

ISSN 1407-7493

COMPUTER SCIENCE
DATORZINĀTNE

2009-7493

TECHNOLOGIES OF COMPUTER
CONTROL
DATORVADĪBAS TEHNOLOĢIJAS

3D OBJECTS VISUALIZATION USING B-SPLINE FUNCTIONS AND RAY TRACING

**3D OBJEKTU VIZUALIZĒŠANA IZMANTOJOT B-SPLINE FUNKCIJAS UN STARU
TRASĒŠANU**

Aleksandrs Sisojevs, PhD student

Riga Technical University, Faculty of Computer Science and Information Technologies, Institute of Computer Control, Engineering and Technology

Address: Meza str. 1, LV-1048, Riga, Latvia

E-pasts: alexiv@inbox.lv

Aleksandrs Glazs, Dr.habil.sc.ing., professor

Riga Technical University, Faculty of Computer Science and Information Technologies, Institute of Computer Control, Engineering and Technology

Address: Meza str. 1, LV-1048, Riga, Latvia

E-Mail: glaz@egle.cs.rtu.lv

Atslēgas vārdi: B-spline, brīvas formas virsma, gradienta metode, optimizācija, staru trasēšana, vizualizēšana

Ievads

Brīvo formu virsmu vizualizēšanas problēma ir aktuālā dažādas jomas zinātnē un tehnikā. Datorgrafikas teorijā eksistē dažas matemātiskas aprakstus, ar kuru palīdzību ir iespējams aprakstīt brīvo formu virsmu [1]. Efektīvais un elastīgais apraksts ir B-spline virsmas.

Mūsdienu situācija 3D datorgrafikā ir sekojoša: attēla sintēze attīstās divos virzienos:

- Pirmajā gadījumā galvenā loma ir ātrdarbībai. Tāda pieeja iekļauj poligonālo grafiku, aparatūras paātrinājumu un grafiskas bibliotēkas (OpenGL, DirectX, Java 3D).
- Otrajā gadījumā pieeja paredzēta, lai dabūtu attēla augstāko precizitātes un realitātes pakāpi. Pieveja iekļauj augsti precīzas vizualizēšanas metodes, fiziski pamatotus apgaismošanas modeļus, reālistiskos teksturēšanas algoritmus utt..

Darbā tiek apskatīta viena no 3D scēnu vizualizēšanas metodēm – staru trasēšana. Staru trasēšanas metode ir populāra augsta kvalitātes vizualizēšanas metode. Dotā metode ļauj vizualizēt 3D scēnas ar augstu precizitātes pakāpi, un rezultātā iegūt augstas kvalitātes attēlu. Dotajā darbā tiek piedāvāta jaunā metode, ar kuras palīdzību ir iespējams vizualizēt B-spline virsmu ar augsto precizitāti un kvalitāti. Eksistē dažādas pieejas, kuras risina šo uzdevumu.

Martin un citi [2] apraksta pieeju apgriezto NURBS virsmu trasēšanu, izmantojot Ņūtona metodi. Tiek izmantota nelineāro vienādojumu sistēma lai atrast krustošanas punktu. Ņūtona metode tiek izmantota lai risināt šo vienādojuma sistēmu.

Nišita un citi [3] apraksta iteratīvo algoritmu, kas balstās uz Beziē atgriezumumu (Bézier clipping). Algoritms izskaitļo krustošanas punktu starp staru un Beziē gabalvirsmu, nosakot tos virsmas

reģionus, kas noteikti nekrustojas ar staru un izgriežot tos no gabalvirsmas. Šai pieejai ir dažādi trūkumi, piemēram nestabils darbs strādājot ar dažām virsmām un risinājuma noteikšanas problēma daudzkārtīgo krustošanas punktu gadījumā.

Efremov un citi [4] piedāvā dažas Nišitas algoritma modifikācijas Beziē un NURBS virsmām. Šinī gadījumā palielinās gabalvirsmu skaits un kā rezultāts, palielinās aprēķinu skaits. Arī, NURBS virsmas gadījumā, pēc būtības notiek nevis virsmas tiešais aprēķins, bet ekvivalentas virsmas (kura apraksta tikai sākotnējās virsmas daļu) aprēķins.

Šajā darbā mēs piedāvājam jaunu pieeju lai noteikt nezināmo parametru vērtības, kuras nepieciešamas B-spline virsmas (kas ir NURBS virsmas neracionālais gadījums) vizualizēšanai. Metode tiek balstīta uz sakņu atrašanas uzdevuma transformācijas uz optimizācijas uzdevumu. Šajā gadījumā optimizācijas uzdevums tiek sadalīts uz diviem soliem: iepriekšēja meklēšana un vērtību precizēšana ar optimizācijas metodēm.

Piedāvāta pieeja

Krustošanas punkta starp staru un parametrisko virsmu noteikšanas matemātiskais uzdevums var būt aprakstīts kā nelineāra vienādojuma sistēma [5]:

$$\begin{cases} Q_X(u, v) = Q_X(t) \\ Q_Y(u, v) = Q_Y(t) \\ Q_Z(u, v) = Q_Z(t) \end{cases}, \quad (1)$$

kur:

$Q_X(u, v)$, $Q_Y(u, v)$, $Q_Z(u, v)$, – virsmas vienādojums,
 $Q_X(t)$, $Q_Y(t)$, $Q_Z(t)$, – stara vienādojums.

Lai izskaitļot nezināmo parametru u , v un t vērtības, sistēmu (1) nepieciešams aplūkot gadījumā kad stars sakrīt ar vienu no koordinātes asīm. Gadījumā, ja stars sakrīt ar Z asi (Oz) vienādojumu sistēma (1) pieņem veidu [5]:

$$\begin{cases} Q_x(u, v) = 0 \\ Q_y(u, v) = 0 \\ Q_z(u, v) = Q_z(t) \end{cases}, \quad (2)$$

Aprakstīsim vienādojumu $Q_x(u, v)$, $Q_y(u, v)$, $Q_z(u, v)$ B-spline virsmas gadījumā. Neracionālo B-spline virsmu var aprakstīt sekojoša veidā [6,7]:

$$Q(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} \cdot N_{i,p}(u) \cdot N_{j,q}(v), \quad (3)$$

kur:

$P_{i,j}$ – kontrolpunktu masīvs;

u, v – parametri;

$N_{i,p}(u)$ – neracionālais B-spline polinoms parametriskajā virzienā u ;

$N_{j,q}(v)$ – neracionālais B-spline polinoms parametriskajā virzienā v ;

n – kontrolpunktu masīva izmērs parametriskajā virzienā u ;

m – kontrolpunktu masīva izmērs parametriskajā virzienā v ;

p – virsmas kārtas parametriskajā virzienā u ;

q – virsmas kārtas parametriskajā virzienā v .

Analoģiski, stara vienādojumu mēs varam reprezentēt sekojošā veidā:

$$Q(t) = \sum_{i=0}^1 P_i \cdot N_{i,1}(t), \quad (4)$$

kur:

P_i – kontrolpunktu masīvs;

t – parametrs;

$N_{i,1}(u)$ – neracionālais B-spline polinoms parametriskajā virzienā t .

Eksistē dažas pieejas, kuras ļauj risināt nelineāro vienādojumu sistēmu. Šajā darbā mēs transformējam vienādojumu sistēmu risināšanas uzdevumu uz optimizācijas uzdevumu. Lai atrast parametru u un v vērtību no vienādojumu sistēmas (2), pietiekami izmantot tikai pirmos divus vienādojumus, tādēļ optimizācijas funkciju var aprakstīt sekojošā veidā:

$$w = (Q_x(u, v))^2 + (Q_y(u, v))^2 \rightarrow \min_{u, v}, \quad (5)$$

Darbā funkcijas w minimizēšanas uzdevums tiek sadalīts uz divām daļām, diviem soļiem:

1. Iepriekšēja meklēšana, kura iekļauj iepriekšējo vērtību kartes izveidi;
2. Parametru precizēšana ar optimizācijas metodēm.

Optimizācijas funkciju w minimumā parametru u un v vērtība sakrīt ar vienādojumu sistēmu (2) sakni.

Iepriekšēja meklēšana

Tādēļ, lai iegūt iepriekšējas parametru u un v vērtības katrā pikselī tiek piedāvāta iepriekšējo vērtību kartes izveide. Karte tiek sastādīta no gabalvirsmas datiem, kuri tiek kodēti RGB kanālos. Sarkans (Red) kanāls ietver gabalvirsmas numuru. Matemātisku saistību var aprakstīt sekojošā veidā:

$$numurs = r - 1, \quad (6)$$

kur:

$numurs$ – gabalvirsmas numurs, kurš maiņas diapazonā [0; 254];

r – sarkana kanāla vērtība, kurā maiņas diapazonā [0; 255].

Gadījumā, ja $r=0$ var teikt kā šajā pikselī stars nekrustojas ne ar kādu gabalvirsmu.

Zaļš (Green) un zils (Blue) kanāli satur viendabīgu gradienta tekstūru, kura uzlikta uz gabalvirsmas. Pieņemsim, ka krāsu vērtība katrā kanālā ir veselais skaitlis no diapazona [0; 225].

Krāsu vērtības stūrains kontrolpunktos var piešķirt tādā veidā, kā ir aprakstīts 1.tabulā.

1. tabula

Table 1.

Stūrains kontrolpunktu krāsu vērtība
Colour values in angular control points

Kontrolpunkts	Krāsu vērtība
$P_{0,0}$	$(r; 0; 0)$
$P_{n,0}$	$(r; 255; 0)$
$P_{0,m}$	$(r; 0; 255)$
$P_{n,m}$	$(r; 255; 255)$

Pārējās krāsu vērtības gradienta tekstūrā vienmērīgi interpolējas un tiek uzliktas uz gabalvirsmas. Šajā gadījumā iepriekšējo parametru vērtību var atrast sekojošā veidā:

$$\begin{cases} u_0 = (u_{\max} - u_{\min}) \cdot \frac{g}{255} + u_{\min} \\ v_0 = (v_{\max} - v_{\min}) \cdot \frac{b}{255} + v_{\min} \end{cases}, \quad (7)$$

kur:

u_0, v_0 – parametru iepriekšēja vērtība pikselī;

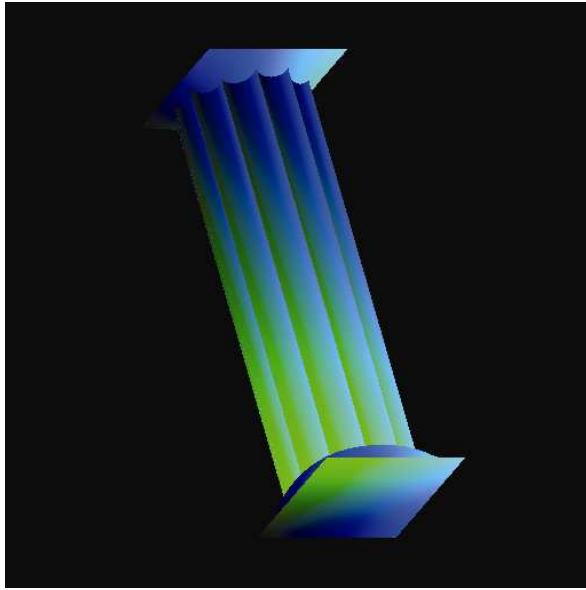
u_{\min}, u_{\max} – parametra u diapazons;

v_{\min}, v_{\max} – parametra v diapazons;

g – zaļa kanāla vērtība;

b – zila kanāla vērtība.

Iepriekšējo vērtību karti ērti izveidot, izmantojot grafiskas bibliotēkas iespējas. Darba tiek izmantota bibliotēka OpenGL. Iepriekšējo vērtību kartes piemērs parādīts 1.att..



1. att. Iepriekšējo vērtību kartes piemērs
Fig. 1. The example of preliminary values map

Parametru precizēšana

Optimizācijai mēs izvēlējamies gradienta metodi [8]. Gradienta metode ir iteratīva metode, kur katrā iterācijas solī notiek meklējamo parametru precizēšana. Matemātiski parametru precizēšanu var aprakstīt sekojoši:

$$\begin{cases} u^{new} = u^{old} - h \cdot \frac{\partial w}{\partial u} \Big|_{(u^{old}; v^{old})} \\ v^{new} = v^{old} - h \cdot \frac{\partial w}{\partial v} \Big|_{(u^{old}; v^{old})} \end{cases}, \quad (8)$$

kur:

u^{new}, v^{new} – parametru vērtība pēc iterācijas soli;

u^{old}, v^{old} – parametru vērtība pirms iterācijas soli;

h – svaru koeficients.

Parciālos atvasinājumus var izskaitļot sekojošā veidā:

$$\begin{aligned} \frac{\partial w}{\partial u} = 2 \cdot \left(Q_X(u, v) \cdot \frac{\partial Q_X(u, v)}{\partial u} + \right. \\ \left. + Q_Y(u, v) \cdot \frac{\partial Q_Y(u, v)}{\partial u} \right), \quad (9) \end{aligned}$$

un analogiski:

$$\begin{aligned} \frac{\partial w}{\partial v} = 2 \cdot \left(Q_X(u, v) \cdot \frac{\partial Q_X(u, v)}{\partial v} + \right. \\ \left. + Q_Y(u, v) \cdot \frac{\partial Q_Y(u, v)}{\partial v} \right), \quad (10) \end{aligned}$$

Matricas formā vienādojumu sistēmu (8) var aprakstīt sekojošā veidā:

$$\begin{aligned} \begin{bmatrix} u^{new} \\ v^{new} \end{bmatrix} = \begin{bmatrix} u^{old} \\ v^{old} \end{bmatrix} - 2 \cdot h \cdot \\ \cdot \begin{bmatrix} \frac{\partial Q_X(u, v)}{\partial u} & \frac{\partial Q_Y(u, v)}{\partial u} \\ \frac{\partial Q_X(u, v)}{\partial v} & \frac{\partial Q_Y(u, v)}{\partial v} \end{bmatrix} \cdot \\ \cdot \begin{bmatrix} Q_X(u, v) \\ Q_Y(u, v) \end{bmatrix}, \quad (11) \end{aligned}$$

Optimizācija tiek veikta līdz brīdim, kad būs izpildīts sekojošais nosacījums:

$$w \leq \varepsilon, \quad (12)$$

kur:

ε – precizitātes sliekšnis (esošā darbā bija pieņemts $\varepsilon < 0,001$).

Iterācijas darbā gaitā notika gadījumi kad $w(u^{new}; v^{new}) > w(u^{old}; v^{old})$. Šajā gadījumā nepieciešami samazināt svaru koeficientu h .

Matemātiski koeficienta aprēķins notika šādā veidā:

$$h^{new} = 0,618 \cdot h^{old} \quad (13)$$

Parametru precizēšanas paātrinājums

Pieeja, kura aprakstītā augšējā nodaļā ir diezgan efektīvā. Tajā pašā laikā, šī pieeja dod pārāk lielu aprēķinu apjomu. Galvenokārt tas ir saistīts ar to, ka gradienta izskaitļošana notika katrā iterācijā. Tagad ir iespējams izmantot gradienta vērtību (gradienta virzienu), kurš iegūts no iepriekšējās iterācijas. To var izdarīt gadījumā, ja iepriekšējā iterācijā tiek izpildīts nosacījums $w(u^{new}, v^{new}) > w(u^{old}, v^{old})$. Tāda pieeja ir zināma literatūrā kā visātrākās nolaišanas metode [8]. Pseudokodā tādu procedūru ir iespējams aprakstīt sekojošā veidā:

$$\Delta u = \text{grad_u}(u_{old}, v_{old})$$

$$\Delta v = \text{grad_v}(u_{old}, v_{old})$$

$$u_{new} = u_{old} - \Delta u$$

$$v_{new} = v_{old} - \Delta v$$

$$w_{old} = \mathbf{w}(u_{old}, v_{old})$$

$$w_{new} = \mathbf{w}(u_{new}, v_{new})$$

while $w_{new} > \varepsilon$ then

(

$$u_{new} = u_{old} - \Delta u$$

$$v_{new} = v_{old} - \Delta v$$

$$w_{new} = \mathbf{w}(u_{new}, v_{new})$$

if $w_{new} \geq w_{old}$ then

(

$$\Delta u = \text{grad_u}(u_{old}, v_{old})$$

$$\Delta v = \text{grad_v}(u_{old}, v_{old})$$

)

else

(

$$u_{old} = u_{new}$$

$$v_{old} = v_{new}$$

)

$$w_{old} = \mathbf{w}(u_{old}, v_{old})$$

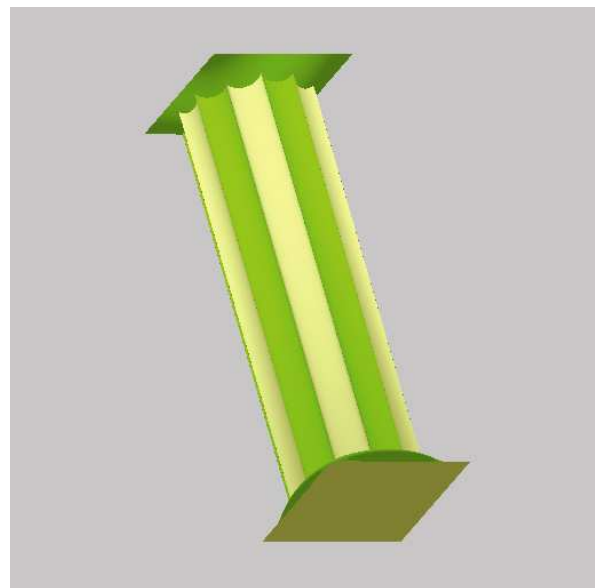
)

Augstāk bija apskatīta metode, kura ļauj atrast virsmas krustošanās punktu ar Z asi. Lai atrastu virsmas krustošanās punktu ar jebkuru staru pieņemsim sekojošu pieeju: transformējam virsmu un staru tāda veidā, lai stara sākums

sakristu ar vērotāja punktu un stara virziens sakristu ar Z asi. Šo transformāciju var izdarīt, izmantojot nobīdes un rotācijas matricu reizinājumu. Šīs problēmas risinājums aprakstīts darbā [9].

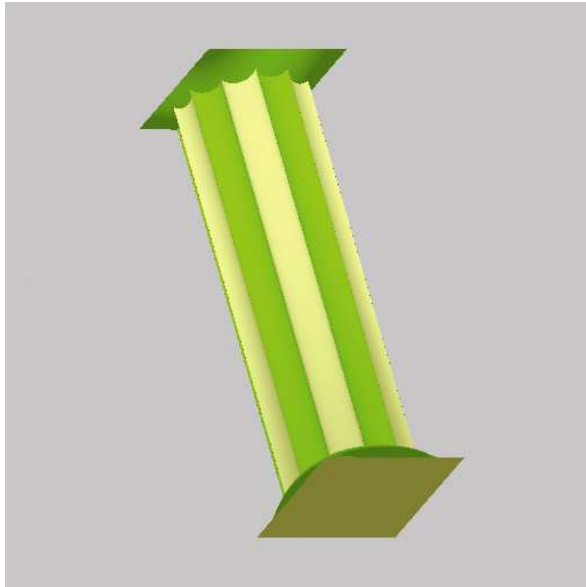
Eksperimentāli rezultāti

Esošā darbā piedāvātas metodes, gan arī metode, kura bija aprakstīta darbā [2] bijā realizētas datorprogrammas veidā. Lai vizualizēt 3D objektu bija izmantotā pieeja 1 stars uz pikseli. Attēlu izmērs eksperimentā bija 512×512 pikseļus. Eksperimenti tiek izveidoti izmantojot datoru ar sekojošo konfigurāciju: centrālais procesors Intel Xeon ar frekvenci 3,2 GHz, operatīvā atmiņa 2 GB, videoadapteris nVidia Quadro FX 1400. Eksperimentā gaitā bija vizualizētas 3 scēnas: pirmās divas scēnās tika vizualizēti eksperimentālie objekti un trešā scēna tika vizualizēts praktiskais objekts, kurš iegūts modelējot medicīnas objektu [10]. Iegūtie attēli tiek parādīti att. 2.-9.



2. att. Objekts, kas iegūts izmantojot piedāvāto metodi (nemodificēto)

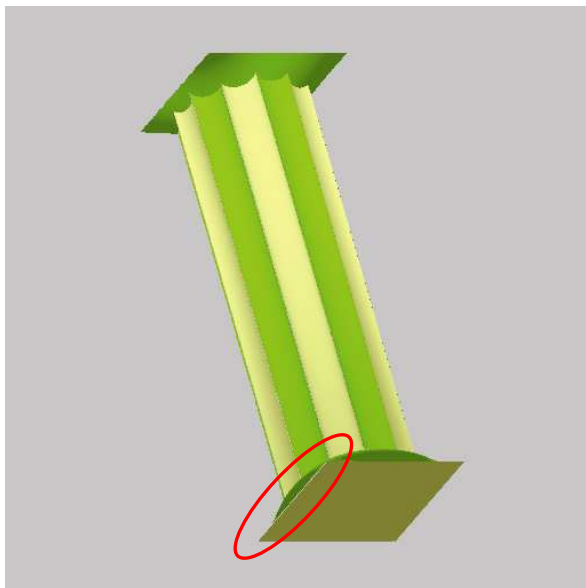
Fig. 2. The object obtained using the proposed method (non-modified)



3. att. Objekts, kas iegūts izmantojot piedāvāto metodi (modificēto)
Fig. 3. The object obtained using the proposed method (modified)



5. att. Objekts, kas iegūts izmantojot piedāvāto metodi (nemodificēto)
Fig. 5. The object obtained using the proposed method (non-modified)



4. att. Objekts, kas iegūts izmantojot metodi Martin un citi
Fig. 4. The object obtained using the method Martin et al.



6. att. Objekts, kas iegūts izmantojot piedāvāto metodi (modificēto)
Fig. 6. The object obtained using the proposed method (modified)



7. att. Objekts, kas iegūts izmantojot metodi Martin un citi

Fig. 7. The object obtained using the method Martin et al.

Kā ir iespējams redzēt no attēliem piedāvāta metode dod labāku rezultātu no attēla kvalitātes viedokļa (nav izkropļošanas uz gabalvirsmas robežām). Attēlu renderēšanas laiks tiek parādīts 2.tabulā.

2. tabula
Table 2.

Attēlu renderēšanas laiks sekundēs
Image rendering time in seconds

Metode	"Kolonna"	"Ūdensroze"
Piedāvāta (modificēta)	31,156	98,313
Piedāvāta (nemodificēta)	74,906	180,234
Martin un citi	60,750	205,600

Kā var redzēt no 2.tabulas attēliem nemodificēta piedāvāta metode dod nestabilu renderēšanas laika samazinājumu (salīdzinājos ar metodi Martin un citi). Tajā pašā laikā modificēta piedāvāta metode dod stabili labāku rezultātu pēc renderēšanas laika mūsu eksperimentā. Lai salīdzināt renderēšanas laiku pārveidosim 2.tabulā laiku procentos, pieņemot par 100% Martina metoša rezultāti.

3.tabula liecina, ka modificēta piedāvāta metode dod aptuveni divkārtējo renderēšanas laika samazināšanu salīdzinājāt ar Martina metodi.

3. tabula
Table 3.

Attēlu renderēšanas laiks procentos
Image rendering time in percentage

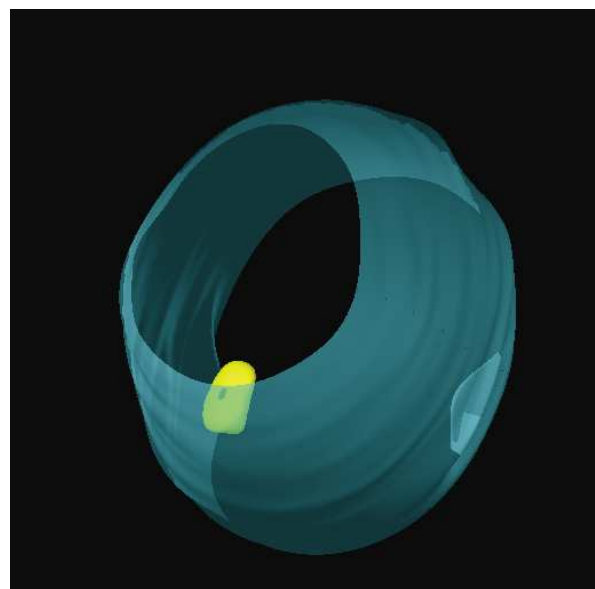
Metode	"Kolonna"	"Ūdensroze"
Piedāvāta (modificēta)	51,3	47,8
Piedāvāta (nemodificēta)	123,3	87,7
Martin un citi	100	100

4. tabula
Table 4.

Attēlu renderēšanas laiks
Image rendering time

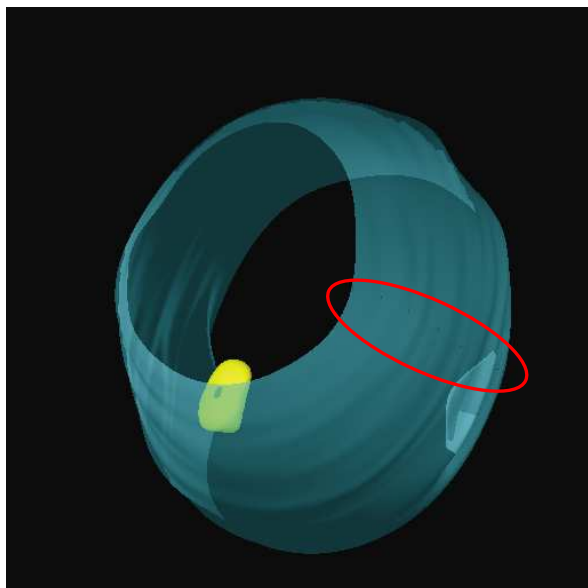
Metode	min.	%
Piedāvāta (modificēta)	17,683	69,98
Piedāvāta (nemodificēta)	24,533	97,09
Martin un citi	25,267	100

Metozu praktiskais pielietojums ir aprobācija uz medicīnas objekta 3D modeļa. Darbā piedāvāta metode, gan nemodificēta, gan modificēta dos ekvivalentus attēlus. Šis attēls parādīts 8.att.. Savukārt attēls, kurš iegūts izmantojot Martina metode parādīts 9.att..



8. att. Medicīnas objekts, iegūts izmantojot piedāvāto metodi

Fig. 8. The medicine object obtained using the proposed method



9. att. Medicīnas objekts, iegūts izmantojot metodi Martin un citi

Fig. 9. The medicin object obtained using the method Martin et al.

Kā ir iespējams redzēt no attēliem piedāvāta metode dod drusku labāku rezultātu no attēla kvalitātes viedokļa (nav izkropļošanas uz gabalvirsmas salaidumā). Attēlu renderēšanas laiki tiek parādīti 4.tabulā (minūtēs) un 5.tabulā (procentos).

Secinājumi

Iegūtie rezultāti, kuri tiek parādīti att. 2.-9. liecina, ka:

Piedāvāta metode (gan nemodificēta, gan modificēta) dod labāku rezultātu no attēla kvalitātes viedokļa. Tāds rezultāts tika iegūts visos trijos eksperimentos.

Nemodificēta piedāvāta metode dod labāko rezultātu pēc renderēšanas laika. Par to liecina eksperimentāli rezultāti otrā un trešā eksperimentā. Bet šis rezultāts nav stabils, par ko liecina pirmā eksperimenta rezultāts. Līdz ar to ir nepieciešama sākotnējās piedāvātās metodes modifikācija, kura arī bija veikta darbā.

Tajā pašā laikā modificētā piedāvātā metode dod stabilu labāku rezultātu pēc renderēšanas laika mūsu eksperimentā. Par to liecina eksperimentu rezultāti visos trijos eksperimentos.

Literatūra

1. Rodgers D., Adams J. Mathematical Elements for Computer Graphic, 2nd Ed., McGraw-Hill, Boston, MA, 1990.
2. Martin, W., Cohen, E., Fish, R., & Shirley, P. Practical Ray Tracing of Trimmed NURBS Surfaces. JGT 5(1): 27-52. 2000.
3. Nishita, T., Sederberg, T.W., and Kakimoto M. Ray tracing trimmed rational surface patches, Computer Graphics, vol.24, no. 4, pp. 337-345, 1990.
4. Efremov, A., Havran, V. & Seidel, H.-P. Robust and numerically stable Bézier clipping method for ray tracing NURBS surfaces. In Proceedings of the 21st Spring Conference on Computer Graphics SCCG'05:127 - 135. 2005
5. Sisojevs A., Glazs A. Visualization of Free – form Surfaces by a Ray Tracing Method// Scientific Proceedings of Riga Technical University, Computer Science, part 5, volume 32. – Riga: RTU, 2007. – 7 – 13 p.
6. Piegl L.A., Tiller W. The NURBS Book (Monographs in Visual Communication).- Springer-Verlag, 1997. – 646 p.
7. Hearn, D. and Baker, M.P. Computer Graphics with OpenGL, 3rd Ed., Prentice Hall, 2004.
8. Himmelblau, D. Applied Nonlinear Programming, – McGraw-Hill Book Company, 1972.
9. Sisojevs A. Staru trasēšana un bilineāru virsmu vizualizācija// Rīgas Tehniskās universitātes zinātniskie raksti, Datorzinātne, sērija 5., sējums 27. – Rīga: RTU, 2006. – 35. – 41. lpp.
10. Krečetova K., Glazs A. Patoloģijas zonu tilpuma noteikšana 3D medicīnas attēlos// Rīgas Tehniskās universitātes zinātniskie raksti, Datorzinātne, sērija 5., sējums 35. – Rīga: RTU, 2008. – 18. – 26. lpp.

Aleksandrs Sisojevs, PhD student

Riga Technical University, Faculty of Computer Science and Information Technologies, Institute of Computer Control, Engineering and Technology
Address: Meza str. 1, LV-1048, Riga, Latvia
E-pasts: alexiv@inbox.lv

Aleksandrs Glazs, Dr.habil.sc.ing., professor

Riga Technical University, Faculty of Computer Science and Information Technologies, Institute of Computer Control, Engineering and Technology

Address: Meza str. 1, LV-1048, Riga, Latvia
E-Mail: glaz@egle.cs.rtu.lv

Sisojevs A., Glazs A. 3D objektu vizualizēšana izmantojot b spline funkcijas un staru trasēšanu

Kopsavilkums – Brīvo formu virsmu vizualizēšanas problēma ar augstu kvalitāti ir aktuālā dažādas jomas zinātnē un tehnikā. Datorgrafikā eksistē vairāki matemātiskie apraksti, ar kuru palīdzību ir iespējams aprakstīt brīvās formas virsmu. Viens no efektīvākiem un elastīgākiem aprakstiem ir B-spline virsmas. 3D attēlu sintēze mūsdienu laikā attīstās divos virzienos: pirmajā gadījumā galvenā loma ir ātrdarbībai un otrajā gadījumā pieeja paredzēta, lai dabūtu attēla augstāko precizitātes un realitātes pakāpi. Darbā tiek apskatīta viena no 3D scēnu vizualizēšanas metodēm – staru trasēšana. Šī metode ir populāra augsta kvalitātes vizualizēšanas metode. Dotajā darbā tiek piedāvāta jaunā metode, ar kuru palīdzību ir iespējams vizualizēt B-spline virsmu ar augsto precizitāti un kvalitāti. Piedāvāta metode (gan nemodificēta, gan modificēta) dod labāku rezultātu no attēla kvalitātes viedokļa. Nemodificēta piedāvāta metode dod labāko rezultātu pēc renderēšanas laika, bet šis rezultāts nav stabils. Līdz ar to ir nepieciešama sākotnēja piedāvāta metoša modifikācija, kura arī bija veikta darbā. Modificēta piedāvāta metode arī dod stabilu labāku rezultātu pēc renderēšanas laikā. Par to liecina eksperimentāli rezultāti eksperimentos.

Sisojevs A., Glazs A. 3D objects visualization using B-spline functions and ray tracing

The problem of free-form surfaces visualization with high quality is actual in various areas of a science and engineering. There are various mathematical descriptions in the computer graphics that describe a free-form surface. One of the most effective and flexible descriptions is B-spline surface. The synthesis of 3D images presently develops in two directions: on one hand the main role is speed and on the other hand the approach is intended to obtain an image with high accuracy and degrees of realness. In this work, one of methods of 3D scenes visualization – ray tracing is described. This method is a popular method of high quality visualization. In this work the new method is proposed that gives the possibilities to render B-spline surface with high accuracy and quality. The proposed method (unmodified and modified) gives the best result from the viewpoint of quality of the image. The unmodified proposed method gives the best result on rendering's time, but this result is unstable. Due to this, the modification of the primary proposed method is necessary, which was also proposed in this work. The modified proposed method gives stably best result on rendering time. The results of experiments testify to it.

Сысоев А., Глаз А. Использование B-spline функций и трассировки лучей для визуализации 3D объектов

Проблема визуализации криволинейных поверхностей с высоким качеством актуальна в различных областях науки и техники. В компьютерной графике существуют различные математические описания, с помощью которых возможно описать поверхность свободной формы. Эффективным и гибким описанием является B-spline поверхности. Синтез 3D изображений в наше время развивается в двух направлениях: в первом случае в главной роли быстроедействие и во втором случае подход предназначен, что бы получить изображение высокой точности и степени реалистичности. В работе описан один из методов визуализации 3D сцен - трассировка лучей. Этот метод является популярным методом визуализации высокого качества. В данной работе предложен новый метод, с помощью которого возможно визуализировать B-spline поверхность с высокой точностью и качеством. Предложенный метод (как немодифицированный, так и модифицированный) дает лучший результат с точки зрения качества изображения. Немодифицированный предложенный метод дает лучший результат по времени рендеринга, но этот результат нестабилен. В связи с этим необходима модификация изначального предложенного метода, которая так же была предложена в работе. Модифицированный предложенный метод так же дает стабильный лучший результат по времени рендеринга. Об этом свидетельствуют результаты экспериментов.