

Diskretizirano spektralno upodabljanje volumetričnih podatkov

Jernej Jezeršek¹, Matija Marolt², Žiga Lesar², Ciril Bohak²

¹ Univerza v Ljubljani, Fakulteta za matematiko in fiziko

² Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

E-pošta: jj13824@student.uni-lj.si, {matija.marolt, ziga.lesar, ciril.bohak}@fri.uni-lj.si

Discretised spectral volume rendering

Spectral volume rendering enables more physically accurate light modeling than traditional approaches that use the RGB color space. In reality the color spectrum is a continuous function of wavelength. In this article we present a representation of color spectra as an approximation of such function. We created our own implementation based on previous works in the field by extending the volumetric path tracing framework (VPT). The results show advantages of spectral volume rendering such as simulation of metamerism and differences of light scattering depending on the wavelength. However those advantages come at a cost of higher computational complexity compared to non-spectral approaches.

1 Uvod

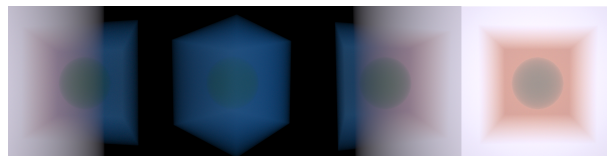
V članku se osredotočamo na spektralno upodabljanje volumetričnih podatkov s sledenjem potem. Običajno se pri upodabljanju sevalnost predstavi s trojčkom vrednosti, ki predstavljajo rdečo, zeleno in modro barvo. To je smiselna izbira, saj človeški vidni sistem in posledično večina tehnologij za zajem in prikaz slik (zasloni in senzorji v kamerah) temelji na tribarvni teoriji. A svetloba je v resnici sestavljena iz zveznega spektra valovnih dolžin in s poenostavitvijo na tri barve ne moremo modelirati vseh njenih lastnosti. Na primer, svetloba se glede na valovno dolžino bolj ali manj sipa v različnih medijih. Modra svetloba se v atmosferi bolj sipa kot rdeča, zato je nebo videti modro. Prav tako se svetloba lomi pod različnimi koti v odvisnosti od valovne dolžine. Zato vidimo mavrične barve, ko svetloba prehaja skozi prizmo.

Če želimo fizikalno natančno modelirati svetlobo, moramo uporabiti spektralno upodabljanje. To pomeni, da namesto treh vrednosti pri izračunih hranimo celoten spekter svetlobe. Praktično je to seveda nemogoče, saj bi morali hraniti neskončno vrednosti, zato uporabimo približke ali pa spekter svetlobe naključno vzorčimo in prispevke fotonov sproti pretvarjamo v tri primarne barve.

Poleg bolj natančnega simuliranja svetlobe spektralno upodabljanje omogoča modeliranje večjega nabora materialov, med drugimi tudi fizikalno nemogočih. S tem je omogočena večja fleksibilnost pri oblikovanju prenosne funkcije in vizualizaciji volumnov [8]. Če sliko shranimo v obliki (diskretiziranega) spektra, lahko osvetlitev kasneje ponovno izračunamo z drugačno svetlobo in na ta

način skrijemo ali pokažemo različne dele volumna brez potrebe po ponovnem upodabljanju [2, 1].

V nadaljevanju predstavimo implementacijo spektralnega upodabljanja v ogrodju VPT s poudarkom na spektralni predstavitvi svetlobe in barvnih pretvorbah pri prikazu na zaslon. Nato opišemo, kako deluje algoritem za upodabljanje volumetričnih podatkov s sledenjem poti in kako smo ga prilagodili za spektralno upodabljanje. Na koncu implementacijo performančno ovrednotimo in podamo kritično mnenje. Prispevek tega dela je implementacija diskretnega spektralnega upodabljanja v spletno volumetrično upodobljevalno ogrodje VPT [7].



Slika 1: Testni volumen, sestavljen iz kocke in krogel. Kocka je iz materiala, ki sipa več modre svetlobe in prepušča več rdeče svetlobe. Zato je kocka videti modra, ko jo osvetlimo iz strani, in bolj rdeča, ko jo osvetlimo od zadaj.

2 Predstavitev spektra

Najbolj natančna predstavitev spektra svetlobe je funkcija $L : \mathbb{R} \rightarrow \mathbb{R}$, ki valovno dolžino λ preslika v sevalnost L . V nadaljevanju bomo s spektrom predstavili sevalnost svetlobe, barvo (albedo) in prosojnost materialov. Zato ne bomo določili enote vrednosti l .

Prostor funkcij L je neskončno dimenzionalen, zato je vse njegove elemente nemogoče hraniti. V praksi lahko z množico baznih funkcij (npr. polinomi določene stopnje, odsekoma linearnimi funkcijami ipd.) definiramo n -dimenzionalen podprostor in spekter hranimo kot vektor koeficientov v linearni kombinaciji baznih funkcij.

Pharr idr. [10] predlagajo uporabo prostora odsekov konstantnih funkcij, ki jih lahko zapišemo kot:

$$H_i(\lambda) = \begin{cases} 1 & \lambda \in [\lambda_{i-1}, \lambda_i] \\ 0 & \text{sicer,} \end{cases}, \quad (1)$$

kjer so za $i \in \{1, \dots, n\}$, $\lambda_0, \dots, \lambda_n$ valovne dolžine, ki določajo n odsekov. Uporabna lastnost te baze je, da

so funkcije H_1, \dots, H_n paroma ortogonalne. To nam omogoča, da lahko za dan spekter $L(\lambda)$ preprosto poiščemo aproksimacijo po metodi najmanjših kvadratov:

$$\underbrace{[H_i, H_j]}_{\text{Gramova matrika}} \underline{\alpha} = [L, H_i]. \quad (2)$$

Gramova matrika ima v tem primeru neničelne elemente le na diagonali. Zato lahko enačbo rešimo z deljenjem po elementih. Če upoštevamo še definicijo baznih funkcij H_i , dobimo:

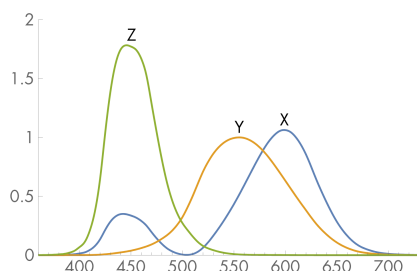
$$\alpha_i = \frac{\langle L, H_i \rangle}{\langle H_i, H_i \rangle} = \frac{\int_{\lambda} L(\lambda) H_i(\lambda) d\lambda}{\int_{\lambda} H_i(\lambda) H_i(\lambda) d\lambda} = \frac{\int_{\lambda_{i-1}}^{\lambda_i} L(\lambda) d\lambda}{\lambda_i - \lambda_{i-1}}. \quad (3)$$

Aproksimacija spektra $L(\lambda)$ po metodi najmanjših kvadratov je torej:

$$L(\lambda) \approx \sum_{i=1}^n \alpha_i H_i(\lambda). \quad (4)$$

2.1 Pretvorba v barvni prostor XYZ

Človeško oko ima tri tipe čepkov, katerih odziv na svetlobni spekter lahko predstavimo v barvnem prostoru XYZ. Funkcije $X(\lambda)$, $Y(\lambda)$, $Z(\lambda)$ predstavljajo odzivnost očesa na dano valovno dolžino svetlobe (slika 2).



Slika 2: Funkcije $X(\lambda)$, $Y(\lambda)$, $Z(\lambda)$, ki predstavljajo odzivnost čepkov v človeškem očesu na dano valovno dolžino svetlobe [10].

Za dani spekter $L(\lambda)$ so koeficienti x , y , z definirani kot:

$$x = \int_{\lambda} X(\lambda) L(\lambda) d\lambda, \quad (5)$$

$$y = \int_{\lambda} Y(\lambda) L(\lambda) d\lambda, \quad (6)$$

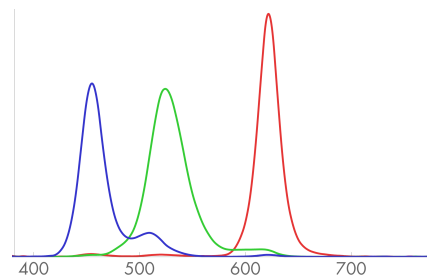
$$z = \int_{\lambda} Z(\lambda) L(\lambda) d\lambda. \quad (7)$$

2.2 Pretvorba v barvni prostor RGB

Če želimo sliko prikazati na zaslonu, moramo spekter pretvoriti v trojico števil, ki določa barvo v barvnem prostoru RGB. Slika 3 prikazuje spekter bele svetlobe, kakršnega izseva tipičen LED zaslon.

To lahko naredimo z linearno preslikavo iz barvnega prostora XYZ v RGB [4]:

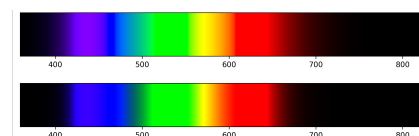
$$\begin{bmatrix} \hat{r} \\ \hat{g} \\ \hat{b} \end{bmatrix} = \begin{bmatrix} 3,2406 & -1,5372 & -0,4986 \\ -0,9689 & 1,8758 & 0,0415 \\ 0,0557 & -0,2040 & 1,0570 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (8)$$



Slika 3: Primer spektra bele svetlobe tipičnega LED zaslona [10].

Preslikava med barvnima prostoroma XYZ in sRGB ni linearna (slika 4), zato moramo opraviti še gama korekcijo nad vsakim barvnim kanalom $c \in \{r, g, b\}$:

$$c = \begin{cases} 12,92\hat{c} & \hat{c} \leq 0,0031308, \\ 1,055\hat{c}^{1/2,4} - 0,055 & \hat{c} > 0,0031308. \end{cases} \quad (9)$$



Slika 4: Zgoraj: pretvorba spektra v RGB z gama korekcijo. Spodaj: pretvorba brez gama korekcije.

3 Upodabljanje volumnov

Za upodabljanje volumnov obstaja več tehnik, ki jih v splošnem delimo na posredne in neposredne. Posredne tehnike volumen pred upodabljanjem pretvorijo v alternativno predstavitev (npr. trikotniško mrežo, oblak točk, predznačeno funkcijo razdalje), medtem ko neposredne tehnike pretvorb ne izvajajo. V tem delu smo se osredotočili na neposredno upodabljanje s sledenjem poti. Algoritem deluje tako, da za vsak piksel izračuna sevalnost svetlobe, podano z integralom virov S , uteženih s prepustnostjo T [9, 6]:

$$L(x, \omega, \lambda) = \int_0^{\infty} T(x, y, \lambda) S(y, \omega, \lambda) dy, \quad (10)$$

kjer je x izhodiščna točka žarka in ω smer žarka. Pri spektralnem upodabljanju so vsi parametri materiala, vsebovani v S , odvisni od valovne dolžine λ , določeni pa so z uporabniško definirano prenosno funkcijo.

Integral rešujemo numerično z metodo Monte Carlo. Za vsak piksel generiramo fotone z naključno valovno dolžino in jih pošljemo skozi volumen. Foton začne svojo pot v prvem presečišču žarka z volumnom, nato se premakne za naključno razdaljo (ki jo vzorčimo iz eksponentne porazdelitve) v smeri žarka. Če foton izstopi iz volumna, vzorčimo sevalnost ozadja, sicer pa naključno izberemo tip interakcije z volumnom (absorpcija, sipanje ali ničelni trk). Podrobnejši opis algoritma je opisan v [6].

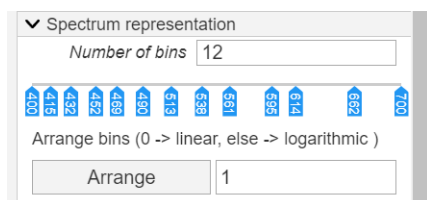
4 Implementacija

Za osnovo implementacije smo uporabili ogrodje za upodabljanje volumetričnih podatkov – VPT [7], ki je bilo prilagojeno za delo z vmesnikom WebGPU. Ta omogoča uporabo računskih senčilnikov, kar bistveno olajša shranjevanje slike v obliki spektra namesto v formatu RGB.

Ogrodje VPT je zasnovano tako, da podpira različne module za upodabljanje in poobdelavo rezultatov. Dodali smo nov modul za spektralno upodabljanje volumetričnih podatkov. Za osnovo smo vzeli obstoječ modul za upodabljanje volumnov s sledenjem poti in večkratnim sipanjem svetlobe.

4.1 Modul za upodabljanje

Obstoječi modul smo razširili tako, da za vsak piksel hrani spekter svetlobe namesto vrednosti RGB. Spekter je predstavljen z linearno kombinacijo odsekoma konstantnih funkcij, kot je opisano v poglavju 2. Število odsekov in meje lahko uporabnik poljubno določi v vmesniku (slika 5) ali pa jih razporedi linearno oz. logaritemsko. Vsak foton prispeva sevalnost natanko enemu odseku spektra.



Slika 5: Uporabniški vmesnik za določanje odsekoma konstantne predstavitev spektra.

4.2 Prenosna funkcija

Prenosno funkcijo smo prilagodili tako, da vrednost volumna in valovno dolžino fotona preslika v tri optične lastnosti materiala:

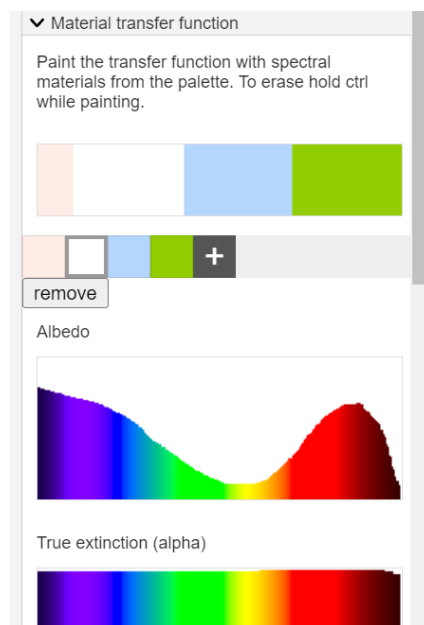
- albedo – poveča verjetnost sipanja in zmanjša verjetnost absorpcije,
- ekstinkcija – poveča intenzivnost interakcij med svetlobo in materialom,
- anizotropija – spremeni porazdelitev smer sipanja.

Odločili smo se za predstavitev prenosne funkcije z uporabo teksture formata RGB, saj ta omogoča največjo fleksibilnost. Tekstura je sestavljena tako, da x os predstavlja valovno dolžino, y os pa surovo vrednost volumna. Kanal R predstavlja albedo, kanal G ekstinkcijo, kanal B pa anizotropijo.

Implementacija omogoča tudi uporabo več različnih materialov. Uporabnik najprej določi preslikavo med surovimi vrednostmi volumna in materiali. Med upodabljanjem se glede na surovo vrednost volumna izbere material, katerega pripadajoča prenosna funkcija se nato uporabi za določanje optičnih lastnosti. Uporabniški vmesnik za določanje materialov in prenosnih funkcij je prikazan na sliki 6.

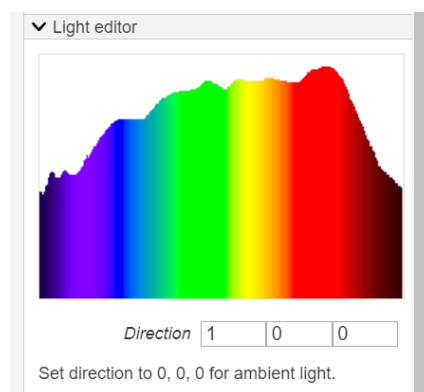
4.3 Osvetlitev

Osvetlitev smo prilagodili tako, da lahko uporabnik določi smer in spekter svetlobe. Smer svetlobe je določena z



Slika 6: Uporabniški vmesnik za določanje materialov in prenosnih funkcij.

vektorjem. Če je dolžina vektorja enaka 0, se smatra, da luč sveti iz vseh smeri. Spekter luči je shranjen v teksturi uporabnik pa ga lahko določi z uporabniškim vmesnikom, ki je prikazan na sliki 7.



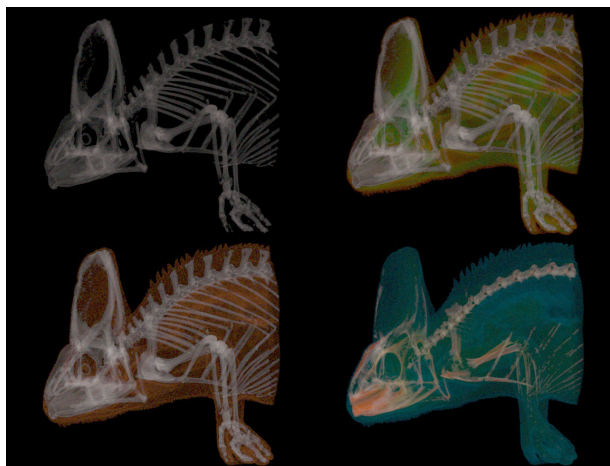
Slika 7: Uporabniški vmesnik za nastavljanje spektra in smeri svetlobe.

5 Rezultati

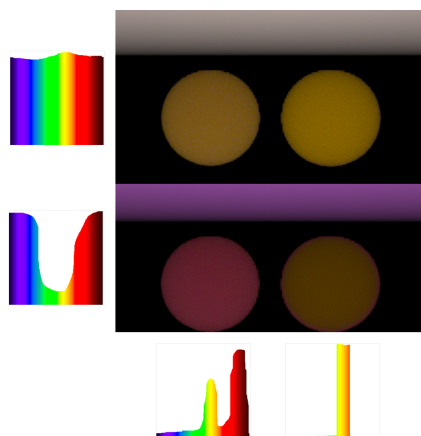
Spektralni upodabljalnik smo preizkusili na sintetičnem testnem volumnu, ki vsebuje dve sferi z različnimi surovimi vrednostmi, ter volumnu kameleona iz repozitorija *Open Scientific Visualization Datasets* [3]. Primer slednjega je prikazan na sliki 8.

Zanimiv pojav, ki ga lahko simuliramo s spektralnim upodabljanjem, je metamerizem [1]. Pri metamerizmu sta dva različna materiala pri enem spektru svetlobe videti enako, pri drugem pa različno. Primer metamerizma je prikazan na sliki 9.

Še en pojav, ki ga lahko simuliramo s spektralnim upodabljanjem, je različno sipanje svetlobe pri različnih



Slika 8: Upodobitev volumna *Chamaeleo calytratus* *Digital Morphology*, 2003 z različnimi prenosnimi funkcijami.



Slika 9: Primer metamerizma. Levo sta prikazana spektra luči spodaj pa sta prikazana spektra barve materialov.

valovnih dolžinah. Na sliki 1 je prikazan testni volumen, ki je sestavljen iz kocke in krogle. Kocka je iz materiala, ki se obnaša podobno kot zračna atmosfera – sipa več modre svetlobe, prepušča pa več rdeče. Posledično je kocka videti modra, ko jo osvetlimo iz strani, in bolj rdeča, ko jo osvetlimo od zadaj.

Spektralni upodabljalnik zahteva več računske zmogljivosti in spomina kot nespektralni. Shranjevanje slike v obliki odsekoma konstantne funkcije zahteva shranjevanje n koeficientov za vsak piksel (kjer je n število odsekov spektra), shranjevanje spektra v obliki RGB pa le tri vrednosti. Prav tako je konvergenca algoritma počasnejša, saj je za prikaz na zaslonu prisoten dodaten integral po valovnih dolžinah. Ocenili smo, da spektralni upodabljalnik potrebuje približno 2 do 3 krat več časa za doseganje enake kvalitete slike kot z uporabo nespektralnega upodabljalnika. Natančen faktor upočasnitve je odvisen od strojne opreme.

Izvorna koda implementacije je dostopna na [5].

6 Zaključek

Spektralno upodabljanje volumetričnih podatkov ponuja natančnejše in bolj realistične vizualizacije, saj upošteva celoten spekter svetlobe namesto zgolj treh barvnih kanalov. Implementacija v ogrodju VPT je pokazala, da lahko ta pristop učinkovito simulira kompleksne optične pojave, kot so metamerizem in neenakomerno sipanje svetlobe. Kljub temu pa spektralno upodabljanje zahteva večje računske zmogljivosti in več pomnilnika, kar podaljša čas upodabljanja. V prihodnosti bi lahko dodali še podporo za učinke, kot je fluorescenca [11].

Zahvala

Članek je nastal v okviru temeljnega raziskovalnega projekta J2-50221 *Vizualizacija celic z združevanjem mikroskopskih podatkov in proceduralno ustvarjenih subceličnih struktur*, ki ga financira Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost RS – iz državnega proračuna.

Literatura

- [1] S. Bergner, T. Moller, M. Tory, and M.S. Drew. A practical approach to spectral volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 11(2):207–216, 2005.
- [2] Steven Bergner, Torsten Moller, Mark S. Drew, and Graham D. Finlayson. Interactive spectral volume rendering. In *IEEE Visualization, 2002*, pages 101–108, 2002.
- [3] Open Scientific Visualization Datasets. <https://klacansky.com/open-scivis-datasets/>.
- [4] IEC. IEC 61966-2-1:1999 Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB, 1999.
- [5] Jernej Jezeršek, Matija Marolt, Žiga Lesar, and Ciril Bohak. VPT-spectral-volume-rendering. <https://github.com/UL-FRI-LGM/VPT-spectral-volume-rendering>.
- [6] Peter Kutz, Ralf Habel, Yining Karl Li, and Jan Novák. Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Trans. Graph.*, 36(4), jul 2017.
- [7] Žiga Lesar, Ciril Bohak, and Matija Marolt. Real-time interactive platform-agnostic volumetric path tracing in webgl 2.0. In *Proceedings of the 23rd International ACM Conference on 3D Web Technology*, New York, NY, USA, 2018. Association for Computing Machinery.
- [8] Herke Jan Noordmans, Hans van der Voort, and Arnold W. M. Smeulders. Spectral volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(3):196–207, 2000.
- [9] Jan Novák, Iliyan Georgiev, Johannes Hanika, Jaroslav Křivánek, and Wojciech Jarosz. Monte carlo methods for physically based volume rendering. In *ACM SIGGRAPH Courses*, August 2018.
- [10] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory To Implementation*. The MIT Press, 2023.
- [11] Wataru Yamamoto, Bisser Raytchev, Toru Tamaki, and Kazufumi Kaneda. Spectral rendering of fluorescence using importance sampling. In *SIGGRAPH Asia 2018 Posters*, 2018.