio varianto tarp šių iteracijų. Kadangi toks rekursyvus dalinimas neapibrėžiamas parametrine paviršiaus lygtimi, todėl gautasis rezultatas negali būti keičiamas vienu tikslumo vienetu. Šiame darbe išnagrinėtas metodas pagal kurį toks tikslumas pasiekiamas.

2. Trečios eilės Bežjė paviršius

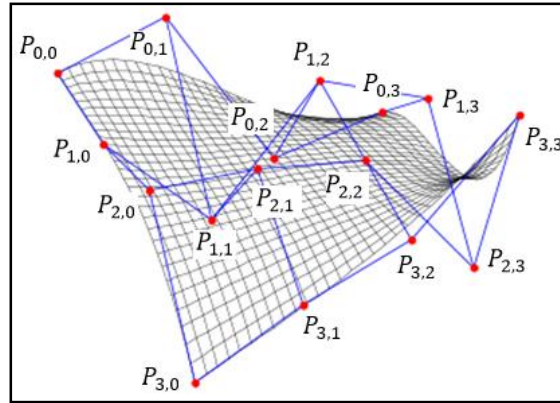
Tegu $B(u, v)$ – trečios eilės Bežjė paviršius, kurio parametrinė forma

$$B(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 \binom{3}{i} u^i (1-u)^{3-i} \binom{3}{j} v^j (1-v)^{3-j} P_{i,j}. \quad (1)$$

Apskaičiavus binominius koeficientus, paviršiaus parametrinę lygtį galima suvesti į matricinę formą:

$$B(u, v) = \begin{pmatrix} 1 & u & u^2 & u^3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix} \begin{pmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{pmatrix} \cdot \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ v \\ v^2 \\ v^3 \end{pmatrix}.$$

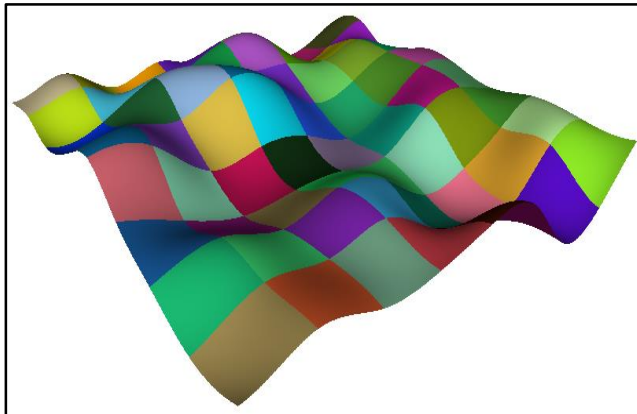
$P_{i,j}$ – kontroliniai taškai, pagal kurių išsidėstymą suformuojamas Bežjė paviršius (arba C2 paviršiaus fragmentas).



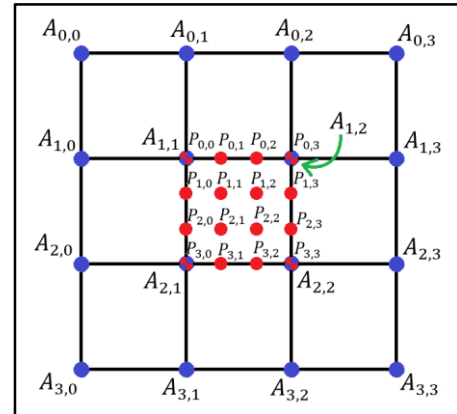
2 pav. Bežjė paviršiaus pavyzdys.

3. C2 paviršiaus konstravimas

C2 paviršių galima sudaryti iš 3 eilės Bežjė paviršių, kurių sandūroje sutaptų antros eilės išvestinės bet kokio tiesaus pjūvio atžvilgiu. 3 paveikslėlyje pateiktas C2 paviršiaus pavyzdys, kuriame skirtingos spalvos paviršius atitinka skirtingą Bežjė paviršių. Pagrindinis uždavinys – apskaičiuoti kontrolinius taškus šiems Bežjė paviršiams.



3 pav. C2 paviršius, sudarytas iš Bežjė paviršių.



4 pav. Kontrolinių Bežjė paviršiaus taškų apskaičiavimas.

Tarkime turime kvadrianguliarių paviršių, kuriame iš kiekvienos vidinės viršūnės viršūnės išeina po 4 briaunas. Tada kiekvienam vidiniam šio paviršiaus keturkampii galima sukonstruoti Bežjė paviršių, kurio kontroliniai taškai apskaičiuojami pagal šias formules:

$$\begin{aligned}
 P_{0,0} &= \frac{1}{36}A_{0,0} + \frac{1}{36}A_{0,2} + \frac{1}{36}A_{2,0} + \frac{1}{36}A_{2,2} + \frac{1}{9}A_{0,1} + \frac{1}{9}A_{1,0} + \frac{1}{9}A_{1,2} + \frac{1}{9}A_{2,1} + \frac{4}{9}A_{1,1}, \\
 P_{0,1} &= \frac{1}{18}A_{0,2} + \frac{1}{18}A_{2,2} + \frac{1}{9}A_{0,1} + \frac{1}{9}A_{2,1} + \frac{2}{9}A_{1,2} + \frac{4}{9}A_{1,1}, \\
 P_{0,2} &= \frac{1}{18}A_{0,1} + \frac{1}{18}A_{2,1} + \frac{1}{9}A_{0,2} + \frac{1}{9}A_{2,2} + \frac{2}{9}A_{1,1} + \frac{4}{9}A_{1,2}, \\
 P_{0,3} &= \frac{1}{36}A_{0,1} + \frac{1}{36}A_{0,3} + \frac{1}{36}A_{2,1} + \frac{1}{36}A_{2,3} + \frac{1}{9}A_{0,2} + \frac{1}{9}A_{1,1} + \frac{1}{9}A_{1,3} + \frac{1}{9}A_{2,2} + \frac{4}{9}A_{1,2}, \\
 P_{1,0} &= \frac{1}{18}A_{2,0} + \frac{1}{18}A_{2,2} + \frac{1}{9}A_{1,0} + \frac{1}{9}A_{1,2} + \frac{2}{9}A_{2,1} + \frac{4}{9}A_{1,1}, \\
 P_{1,1} &= \frac{1}{9}A_{2,2} + \frac{2}{9}A_{1,2} + \frac{2}{9}A_{2,1} + \frac{4}{9}A_{1,1},
 \end{aligned}$$

$$\begin{aligned}
 P_{1,2} &= \frac{1}{9}A_{2,1} + \frac{2}{9}A_{1,1} + \frac{2}{9}A_{2,2} + \frac{4}{9}A_{1,2}, \\
 P_{1,3} &= \frac{1}{18}A_{2,1} + \frac{1}{18}A_{2,3} + \frac{1}{9}A_{1,1} + \frac{1}{9}A_{1,3} + \frac{2}{9}A_{2,2} + \frac{4}{9}A_{1,2}, \\
 P_{2,0} &= \frac{1}{18}A_{1,0} + \frac{1}{18}A_{1,2} + \frac{1}{9}A_{2,0} + \frac{1}{9}A_{2,2} + \frac{2}{9}A_{1,1} + \frac{4}{9}A_{2,1}, \\
 P_{2,1} &= \frac{1}{9}A_{1,2} + \frac{2}{9}A_{1,1} + \frac{2}{9}A_{2,2} + \frac{4}{9}A_{2,1}, \\
 P_{2,2} &= \frac{1}{9}A_{1,1} + \frac{2}{9}A_{1,2} + \frac{2}{9}A_{2,1} + \frac{4}{9}A_{2,2}, \\
 P_{2,3} &= \frac{1}{18}A_{1,1} + \frac{1}{18}A_{1,3} + \frac{1}{9}A_{2,1} + \frac{1}{9}A_{2,3} + \frac{2}{9}A_{1,2} + \frac{4}{9}A_{2,2}, \\
 P_{3,0} &= \frac{1}{36}A_{1,0} + \frac{1}{36}A_{1,2} + \frac{1}{36}A_{3,0} + \frac{1}{36}A_{3,2} + \frac{1}{9}A_{1,1} + \frac{1}{9}A_{2,0} + \frac{1}{9}A_{2,2} + \frac{1}{9}A_{3,1} + \frac{4}{9}A_{2,1}, \\
 P_{3,1} &= \frac{1}{18}A_{1,2} + \frac{1}{18}A_{3,2} + \frac{1}{9}A_{1,1} + \frac{1}{9}A_{3,1} + \frac{2}{9}A_{2,2} + \frac{4}{9}A_{2,1}, \\
 P_{3,2} &= \frac{1}{18}A_{1,1} + \frac{1}{18}A_{3,1} + \frac{1}{9}A_{1,2} + \frac{1}{9}A_{3,2} + \frac{2}{9}A_{2,1} + \frac{4}{9}A_{2,2}, \\
 P_{3,3} &= \frac{1}{36}A_{1,1} + \frac{1}{36}A_{1,3} + \frac{1}{36}A_{3,1} + \frac{1}{36}A_{3,3} + \frac{1}{9}A_{1,2} + \frac{1}{9}A_{2,1} + \frac{1}{9}A_{2,3} + \frac{1}{9}A_{3,2} + \frac{4}{9}A_{2,2},
 \end{aligned}$$

čia koeficientai prie $A_{i,j}$, $i, j = 0, \dots, 3$ atitinka tenzorinės dviejų trečios eilės Bezjė kreivių sandaugos koeficientus. Šis metodas taikomas tada, kai 4 paveikslėlyje pavaizduoti taškai $A_{1,1}$, $A_{1,2}$, $A_{2,1}$ ir $A_{2,2}$ nėra ypatingieji, t. y. turi po 4 incidentias briaunas. Kitu atveju šis metodas netinka, nes taškų skaičius, skirtas apskaičiuoti kontroliniams taškams, nelygus 16.

Tačiau kvadrianguliarių objektų viršūnės nebūtinai turi po 4 incidentias briaunas. Tokiu atveju šios viršūnės vadinamos ypatingaisiais taškais, kurių aplinkoje sukonstruoti C2 paviršius minėtu metodu negalima. Darbuose (Karčiauskas, Peters, 2006), (Nießner, Loop, 2012) nagrinėjami metodai šiai problemai spręsti. Ypatingųjų taškų srityse realizuojamas Catmull-Clark paviršių dalinimo algoritmas. Tokiu atveju kvadrianguliarus paviršiaus keturkampiai, kuriems yra taikomas šis dalinimas pakeičia savo viršūnių koordinatas pagal formulę:

$$v^{i+1} = \frac{1}{2}v^i + \frac{1}{16}\sum_{j=1}^4 e_j^i + \frac{1}{16}\sum_{j=1}^4 f_j^{i+1}, \quad (2)$$

čia e_j^i – viršūnės gretimos viršūnei v^i , f_j^{i+1} – keturkampio, gretimo viršūnei v^i , centras.

Taip pat apskaičiuojami nauji taškai, atitinkantys briaunas:

$$e_j^{i+1} = \frac{1}{4}(v^i + e_j^i + f_{j-1}^{i+1} + f_j^{i+1}), \quad (3)$$

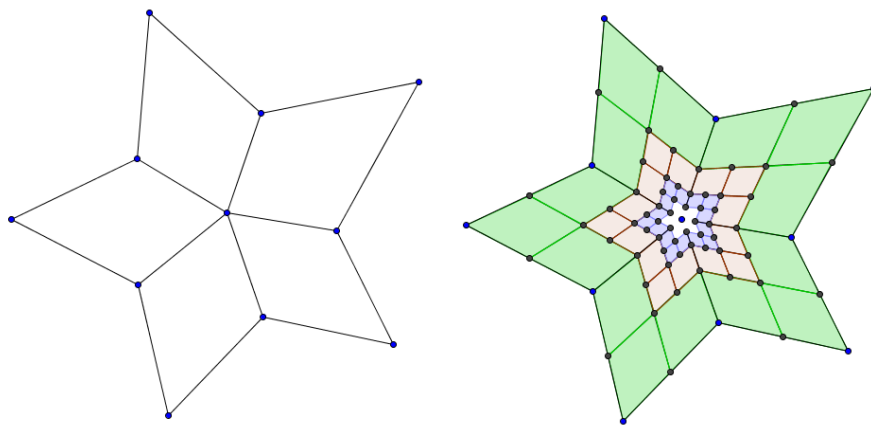
čia f_{j-1}^{i+1} ir f_j^{i+1} gretimų briaunai $v^i e_j^i$ keturkampių centrai.

Jei kvadrianguliarus paviršiaus viršūnės ir briaunos yra kraštinės, tada taikomos šios formulės naujo paviršiaus viršūnėms apskaičiuoti:

$$e_j^{i+1} = \frac{1}{2}(v^i + e_j^i), \quad (4)$$

$$v_j^{i+1} = \frac{1}{8} (e_j^i + 6v^i + e_k^i). \quad (5)$$

Pagal (2, 3, 4, 5) lygybes kvadrianguliaram paviršiui apskaičiavę naujų viršūnių koordinates ir sujungę gautas viršūnes briaunomis iš esmės atliekame vieną Catmull-Clark algoritmo iteraciją. Kadangi Catmull-Clark algoritmo rezultatas konverguoja į C2 paviršių, o mūsų tikslas sukonstruoti C2 paviršių iš atskirų Bezjė paviršių, tai galima realizuoti vieną Catmull-Clark iteraciją pradiniam kvadrianguliaram paviršiui ir po to taikyti minėtą kontrolinių taškų apskaičiavimo metodą. Problema yra tik išliekantys ypatingieji taškai savo pozicijose, tačiau, paeiliui taikant Catmull-Clark algoritmą ir kontrolinių taškų apskaičiavimo metodą, galima vis mažesniais Bezjė paviršiais konstruoti C2 paviršių ypatingųjų taškų srityse (5 pav.). Pirmoje dalyje pateiktas ypatingasis taškas, į kurį remiasi 5 keturkampiai, antroje dalyje pavaizduotas šios srities užpildymas 3 eilės Bezjė paviršiais taikant 3 minėto metodo iteracijas (skirtingos spalvos rodo skirtingų iteracijų rezultatus).



5 pav. Bezjė paviršių generavimas ypatingojo taško aplinkoje.

Kadangi šis procesas yra begalinis, todėl praktikoje užtenka realizuoti 4-5 minėto metodo iteracijas, o centro tuštumą užpildyti papildoma detale (Wagnera, Hormannb, Greiner, 2003). Jei tokio pobūdžio modelis atlieka vaidmenį, pavyzdžiui, animaciniame filme, plika akimi tokie užpildymai nematomi.

Šiame darbe išnagrinėtas kitas būdas ypatingojo paviršiaus taško srities užpildymo problemai spręsti. Akivaizdu, kad aproksimuojant pradinį modelį C2 paviršiumi galima pasirinkti norimą rezultato tikslumą, t. y. Bezjė tinklo UV tankumą. Jei šis tankumas būtų traktuojamas kaip pradinio kvadrianguliaraus paviršiaus briaunoje esančių taškų skaičius, akivaizdu, kad begalinis dalinimas minėtu metodu nebūtų prasmingas, nes tokiu atveju egzistuotų be galo daug nereikalingų Bezjė paviršių, kurių parametrinės lygtys nebūtų reikalingos UV tinklo taškams apskaičiuoti. Todėl naudinga remtis straipsniu (Peters & Reift, 1996), pagal kurį centrinis ypatingasis taškas A konverguoja į A' :

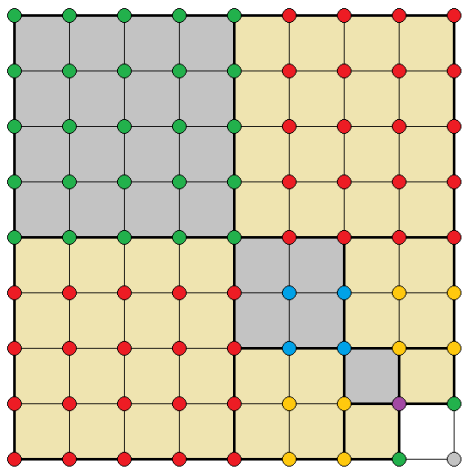
$$A' = \frac{m}{m+5} A + \frac{4}{m(m+5)} \sum_{i=1}^m B_i + \frac{1}{m(m+5)} \sum_{i=1}^m C_i \quad (6)$$

čia B_i – taškai, esantys vienos briaunos atstumu nuo ypatingojo taško A , C_i – taškai, esantys dviejų briaunų atstumu nuo ypatingojo taško A , m – ypatingojo taško A eilė (pavyzdžiui, 5 paveikslėlyje $m = 5$). Likusius UV tinklo taškus galima apskaičiuoti begalinį Catmull-Clark dalinimo ir vis mažesnių Bežjė paviršių formavimo procesą pakeitus baigtiniu.

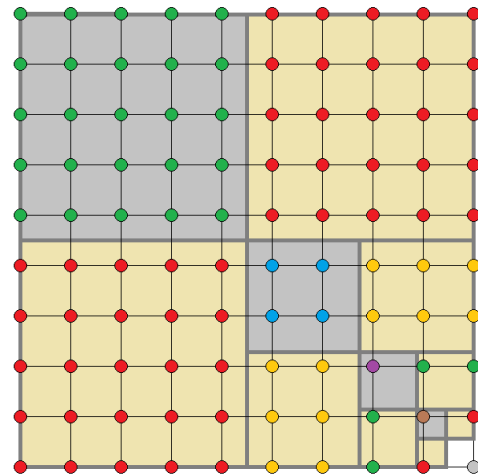
Pirmiausia apskaičiuokime kiek dalinimo iteracijų turime realizuoti, jei UV tinklo kraštinėje yra n taškų:

$$\lceil \log_2(n - 1) \rceil + 1 \quad (7)$$

Pagal (7) lygybės natūralią reikšmę reikia iš dalinti į ypatingąjį tašką besiremiantį keturkampį. Dalinimo pavyzdžiai pateikti 6 ir 7 paveikslėliuose.



6 pav. UV tinklas, kai $n=9$.



7 pav. UV tinklas, kai $n=10$.

Kiekvienam naujai sukonstruotam keturkampiiui, pažymėtam 6 ir 7 paveikslėliuose skirtinga spalva reikia apskaičiuoti tokias $u, v \in [0, 1]$ reikšmes, kurias įsistačius į (1) lygtį apskaičiuoti Bežjė paviršiaus taškai atitiktų UV tinklo taškus. Tai galima padaryti įgyvendinus šį algoritmą:

1 algoritmas. *Tinklas*(n)

1. $d \rightarrow \lceil \log_2(n - 1) \rceil + 1$

2. $m_1 \rightarrow \lfloor \frac{n+2}{2} \rfloor$

3. $m_2 \rightarrow n + 1 - m_1$

4. for $i \rightarrow 1$ to d do

5. $p \rightarrow \frac{2^i m_2}{n} - 1$

6. for $j \rightarrow 1$ to m_1 do

7. $T_1[i][m_1 - j + 1] \rightarrow 1 - p - \frac{(j-1)2^i}{n}$

8. for $k \rightarrow 1$ to m_2 do

9. $T_2[i][m_2 - k + 1] \rightarrow 1 - \frac{(k-1)2^i}{n}$

10. $m_1 \rightarrow \left\lfloor \frac{m_2}{2} \right\rfloor$
11. $m_2 \rightarrow m_2 - m_1$
12. return T_1, T_2

Pagal 6 paveikslėlį *Tinklas(n)* algoritmo rezultatas lygus:

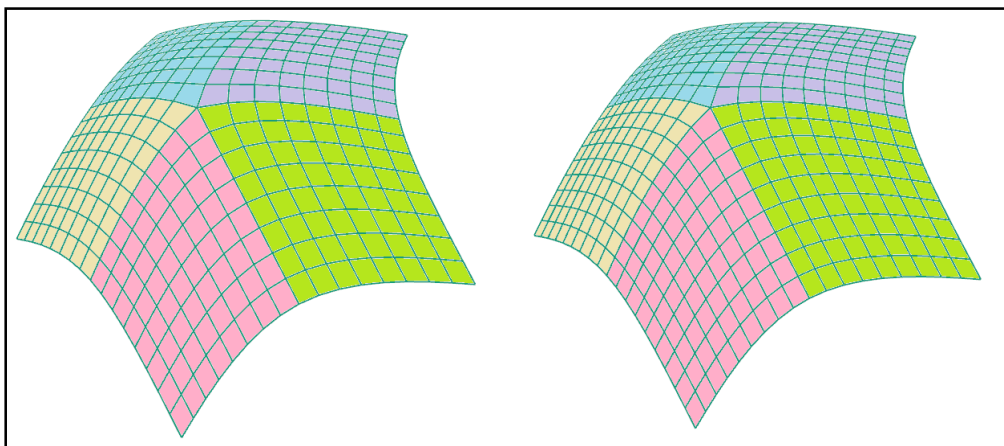
$$T_1 = \begin{pmatrix} 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \\ \frac{1}{2} & 1 & - & - & - \\ \frac{1}{2} & - & - & - & - \\ 1 & - & - & - & - \end{pmatrix}, \quad T_2 = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \\ \frac{1}{2} & 1 & - & - \\ \frac{1}{2} & - & - & - \\ 1 & - & - & - \end{pmatrix},$$

T_1 matricos kiekviena eilutė atitinka 6 paveikslėlyje kvadrato, einančio per įstrižainę, $u, v \in [0, 1]$ reikšmes, T_2 matricos kiekviena eilutė analogiškai atitinka kitą iš dešinės į kairę išdėstytą kvadratų $u, v \in [0, 1]$ reikšmes.

Pagal 7 paveikslėlį *Tinklas(n)* algoritmo rezultatas lygus:

$$T_1 = \begin{pmatrix} 0 & \frac{2}{9} & \frac{4}{9} & \frac{2}{3} & \frac{8}{9} \\ \frac{2}{9} & \frac{2}{3} & - & - & - \\ \frac{2}{9} & \frac{2}{3} & - & - & - \\ \frac{2}{9} & - & - & - & - \\ \frac{2}{9} & - & - & - & - \end{pmatrix}, \quad T_2 = \begin{pmatrix} \frac{1}{9} & \frac{1}{3} & \frac{5}{9} & \frac{7}{9} & 1 \\ \frac{1}{9} & \frac{2}{3} & \frac{5}{9} & 1 & - \\ \frac{1}{9} & \frac{2}{3} & 1 & - & - \\ \frac{1}{9} & 1 & - & - & - \\ 1 & - & - & - & - \end{pmatrix}.$$

Tolimesnėje C2 paviršiaus formavimo eigoje matricų T_1 ir T_2 komponentės naudojamos $u, v \in [0, 1]$ parametrų reikšmės galutinio rezultato taškams apskaičiuoti. Šie taškai turi būti išrikiuojami bei suformuojami nauji keturkampiai. 8 paveikslėlyje pateiktas išrikiuotų bei susietų taškų pavyzdys, atitinkantis 5 paveikslėlyje esančią konstrukciją, kuriai apskaičiuotas C2 paviršius pagal 6 ir 7 paveikslėlių schemas.



8 pav. C2 paviršiaus formavimas ypatingųjų taškų srityse.

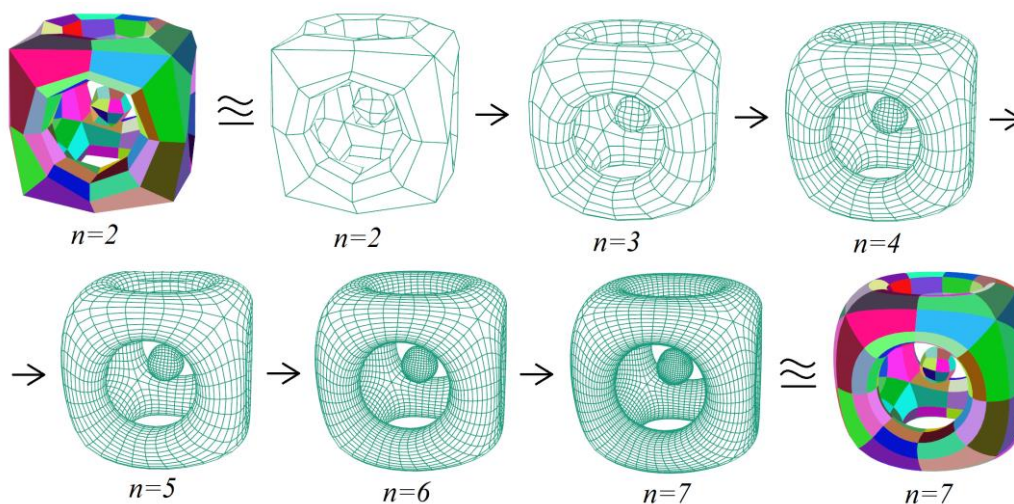
Siekiant apskaičiuoti C2 paviršių bet kokiam pradiniam kvadrianguliariam paviršiui belieka susieti visus šiame straipsnyje aprašytus metodus. Bendra šio algoritmo idėja:

2 algoritmas. C2_pavrsius(n)

1. Inicializacija,
2. Tinklas(n),
3. Realizuok $\lceil \log_2(n - 1) \rceil + 1$ kartų Catmull-Clark algoritmą keturkampiams, kurių viršūnės – ypatingieji taškai,
4. Apskaičiuok visų Bežjė paviršių lygtis,
5. Į šias lygtis įsistatyk atitinkamas $u, v \in [0, 1]$ parametrų reikšmes ir apskaičiuok C2 paviršiaus taškus,
6. Susiedamas šiuos taškus po 4 išsaugok rezultatą, kuris atitinka C2 paviršiaus aproksimaciją.

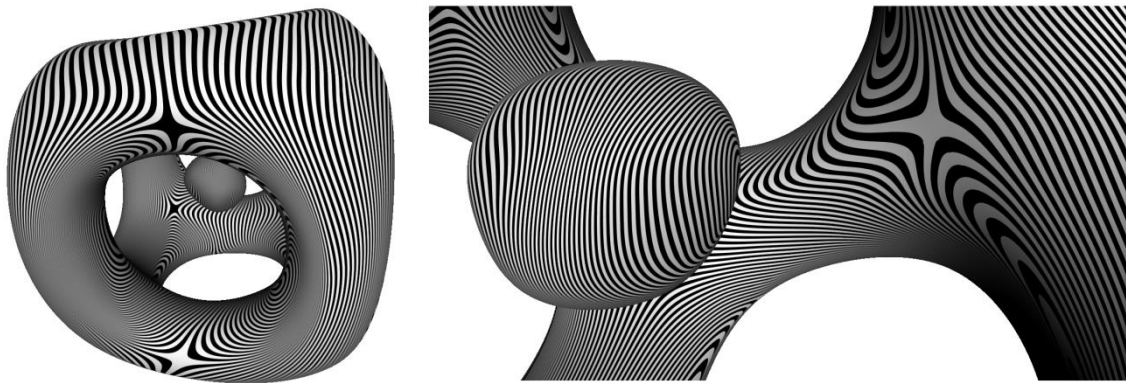
4. Programinė įranga

2 algoritmas buvo realizuotas Maple programos aplinkoje. Įvesties ir išvesties duomenims pasirinktas „OFF“ skaitmeninių modelių formatas. 9 paveikslėlyje pavaizduoti galimi 2 algoritmo rezultatai, atitinkantys C2 paviršiaus aproksimaciją, kai paviršiaus detalumo parametras n kinta nuo 2 iki 7. Jei palygintume šiuos rezultatus su standartinio Catmull-Clark algoritmo rezultatais, tai $n = 2$ ir $n = 4$ atvejais rezultatas būtų identiškas, kitų rezultatų (kai $n = 3, 5, 6, 7$) taikant Catmull-Clark algoritmą gauti neįmanoma.



9 pav. Paviršiaus dalinimas

Šiame darbe pasiūlytas paviršių dalinimo algoritmas už Loop-Schaefer algoritmą (Loop, Schaefer, 2008) pranašesnis tuo, kad gąžinamas C2 paviršius, tuo tarpu Loop-Schaefer algoritmas ypatingųjų taškų srityse generuoja Bežjė paviršius, tačiau jų sandūroje nesutampa antros eilės išvestinės. 10 paveikslėlyje pateiktas $C2_pavrsius(100)$ algoritmo rezultatas, kurio paviršius padengtas atspindžio linijomis (*angl. Reflection lines*). Šios linijos parodo kokybišką rezultatą (Langer, Seidel, 2007).



10 pav. Paviršiaus atspindžio linijos

5. Išvados

Šiame straipsnyje pasiūlytas C2 paviršiaus konstravimo metodas kvadrianguliams paviršiams, neturintiems gretimų ypatingųjų taškų. Palyginus su standartiniais Bezjė paviršių ir C2 paviršių formavimo metodais, šio metodo pranašumas yra galimybė rezultato tikslumą keisti vienetine tinklo UV, atitinkančio pradinio kvadriangulio objekto sieną, reikšme.

Pagal pasiūlytą *Tinklas(n)* metodą galima nebūtinai u , v reikšmes skaičiuoti griežtai UV tinklo reikšmėms. Jei ypatingasis paviršiaus taškas 5 ir aukštesnės eilės, C2 paviršiaus taškai turi savybę tokių ypatingųjų taškų aplinkose išsidėstyti rečiau. Todėl teoriškai galima *Tinklas(n)* metodą papildyti naujomis taisyklėmis, pagal kurias u , v parametrai būtų apskaičiuojami taip, kad C2 paviršiaus taškai išsidėstytų tolygiai visame paviršiuje, įskaitant ypatingųjų taškų sritis.

C2_pavrsius(n) algoritmas buvo realizuotas programiškai, gauti praktiniai rezultatai patvirtina šiame straipsnyje suformuluotą teoriją.

Darbe pasiūlyta Bezjė paviršių aproksimavimo metodika gali būti taikoma optimizavimo uždaviniuose, kuriuose pagrindinis tikslas maksimizuoti C2 paviršiaus detalumą, kai:

- kompiuterio pastovioji atmintis, skirta *C2_pavrsius(n)* algoritmo rezultato išsaugojimui, turi tam tikrus apribojimus;
- yra žinoma perteklinė informacijos riba, pavyzdžiui, naudojant 3D spausdintuvą trimačiams objektams suformuoti, papildomas C2 paviršiaus taškų skaičius neturi įtakos rezultato kokybei.

Literatūra

- Catmull, E., Clark, J. (1978 m. November). Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes. *Computer-Aided Design*, 10(6), 350-355.
- Jiao, X., Bayyana, N. R. (2008 m. February). Identification of C1 and C2 Discontinuities for Surface Meshes in CAD. *Computer-Aided Design*, 40(2).

- Karčiauskas, K., Peters, J. (2006 m. June 29-July 5). Parameterization Transition for Guided C2 Surfaces of Low Degree. *In: Sixth AFA Conference on Curves and Surfaces Avignon*, 183-192.
- Langer, T., Seidel, H.-P. (2007). Mean Value Bezier Surfaces. *Mathematics of Surfaces XII*, 4647, 263-274.
- Loop, C., Schaefer, S. (2008 m. March). Approximating Catmull-Clark Subdivision Surfaces with Bicubic Patches. *ACM Transactions on Graphics*, 27(1), 8:1-8:11.
- Nießner, M., Loop, C. (2012). Feature Adaptive GPU Rendering of Catmull-Clark Subdivision Surfaces. *ACM Transactions on Graphics*, 31(1), 1-11.
- Peters, J., Reift, U. (1996 m. April). Analysis of Algorithms Generalizing B-Spline Subdivision. *SIAM Journal on Numerical Analysis*, 35(2), 728-748.
- Róth, Á., Juhász, I. (2009). Quadrilateral mesh generation from point clouds by a Monte Carlo method. (M. Chen, V. Skala, Mont.) *Visualization and Computer Vision in co-operation with EUROGRAPHICS*, 97-104.
- Wagner, M., Hormann, K., Greiner, G. (2003). *C2-continuous surface reconstruction with piecewise polynomial patches*. California, USA: California Institute of Technology.

M. Sabaliauskas yra trečio kurso informatikos inžinerijos Vilniaus universiteto (VU) Matematikos ir informatikos instituto doktorantas. 2010 m. baigė VU matematikos ir informatikos fakulteto matematikos ir matematikos taikymų bakalauro studijas, 2012 m. tame pačiame fakultete baigė matematikos ir informatikos dėstyto magistrantūros studijas. Interesų sritys: 3D skaitmeninių modelių apdorojimo algoritmai, optimizavimo metodai.

AN ALGORITHM FOR APPROXIMATION OF BAZIER SURFACES FOR SPECIAL CASE OF QUADRIANGULAR GRID

Martynas Sabaliauskas

Summary

This paper proposes an algorithm for construction of C2 surface. The input is a 3D quadrangular surface which doesn't have adjacent extraordinary points. The difference between this algorithm and the regular C2 forming algorithms is a possibility to choose the natural number of points of the output grid. For example using the Catmull-Clark algorithm the same possibility corresponds exponential natural numbers where the basis is 2. The changing infinite Bezier surfaces generation process to finite calculation of necessary points leads the ability to get more results. The C2 surface generation algorithm was realized and the quality results of output surfaces were performed using reflection lines.

Keywords: Bezier surface, C2 surface, approximation, algorithm.