Lietuvos matematikos rinkinys
Lietuvos matematikų draugijos darbai, ser. B
53 t., 2012, 145–150

ISSN 0132-2818
www.mii.lt/LMR/

Skaitmeninio vaizdo elementų programavimo aspektai

Rima Birškytė, Olegas Ramašauskas

Klaipėdos universitetas, Gamtos ir matematikos mokslų fakultetas
H. Manto g. 84, LT-92291 Klaipėda
E. paštas: rima.birskyte@gmail.com, olegas@ik.ku.lt

Santrauka. Straipsnis yra skirtas matematikos ir informatikos studijų dalykų įsisavinimui vizualinio objektinio programavimo priemonėmis. Kompiuterių grafikos uždavinių sprendimui patogu naudoti naujus skaičiuojamosios matematikos programinius paketus, kurie jau yra praturtinti skaitmeninių vaizdų apdorojimo įrankiais, naujomis funkcijomis ir objektais, leidžiančiais sukurti, spalvinti ir animuoti įvairius realaus pasaulio objektus kompiuterio ekrane, simuliuoti ir naudoti virtualiosios realybės produktus.

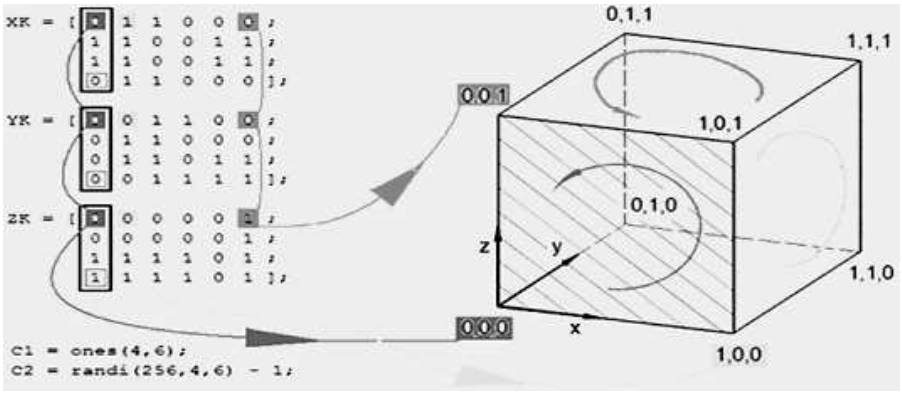
Raktiniai žodžiai: skaitmeninis vaizdas, programavimas, kompiuterių grafika, matrica.

1 Įvadas

Kompiuterinės matematinės programos yra sukurtos sudėtingų skaičiavimų palengvinimui ir abstrakčių duomenų vizualizavimui [3] ir gali būti plačiai taikomos įvairiose mokslo ir studijų srityse. Tokias programas lengviau įsisavinti negu tradicines programavimo kalbas [2]. Matematinių programų (Maple, Mathematica, Octave, programa R ir pan.) pagalba galima atlikti daugelį tradicinių užduočių, pavyzdžiui: išspręsti lygčių sistemas, diferencialines lygtis, pavaizduoti funkcijas, sukurti įvairius modelius, ištirti jų savybes, todėl tokių nuolat atnaujinamų ir papildomų naujomis galimybėmis programų sistemų studijavimas yra labai aktualus. Informatikos ir tradicinės matematikos uždavinių analizei naudojama keletas populiarių paketų, tačiau vis didesnį populiarumą įgyja sudėtingas, kompleksinis, nuolat atnaujinamas įrankis – Matlab programų paketas [4]. Jo galimybės dažnai išnaudojamos tik iš dalies (ir tai savaime suprantama, programa – didelė, sudėtinga, skirta ne tik teoriniam darbui, bet ir inžinerijai, elektronikai, ekonomikai, vaizdų apdorojimui ir t. t.). Siekiant atskleisti naujų programos versijų galimybių veikimą, studijuojant matricinių skaičiavimų taikymus skaitmeninio vaizdo elementų kūrimui bei apdorojimui, buvo pasirinktas tyrimui pakankamai vaizdus vienetinio kubo programavimo uždavinys, sprendžiamas naudojantis koordinačių aprašymo ir briaunų jungimo metodais [1]. Tai leistų vaizdžiai supažindinti besimokančiuosius su skaitmeninio vaizdo elementų aprašymo metodais bei programavimo aspektais, naudingais mokantis kompiuterinės grafikos programavimo pagrindų.

2 Koordinačių aprašymo metodas

Aprašomas metodas priklauso žemesnio lygmens programavimo sprendimams, tačiau leidžia lanksčiai manipuliuoti kiekvienu paveikslo elementu. Metodo esmę sudaro



1 pav. Viršūnių koordinačių ir spalvų (čia $C1$, $C2$) specifikavimo schema.

nuoseklus kuriamųjų objektų koordinačių vektorių (matricų) specifikavimas, kuriais aprašyti taškai ir linijos vidinių sistemos funkcijų pagalba sujungiami ir nupiešiami (angl. rendering) kaip paveikslo elementai: viršūnės, briaunos ar paviršiai, sudarantys sudėtingesnius objektus, algoritmas vaizdžiai pateiktas 1-jame paveiksle.

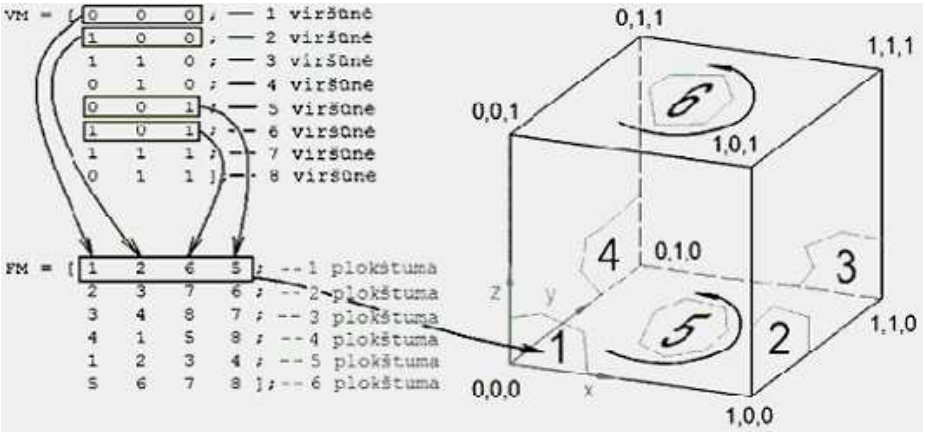
Pavyzdžiui, klasikinis uždavinys, kai sukuriamas ir uždažomas vienetinis kubas, kurį sudaro 8 viršūnės ir 6 plokštumos, iš viso reikalauja 24 viršūnių koordinačių aprašymo, kadangi kiekviena iš 6 plokštumų aprašoma 4-is viršūnėmis, neatsižvelgiant į tai, kad gretimose plokštumose dalis jų sutampa. Kita vertus, toks mechanizmas leidžia kuriamojo objekto poligonų plokštumas išdėstyti erdvėje nežiūrint ar jos liečiasi, ar kertasi, ar prasilenkia. Tokiam sprendimui naudotinos trys $M \times N = 4 \times 6$ koordinačių reikšmių matricos, talpinančios x, y, ir z koordinačių masyvus, kurių perrinkimą ir vizualizavimą galima atlikti Matlab patch komanda, laikantis 1 paveiksle parodyto įrašų nuoseklumo.

Jei kuriamojo vienetinio kubo spalvinį sprendimą nurodysime parametru 'w', tai objektas bus sugeneruotas baltos spalvos su juodomis linijomis, kaip pirmajame 3-ojo paveikslo pavyzdyje. Spalvą galima nurodyti paletėmis, pavyzdžiui, hsw(M) atspalviosodrumo vertėmis ($M \times 3$) arba pasirinktu spalvų sąrašu vektoriumi, arba koordinačių matricų dydžio spalvų matrica ($C1$ ar $C2$). Rezultatai matyti antrajame 3-ojo paveikslo pavyzdyje.

3 Briaunų jungimo metodas

Tai aukštesniojo programavimo lygmens metodas. Jis remiasi tuo principu, kad sutampantys elementai (viršūnės ar plokštumos) neturėtų būti pakartotinai aprašinėjami. Kadangi kiekviena kubo plokštuma dalinasi viršūnėmis su likusiomis keturiomis, tai kur kas efektyvesnis sprendimas būtų toks, kai kiekviena viršūnė deklaruojama tik vieną kartą ir specifikuojamas jų sujungimo eiliškumas, leidžiantis suformuoti visas tarp viršūnių nubrėžtomis briaunomis besiribojančias plokštumas ir išgauti norimą vienetinio kubo atvaizdą.

Tokio uždavinio sprendimui pakanka sudaryti nebe tris, o dvi duomenų matricas. Viena aprašo visas aštuonias kubo viršūnių koordinates x, y ir z, todėl jos dydis yra $M \times N = 8 \times 3$, antroji nurodo šešias plokštumas, kurias tos viršūnės tenkina, todėl jos



2 pav. Viršūnių ir plokštumų specifikavimo apėjimo tvarka.

dydis yra $M \times N = 6 \times 4$. Matyti, kad abi matricos turi vienodai elementų, skiriasi tiktai jų apdorojimo tvarka, kuri vaizdžiai parodyta 2-ojo paveikslo mnemoninėje schemoje. Svarbu tai, kad tiek šio vienetinio kubo, tiek jo briaunų, tiek kiekvienos plokštumos viršūnių apėjimo tvarka yra prieš laikrodžio rodyklę.

Kadangi šio metodo formalioji sintaksė griežtai reikalauja tiktai viršūnių ir paviršių sąrašo, todėl vaizduojamųjų objektų spalvinimas atliekamas (jeigu atliekamas!) parametriškai, tolygiuoju (iš angl. *flat*) arba interpo-liaciniu (iš angl. santr. *interp*) būdu, nurodant plokštumų, briaunų ar viršūnių spalvų paletę ir jos pritaikymą funkcijos parametrų eilutėje. Analogiškai galėtų būti kuriami kiti trimačiai geometriniai objektai.

4 Vienetinio kubo programavimas

Vizualinių objektų programavimui naujesnėse Matlab programų versijose (nuo 7.x) yra įdiegtos kompleksinės vaizdų apdorojimo priemonės (angl., *Computer Vision System Toolbox 4.0, Simulink 3D Animation, Image Acquisition Toolbox*), todėl aiškinant programavimo aspektus vienetinis kubas kuriamas Matlab grafinėje aplinkoje. Sudarome scenarijų, kuris turės generuoti ketvertą trimačių objektų pavyzdžių, du iš jų koordinačių aprašymo, likusius du – briaunų jungimo metodu, vaizdumo pagerinimui naudojant skirtingus spalvinimo įrankius. Erdvinių vaizdų išgavimą (naudojant žemojo lygio sintaksę) pradedame nuo koordinačių matricų sudarymo, kurios skaitomos stulpeliais, iš viršaus žemyn, atitinkamus elementus sujungiant į koordinačių triadas (čia ir toliau tekste kodo pavyzdžiams naudojama Matlab notifikacija).

```
K = [ 0 1 1 0 0 0 ;% 4x6 x-koordinačių matrica,
      1 1 0 0 1 1 ;
      1 1 0 0 1 1 ;
      0 1 1 0 0 0 ];
YK = [ 0 0 1 1 0 0 ;% 4x6 y-koordinačių matrica,
       0 1 1 0 0 0 ;
       0 1 1 0 1 1 ;
       0 0 1 1 1 1 ];
```

```
ZK = [ 0 0 0 0 0 0 1 ;% 4x6 z-koordinačių matrica,
       0 0 0 0 0 0 1 ;
       1 1 1 1 0 1 ;
       1 1 1 1 0 1 ]; % sudarytos trys 24 elementų koordinačių matricos.
```

Pirmojo ir antrojo pavyzdžių vizualizacija grafiniame lange $f1$ su dviem polangiais $h1$ ir $h2$

```
h1 = subplot(2,2,1,'Parent',f1,...
                    'PlotBoxAspectRatio',[1 1 1]);
    p1 = patch(XK,YK,ZK,'w'); % juodai baltas vaizdavimas,
    title('1 pavyzdys','FontWeight','bold');
    view(3); % polangyje rodomas trimatis paveikslas p1.
```

Antrajame pavyzdyje spalvų išgavimui galima išbandyti įvairias alternatyvas, pavyzdžiui:

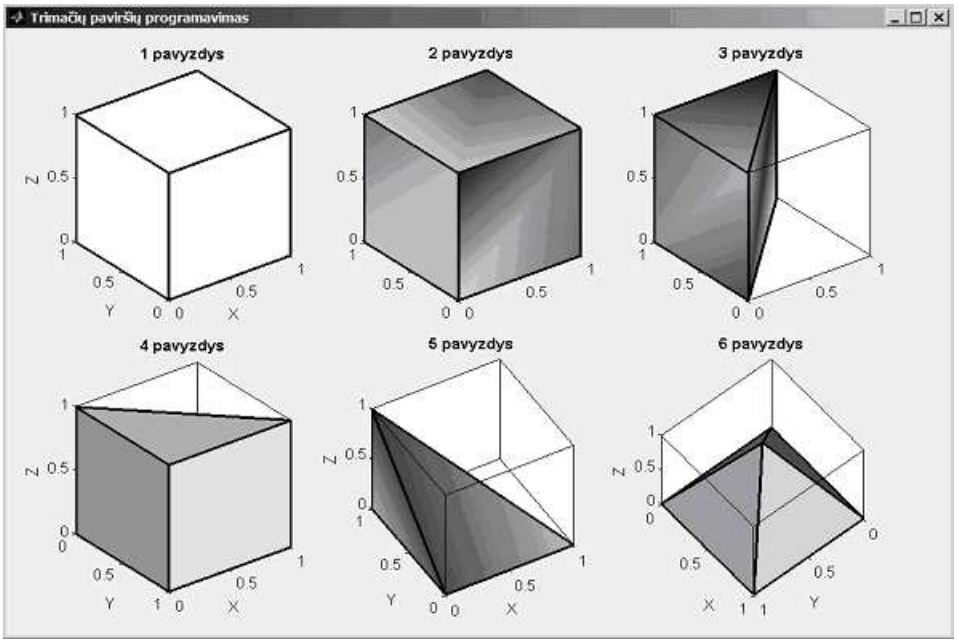
```
C1 = ones(4,6);% vienetų matrica, vienspalvė,
C2 = randi(256,4,6) - 1;% 4x6 sveikųjų matrica [0 255],
C3 = randi(256,1,6) - 1;% 1x6 atsitiktinių bitų 255 spalvų vektorius.
```

Pastaba: visuose pavyzdžiuose išvedimui naudojamas tas pats ekrano parametrais aprašomas grafinis langas $f1 = figure(\ldots)$ su šešetu polangių $h1$-6. Aprašymo metodas standartinis, todėl plačiau nenagrinėjamas.

```
h2 = subplot(2,2,2,'Parent',f1,...
                    'PlotBoxAspectRatio',[1 1 1]);
    p2 = patch(XK,YK,ZK,C2);% parinktas C2, alternatyvos C1 ir C3,
    title('2 pavyzdys','FontWeight','bold');
    view(3); % polangyje rodomas trimatis paveikslas p2,
axis([h1 h2],'square');% užbaigiamas koordinačių metodas.
```

Sekantys pavyzdžiai iliustruoja briaunų jungimo metodą, kai vykdomas viršūnių apėjimas nustatyta kryptimi, generuojamos poligonų plokštumų briaunos ir vizualizuojami iš jų sudaryti 3D paviršiai. Nors galima pasirinkti juodai-baltą arba pilkosios skalės vaizdą, programuojamas tolygus arba interpoliacinio tipo spalvotas gradientinis paviršių užpildymas, parodytas 3 paveiksle. Uždavinyje naudotos $hsv(5)$, $hsv(6)$, $hsv(8)$ paletės.

```
VM = [0 0 0; % Viršūnių masyvas MxN=8x3 (angl. Vertice's),
      1 0 0; % pašalinę 2 ir 6 eilutes sukurtume trikampę prizmę,
      1 1 0;
      0 1 0;
      0 0 1;
      1 0 1;
      1 1 1;
      0 1 1]; % Viršūnių masyvo (VM matricos) pabaiga.
FM = [1 2 6 5; % Paviršių masyvas MxN=6x4 (angl. Face's),
      2 3 7 6; % Pastaba: pašalinę 6 eilutę ir perrašę viršūnių
      3 4 8 7; % apėjimą, pavaizduotume vienetinę trikampę prizmę
      4 1 5 8; % naujo paviršių masyvo MxN=5x4 pagalba;
      1 2 3 4;
      5 6 7 8]; % Paviršių masyvo (FM matricos) pabaiga.
```



3 pav. Vienetinės 3D figūros, pateiktos dviem programavimo metodais.

Trečiojo ir ketvirtojo pavyzdžių vizualizacija pateikta grafinio lango $f1$ apatiniuose polangiuose $h3$ ir $h4$.

```
h3 = subplot(2,2,3,'Parent',f1);
p3 = patch('Vertices',VM,...
           'Faces',FM,...
           'FaceVertexCData',hsv(6),...
           'FaceColor','flat');% tolygus gradientinis užpildas,
     title('3 pavyzdys','FontWeight','bold');
     view(3); % polangyje rodomas trimatis paveikslas p3,
h4 = subplot(2,2,4,'Parent',f1); p4 = patch('Vertices',VM,...
           'Faces',FM,...
           'FaceVertexCData',hsv(8),...
           'FaceColor','interp');% interpoliacinis gradientinis užpildas
     title('4 pavyzdys','FontWeight','bold');
     view(3); % polangyje rodomas trimatis paveikslas p4,
axis([h3 h4],'square');% likę tušti polangiai užpildomi analogiškai.
```

Tiek vienetinio kubo, tiek kitų nesudėtingų trimačių paviršinių objektų (prizmės, piramidės) programinis kūrimas funkcijos patch pagalba aiškus iš 1–2 paveiksluose pateiktų mnemoninių schemų.

5 Išvados

Tyrimas parodė, kad naujos Matlab programų pakete įdiegtos vaizdo apdorojimo priemonės (angl., *Computer Vision System Toolbox 4.0, Simulink 3D Animation, Image*

Acquisition Toolbox) leidžia tvarkyti įvairiais būdais išgautus statinius skaitmeninius vaizdus. Darbe aprašyti metodai, paaiškinantys kaip įsisavinus skaitmeninio vaizdo suformavimo pagrindus ir pritaikius naujas objektiškai orientuotas programos galimybes įmanoma nesunkiai kurti, programuoti, pavaizduoti ir įrašyti spalvotus, vienspalvius ar nespalvotus (pilkosios skalės ar binarinius) paveikslus, nebereikia aprašinėti atskirų komponentų.

Darbo rezultatai gali būti panaudoti ne tik informatikos specialybės studentų pratyboms, bet ir skaičiuojamosios matematikos priemonių studijoms tiek vidurinio, tiek aukštojo mokslo įstaigose, kadangi aprašytasis Matlab įrankynas turi naujas sąsajos kūrimo galimybes, visas reikalingas grafinio programavimo konstrukcijas bei matematinį aparatą.

Literatūra

[1] I. Bilinskis, A. Skageris ir K. Sudars. Method for fast and complexity-reduced asymmetric image compression electronics and electrical engineering. *Technologija*, **4**(110):117–120, 2011.

[2] A.A. Bielskis. *Trimatė grafika: programavimo pagrindai*. Smaltija, Kaunas, 2000. 199 p.

[3] A. Gorban, B. Kegl, C.D. Wunsch and A. Zinovyev. *Principal Manifolds for Data Visualization and Dimension Reduction*. Springer-Verlag, Berlin, Heidelberg, NY, 2008. ISBN 978-3-540-73749-0.

[4] MATLAB Programming//Handle Graphics. Adresas internete: http://en.wikibooks.org/wiki/MATLAB_Programing/Handle_Graphics.

SUMMARY

Pixel programming aspects of digital image

R. Birškytė, O. Ramašauskas

This article describes a realization of the mathematics and informatics subjects by using visual objective programming facilities. For solving advanced computer graphics tasks, seems comfortable to use new estimated mathematics programs and packages, which already are enriched with digital image processing tools, allowing to create, paint and animate a various objects of the real world on computer screen, to extend the products of virtual reality. The described image reconstruction algorithms have been evaluated on the basis of computer simulations.

Keywords: Digital image, programming, computer graphics, matrix.