

NON-REAL-TIME I REAL-TIME METODE RENDEROVANJA: UPOREDNA STUDIJA SLUČAJA RENDEROVANJA KAUSTIKE ZA VODU I STAKLO**NON-REAL-TIME AND REAL-TIME RENDERING METHODS: A COMPARATIVE CASE STUDY OF WATER AND GLASS CAUSTIC RENDERING**Milica Savić, Ivana Vasiljević, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSKA GRAFIKA**

Kratka sadržaj – Tema ovog istraživanja jeste kaustika i algoritmi za renderovanje koji su se pokazali kao najefikasniji kada je ovaj efekat u pitanju. Istraživanje obuhvata testiranje spomenutih algoritama upotrebom različitih renderer-a u programima „Blender“ i „3Ds Max“ kao primer fotorealističnog renderovanja kaustike u non-real-time računarskoj grafici. Radi testiranja real-time renderovanja kaustike, korišćen je „Unreal Engine“.

Ključne reči: Kaustika, BRDF, algoritmi, testiranje, 3Ds Max, Blender, Unreal Engine

Abstract – The aim of this research is to explore caustics and rendering algorithms that are shown to be the most efficient for this effect. The research also includes testing these algorithms using different renderers in the software “Blender” and “3Ds Max” for photorealistic caustics rendering in non-real-time computer graphics. “Unreal Engine” was used for testing real-time caustics rendering.

Keywords: Caustics, BRDF, algorithms, tests, 3Ds Max, Blender, Unreal Engine

1. UVOD

Svetlost, njena priroda i efekti koje može da stvori u različitim okolnostima oduvek su bili fascinantni i primamljivi za istraživanje. Jedan takav intrigantan proizvod igre svetlosti je **kaustika** (eng. *caustics*), fenomen koji nastaje pri interakciji svetlosti sa nekom zakrivljenom površinom ili objektom, najčešće sačinjenih od transparentnih materijala poput stakla ili vode. Pri takvoj interakciji svetlost se reflektuje ili refraktuje i tada nastaju upečatljivi svetlosni šabloni, odnosno kaustika.

Ovaj dokument pruža objašnjenje algoritama za renderovanje, koji su se kroz teoriju i praktičan rad pokazali kao najefikasniji za realistično renderovanje scena sa kompleksnim osvetljenjem i efektima kao što je kaustika.

Na samom kraju dat je uvid u rezultate testiranja renderovanja kaustike upotrebom *non-real-time* i *real-time* metoda.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Ivana Vasiljević, docent.

2. TEORIJSKE OSNOVE NASTANKA KAUSTIKE I NAČINA NJENOG PRIKAZA U RAČUNARSKOJ GRAFICI

U ovom poglavlju opisani su glavni teorijski koncepti koji su važni za razumevanje načina nastanka kaustike, kao i za njeno kreiranje u ovom projektu. Kaustika je fizička pojava i njeno renderovanje, koje je neophodno za postizanje fotorealizma na renderima zasniva se na njenim fizičkim karakteristikama. Ovakav pristup renderovanju u 3D računarskoj grafici naziva se renderovanje na osnovu fizičkih karakteristika (eng. *physically-based rendering*) i ono podrazumeva kreiranje fotorealističnih kompjuterski generisanih slika tako što se simulira ponašanje svetla po modelu iz stvarnog sveta.

2.1. Interakcija svetla i površi

Način na koji svetlo interaguje sa površinom na koju nailazi odlučuje kako će posmatrana površina izgledati [1]. Dve vrste ove interakcija su apsorpcija i raspršivanje svetlosti. U nastavku teksta ukratko su opisane funkcije i fizički zakoni koji su ključni za renderovanje kaustike.

2.2. Bidirekzione (dvosmerne) funkcije

Kao posebno svojstvo nekog materijala izdvaja se reflektivnost, kojom se pokazuje njegova efikasnost u reflektovanju energije radijansa [2]. Bilo je neophodno smisliti način njegovog jasnog prikazivanja na sceni. Tako je nastala **Bidirectional Reflectance Distribution Function**, skraćeno **BRDF** – jedan od centralnih pojmova računarske grafike i to je 4-dimenziona funkcija pomoću koje se mogu u potpunosti opisati reflektivna svojstva bilo koje tačke na sceni. Kažemo da je bidirekciona, odnosno dvosmerna, jer se za reprezentaciju ovih svojstava posmatraju dva pravca: upadni (eng. *Illumination Direction*) - iz kog svetlost dolazi i reflektovani (eng. *Viewing Direction*) - pravac odbijanja upadne svetlosti. Formula BRDF funkcije glasi:

$$f(\theta_i, \Phi_i, \theta_r, \Phi_r) = \frac{L(\theta_r, \Phi_r)}{E(\theta_i, \Phi_i)} \quad (1)$$

Gde su: θ_i, Φ_i - uglovi upadnog pravca; θ_r, Φ_r - uglovi reflektovanog pravca; L- radijansa površine, E –iradijansa izvora.

Pored BRDF-a, postoji i **BTDF** (eng. **Bidirectional Transmittance Distribution Function**). Ova funkcija je slična kao i BRDF, ali se primenjuje na suprotnu stranu površi [3]. Kada se BRDF i BTDF zajedno posmatraju, onda oni čine **BSDF** (eng. **Bidirectional Scattering Distribution Function**), matematičku operaciju za izračunavanje raspršivanja svetlosti.

2.3. Refleksija i refrakcija

Refleksija i refrakcija su dva prirodna fenomena koji su ključni za kreiranje fotorealističnih scena, kao i kaustike, s obzirom da nju stvaraju površine prozirnih materijala poput vode i stakla, koje ujedno i odbijaju i prelamaju svetlo. Takođe, formule ovih zakona predstavljaju srž *Ray tracing* algoritma. **Zakon odbijanja (refleksije) glasi:** svetlosni zraci se na graničnoj površini između dve optički različite sredine reflektuju pod istim uglom pod kojim na nju i padaju. Pri tome upadni zrak, odbojni zrak i normala na površinu leže u istoj ravni [4]:

$$\alpha = \alpha' \quad (2)$$

Zakon prelamanja (refrakcije) svetlosti: kada svetlosni zrak prelazi iz sredine u kojoj je brzina svetlosti c_1 i indeks prelamanja n_1 u sredinu u kojoj se svetlost kreće brzinom c_2 i čiji je indeks prelamanja n_2 , onda je prema zakonu prelamanja [4]:

$$\frac{\sin \alpha}{\cos \beta} = \frac{c_1}{c_2} = \frac{n_2}{n_1} = n_{2/1} \quad (3)$$

Ovde je c brzina svetlosti i iznosi približno 300 m/s, dok $n_{2/1}$ predstavlja indeks prelamanja druge sredine u odnosu na prvu. **Indeks prelamanja** (eng. *index of refraction*) ujedno je i ključni pojam kada se radi o refrakciji. Ovo je komponenta koja određuje u kojoj meri će se svetlosni zrak saviti ili prelomiti pri ulasku u materijal od kojeg je sačinjena površina na koju nailazi. Zahvaljujući ovom parametru dešavaju se pojave kao što su disperzija, hromatska aberacija, kaustika itd.

2.4. Path Tracing

Ray tracing (praćenje zraka) predstavlja *non-real-time* metodu za renderovanje u 3D računarskoj grafici, kojom se praćenjem pojedinačnih svetlosnih zraka simulira realistično osvetljenje scene i objekata na njoj, što rezultuje renderovanje fizički tačnih refleksija, refrakcija i senki. Renderovanje ovom metodom vrši se tako što se prati svetlosna putanja koja ide iz pogleda kamere, kroz 2D ravan slike, na 3D scenu i nazad u svetlosni izvor. Postupak se ponavlja rekurzivno dok se ne postigne željeni nivo realizma na renderu i može da potraje dosta dugo. **Path Tracing** se može shvatiti kao ekstenzija *Ray tracing* algoritma, a njen rad se zasniva na upotrebi Monte Karlo metoda za rešavanje jednačine renderovanja [5]. Razlog ovome leži u tome što je kod ove metode neophodno ispratiti putanje i do nekoliko hiljada svetlosnih zraka koji polaze iz svetlosnog izvora istovremeno, a koji se potom nasumično odbijaju i generišu u raznim pravcima pri njihovoj interakciji sa objektima na sceni. Umesto da se prati svaki zasebno, uzima se uzorak svih zraka, koji se zasniva na proceni putanje kojom će se svetlost najverovatnije kretati [6].

2.5. Bidirectional Path Tracing

Bidirectional Path Tracing je varijanta *path tracing*-a, koja radi na osnovu kombinacije **Backwards Path Tracing** i **Forwards Path Tracing (Light Tracing)** pristupa. Prvi pristup funkcioniše tako što se prate svetlosni zraci čije putanje polaze iz kamere i kreću se kroz scenu dok ne naiđu na svetlosni izvor. Ovaj pristup je značajan za pravilno renderovanje indirektnih

svetlosnih efekata kao što su refleksija i refrakcija. Kod drugog pristupa, *Light Tracing*-a, prate se svetlosni zraci koji potiču iz svetlosnih izvora i kreću se po sceni sve dok ne dođe do kontakta sa kamerom.

Sa ovim pristupom omogućava se pravilan prikaz efekata direktnog svetla, kao što su senke. Dakle, kombinacija ova dva pristupa u jedan, odnosno *Bidirectional Path Tracing*, omogućava preciznije i efikasnije prikazivanje efekata globalnog osvetljenja, a ono što je primećeno na rezultatima koje daje jeste manja količina šuma nego sa običnim *path tracing* algoritmom. Pošto ima mnogo veći broj svetlosnih zraka za praćenje, odnosno prati dva različita seta svetlosnih putanja, mana mu je što može da zahteva značajne računске resurse (uzevši u obzir i da se vrše proračuni efekata ovih zraka na svaki piksel na sceni). Zbog toga se ova metoda primenjuje samo tamo gde je potreban izuzetno visok kvalitet i fotorealizam.

2.6. Photon Mapping

Photon Mapping (mapiranje fotona) predstavlja još jednu varijantu *Ray tracing*-a, koja je specijalizovana za dodavanje realističnijeg osvetljenja u 3D scenu, tako što simulira ponašanje fotona, pri njihovoj interakciji sa različitim površinama i materijalima. Ovaj algoritam radi u dva prolaza. Prvi prolaz je **Photon Tracing (praćenje fotona)**, u okviru kojeg se generiše fotonska mapa i sastoji se od tri procesa: **Photon Emission** – emisija diskretnih fotona iz svetlosnih izvora, **Photon Scattering** – proces raspršivanja fotona i na kraju **Photon Storing** – skladištenje fotona u određenu strukturu podataka pre renderovanja sa podacima o njihovoj poziciji, pravcu i intenzitetu, odnosno boji. Drugi prolaz je **Rendering** i u okviru njega se upotrebom *Photon Mapping* algoritma računaju efekti indirektnog osvetljenja (kaustika).

2.7. MIP mape

Kada je u pitanju *real-time* renderovanje, pogotovo scena koje imaju kompleksne efekte globalnog osvetljenja, od velike pomoći za optimizaciju i poboljšanje performansi samog procesa su svakako **MIP mape** (eng. *Mipmaps, MIP maps*). MIP ili *MIP level* predstavlja zapravo verziju teksture sa određenom rezolucijom, dok su *MIP mape* setovi u kojima se smeštaju *mip*-ovi i one sadrže verzije originalne teksture gde svaka ima manju rezoluciju nego njena prethodnica. Svoju primenu *mip*-mape nalaze najčešće u *real-time* 3D aplikacijama igre u slučajevima kada objekti imaju različitu udaljenost od kamere. MIP veće rezolucije biće korišćen za objekte bliže kameri, dok će oni sa manjom biti primenjeni na objekte koji su od iste udaljeni. U ovome se ogledaju benefiti MIP mapa koji se odnose na optimizaciju i poboljšanje performansi, jer se na ovaj način ubrzava proces renderovanja, pošto se smanjuje broj numeričkih operacija koje bi grafički procesor inače radio na teksturi u njenoj originalnoj rezoluciji.

3. TESTIRANJE RENDEROVANJA KAUSTIKE

Testiranje renderovanja kaustike stakla i vode vršeno je upotrebom različitih renderera u programima „Blender”, „3Ds Max” i „Unreal Engine 5”, nakon čega su rezultati upoređeni.

3.1. „Blender“

„Blender“ je besplatan program otvorenog koda, koji koristi tehnologije 3D računarske grafike, a za renderovanje kaustike u okviru njega su korišćeni njegov podrazumevani *renderer* – „Cycles“ i *plugin* – „LuxCore“.

3.1.1. „Cycles“

„Cycles“ je fizički zasnovan *renderer*, koji dolazi uz „Blender“, a čija je glavna odlika upotreba *Path Tracing* algoritma, kao i primena BSDF funkcije u podešavanjima materijala. Kako bi se kaustika renderovala, neophodno je da se na sceni nalaze tri ključna objekta – bar jedan koji će je stvarati, bar jedan koji će je primati, tj. podloga i svetlosni izvor. Za svaki od ovih objekata u okviru podešavanja za renderovanje postoje opcije preko kojih se dodeljuju pomenute uloge. Osim toga, s obzirom da je cilj testiranja renderovanja kaustike stakla i vode, neophodno je da se materijalima odrede indeksi prelamanja, što je za staklo konstanta vrednosti 1.50, a za vodu 1.33.

3.1.2. „LuxCore“

„LuxCore“ je fizički-zasnovan, *plugin renderer*, ali radi na osnovu *bidirectional path tracing* algoritma. Zahvaljujući ovome u stanju je da proizvede slike fotografskog kvaliteta i da prikaže širok spektar svetlosnih fenomena, uključujući i kaustiku. Najbitnije je bilo postaviti iste indekse prelamanja kao i za „Cycles“, dok su podešavanja za samo renderovanje malo kompleksnija, te je potrebno uključiti i adekvatno postaviti parametre opcija kao što su *Light Tracing*, *Clamping* ili *Denoiser* kako bi se dobili što tačniji i čistiji rezultati.

3.2. „3Ds Max“

Zadatak je zamišljen tako da se, pored besplatnog softvera kakav je „Blender“, uradi i poređenje rezultata dobijenih upotrebom nekog profesionalnog softvera i alata za renderovanje koji su njemu na raspolaganju. Takav softver je svakako „3Ds Max“ koju razvija i proizvodi kompanija „Autodesk“. *Renderer*-i koji su odabrani za testiranje u okviru njega su „V-Ray“ i „Corona“

3.2.1. „V-Ray“

„V-Ray“ predstavlja komercijalni *plugin* za renderovanje, koji se najčešće koristi u okviru „3Ds Max“-a. Podešavanja materijala su, po parametrima, slična kao i u „Blender“-u, s tim da treba naglasiti da se za neophodne proračune interakcije svetla i površi primenjuje BRDF funkcija. Pored toga, posebno je zanimljiv *Caustics* odeljak u okviru GI menija u podešavanjima za renderovanje, gde se vidi da „V-Ray“ koristi *Photon Mapping* algoritam za renderovanje.

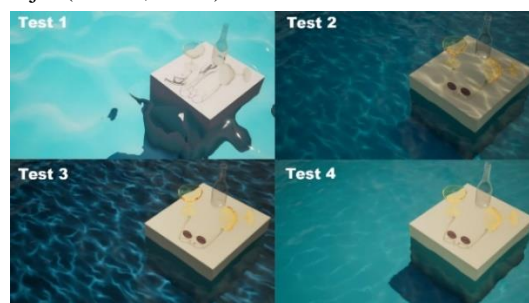
3.2.2. „Corona“

„Corona“ *renderer* takođe predstavlja *plugin* za renderovanje koji je u ovom testiranju korišćen u okviru „3Ds Max“-a. Njegova glavna odlika je svakako činjenica da za proces renderovanja koristi CPU za razliku od svih ostalih korišćenih *renderer*-a kod kojih je za to zadužen GPU. Osim toga, ima mogućnost renderovanja *Path Tracing* algoritmom, kao i neodređenom pristrasnom tehnikom. Kao i kod „V-Ray“-

a i ovde se primenjuje BRDF funkcija, a pošto je utvrđeno da nepristrasna tehnika ne daje dobre rezultate, render sa prikazom kaustike dobijen je korišćenjem *Path Tracing*-a.

3.3. „Unreal Engine“

„Unreal Engine“ predstavlja trenutno najpopularniji softver kada je u pitanju *real-time* računarska grafika, posebno u oblasti 3D video-igara. Zbog inovativnih alata, koje nudi, odabran je za potrebe ovog rada kao primer *real-time* renderovanja kaustike. Međutim zbog nekompatibilnosti grafičke kartice računara na kome je testiranje vršeno sa „Unreal Engine“-om renderovanje scene upotrebom *Path Tracing* algoritma nije bilo moguće. Rešenje problema pronađeno je prvo u testiranju *Water plugin* opcije, koje ovaj softver ima ugrađenog u sebi, a koji nudi automatske opcije za kreiranje vodenih površina, kao i efekta kaustike koji u ovom slučaju svakako izgleda više stilizovano nego realistično (slika 1, test 1). Nakon toga, kao još jedno zamensko rešenje vršeno je testiranje kreiranja lažne kaustike, koja je pomenuta ranije u tekstu. Za ove potrebe, sa interneta je preuzeta slika sa kaustičnim šarama od koje je potom u programu generisan materijal, nakon čega se on primenjuje na svetlosni izvor (slika 1, test 2). Osim toga, primećeno je da u „Unreal Engine“-u postoji vodeni materijal, koji u sebi sadrži parametar za simulaciju kaustike. Zbog toga je još testirana i kombinacija ova dva materijala na vodenu površinu (rezultat slici 1, test 3). Poslednje je u ovom programu urađeno renderovanje površine vode, koja na sebi ima primenjen samo vodeni materijal (slika 1, test 4).



Slika 2. Testiranje *real-time* renderovanja kaustike

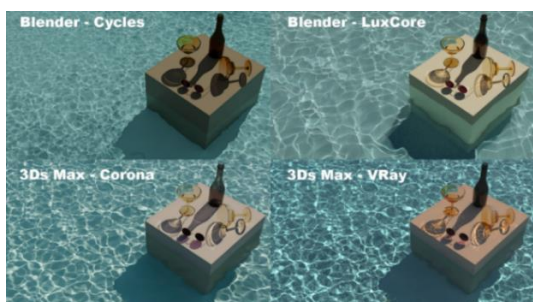
4. ANALIZA DOBIJENIH REZULTATA

U ovom delu rada vršena je analiza rezultata dobijenih *non-real-time* metodama renderovanja. Za početak, na slici 2 prikazani su dobijeni rezultati renderovani pomoću ranije opisanih *renderer*-a.

Ono što je primećeno da uprkos postavljanju parametara materijala na iste vrednosti i što sličnijem podešavanju kamera i svetlosnih izvora nisu dobijeni isti rezultati. Prvi zaključak zašto dolazi do ovoga je prosto činjenica da korišćeni *renderer*-i u svom radu koriste različite algoritme, te samim tim se vrše različiti proračuni i različito se tretira i vidi svetlost.

Druga uočena razlika između „Blender“-a i „3Ds Max“-a jeste ta da „Blender“ u svojim *shader*-ima koristi BSDF funkciju, koja za proračune uzima u obzir refleksiju, refrakciju i transmisiju svetlosti, dok se u „3Ds Max“-u primećuje upotreba BRDF-a, gde je fokus na refleksiji svetlosti. Uprkos razlikama, neophodno je nekako odrediti koji od testiranih *renderer*-a predstavlja najbolji izbor za renderovanje kaustike. Kao najvažniji parametri izdvojeni

su tehnika renderovanja koja se koristi, vreme koje je potrebno za dobijanje rezultata (videti tabelu 1) i njegov finalni izgled.



Slika 2. Rezultati non-real-time renderovanja kaustike

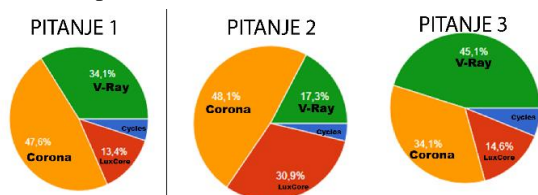
Tabela 1. Tabela prikaz ključnih parametara analize

Renderer (softver)	Tehnika renderovanja	Vreme renderovanja
„Cycles“ („Blender“)	Path Tracing	45 min
„LuxCore“ („Blender“)	Bidirectional Path Tracing	približno 3h
„V-Ray“ („3Ds Max“)	Photon Mapping	preko 5h
„Corona“ („3Ds Max“)	Path Tracing	2h 30 min

Pored toga, radi izvođenja objektivnog zaključka nad 82 ispitanika je sprovedena anketa u kojoj je bilo neophodno dati odgovor na sledeća tri pitanja:

1. Koji od ponuđenih rezultata Vam na prvi pogled izgleda najrealističnije (najbolje)?
2. Fokusirajte se sada samo na staklene objekte i njihove senke. Koji renderer je u ovom slučaju dao najrealističniji (najbolji) rezultat?
3. Kada je u pitanju voda, na kojoj slici izgleda najrealističnije (najbolje)?

Rezultati su prikazani na slici 3.



Slika 3. Rezultati sprovedene ankete

Dakle, prema mišljenju ispitanika najrealističniji render proizvela je „Corona“, koja se pokazala kao najbolja i kada je u pitanju samo staklo. Kada se posmatra samo voda, najubedljiviji u ovom testiranju bio je „V-Ray“.

Ako se obrati pažnja i na ostatak rezultata ankete, a i kad se uzme u obzir prethodno izlaganje i analiza, dolazi se do zaključka da je za potrebe renderovanja realističnih prikaza, kao i same kaustike najbolje odabrati „3Ds Max“ i neki od *renderer*-a namenjenih ovom softveru. Konačnu odluku o korišćenju svakako treba da donese korisnik u zavisnosti od toga kojem cilju teži.

5. ZAKLJUČAK

U ovom dokumentu dati su osnovni teorijski koncepti nastanka efekta kaustike u računarskoj grafici kao i kratki opisi algoritama na kojima se zasniva fizički zasnovano renderovanje, a koji su implementirani u *renderer*-ima korišćenim za testiranja u praktičnom delu rada. Ovi *renderer*-i, kao i postupak rada u njima su takođe opisani u kratkim crtama, nakon čega je usledilo predstavljanje i analiza dobijenih rezultata. U svrhe izvođenja objektivnog zaključka koji softver se preporučuje kao najbolja opcija za realistično renderovanje kaustike sprovedena je kratka anketa nad 82 ispitanika, kojima su najrealističniji utisak dali renderi koji su proizveli „Corona“ i „V-Ray“ pogoni za renderovanje, oba namenjena za „3Ds Max“.

Uzimajući u razmatranje i vremena neophodna za dobijanje rezultata, koja su prikazana u tabeli 1, u ovom testiranju se kao najbolji *renderer* po svojoj efikasnosti i realističnosti finalnog rendera pokazala „Corona“. Zbog neadekvatnih rezultata, „Unreal Engine“ je izuzet iz ove analize, ali je poslužio kao primer kako se zaobilaznim metodama može kreirati lažna kaustike u *real-time* računarskoj grafici.

6. LITERATURA

- [1] Larsen, S. (2006). Real-time Caustics. <https://www2.imm.dtu.dk/pubdb/edoc/imm4599.pdf> (pristupljeno u septembru 2023.)
- [2] Synopsys. <https://www.synopsys.com/glossary/what-is-reflectance-and-transmittance.html> (pristupljeno u oktobru 2023.)
- [3] F O Bartell, E. L. Dereniak, W. L. Wolfe "The Theory And Measurement Of Bidirectional Reflectance Distribution Function (Brdf) And Bidirectional Transmittance Distribution Function (BTDF)", Proc. SPIE 0257, Radiation Scattering in Optical Systems, (03.03.1981); <https://doi.org/10.1117/12.959611> (pristupljeno u oktobru 2023.)
- [4] Budinski-Petković, Lj (2015). Fizika. Novi Sad: FTN Izdavaštvo
- [5] NVIDIA. <https://blogs.nvidia.com/blog/what-is-path-tracing/> (pristupljeno u oktobru 2023.)
- [6] Digital Trends. <https://www.digitaltrends.com/computing/ray-tracing-vs-path-tracing/> (pristupljeno u oktobru 2023.)

Kratka biografija:



Milica Savić rođena je u Novom Sadu 2000. Master rad na Fakultetu tehničkih nauka iz oblasti Računarska grafika odbranila je 2024. godine.

kontakt: milicassavic000@gmail.com



docent dr Ivana Vasiljević doktorirala je na Fakultetu tehničkih nauka u Novom Sadu u decembru 2021. god. i tako je postala prvi doktor nauka iz naučne oblasti Računarska grafika.

kontakt: ivanav@uns.ac.rs