

# Rejtresing u Poenkareovom sfernom modelu hiperboličkog prostora

---

*Boris Ajdin*

Matematički fakultet  
Univerzitet u Beogradu

## Sadržaj

Zahvalnost.....	4
Apstrakt.....	6
1. Uvod.....	7
2. Hiperbolička geometrija.....	8
2.1 Istorijat i razvoj.....	8
2.2 Poenkareov disk model.....	11
2.3 Poenkareov sferni model.....	13
3. Svetlost.....	15
3.1 Optika.....	15
3.2 Ljudski vizuelni sistem.....	17
4. Svetlost u računarskoj grafici.....	19
4.1 Lokalni modeli osvetljenja.....	19
4.2 Analitički BRDF modeli.....	21
4.3 Globalni model osvetljenja i jednačina renderovanja.....	23
5. Rejtreesing algoritam.....	27
5.1 Rejkasting algoritam.....	27
5.2 Rekurzivni rejtreesing algoritam.....	29
5.3 Tehnike ubrzavanja.....	31
5.3.1 Paralelizacija rejtreesing algoritma.....	32
5.3.2 Hijerarhije graničnih tela.....	34
6. Hiperbolički rejtreesing.....	37
6.1 Postavljeni ciljevi.....	37
6.2 Osnovni elementi hiperboličkog rejtreesinga.....	38
6.3 Osnovne geometrijske primitive.....	39
6.4 Mreža piksela i tipovi projektovanja.....	42
6.4.1 Projektovanje objekata $h$ -prostora na $h$ -ravan.....	43
6.4.2 Mreže piksela u Poenkareovom sfernom modelu.....	44
7. Implementacija.....	48
7.1 Opis biblioteke HyperGeomLib.....	48
7.1.1 Izbor programskog jezika i korišćenih biblioteka.....	48

7.1.2	Dizajn biblioteke i hijerarhija klasa .....	48
7.2	Opisivanje scene .....	52
7.2.1	Tokeni gramatike ulazne datoteke.....	53
7.2.2	Gramatika ulazne datoteke.....	53
8.	Primeri hiperboličkih scena.....	55
8.1	Kreiranje geometrijskih objekata.....	55
8.2	Ilustracija rejtrensing algoritma – refleksija i refrakcija.....	57
8.3	Transformacije scene i pozicije posmatrača .....	59
8.4	Ilustracije osobina hiperboličke geometrije.....	60
8.5	Efikasnost izračunavanja.....	63
9.	Pregled relevantnih radova.....	65
9.1	Prve vizualizacije hiperboličkog prostora.....	65
9.2	Napredne vizualizacije .....	67
10.	Zaključci i dalji rad.....	69
10.1	Zaključci.....	69
10.2	Dalji rad.....	70
Dodatak A.	Primer ulazne datoteke.....	72
Bibliografija	.....	75

## Zahvalnost

Želeo bih da se ovim putem zahvalim svima koji su mi pomogli u veoma dugom procesu kreiranja ove teze, svestan činjenice da svega ovoga bez tih ljudi verovatno ne bi ni bilo.

Mentoru za izradu teze i idejnom tvorcu projekta, profesoru Predragu Janičiću, dugujem veliku zahvalnost na iskazanoj podršci, znanju i entuzijazmu, bez kojih bi izrada projekta bila vremenski neograničena. Profesor Janičić nije odustajao od ideja koje smo imali na početku, insistirajući na tome da se rad privede kraju, „gurajući“ stvari u željenom pravcu i pokazujući gotovo nadljudsko strpljenje u određenim trenucima. No, najviše zahvalnosti profesoru Janičiću dugujem za ilustraciju lepota fantastičnog sveta računarske grafike kojom se sada bavim, a do čega samostalno nikada ne bih uspeo stići.

Profesoru Milanu Tubi i profesoru Srđanu Vukmiroviću, članovima komisije za pregledanje i odbranu teze, zahvalnost dugujem na brzini kojom su pregledali tezu i izneli sugestije i kritike, kao i na plodnoj diskusiji o samoj temi rada. Svojim doprinosom pomogli su da se ovaj tekst približi nekoj svojoj „savršenoj“ verziji, ujedno potvrđujući postignute rezultate projekta.

Mr Aleksandru Samardžiću bih želeo da se zahvalim na brojnim savetima koji su mi olakšali rad na projektu. Bez njegovog ogromnog znanja računarske grafike i saveta kako rešiti brojne praktične probleme koji su se tokom rada pojavljivali ceo proces izrade projekta bio bi daleko komplikovaniji i teži za mene. Profesoru Srđanu Vukmiroviću dugujem zahvalnost na pomoći u vezi sa tehničkim detaljima o Poenkareovom sfernom modelu. Saveti profesora Vukmirovića su centralni deo softvera koji je napisan kao deo projekta, prevashodno o metričkim svojstvima hiperboličke geometrije. Profesorki Nedi Bokan veliku zahvalnost dugujem na podršci ukazanoj kroz rad „Grupe za geometriju, obrazovanje i vizualizacije sa primenama“ Matematičkog fakulteta, koja je iskazala interesovanje za projekat i pomogla njegovo predstavljanje na konferenciji *Workshop on Multimedia Technology for Mathematics and Computer Science Education, Belgrade, November 10-11, 2005* i u *DAAD Spring School "Visualization and Geometry", Berlin, 10-13 April 2006*. Profesorka Bokan je kao dekan Matematičkog fakulteta iskazivala interesovanje i pružala podršku studentskim projektima, te ovaj rad nije u tome izuzetak.

Profesoru Zoranu Lučiću i profesorki Miroslavi Antić sam zahvalan što su uspeli da mi privuku pažnju svojim predavanjima o hiperboličkoj geometriji. Interesantne istorijske anegdote o njenom nastanku, teorija i praktični detalji kojima su ta predavanja bila protkana otvorila su za mene jedan novi svet, koji je, iako veoma stran, veoma lep. Stoga i ne čudi entuzijazam sa kojim sam prihvatio rad na projektu hiperboličkog rejtrejsinga. Profesor Konrad Poltier sa Slobodnog Univerziteta u Berlinu pratio je projekat od njegovog samog početka i izneo niz korisnih sugestija od kojih su neke inkorporirane u finalnoj verziji. Dr. Džefri Viks, jedan od pionira vizualizacija hiperboličkog prostora, pomogao mi je demistifikacijom procesa formiranja slike u virtuelnom hiperboličkom okruženju i svojim entuzijazmom učvrstio veru da uloženi rad na projektu nije uzaludan.

Ogromnu zahvalnost dugujem mojim kolegincama i prijateljicama, Jeleni Novičić i Radmili Stamenčić, koje su učestvovala u radu na projektu u njegovim inicijalnim fazama. Rad sa Jelenom i Radmilom je uvek bio prijatan, iako ne uvek i lak, a rezultati su govorili sami za sebe. U trenucima

euforije, kada bi mi se činilo da je „sve lako uraditi“, Jelena i Radmila bi me spuštale na zemlju i pokazivale da je neophodan veliki trud da bi se došlo do zadovoljavajućih rezultata.

Koni Ligl mi je bila velika podrška, najviše u trenucima kada je izrada projekta nailazila na, naizgled, nepremostive prepreke i kada sam ostajao bez motivacije da istrajem u radu. Bez njenog entuzijazma, želje da bude od pomoći i tolerisanja svih mojih hirova ovaj rad bi još uvek bio u izradi.

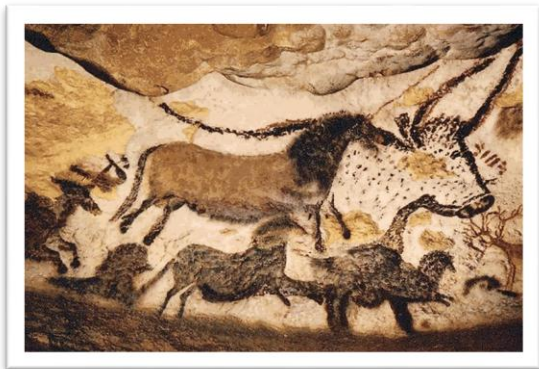
Na posletku, zahvalio bih svojoj porodici na svemu što su učinili za mene tokom svih ovih godina. Bez njihovih reči podrške, otrežnjenja, ponekad i grdnje sigurno ne bih bio u prilici da pišem ove redove, jer bih mnogo ranije poklekao pred malim i velikim preprekama na koje sam nailazio. Nadam se da se vaša vera u mene na kraju isplatila.

## Apstrakt

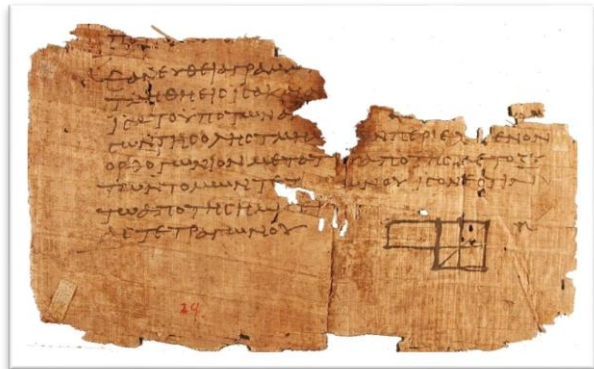
Intenzivan razvoj računarstva od polovine XX veka doprineo je razvoju uređaja i algoritama koji za osnovni cilj imaju grafički prikaz i vizualizaciju sveta koji nas okružuje. Jedan od najstarijih i najpoznatijih algoritama računarske grafike je rejtrensing algoritam koji rešava problem globalnog osvetljenja i omogućava kreiranje foto-realističnih prikaza trodimenzionih scena. Koristeći parametre scene poput geometrijskog opisa objekata prisutnih u sceni, parametre materijala i osvetljenja, rejtrensing algoritam izračunava koji su objekti vidljivi za posmatrača u sceni, kao i višestruke interakcije svetlosti sa datim objektima, simulirajući komplikovane fizičke reakcije iz stvarnog sveta. S druge strane, hiperbolički prostor je intrigantna geometrijska struktura koja se često kosi sa našom intuicijom, po prirodi euklidskom. U nameri da kreiramo alat za vizualizaciju hiperboličkog prostora i ispitivanje njegovih osobina, razvili smo verziju rejtrensing algoritma u hiperboličkom prostoru koja nije do sada postojala. U daljem tekstu biće prikazani razvijeni algoritam, određeni problemi sa kojima smo se susreli, predložena rešenja i dosadašnji rezultati. Ova teza je nastavak rada o hiperboličkom rejtrensing algoritmu koji je nagrađen godišnjom nagradom Univerziteta u Beogradu 2006. godine kao najbolji studentski naučni rad iz oblasti tehničkih nauka (Ajdin et al., 2006).

## 1. Uvod

Od praistorijskih vremena ljudska vrsta izražava potrebu za vizualizacijom sveta koji nas okružuje. Arheološka otkrića u pećini Chauvet (slika 1) u Francuskoj uključuju i pećinske crteže stare čak 32.000 godina – smatra se da su to i najstariji sačuvani crteži ikada napravljeni. Vizualizacija ima poseban značaj u geometriji, nauci koja „izučava prostorne odnose“. Jedan od najstarijih sačuvanih geometrijskih dijagrama potiče od *Euklida* iz Aleksandrije<sup>1</sup>. Nekada samo jedna od ilustracija u delu *Elementi* (grčki: Στοιχεῖα), jedne od najznačajnijih knjiga u istoriji matematike, ovaj dijagram predstavlja geometrijsku interpretaciju poznatog algebarskog identiteta (slika 2).



Slika 1: prizor iz pećine Chauvet.



Slika 2: dijagram iz II knjige Euklidovog dela *Elementi*.

Sve do pojave računara kao alata za vizualizaciju mogućnosti ilustrovanja geometrije pomoću crteža i dijagrama bile su limitirane. Poseban značaj računarska vizualizacija ima upravo u modernom razvoju neeuklidskih geometrija. No, alati za prikaz trodimenzionalnih neeuklidskih prostora su i danas retki, te većina hiperboličkih vizualizacija pripada klasi dvodimenzionalnih vizualizacija.

U ovom radu biće prezentovan *rejtrensing* algoritam (eng. Ray Tracing), odnosno njegova modifikovana verzija prilagođena hiperboličkom prostoru. Pre toga, u kratkim crtama biće iznete i osnovne osobine kako hiperboličkog prostora, tako i fizičkih osobina svetlosti, neophodne za razumevanje rejtrensing algoritma. Sva neophodna izračunavanja izvedena su vodeći se zakonima koji važe u hiperboličkoj geometriji, bez geometrijskih aproksimacija. Kao model prostora u kojem je implementiran algoritam izabran je *Poenkareov<sup>2</sup> sferni model* hiperboličkog prostora. Iako Poenkareov model, gledano iz ugla računarske grafike i rejtrensing algoritma, nije najpodesniji izbor, takav izbor je načinjen prevashodno imajući u vidu da je model konforman i ograničen. Drugim rečima, Poenkareov sferni model intuitivno je najjednostavniji za razumevanje. Budući da je cilj ovog projekta pojednostavljivanje razumevanja hiperboličke geometrije korisnika koji sa njom nisu direktno upoznati, prednost je data intuitivnom razumevanju modela u odnosu na jednostavnost i efikasnost programiranja u modelu.

<sup>1</sup> Euklid, grčki matematičar; oko 365 p.n.e. – 275 p.n.e.

<sup>2</sup> Žil Anri Poenkare, francuski matematičar; 1854 – 1912.

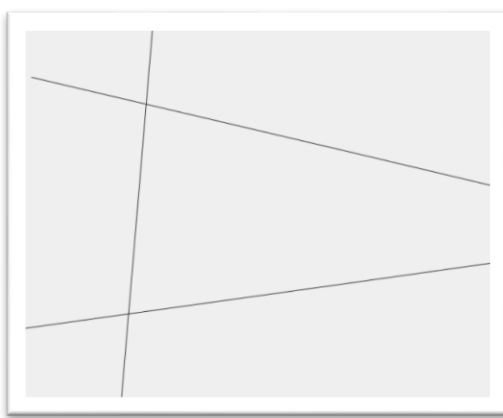
## 2. Hiperbolička geometrija

U ovoj glavi biće ukratko prezentovani istorijat razvoja hiperboličke geometrije i neke od njenih osobina, kao i Poenkareov sferni model hiperboličke geometrije (Lučić, 1997), (Coxeter, 1965).

### 2.1 Istorijat i razvoj

Geometrija se pojavila uporedo sa pojavom prvih civilizacija, kao rezultat potrebe da se opiše i bolje razume svet koji nas okružuje. Inicijalno se geometrija razvijala kao sasvim praktična disciplina: čovečanstvu su bila potrebna precizna merenja dužina, površina i zapremina; osećala se potreba za razumevanjem odnosa (npr. u trgovini) i to je najčešće izražavano uz pomoć geometrije; na posletku, geometrija je najprirodniji način da se na logički valjan način opišu objekti prisutni u svakodnevnom životu. Do kvalitativne promene viđenja značaja geometrije dolazi sa objavljivanjem Euklidove knjige *Elementi* u III veku p.n.e. Iako je većina tvrđenja koja Euklid iznosi bila poznata i njegovim prethodnicima, ono što *Elemente* čini jednim od najvažnijih dela u celokupnoj istoriji matematike je Euklidov pristup uvođenju geometrije – po prvi put je pokušano da se neka matematička teorija uvede sistematski pomoću aksioma, tvrđenja koja se ne dokazuju i iz kojih se izvode sva ostala. Drugim rečima, tek je Euklid uvideo mogućnost da se geometrija izdigne iznad jednostavne praktične discipline u redove apstraktnog konstrukta koji danas nazivamo matematičkom teorijom. Prateći Euklidovu viziju, više od dva milenijuma razvoja klasične geometrije iznedrilo je ogroman broj izvanrednih naučnika koji su dali svoj doprinos toj nauci. Taj proces je u mnogome zaokružen objavljivanjem dela *Osnovi geometrije* iz 1899. godine nemačkog matematičara *David Hilberta*<sup>1</sup>.

Neeuklidske geometrije su, uslovno rečeno, stare koliko i sama euklidska geometrija. Već se u *Elementima* nazire ideja da se euklidska geometrija mora aksiomatski definisati, tj. da ona ne važi „po prirodi stvari“. V *Euklidov postulat* (slika 3), koji Euklid ne dokazuje, tvrdi da će se dve prave jedne ravni, koje seku treću pravu te iste ravni, seći sa one strane treće prave sa koje je zbir uglova preseka te dve prave sa trećom manji od opruženog ugla.



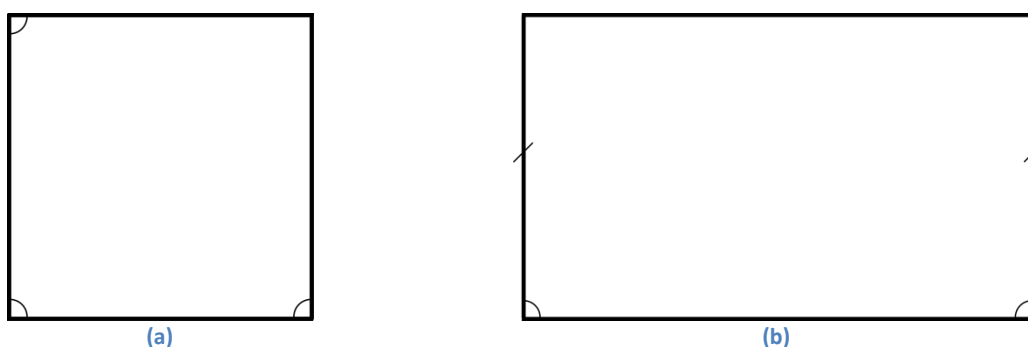
Slika 3: V Euklidov postulat.

<sup>1</sup> David Hilbert, nemački matematičar; 1862. – 1943.



Euklidovi nasljednici postali su i njegovi prvi kritičari, prevashodno kritikujući postojanje V postulata. Kao razlog kritike najčešće se navodi činjenica da su ostale aksiome i postulati na osnovu kojih Euklid gradi geometriju daleko jednostavniji, te V postulat odudara kao daleko kompleksnije tvrđenje. Međutim, stvarni razlog se krio u tadašnjoj geometrijskoj dogmi da je jedina moguća geometrija upravo Euklidova i da V postulat mora biti moguće izvesti iz ostalih aksioma i postulata. Tvrditi drugačije je bilo nezamislivo i ta nekonzistentnost primetna u *Elementima* morala je biti otklonjena. Koristeći *Elemente* kao *spiritus movens*, generacije matematičara su u naredna dva milenijuma pokušavale da dokažu V postulat ili neki od njegovih ekvivalenata, polazeći od preostalih Euklidovih aksioma i postulata.

Abu Ali al-Hasan ibn al-Hasan ibn al-Hajtam<sup>1</sup> može se svrstati u red najplodnijih naučnika i istraživača svih vremena. Ovaj vrsni matematičar, fizičar, inženjer, astronom, filozof, oftamolog, psiholog – generalno rečeno multi-disciplinarni stvaralac, pokušao je da dokaže V Euklidov postulat pretpostavljajući postojanje neeuklidskih geometrija i dovodeći takve pretpostavke do kontradikcije. Njegov pokušaj zasnivao se na uvođenju novih postulata, sličnih aksiomi Plejfera<sup>2</sup>, koja je ekvivalentna V Euklidovom postulatu, te je samim tim i sam al-Hajtamov dokaz pogrešan. Ipak, u pokušajima da dokaže V postulat svođenjem na kontradikciju al-Hajtam je izveo neke od prvih teorema hiperboličke i eliptičke geometrije. Svoje dokaze je izvodio razmatrajući takozvani *Ibn al-Hajtamov četvorougao*, koji je u zapadnom svetu popularisao Lambert<sup>3</sup> te se taj četvorougao naziva i *Ibn al-Hajtam-Lambertovim četvorouglom* (slika 4a). Ibn al-Hajtam-Lambertov četvorougao sadrži tri prava unutrašnja ugla, dok je četvrti ugao prav, oštar ili tup, u zavisnosti od toga da li razmatramo taj četvorougao u euklidskoj, hiperboličkoj ili eliptičkoj geometriji.



Slika 4: (a) ilustracija Ibn al-Hajtam-Lambertovog četvorougla; (b) ilustracija Khajam-Sakerijevog četvorougla.

Omar Khajam<sup>4</sup>, matematičar, astronom, filozof i pesnik, dokazivao je istinitost V Euklidovog postulata polazeći od ekvivalentnog tvrđenja koje glasi:

*Dve konvergentne prave se seku i nemoguće je da dve konvergentne prave divergiraju u pravcu u kojem konvergiraju.*

<sup>1</sup> Abu Ali al-Hasan ibn al-Hasan ibn al-Hajtam, arapski/persijski naučnik; 965 – 1039.

<sup>2</sup> Džon Plejfer, škotski matematičar; 1748 – 1819.

<sup>3</sup> Johan Hajnrih Lambert, švajcarski matematičar; 1728 – 1777.

<sup>4</sup> Omar Khajam, persijski naučnik; 1048 – 1123.

Ovo tvrđenje, pomalo zbunjujuće, zapravo kazuje da se dve prave ili seku ili su paralelne. Jasno je da je dokazati istinitost nekog tvrđenja polazeći od njegovog ekvivalenta jednostavno, te je i Khajam „dokazao“ da je euklidska geometrija jedina moguća. No, značaj Khajamovog pristupa ogleđa se u definisanju *Khajamovog četvorougla*, poznatijeg u zapadnoj literaturi i kao *Sakerijev<sup>1</sup> četvorougao*, odnosno *Khajam-Sakerijev četvorougao* (slika 4b).

Kod Khajam-Sakerijevog četvorougla dve naspramne stranice su podudarne i upravne na trećoj, pri čemu su sve stranice proizvoljne dužine. Definisan na ovaj način četvorougao ima dva prava unutrašnja ugla, dok su preostala dva podudarni uglovi. Ako se pretpostavi da su oni takođe pravi uglovi, onda se iz te pretpostavke može dokazati V Euklidov postulat. Omar Khajam koristi tu pretpostavku i svoju novouvedenu aksiomu da pravilno dokaže validnost V Euklidovog postulata – naravno, polazeći od pogrešne pretpostavke.

Sakeri je nastavio putem Omara Khajama, pokušavajući da dokaže da je zbir unutrašnjih uglova trougla jednak opruženom uglu, što je ekvivalent V Euklidovom postulatu. Pretpostavljajući suprotno, Sakeri lako dokazuje da zbir uglova ne može biti veći od opruženog ugla. Ukoliko bi to bilo moguće, tada bi prave u prostoru bile konačne, što je u kontradikciji sa II Euklidovim postulatom<sup>2</sup>. Međutim, dokazati da zbir uglova ne može biti manji od opruženog ugla pokazalo se kao daleko veći problem. Sakeri je uporno dokazivao nova tvrđenja tražeći u njima kontradikciju, ali sve što je iz njegovog rada proizašlo su zapravo brojne teoreme hiperboličke geometrije. Svoj rad na ovoj temi Sakeri zaključuje rečima: „hipoteza oštrog ugla je apsolutno netačna, jer je u suprotnosti sa prirodom pravih linija“.

Konačan rasplet ovog matematičkog trilera dat je tek polovinom XIX veka u radovima *Boljaja<sup>3</sup>*, *Gausa<sup>4</sup>* i *Lobačevskog<sup>5</sup>*. Pokazano je da je V euklidov postulat nezavisno tvrđenje koje ne može biti izvedeno iz ostalih aksioma i postulata, već im se mora pridružiti kao tvrđenje koje se ne dokazuje. Po prvi put se pojavila ideja da je moguće postojanje paralelnih geometrija, sa mnogim sličnostima i razlikama – da geometrijski „očigledno“ ne znači i jedino moguće. Pretpostavivši da je moguće razmatrati geometriju u kojoj ne važi V Euklidov postulat, Boljaj, Gaus i Lobačevski stvorili su takozvanu *hiperboličku geometriju*, predivan teorijski konstrukt, problematičan kao suprotnost milenijumskim dogmama i izuzetno neintuitivan. U čast njenih izumitelja ta geometrija se danas još naziva i geometrijom Lobačevskog i geometrijom Boljaj-Gaus-Lobačevskog.

Jedno od osnovnih pitanja vezanih za prirodu raznih geometrija je i pitanje njihovog odnosa. Jasno je da se euklidska geometrija razlikuje netrivialno od neeuklidskih geometrija, ali na prvi pogled nije moguće odmah reći o kojim se razlikama radi. Ispostavlja se da se, na aksiomatskom nivou, euklidska i hiperbolička geometrija razlikuju jedino u aksiomi paralelnosti – sve ostale aksiome su zajedničke za ove dve geometrije. Stoga se u izučavanju klasične geometrije obično najpre uvodi takozvana *apsolutna geometrija*, tj. geometrija bez aksiome paralelnosti (aksiome se obično dele u 5 grupa: aksiome incidencije, rasporeda, neprekidnosti, podudarnosti i paralelnosti). Nakon što se izvedu sva geometrijska tvrđenja apsolutne geometrije, koja važe kako u euklidskoj tako i u hiperboličkoj

---

<sup>1</sup> Dovani Điolamo Sakeri, italijanski matematičar; 1667 – 1733.

<sup>2</sup> Neka se pretpostavi...i da ograničena prava može biti produžena u svom pravcu neprekidno.

<sup>3</sup> Janoš Boljaj, mađarski matematičar; 1802 – 1860.

<sup>4</sup> Johan Karl Fridrih Gaus, nemački naučnik; 1777 – 1855.

<sup>5</sup> Nikolaj Ivanovič Lobačevski, ruski matematičar; 1792 – 1856.

geometriji, uvode se zasebne aksiome paralelnosti i pristupa se izvođenju tvrđenja specifičnih za datu geometriju. Valja spomenuti da se navedeni pristup ne može direktno primeniti na eliptičku geometriju s obzirom na to da u toj geometriji ne važe sve aksiome apsolutne geometrije.

Važno pitanje je i pitanje neprotivrečnosti navedenih geometrija, tj. pitanje valjanosti sistema aksioma kojima se one definišu. Odgovor na ovo pitanje doneo je Eugenio Beltrami<sup>1</sup> 1868. godine definišući model hiperboličke geometrije unutar euklidske geometrije. Time je pokazano da eventualna nekonzistentnost hiperboličke geometrije nužno dovodi i do nekonzistentnosti euklidske geometrije. Nešto ranije sam Lobačevski povezo je hiperboličku i sfernu geometriju, a time i euklidsku. Ova dva rezultata su dokaz *ekvivalentnosti* euklidske i hiperboličke geometrije – ili su obe konzistentne ili su obe nekonzistentne. Eugenio Beltrami je zaslužan i za definisanje najpoznatijih modela hiperboličke ravni, *Klajnovog<sup>2</sup> disk modela*, *Poenkareovog disk modela* i *Poenkareovog polu-ravanskog modela*. Ovi modeli obično nose imena matematičara koji su definisali validnu *metriku* u tim modelima.

Dati modeli mogu se prirodno proširiti na treću dimenziju, te postoje *Klajnov sferni model*, *Poenkareov sferni model* i *Poenkareov polu-prostorni model*. Sledi kraći prikaz Poenkareovog sfernog modela. Prethodno će biti definisan Poenkareov disk model, s obzirom na to da je u dve dimenzije jednostavnije prikazati osnovne koncepte na kojima se ti modeli zasnivaju.

## 2.2 Poenkareov disk model

Neka je dat krug u ravni i neka je njegov centar ujedno i centar Dekartovog<sup>3</sup> pravougaonog koordinatnog sistema u toj ravni (određenosti radi neka je poluprečnik kruga 1). Taj krug zvaćemo *apsolutom*, unutrašnjost kruga zvaćemo *h-ravan*, a sve tačke *h-ravni* zvaćemo *h-tačke*. Ako je neki krug (prava) upravan na apsolutu segment tog kruga (prave) unutar apsolute zvaćemo *h-pravom* (prava upravna na apsolutu sadrži središte apsolute). Segment kruga (duž prave) upravnog na apsolutu, čija su temena *h-tačke* date *h-ravni* zvaćemo *h-duž*, a segment kruga (duž prave) upravnog na apsolutu čija su temena, redom, *h-tačka* date *h-ravni* i element apsolute zvaćemo *h-polupravom*. Prvo od tih dvaju temena zvaćemo *temenom*, a drugo *krajem* date *h-poluprave*. Slično kao u euklidskoj geometriji mogu se definisati i *h-ugao*, *h-trougao*, *h-poligonska linija*, *h-poligon* i *h-poligonska površ*.

Razmotrimo inverziju u odnosu na krug upravan na apsolutu ili refleksiju u odnosu na pravu normalnu na apsolutu. Tom inverzijom, odnosno refleksijom, *h-ravan* se preslikava sama na sebe. To je motivacija za definisanje *h-refleksije* kao restrikcije te inverzije, odnosno refleksije, na *h-ravan*. Ova definicija omogućava definisanje i *simetrale h-duži* kao *h-prave* ortogonalne na datu *h-duž* kojom je definisana *h-refleksija* koja preslikava jedno teme date *h-duži* u drugo.

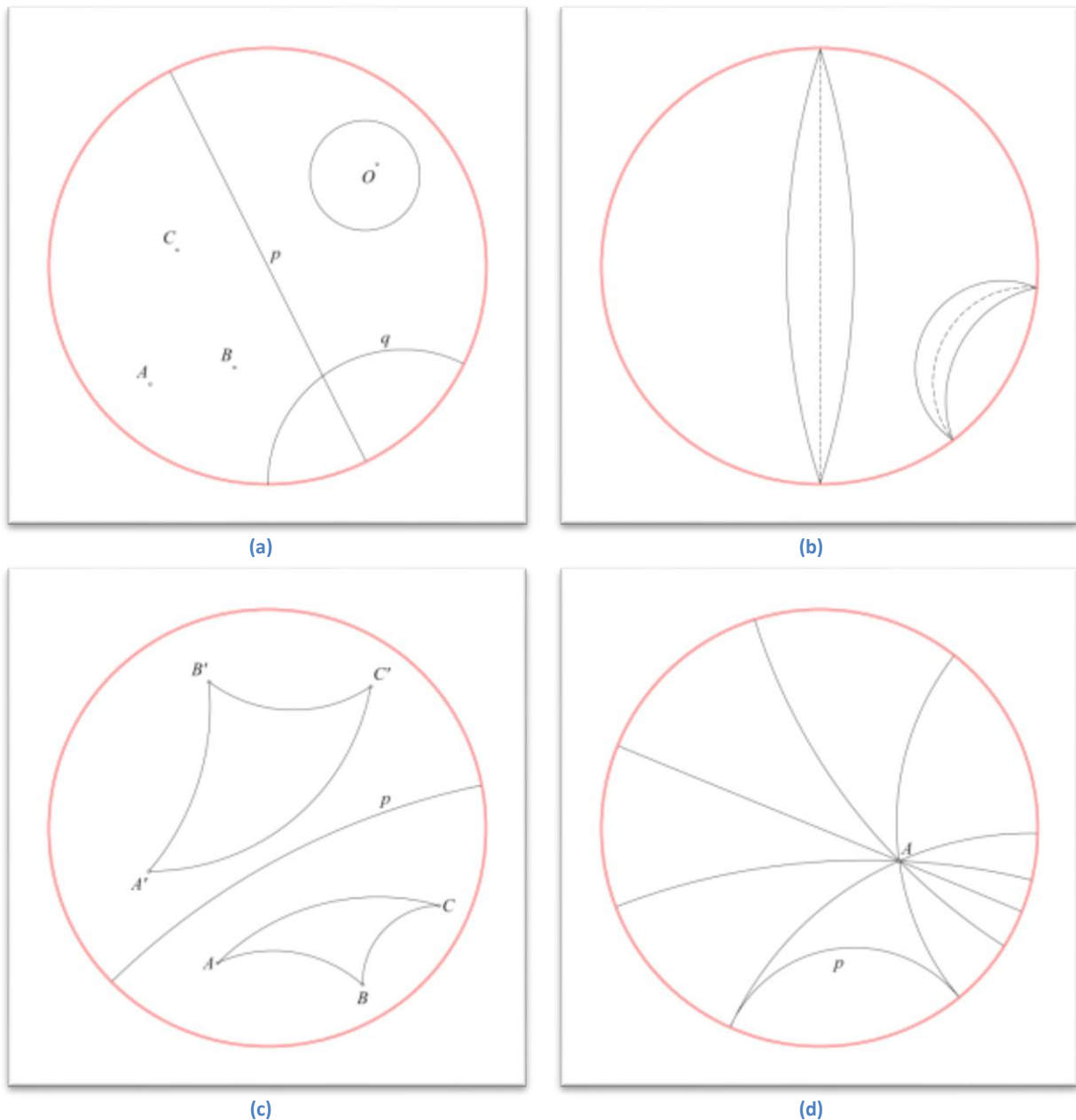
Definisanje *h-refleksije* omogućava definisanje i podudarnosti u hiperboličkom smislu: neka su dati parovi *h-tačaka*  $(A, B)$  i  $(C, D)$ ; reći ćemo da su ti parovi *h-podudarni* ako postoji niz *h-refleksija* koji

<sup>1</sup> Eugenio Beltrami, italijanski matematičar; 1835 – 1899.

<sup>2</sup> Feliks Kristijan Klajn, nemački matematičar; 1849 – 1925.

<sup>3</sup> Rene Dekart, francuski matematičar; 1596 – 1650.

preslikava par  $(A, B)$  u par  $(C, D)$ . Proizvod tih  $h$ -refleksija zvaćemo  $h$ -izometrijom. Među svim izometrijama izdvajaju se  $h$ -translacija i  $h$ -rotacija:  $h$ -translacija za datu  $h$ -duž je kompozicija dve  $h$ -refleksije, prve u odnosu na  $h$ -pravu ortogonalnu na datu  $h$ -duž u početnoj  $h$ -tački i druge u odnosu na simetralu date  $h$ -duži;  $h$ -rotacija oko date  $h$ -tačke za dati  $h$ -ugao  $\alpha$  je kompozicija dve  $h$ -refleksije definisane dvema  $h$ -pravama koje sadrže datu  $h$ -tačku i zahvataju  $h$ -ugao  $\alpha/2$ , pri čemu je orijentacija  $h$ -ugla koje zahvataju te  $h$ -prave jednaka orijentaciji datog  $h$ -ugla.



Slika 5: ilustracija Poenkareovog disk modela; (a) osnovni elementi modela –  $h$ -tačke,  $h$ -prave i  $h$ -krug, (b)  $h$ -cilindar u modelu, (c)  $h$ -refleksija  $h$ -trougla, (d) ilustracija aksiome Lobačevskog.

U Poenkarevom disk modelu se može prirodno uvesti metrika preko *inverzivnog rastojanja*, što je inspirisano definicijom  $h$ -izometrije kao kompozicije  $h$ -refleksija (koje su predstavljene pomoću inverzije u odnosu na krug, odnosno refleksije u odnosu na pravu). To međutim nije jedini način da

se uvede metrika u Poenkareov model hiperboličke ravni. Definicija rastojanja omogućava definisanje *h-kruga* kao skupa svih *h*-tačaka podjednako udaljenih od date *h*-tačke. U Poenkareovom modelu je *h*-krug i u euklidskom smislu krug, međutim euklidski i hiperbolički centar kruga ne moraju nužno biti jednaki. Pomoću metrike može se definisati i *h-cilindar* kao skup *h*-tačaka podjednako udaljenih od date *h*-prave, koju nazivamo *osom h-cilindra*. U Poenkareovom disk modelu *h*-cilindar je unija dva kružna luka ili kružnog luka i duži.

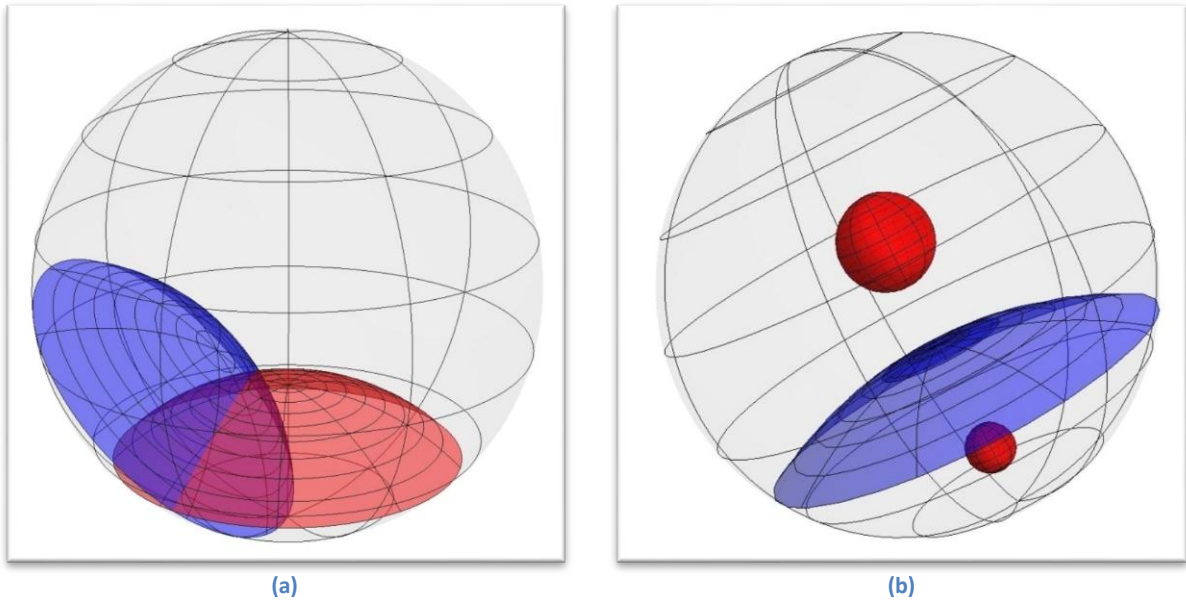
Važna osobina Poenkareovog disk modela je njegova *konformnost*, tj. važi da model čuva uglove. Drugim rečima, mera *h*-ugla između dve *h*-prave jednaka je euklidskoj meri ugla između dva kruga (ili kruga i prave ili dve prave) koji sadrže date *h*-prave. To čini Poenkareov model intuitivnijim u odnosu na neke druge modele gde je mera *h*-ugla komplikovanija. S druge strane, Poenkareov model *nije projektivan model*, tj. u modelu se ne mogu direktno koristiti projektivne koordinate i matrice transformacije, koje značajno pojednostavljuju i ubrzavaju operacije sa modelom na računaru. Za ilustraciju osnovnih elemenata Poenkareovog disk modela pogledati sliku 5.

## 2.3 Poenkareov sferni model

U ovom odeljku biće definisan Poenkareov sferni model, koji predstavlja model hiperboličkog trodimenzionalnog prostora. Posebna pažnja biće posvećena definicijama koje se razlikuju od odgovarajućih definicija u Poenkareovom disk modelu, kao i definicijama koje nisu mogle biti uvedene u disk modelu.

U sfernom modelu apsoluta je sfera čiji je centar ujedno i centar Dekartovog pravougaonog koordinatnog sistema (slično kao u disk modelu, uzima se određenosti radi sfera poluprečnika 1). Unutrašnjost apsolute zvaćemo *h-prostor*, a sve tačke *h*-prostora zvaćemo *h-tačke*. Ako je neki krug (prava) upravna na apsolutu, segment tog kruga (prave) koji se nalazi unutar apsolute zvaćemo *h-prava*. Ako je neka sfera (ravan) upravna na apsolutu deo te sfere (ravni) koji se nalazi unutar apsolute zvaćemo *h-ravan* (ravan upravna na apsolutu sadrži središte apsolute). Inverzijom u odnosu na sferu upravnu na apsolutu (refleksijom u odnosu na ravan upravnu na apsolutu) se unutrašnjost apsolute preslikava sama na sebe. To je motivacija da se definiše *h-ravanska refleksija* u sfernom modelu, kraće *h-refleksija*, kao restrikcija te inverzije (refleksije) na *h-prostor*. Analogno definiciji simetrale *h-duži* u disk modelu, definiše se *simetralna h-ravan h-duži* (kraće *simetralna h-ravan*) kao *h-ravan upravna na datu h-duž* kojom se jedno teme date *h-duži* preslikava u drugo. Pored definicija *h*-ugla, *h*-poligona i *h*-poligonske površi, u sfernom modelu mogu se definisati i *h-poliedar* i *h-poliedarska površ*, slično kao u euklidskoj geometriji.

Definicija *h*-izometrije ekvivalentna je odgovarajućoj definiciji u disk modelu. Specijalno, *h-translacija* za datu *h*-duž se definiše kao kompozicija dve *h*-refleksije, prve u odnosu na *h-ravan normalnu na datu h-duž* u početnoj *h*-tački i druge u odnosu na simetralnu *h-ravan* date *h*-duži. *H-rotacija* oko date *h*-duži za dati *h*-ugao  $\alpha$  je kompozicija dve *h*-refleksije u odnosu na *h*-ravni koje sadrže datu *h*-duž i zahvataju *h*-ugao  $\alpha/2$ , čuvajući orijentaciju *h*-ugla.



Slika 6: ilustracija Poenkareovog sfernog modela; (a) osnovni elementi modela –  $h$ -ravni i  $h$ -prava kao njihov presek, (b) ilustracija  $h$ -refleksije  $h$ -sfere u odnosu na datu  $h$ -ravan.

U Poenkareovom sfernom modelu može se definisati metrika, slično kao u disk modelu, tj. funkcija rastojanja između  $h$ -tačaka. Postoji više funkcija koje zadovoljavaju uslove da se nazivaju metrikom u prostoru, a u ovom radu koristi se metrika data jednačinom

$$d(X, Y) = \ln \frac{1 + p(X, Y)}{1 - p(X, Y)}$$

$$p(X, Y) = \sqrt{\frac{|\vec{x} - \vec{y}|^2}{(1 - |\vec{x}|^2)(1 - |\vec{y}|^2) + |\vec{x} - \vec{y}|^2}}$$

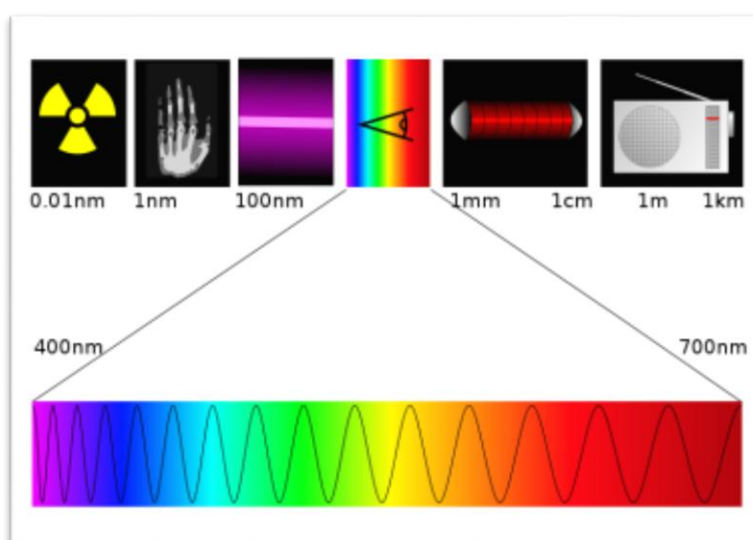
Jednačina 1: funkcija rastojanja  $d$  u Poenkareovom sfernom modelu hiperboličkog prostora.

U navedenoj formuli  $X, Y$  su dekartovske tačke koje odgovaraju datim  $h$ -tačkama modela,  $\vec{x}, \vec{y}$  su njihovi radijus vektori, a vrednost  $|\vec{x}|$  je intenzitet vektora  $\vec{x}$ . Uvođenje metrike omogućava da se, pored  $h$ -krugova, definišu i  $h$ -sfere kao skupovi  $h$ -tačaka u  $h$ -prostoru podjednako udaljenih od date  $h$ -tačke, koja se naziva centrom  $h$ -sfere. U modelu su  $h$ -krugovi i  $h$ -sfere i u euklidskom smislu krugovi i sfere, ali njihovi centri ne moraju nužno biti i centri u euklidskom smislu. Štaviše, centar  $h$ -kruga ne mora pripadati euklidskoj ravni koja je određena datim krugom. Pomoću metrike može se definisati i  $h$ -cilindar kao skup svih  $h$ -tačaka podjednako udaljenih od date  $h$ -prave. Definišemo i  $h$ -valjak kao skup  $h$ -tačaka podjednako udaljenih od date  $h$ -duži.

Poput Poenkareovog disk modela i Poenkareov sferni model je konforman, tj. mera  $h$ -ugla u modelu jednaka je meri odgovarajućeg ugla u euklidskom smislu. Na primer, mera  $h$ -ugla između dve  $h$ -ravni jednaka je meri ugla između dve sfere (ili sfere i ravni ili dve ravni) kojima su određene date  $h$ -ravni.

### 3. Svetlost

U fizici se pojam svetlosti koristi kao sinonim za *elektromagnetno zračenje*, koje se manifestuje u vidu talasa koji se prostiru kroz vakuum ili materiju i sadrže oscilujuće električno i magnetno polje, uzajamno ortogonalne i ortogonalne u odnosu na pravac prostiranja. Elektromagnetno zračenje prema rastućoj talasnoj dužini deli se na *gama zračenje*, *x-zračenje* (poznatije kao *rendgensko<sup>1</sup> zračenje*), *ultravioletno zračenje*, *vidljivu svetlost*, *infracrveno*, *mikrotalasno* i *radio zračenje* (slika 7). Vidljiva svetlost je elektromagnetno zračenje čija je talasna dužina između 400 i 700 nanometara (dok određeni izvori navode i raspon od 380 do 750 nm) i odgovara svetlosti koju ljudsko oko može detektovati (u daljem tekstu pojam svetlosti koristi se kao sinonim za vidljivu svetlost). Pored talasne dužine svetlost se najčešće karakteriše i *intenzitetom*. U ovoj glavi izložene su fizičke osobine svetlosti i proces formiranja slike (Crowell, 2008).

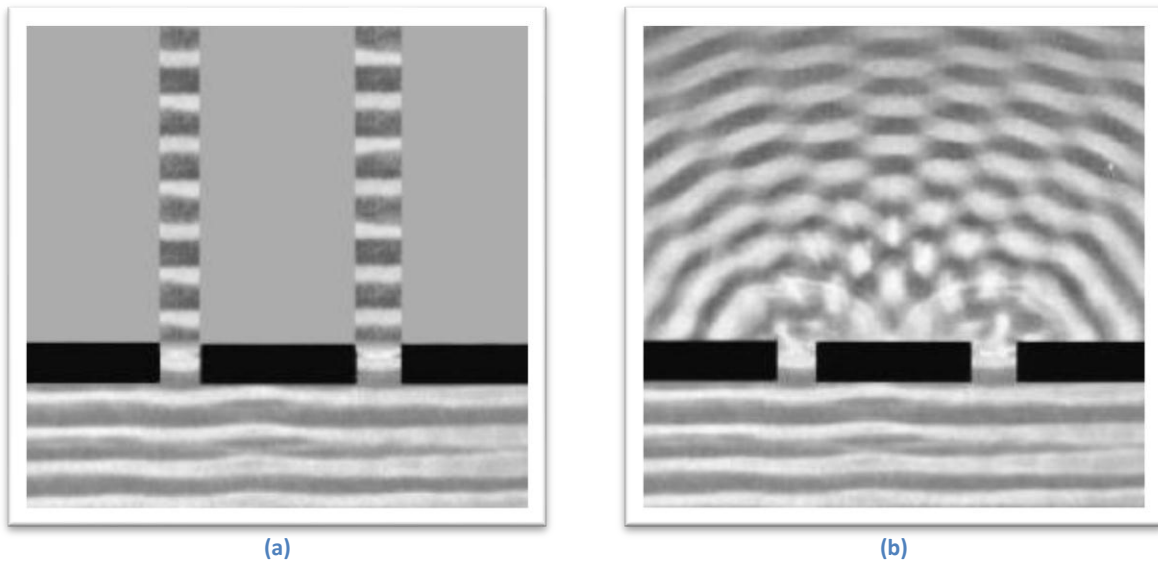


Slika 7: podela elektromagnetnog zračenja prema talasnoj dužini.

#### 3.1 Optika

U tradicionalnoj optici svetlost je predstavljana *zracima*, uzevši da se svetlost u vakuumu prostire pravolinijski. Pomoću tradicionalne optike moguće je opisati osnovne osobine svetlosti, kao što su pomenuto pravolinijsko prostiranje, *refleksija*, *refrakcija*, pa čak i *zakrivljeno prostiranje* unutar određenih transparentnih objekata (kod kojih je koeficijent refrakcije funkcija pozicije u objektu). Generalno se tradicionalna optika može koristiti kada su strukture sa kojima svetlost interaguje veće za nekoliko redova veličine od talasne dužine svetlosti. Međutim, brojni svetlosni efekti nastaju upravo kao rezultat interakcije svetlosti sa objektima čije su dimenzije usporedive sa talasnom dužinom svetlosti. Na primer, da bi se opisali *difrakcija* ili *polarizacija* svetlosti mora se koristiti neki drugi, opštiji model svetlosti.

<sup>1</sup> Vilhelm Konrad Rendgen, nemački fizičar; 1845 – 1923.



Slika 8: interakcija svetlosti sa objektom čije su dimenzije usporedive sa talasnom dužinom svetlosti: (a) tradicionalna optika ne može da opiše difrakciju koja nastaje u datoj interakciji; (b) talasni model prostiranja svetlosti korektno predviđa širenje talasa nakon prolaska svetlosti kroz proreze.

Model koji bolje opisuje ponašanje svetlosti u prirodi je takozvani *talasni model* (model koji pripada *talasnoj optici*) Roberta Huka<sup>1</sup>, koji svetlost tretira kao talas određene talasne dužine i pravca prostiranja. Talasni model je u stanju da opiše interakcije svetlosti sa objektima čije su dimenzije slične talasnoj dužini svetlosti ili manje (slika 8). S druge strane, talasni model je komplikovaniji od tradicionalnog modela – pored vektora prostiranja svetlosti potrebno je imati i podatak o talasnoj dužini svetlosti. I sam opis interakcije svetlosti sa objektima potrebno je proširiti da bi se obuhvatila interakcija talasa sa tim objektima. Međutim, ni talasni model nije u stanju da opiše sve osobine svetlosti. Na primer, *foto-električni efekat* ili *zračenje crnih tela* mogu se objasniti korišćenjem modela koji uzima u obzir *dualnu prirodu* svetlosti: svetlost se može ponašati i kao talas i kao čestica. Ajnštajn<sup>2</sup> je 1905. opisao foto-električni efekat uvodeći koncept dualne prirode svetlosti, proširujući rezultate do kojih je došao Maks Plank<sup>3</sup> nekoliko godina ranije. Time je početak XX veka u fizici obeležen pojavom *kvantne optike*, koja predstavlja sveobuhvatni model ponašanja svetlosti u prirodi.

Prilikom kreiranja algoritama računarske grafike koji se bave foto-realističnim renderovanjem obično se koristi tradicionalna optika, primenjena na vidljiv spektar svetlosti. Iako ona ne može da opiše sve interakcije svetlosti u prirodi, scene koje se najčešće predstavljaju na računaru ne sadrže objekte čije bi interakcije sa svetlošću zahtevale primenu talasne ili kvantne optike. Stoga je u nastavku podrazumevano predstavljanje svetlosti zracima. Postoje i algoritmi koji za renderovanje koriste Maksvelove<sup>4</sup> jednačine talasne optike, ali kompleksnost datih jednačina dovodi do drastičnih usporenja u procesu kreiranja slike.

<sup>1</sup> Robert Huk, engleski matematičar; 1635 – 1703.

<sup>2</sup> Albert Ajnštajn, nemački fizičar; 1879 – 1955.

<sup>3</sup> Maks Plank, nemački fizičar; 1858 – 1947.

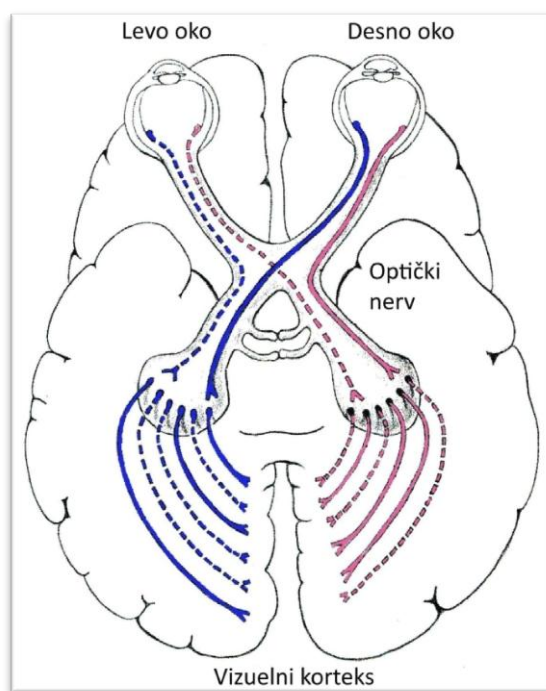
<sup>4</sup> Džejs Klerk Maksvel, škotski fizičar; 1831 – 1879.



## 3.2 Ljudski vizuelni sistem

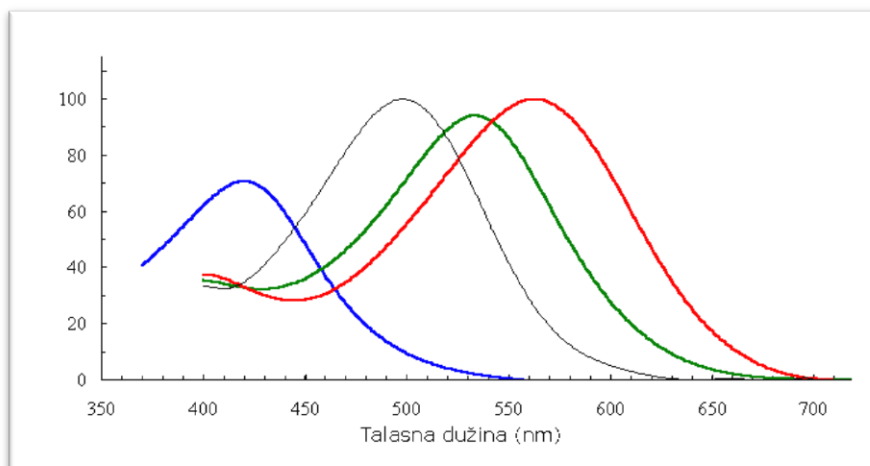
Nakon višestrukih interakcija svetlosti sa objektima koji se, zajedno sa posmatračem, nalaze u sceni, određeni procenat svetlosti nalazi svoj put i do očiju posmatrača. Međutim, ljudski vizuelni sistem poseduje određene specifičnosti koje se u računarskoj grafici koriste kako bi se aproksimacijama realnog fizičkog sveta smanjila kompleksnost problema osvetljenja, ne umanjujući pritom realističnost prikaza.

Vizuelni sistem većine sisara sastoji se iz sledećih osnovnih delova: *oko*, *optički nerv* i *vizuelni korteks* (slika 9). Oko je organ koji pretvara svetlost, čija talasna dužina pripada određenom intervalu, u električni signal. Taj signal se dalje pomoću optičkog nerva šalje u mozak, u kojem se formira finalna slika. Ljudsko oko poseduje tri vrste foto-receptora, odgovornih za detekciju boja („čepići“), svetlosti slabog intenziteta („štapići“) i ambijentalne svetlosti (*foto-osetljive ganglionske ćelije*). Do otkrića treće grupe foto-receptora, tj. ganglionskih ćelija, došlo se 1991. godine razmatranjem stimulacije očiju glodara, a prethodno je smatrano da se ljudski vid sastoji isključivo od „čepića“ i „štapića“. U daljoj analizi vizuelnog sistema biće podrazumevana valjanost starog modela, s obzirom na to da postojanje ganglionskih ćelija praktično nema uticaja na prikaz slika u računarskoj grafici.



Slika 9: vizuelni sistem homo sapiensa.

Kao što je već napomenuto, foto-receptori u ljudskom oku pretvaraju svetlost u električni signal, ukoliko talasna dužina svetlosti pripada određenom intervalu. Međutim, funkcija odziva foto-receptora unutar tog intervala nije konstanta, već zavisi od specifične talasne dužine unutar samog intervala (slika 10). Boje se detektuju pomoću tri vrste „čepića“, odgovornih redom za svetlost kraćih, srednjih i dužih talasnih dužina.



Slika 10: funkcije odziva foto-receptora.

S obzirom na to da je ljudski vizuelni sistem opremljen tri-hromatskim sensorima, u računarskoj grafici se ova činjenica uzima u obzir u cilju pojednostavljivanja problema osvetljenosti – umesto da se svetlost predstavlja pomoću analitičke funkcije talasne dužine ili njene gusto tabelirane verzije, u grafici je svetlost najčešće predstavljena pomoću tri vrednosti koje odgovaraju odzivu svake od tri vrste „čepića“. To su vrednosti koje se identifikuju kao osnovne boje: *plava*, *zelena* i *crvena*. Data aproksimacija omogućava pojednostavljivanje procesa renderovanja, uz smanjenje opštosti metoda i, u određenim slučajevima, kvaliteta finalne slike. Određene kompleksne interakcije svetlosti sa okolinom, poput *fluorescentnosti*, ne mogu se adekvatno predstaviti tri-hromatskim modelom. Čak i u slučaju da su svi objekti prisutni u datoj sceni „jednostavni“, u kontekstu njihovih interakcija sa svetlošću, ukoliko se želi tačan rezultat renderovanja mora se posegnuti ka takozvanom *multi-spektralnom renderovanju*, gde je svetlost predstavljena daleko preciznije u spektralnom smislu. Pored toga, ukoliko je nemoguće izmeriti vrednosti sve tri komponente boje tri-hromatskog modela, tada se obično zelena komponenta gušće uzorkuje od plave i crvene komponente. Razlog za to leži u navedenoj činjenici da je funkcija spektralnog odziva „čepića“ srednjih talasnih dužina slična funkciji odziva „štapića“ i da najbolje aproksimira osetljivost ljudskog oka na intenzitet svetlosti, koja je daleko veća od osetljivosti na obojenost svetlosti.

## 4. Svetlost u računarskoj grafici

*Spiritus movens* ogromnih istraživačkih napora u računarskoj grafici je problem *globalne iluminacije*. Fizičke osobine svetlosti, iako komplikovane (npr. dualna priroda svetlosti) mogu se za potrebe računarske grafike relativno jednostavno konceptualno opisati, ali je njeno modelovanje na računaru praktično nemoguće. Stoga je razvijen veliki broj metoda za predstavljanje svetlosti i njenih interakcija sa objektima u sceni koji bi pojednostavili sam problem i omogućili da se na računaru generišu foto-realistične slike. U ovoj glavi biće izloženi osnovni lokalni modeli interakcije svetlosti sa objektima, proces formiranja slike i globalni model osvetljenja (Foley et al., 1990), (Kajiya, 1986), (Weyrich et al., 2009).

### 4.1 Lokalni modeli osvetljenja

*Lokalni modeli osvetljenja* predstavljaju modele interakcije svetlosti sa određenim objektom. Nasuprot lokalnim modelima su *globalni modeli osvetljenja*, gde se razmatraju višestruke interakcije svetlosti sa svim objektima u sceni, pri čemu je svaka pojedinačna interakcija predstavljena određenim lokalnim modelom. U ovom poglavlju biće predstavljeni osnovni koncepti koji stoje iza najčešće korišćenih lokalnih modela osvetljenja.

*Funkcija distribucije bidirekcionne refleksije* (eng. *Bidirectional Reflection Distribution Function – BRDF*), je 4-dimenziona funkcija koja opisuje distribuciju refleksije svetlosti od neprozirnih objekata, uz pretpostavku da ne dolazi do ispod-površinskog rasipanja svetlosti (Nicolodemus, 1965). Ova funkcija je definisana na prostoru parova svih upadnih i reflektovanih zraka, definisanih u odnosu na normalu na površ objekta u tački preseka sa tim zracima. Njen rezultat predstavlja odnos upadne i reflektovane jačine svetlosti predstavljene tim zracima, gde je jedinica mere funkcije  $sr^{-1}$ . Funkcija je 4-dimenzionalna pošto se svaki od zraka može predstaviti dvema vrednostima: azimutalnim uglom i elevacijom. Formula kojom se matematički opisuje BRDF je:

$$f_r(w_u, w_r) = \frac{dL_r(w_r)}{dE_u(w_u)}$$

**Jednačina 2: jednačina BRDFa. Indeks 'u' označava upadni zrak svetlosti, dok je indeks 'r' oznaka za reflektovani zrak.**

U datoj formuli  $L$  je *radijansa*, tj. intenzitet reflektovane svetlosti, dok je  $E$  *iradijansa*, tj. intenzitet svetlosti koja pada na dati objekat. Data jednačina BRDFa sama po sebi ne podrazumeva da će njome opisane osobine refleksije svetlosti podlegati osnovnim fizičkim zakonima koji važe u prirodi. Ukoliko se želi model refleksije svetlosti koji podleže zakonima fizike, mora se jednačina BRDFa specijalizovati uvođenjem dodatnih zahteva. To su redom *Helmholzov<sup>1</sup> reciprocitet* i *princip očuvanja energije*:

---

<sup>1</sup> Herman Ludvig Ferdinand fon Helmholtz, nemački fizičar; 1821 – 1894.

$$f_r(w_u, w_r) = f_r(w_r, w_u)$$

$$\forall w_u, \int_{\Omega} f_r(w_u, w_r) \cos \theta_i dw_r \leq 1$$

Jednačina 3: Helmholtcov reciprocitet i princip očuvanja energije.

Koncept BRDFa je veoma važan u računarskoj grafici pošto omogućava da se kompleksne interakcije svetlosti sa objektom jednostavno matematički modeluju. Primetimo da su pretpostavke o pojednostavljenim osobinama objekata, koje su neophodne za uvođenje koncepta BRDFa, jake i u najvećem broju slučajeva objekti koji se žele modelovati posedovaće kompleksnije fizičke osobine. Pre svega, BRDF ne uzima u obzir da objekti u prirodi najčešće nisu homogeni, tj. da postoje određene varijacije njihovih reflektivnih osobina, koje zavise od pozicije gde se odvija razmatrana interakcija svetlosti i površine objekta. Stoga se uvodi koncept *prostorno varijabilne funkcije distribucije refleksije svetlosti* (eng. *Spatially-varying BRDF – svBRDF*). svBRDF je 6-dimenziona funkcija koja, pored zavisnosti od upadnog i reflektovanog zraka svetlosti, zavisi i od pozicije na objektu (koja se može predstaviti parametarski pomoću dve koordinate). svBRDF se tada obično zapisuje kao  $f_r(w_u, w_r, x)$ . Drugim rečima, pretpostavka na kojoj se bazira svBRDF je da se u svakoj tački objekta njegove reflektivne osobine mogu predstaviti odgovarajućim BRDF-om (Matusik et al., 2009).

Mnogi objekti, međutim, ne zadovoljavaju datu pretpostavku, te se svBRDF koncept proširuje takozvanom *bidirekcionom teksturnom funkcijom* (eng. *Bidirectional Texture Function – BTF*). BTF je parametrizovan na identičan način kao i svBRDF, no u BTF su inkorporirani određeni globalni efekti, poput ispod-površinskog rasipanja i samo-senčenja (Dana et al., 1996). Drugim rečima, BTF eliminiše pretpostavku da se objekat može predstaviti kolekcijom BRDF-ova. Konačno, najopštiji model lokalnog osvetljenja koji se koristi u računarskoj grafici je koncept *funkcije distribucije bidirekciono refleksije sa površinskim rasipanjem* (eng. *Bidirectional Surface Scattering Reflection Distribution Function – BSSRDF*). To je 8-dimenziona funkcija koja uzima u obzir da svetlost incidentna na objektu ne mora da se reflektuje u istoj tački – nakon ispod-površinskog rasipanja svetlost može napustiti objekat na drugoj lokaciji (Jensen et al., 2001). Primer takve interakcije svetlosti i objekta su *translucentni* objekti, poput voska. BSSRDF se zapisuje kao  $f_r(w_u, x_u, w_r, x_r)$ . Iako je jasno da BRDF predstavlja refleksiju svetlosti u odnosu na dati objekat na veoma ograničen način, iz praktičnih razloga biće podrazumevano da BRDF predstavlja standardnu funkciju refleksije svetlosti.

BRDF se može predstaviti pomoću analitičke funkcije ili pomoću tabeliranih vrednosti izmerenih npr. *goniorefleksimetrom*. Oba pristupa imaju svojih prednosti i mana – analitički pristup je kompaktniji i izbegava se potreba za interpolacijom, ali s druge strane izračunavanje vrednosti funkcije može biti vremenski zahtevno; pored toga, analitičke funkcije obično nisu u stanju da verno predstavljaju varijacije koje BRDF poseduje, a koje nisu glatke, pa čak ni neprekidne. Tabelarni pristup je memorijski daleko zahtevniji, podložan je vidljivim artefaktima koji su rezultat interpolacije (ukoliko je gustina merenja BRDFa nedovoljna) i sadrži više šuma u signalu, ali se ovim pristupom mogu dobiti realističniji prikazi BRDFa materijala. S obzirom na to da su za potrebe ovog rada analitički BRDF modeli adekvatni, u nastavku teksta pod BRDF modelom biće uvek podrazumevan analitički model.

## 4.2 Analitički BRDF modeli

Najjednostavniji analitički BRDF model je *difuzni*, poznatiji i kao *Lambertov model*. Za objekat se kaže da poseduje osobine Lambertove refleksije ukoliko je jačina reflektovane svetlosti *izotropna*, tj. ukoliko je jačina reflektovane svetlosti jednaka bez obzira na ugao posmatranja. Drugim rečima, svetlost incidentna na objektu se rasipa prilikom refleksije podjednako u svim pravcima na hemisferi. Primer objekta koji se može nazvati difuznim je školska kreda. Međutim, većina objekata se ne može okarakterisati čistom difuznom refleksijom, već je potrebno u model uvesti i pojam *sjajnosti*, odnosno *spekularne refleksije*, koji predstavlja refleksije svetlosti incidentne na objektu koja se ne rasipa podjednako u svim pravcima, već postoji jedan ili više dominantnih pravaca refleksije. Primer spekularnog materijala je polirani metal.

Prvi BRDF model koji eksplicitno tretira spekularnu refleksiju je *Fongov*<sup>1</sup> empirijski model iz 1973. godine (Phong, 1975). Taj model zapravo kombinuje tri komponente refleksije: *ambijentalnu*, *difuznu* i *spekularnu* (ambijentalna komponenta je pokušaj da se simuliraju višestruke uzastopne interakcije svetlosti sa difuznim objektima u sceni, koje je gotovo nemoguće praktično izračunati – videti poglavlje 4.3). Fongova jednačina lokalnog osvetljenja može se zapisati na sledeći način:

$$I_{\lambda}^r = I_{a\lambda} k_a O_{d\lambda} + \sum_{1 \leq i \leq m} S_i f_{att} I_{i\lambda} (k_d O_{d\lambda} \cos \theta + k_s O_{s\lambda} \cos^n \alpha)$$

Jednačina 4: Fongov BRDF model.

Data formula primenjuje se za crvenu, zelenu i plavu komponentu pojedinačno: kombinovanjem rezultata dobija se konačna boja piksela slike. U datoj formuli  $\lambda$  je oznaka za komponentu boje koja se trenutno računa,  $I_{\lambda}^r$  je rezultujući intenzitet svetlosti,  $I_{a\lambda}$  je intenzitet ambijentalnog osvetljenja scene, dok je  $I_{i\lambda}$  intenzitet svetlosnog izvora. Koeficijenti refleksije materijala su dati sa  $k_a$ ,  $k_d$  i  $k_s$ , za ambijentalnu, difuznu i spekularnu refleksiju respektivno, dok je boja objekta data sa  $O_{d\lambda}$  i  $O_{s\lambda}$  za difuznu i spekularnu refleksiju i za svaku komponentu boje posebno. Pošto svetlost slabi sa povećanjem udaljenosti, komponenta  $f_{att}$  sadrži koeficijent slabljenja svetlosti. Ukoliko je izvor svetla blokiran nekim objektom koeficijent  $S_i$  jednak je 0 (taj svetlosni izvor ne utiče na boju objekta); inače ima vrednost 1. Ugao  $\theta$  je ugao između pravca izvora svetlosti i normale na objekat u razmatranoj tački i njime je određen intenzitet difuzne refleksije, dok je ugao  $\alpha$  ugao između pravca izvora svetlosti i reflektovanog zraka pogleda i određuje intenzitet spekularne refleksije. Pošto se razni objekti razlikuju po sjajnosti, vrednost  $\cos \alpha$  stepenuje se koeficijentom sjajnosti  $n$ , pri čemu sjajniji objekti imaju veći koeficijent sjajnosti. Parametar  $m$  predstavlja broj svetlosnih izvora u sceni, a boja trenutnog piksela dobija se sabiranjem rezultata osvetljenja razmatranog objekta od svakog izvora svetla ponaosob. Često se koristi i modifikovana verzija formule u kojoj se boja objekta i koeficijenti refleksije kombinuju u jedinstvene koeficijente refleksije definisane za svaku komponentu boje i za ambijentalnu, difuznu i spekularnu refleksiju posebno.

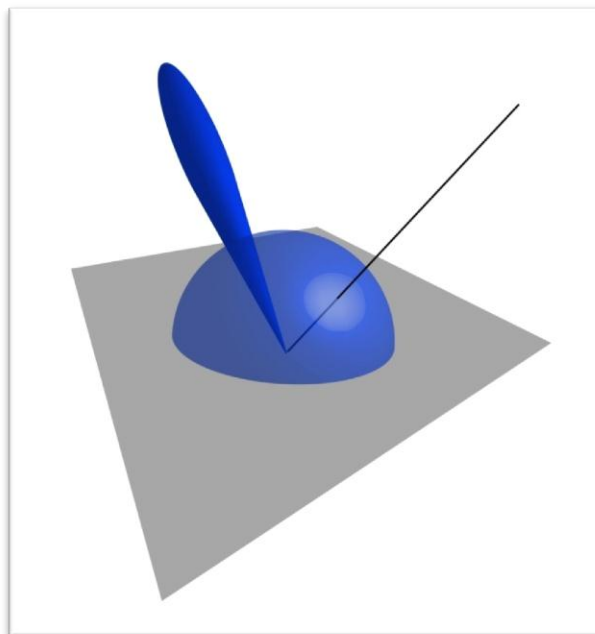
<sup>1</sup> Fong Bui Tuong, američki informatičar; 1942 – 1975.

Objasnimo ukratko i pojam slabljenja svetlosti koji se koristi u formuli Fongovog lokalnog modela osvetljenja. Razmotrimo tačkasti izvor svetlosti u vakuumu koji emituje kratak impuls svetlosti intenziteta  $I$  jednako u svim pravcima. Svetlost se ravnomerno širi kroz vakuum obrazujući takozvani *sferni talasni front*, odnosno talas svetlosti koji se u sfernom obliku udaljava od izvora svetla. Pošto se svetlost kreće kroz vakuum, ukupan intenzitet sfernog talasnog fronta svetlosti jednak je početnom intenzitetu svetlosti  $I$  i jednako je raspoređen po celoj površini talasnog fronta. Razmotrimo tačku prostora na udaljenosti  $r$  od izvora svetlosti. U toj tački intenzitet osvetljenja obrnuto je proporcionalan površini sfere talasnog fronta. Kako je površina sfernog talasnog fronta svetlosti koja je prešla rastojanje  $r$  od izvora jednaka  $4r^2\pi$ , dobija se da je koeficijent slabljenja osvetljenja u razmatranoj tački jednak:

$$f_{att} = \frac{1}{4r^2\pi}$$

**Jednačina 5: koeficijent slabljenja svetlosti u prostoru.**

Prednost Fongovog modela u odnosu na većinu drugih modela je njegova konceptualna jednostavnost. U poređenju sa sveobuhvatnijim modelima, Fongov model je i računski jednostavniji. Nedostatak Fongovog modela je njegova *izotropnost* – spekularna refleksija svetlosti ne zavisi od toga sa koje strane se posmatra dati objekat. Taj nedostatak modela ogleda se u krajnjem rezultatu, pošto objekti sa materijalima koji se opisuju Fongovim BRDF-om izgledaju isuviše „plastično“ – Fongov model i njegovi derivati nisu u stanju da pruže adekvatan nivo fizičkog realizma. Ilustracija Fongovog modela data je na slici 11.



**Slika 11: ilustracija Fongovog BRDF modela - upadni zrak svetlosti doprinosi difuznoj i spekularnoj komponenti refleksije.**

Osnovni nedostatak Fongovog modela i njegovih derivata je pomenuta izotropnost. Stoga su kreirani *anizotropni* BRDF modeli koji pokušavaju da prevaziđu pomenuto ograničenje Fongovog modela.

U *Vardovom*<sup>1</sup> *anizotropnom modelu* spekularnost se definiše pomoću dva parametra,  $\alpha_x$  i  $\alpha_y$ , koji određuju jačinu spekularnosti u datim glavnim pravcima, pri čemu su vektori pravaca  $\vec{x}$  i  $\vec{y}$  međusobno ortogonalni. Spekularnost u Fongovom modelu je određena samo jednim parametrom  $\alpha$ , što nije dovoljno da se predstavi njena anizotropnost.

Anizotropnost BRDF modela se uvodi i pomoću koncepta *mikro-ravni*, tj. ideje da je površina objekta nije savršeno glatka, već je prekrivena brojnim planarnim segmentima. Dimenzije tih segmenata su zanemarljive u odnosu na dimenzije objekta i ne utiču na vidljivu geometriju samog objekta. Međutim, prilikom njihove interakcije sa svetlošću pravac normale na svaku od mikro-ravni utiče na to kako će se svetlost od date mikro-ravni reflektovati. Jasno je da je broj takvih mikro-ravni preveliki da bi se mogao eksplicitno predstaviti na računaru. Umesto toga, mikro-ravni predstavljaju se statističkim modelom njihove distribucije po površi objekta. Pored toga, potrebno je definisati i *geometrijsko slabljenje*, tj pojavu da deo svetlosti reflektovan od neke mikro-ravni može biti blokiran od strane druge mikro-ravni. Najpoznatiji model koji koristi mikro-ravni je *Kuk<sup>2</sup>-Torensov<sup>3</sup> model*. U ovom radu koristi se Fongov model, s obzirom na to da kreiranje foto-realističnih slika nije primarni cilj ovog rada (upitno je i da li termin „foto-realistično“ ima smisla koristiti kada se govori o renderovanju u hiperboličkom prostoru).

### 4.3 Globalni model osvetljenja i jednačina renderovanja

*Jednačina renderovanja* (rendering jednačina) predstavlja jedan od osnovnih konceptata u računarskoj grafici. Uvedena je 1986. godine od strane Džems Kadžije<sup>4</sup> i Dejvida Imela<sup>5</sup> i u njoj je matematički opisano globalno ponašanje svetlosti u sceni, ukoliko su poznati modeli interakcije svetlosti sa svim objektima prisutnim u sceni pojedinačno, tj. njihovi BRDF-ovi (Kajiya, 1986). Model ponašanja svetlosti koji je opisan jednačinom renderovanja je fizički korektan i u teoriji je moguće generisati savršeno realistične slike na računaru ukoliko se pronađe njeno tačno rešenje, uz uslov da su svi BRDF-ovi prisutni u sceni takođe fizički korektni. Sama jednačina se najčešće zapisuje u sledećem obliku:

$$L_i(x, \vec{w}_i) = L_e(x, \vec{w}_i) + \int_{\Omega} f_r(x, \vec{w}_u, \vec{w}_i) L_u(x, \vec{w}_u) (\vec{w}_u \cdot \vec{n}) d\vec{w}_u$$

Jednačina 6: jednačina renderovanja.

<sup>1</sup> Gred Vard Larson, američki informatičar.

<sup>2</sup> Robert L. Kuk, američki informatičar; 1952 –.

<sup>3</sup> Kenet E. Torens, američki informatičar.

<sup>4</sup> Džems T. Kadžija, američki informatičar.

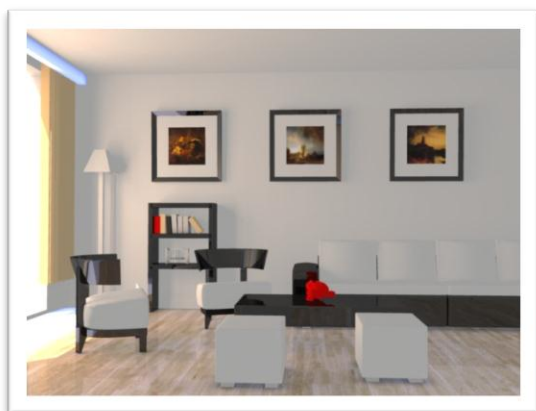
<sup>5</sup> Dejvid Imel, američki informatičar.

$L_i(x, \vec{w}_i)$	Ukupna jačina svetlosti koja napušta scenu u tački $x$ i pravcu $\vec{w}_i$
$L_e(x, \vec{w}_i)$	Jačina emitovane svetlosti koja napušta scenu u tački $x$ i pravcu $\vec{w}_i$
$f_r(x, \vec{w}_u, \vec{w}_i)$	BRDF u tački $x$ za ulazni pravac $\vec{w}_u$ i izlazni pravac $\vec{w}_i$
$L_u(x, \vec{w}_u)$	Jačina ulazne svetlosti u tački $x$ i pravcu $\vec{w}_u$
$(\vec{w}_u \cdot \vec{n})$	Koeficijent slabljenja ulazne svetlosti (zavisnost od upadnog ugla)
$\int_{\Omega} \dots d\vec{w}_u$	Integral po hemisferi u tački $x$ za sve ulazne pravce $\vec{w}_u$ (infinitezimalna suma refleksija svih ulaznih zraka u datoj tački i izlaznom pravcu)

Tabela 1: opis elemenata rendering jednačine.

Tabela 1 opisuje elemente rendering jednačine. Primetimo da su u navedenoj jednačini nepoznate ukupna jačina svetlosti koja napušta scenu, kao i ukupna jačina ulazne svetlosti u sceni – izračunavanje jedne od ove dve vrednosti podrazumeva istovremeno izračunavanje i druge vrednosti. Ovakav tip integrala naziva se *Fredholmov<sup>1</sup> integral II vrste* i njegovo analitičko rešavanje je praktično nemoguće. Stoga se u rešavanju rendering jednačine najčešće pristupa matematičkim trikovima ili fizičkim aproksimacijama u cilju pojednostavlivanja problema.

Jedan od veoma popularnih metoda rešavanja rendering jednačine je *radiositi* (eng. Radiosity) metod, koji podrazumeva da svi objekti u sceni ispoljavaju osobine difuznih reflektora. Uz datu aproksimaciju se rendering jednačina pojednostavljuje, pošto se oslobađa zavisnosti od pravaca prostiranja svetlosti – radiositi metod tada definiše konačan broj segmenata prostora i izračunava transport energije svetlosti između njih. Mana radiositi metoda je u samoj aproksimaciji prostora difuznim objektima, s obzirom na to da su sjajni objekti neizbežni u većini realističnih scena. S druge strane, prednost ovog metoda je u tome što izračunati transport energije između segmenata prostora ne zavisi od pozicije posmatrača, zbog čega se radiositi često koristi kada je potrebno generisati interaktivan prolaz kroz velike trodimenzionalne scene. Slika 12a ilustruje tipičan konačan izgled scene renderovane radiositi algoritmom – svi objekti u datoj sceni su difuzni reflektori i ne modeluje se spekularna refleksija.



(a)



(b)

Slika 12: ilustracija algoritama globalne iluminacije – (a) radiositi algoritam; (b) algoritam klase Monte Karlo.

<sup>1</sup> Erik Ivar Fredholm, švedski matematičar; 1866 – 1927.



U rešavanju rendering jednačine mogu se koristiti i probabilističke metode – umesto da se prostiranje svetlosti i njene interakcije sa objektima u sceni pokušaju eksplicitno izračunati za svaki mogući pravac u sceni, može se statistički izdvojiti uzorak zraka koje će biti praćen, počevši od nekog zdatog skupa zraka. Takvi metodi se obično nazivaju *Monte Karlo metodima*, kojima pripada i sam rejtrensing algoritam. Za uspeh Monte Karlo metoda ključan je izbor distribucije zraka koji će biti korišćeni prilikom izračunavanja: pogrešan izbor funkcije distribucije dovodi do rezultata koji odstupaju od matematičkog očekivanja. S obzirom na to da je spekularna komponenta BRDF-ova objekata obično definisana pomoću malog broja dominantnih pravaca jednostavno je kreirati funkciju distribucije zraka koja će uvek uzeti te pravce u razmatranje. Stoga je kvalitet prikaza sjajnih objekata pomoću Monte Karlo metoda gotovo po pravilu blizu foto-realističnosti. Problem nastaje kod modeliranja višestrukih uzastopnih difuznih refleksija, jer je za njihovo modelovanje neophodan ili veliki broj zraka ili optimalan izbor manjeg broja zraka. Dati metodi zapravo predstavljaju varijante rešenja Fridholmovog integrala poznatijeg kao *Liuvil<sup>1</sup>-Nojmanov<sup>2</sup> red*, koji konvergira ka rešenju u beskonačnosti. Rezultat korišćenja takvog metoda ilustrovan je na slici 12b. Spekularna refleksija je verno prikazana, što nije moguće postići radiositi metodom.



Slika 13: dvoprolazni rendering algoritam.

Već je napomenuto da rešenja rendering jednačine koja pripadaju klasi Monte Karlo algoritama polaze od zdatog skupa zraka, definisanog od strane korisnika. Taj skup zraka bi trebalo da bude neki podskup skupa svih zraka koji napuštaju izvore svetlosti – svetlost napušta izvor i nakon višestrukih interakcija sa objektima u sceni određeni procenat zraka prolazi kroz vidnu ravan i završava u tački pogleda. Međutim, problem leži u tome što je nemoguće unapred odrediti koji zraci su od važnosti za datu sceni, tj. koji zraci će završiti u vidnoj tački i doprineti konačnom izgledu slike. Stoga se može razmatrati obrnuta putanja svetlosti – prate se zraci iz vidne tačke u sceni i vrši analiza osvetljenosti objekata u obrnutom smeru. Prednost ovog principa je što se prate samo oni zraci za koje smo sigurni da direktno doprinose konačnom izgledu slike. S druge strane, očigledna mana leži u

<sup>1</sup> Jozef Liuvil, francuski matematičar; 1809 – 1882.

<sup>2</sup> Karl Gotfrid Nojman, nemački matematičar; 1832 – 1925.

činjenici da je teško predvideti koji zraci svetlosti određuju ambijent slike, tj. difuznu komponentu osvetljenja u sceni. Stoga se za vrhunske rezultate obično koriste takozvani *dvoprolazni algoritmi*: prvo se odredi globalna distribucija iluminacije praćenjem određenog broja zraka ili fotona iz izvora svetlosti do objekata u sceni, a potom se iz vidne tačke za svaki piksel određuje vidljivost i direktna iluminacija piksela u toj tački, uzevši u obzir vrednosti određene u prvom prolazu. Ovakvi metodi su računski zahtevniji od jednoprolaznih metoda, ali daju superiorne rezultate (slika 13).

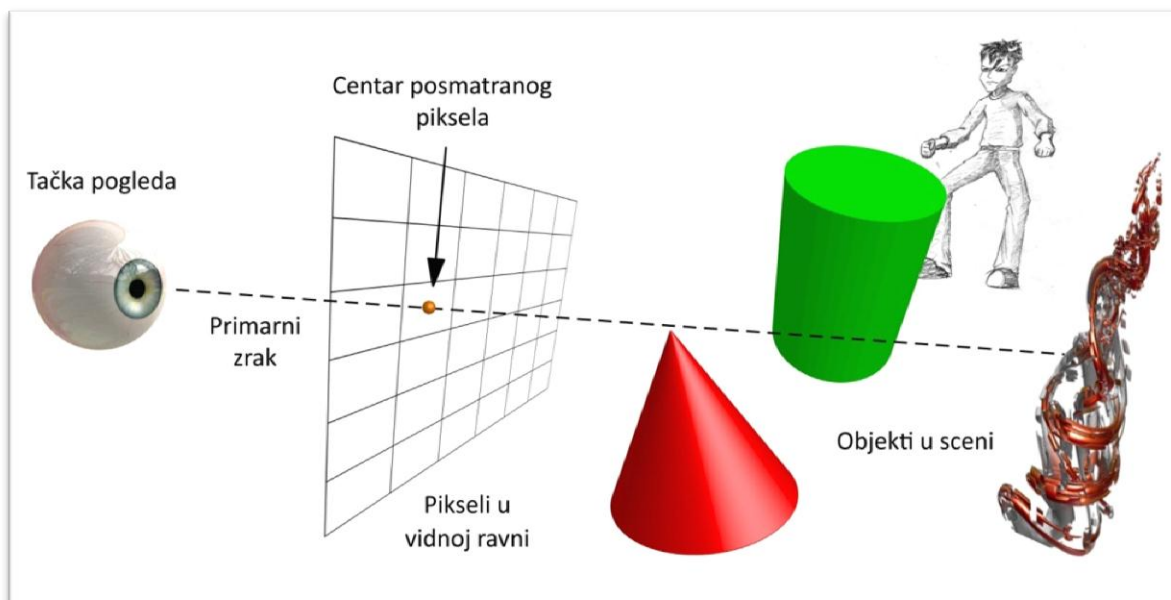
Primetimo da jednačina 6 već poseduje određene aproksimacije. U navedenom obliku jednačine ignoriše se pojam vremena, kao i zavisnost rezultata od talasne dužine svetlosti. Nedostatak zavisnosti od talasne dužine svetlosti je zapravo ekvivalent predstavljanju svetlost kao *monohromatske*, tj. nebojene. U tom slučaju svetlost je potpuno opisana svojim intenzitetom i pravcem prostiranja. Da bi se omogućilo i generisanje boja, za svaku od tri osnovne boje (plavu, crvenu i zelenu) rendering jednačina se primenjuje posebno. Ukoliko je cilj fizički korektno tretiranje pojma obojenosti svetla, tada se mora uključiti i zavisnost od talasne dužine u jednačinu, čime se dobija *spektralna jednačina renderovanja*, koja je u stanju da opiše i kompleksne promene boja usled npr. fluorescentnosti objekta. Nedostatak zavisnosti jednačine renderovanja od vremena kao parametra donosi dva suštinska ograničenja: prvo je da se ne mogu opisati određene pojave poput *fosforescentosti*, kod kojih je emisija svetlosti funkcija akumulirane svetlosti i vremena; drugo ograničenje je pretpostavka da se svetlost kroz vakuum prostire beskonačno velikom brzinom. U najvećem broju slučajeva scene koje želimo da renderujemo ne sadrže objekte čija brzina je dovoljno velika da se primete relativistički efekti. Takođe, ni posmatrač u sceni, odnosno kamera, obično se ne kreću relativističkim brzinama. U slučaju da se relativistički efekti u sceni žele eksplicitno predstaviti mora se parametar vremena, tj. ograničenost brzine svetlosti, uzeti u razmatranje.

## 5. Rejtresing algoritam

U prethodnoj glavi je navedeno da je rejtresing algoritam predstavnik klase Monte Karlo algoritama koji predstavljaju aproksimacije rešenja jednačine renderovanja. Prisetimo se da su ti algoritmi zapravo različite varijante rešenja u obliku Luivil-Nojmanovog reda, koji u beskonačnosti konvergira ka tačnom rešenju (u slučaju jednačine renderovanja, tačno rešenje je fizički korektna slika generisana na osnovu datog opisa scene). Međutim, sa istorijskog stanovišta, nastanak rejtresing algoritma nije direktno povezan sa naporima da se reši rendering jednačina. Štaviše, sam algoritam je nastao pre nego što je jednačina uopšte bila formulisana. U ovoj glavi biće izložen rejtresing algoritam induktivnim metodom, tj. polazeći od jednostavnijih algoritama i njihovim uopštavanjem. Razlog zbog kojeg je odabran dati pristup je intuitivnije predstavljanje algoritma i njegovo lakše razmevanje. Prvo će biti predstavljen *rejkasting* algoritam (eng. Ray Casting), čijom generalizacijom je nastao rejtresing algoritam.

### 5.1 Rejkasting algoritam

Rejkasting je algoritam računarske grafike originalno nastao da bi se omogućilo određivanje vidljivih površina iz date tačke pogleda u sceni. Razvijen od strane Artura Apela<sup>1</sup> 1968. godine, svoj sadašnji naziv dobio je 1982. godine. Osnovna ideja rejkasting algoritma je da se iz tačke pogleda kroz svaki piksel virtualnog prozora pušta zrak vidljivosti i detektuje prvi objekat u sceni koji taj zrak seče – to je objekat koji bi dati posmatrač video u tako definisanoj sceni (Appel, 1968). Slika 14 ilustruje princip funkcionisanja rejkasting algoritma.



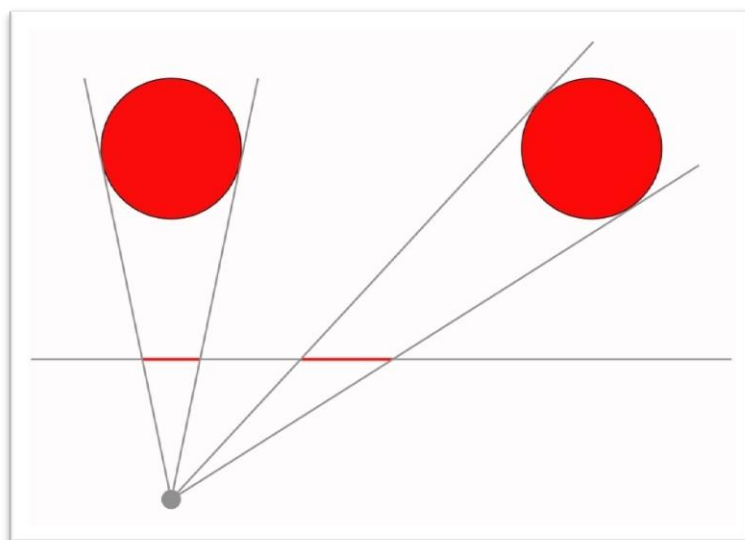
Slika 14: ilustracija rejkasting algoritma: iz tačke pogleda pušta se zrak kroz centar svakog piksela i traži najbliži presek sa objektima u sceni.

<sup>1</sup> Artur Appel, američki informatičar.

U sceni se, pored objekata koji se žele predstaviti, definišu i *tačka pogleda*, u kojoj se nalazi posmatrač i *prozor vidljivosti*, koji definiše horizontalne i vertikalne uglove vidljivosti. Taj prozor pripada takozvanoj *vidnoj ravni* i u njemu se definiše mreža piksela koji se nakon kreiranja slike prikazuju na ekranu. Sada se za tako definisanu tačku pogleda i piksele u vidnoj ravni postavljaju zraci pogleda, koji se nazivaju i *primarnim zracima*, duž kojih se traži presek sa objektima u sceni. Objekat koji je najbliži tački pogleda je objekat koji posmatrač u sceni vidi iz date tačke pogleda. Rejkasting spada u red *image-precision* algoritama određivanja vidljivosti, tj. algoritama koji vidljivost određuju za svaki piksel slike pojedinačno. Pored njih postoje i takozvani *object-precision* algoritmi kod kojih se kreiraju mape objekata ili delova objekata koji su vidljivi, a potom se vrši njihovo projektovanje na vidnu ravan da bi se odredila boja za svaki piksel pojedinačno.

Poštujući Helmholtcov reciprocitet, može se u tački preseka objekta sa primarnim zrakom izračunati i senčenje objekta, tj. primeniti jednačina lokalnog osvetljenja. Helmholtcov reciprocitet kaže da se percepirane reflektivne osobine objekta ne menjaju ako izvor svetlosti i posmatrač zamene svoje pozicije u sceni. Stoga je moguće izračunati osvetljenje objekta polazeći od zraka pogleda umesto zraka svetlosti. Međutim, kombinujući rejkasting algoritam i lokalne modele osvetljenja moguće je izračunati samo direktnu iluminaciju, bez izračunavanja međurefleksija u sceni.

Dodajmo da kod kreiranja vizualizacija virtualnih scena metodom centralnog projektovanja postoji dodatni problem koji se ogleda u deformaciji objekata koja zavisi od njihovog položaja u vidnom polju u sceni. Naime, centralno projektovanje „izdužuje“ objekte koji se ne nalaze na centralnom pravcu pogleda u sceni (tj. centru mreže piksela u vidnoj ravni), pri čemu je pravac izduživanja kolinearan sa vektorom pomeraja objekta od centralnog pravca pogleda (slika 15). Ovaj problem može se korigovati korišćenjem naprednih modela kamere.

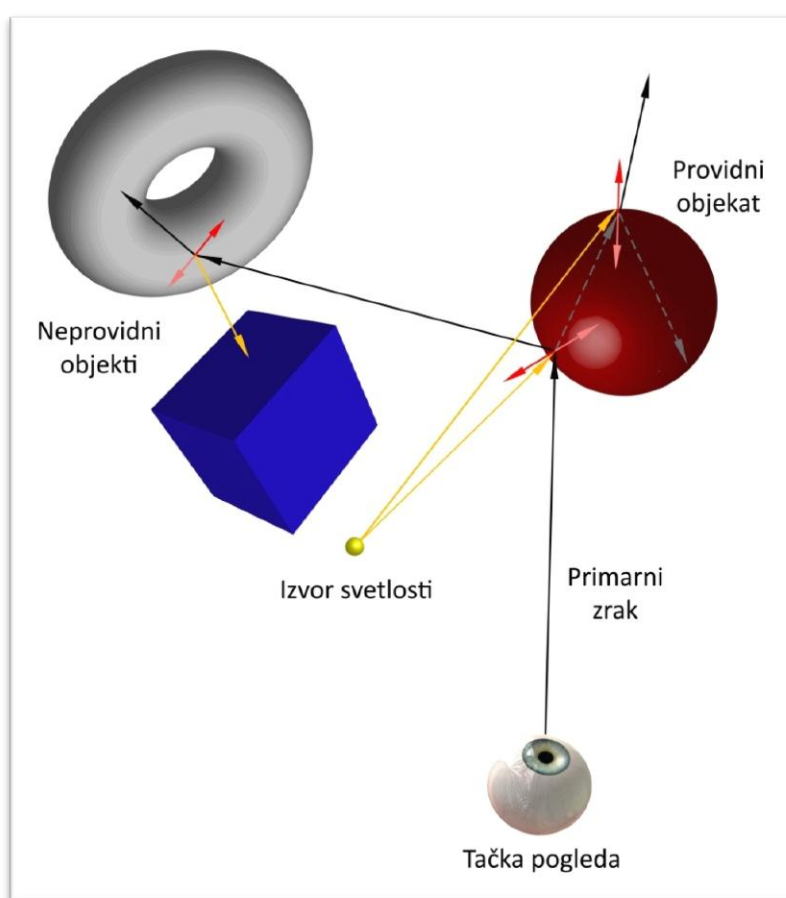


Slika 15: ilustracija izduživanja prilikom centralne projekcije objekta.

S obzirom na to da rejkasting algoritam pruža mogućnosti da se, barem konceptualno, jednostavno opiše proces formiranja slike u sceni, logično je bilo da se on pokuša nadograditi tako što će se u izračunavanje osvetljenja uključiti i indirektna iluminacija. Ta nadogradnja predstavljena je *rekurzivnim rejtrecsing algoritmom*.

## 5.2 Rekurzivni rejtrensing algoritam

Tarner Vitid<sup>1</sup> je 1980-te godine, kao istraživač u *Bel laboratorijama* u Nju Džersiju, objavio rad u kojem se eksplicitno uvodi rekurzivni rejtrensing algoritam, polazeći od postojećeg rejkasting algoritma (Whitted, 1980). Princip funkcionisanja algoritma je sledeći: koristeći rejkasting algoritam pronalazi se tačka preseka primarnog zraka sa najbližim objektom u sceni; potom se izračunavaju direktna iluminacija, koristeći lokalni model osvetljenja, kao i *reflektovani* i *refraktovani*, odnosno prelomljeni zrak svetlosti. Potom se rekurzivno poziva rejkasting algoritam, pri čemu tačka preseka primarnog zraka sa objektom postaje nova tačka pogleda, a reflektovani odnosno prelomljeni zraci su novi primarni zraci (rekurzivno se pozivaju dve instance algoritma). Na kraju se doprinosi direktna iluminacije i iluminacija dobijenih kroz reflektovani i prelomljeni zrak sabiraju, što daje totalnu jačinu svetlosti u datoj tački. Pomenuta procedura se ponavlja za svaki piksel slike posebno (slika 16).



Slika 16: ilustracija rekurzivnog rejtrensing algoritma.

Nakon što se odredi presečna tačka primarnog zraka i objekta najbližeg tački pogleda, određuje se direktna iluminacija, ukoliko zrak svetlosti nije blokiran neprovidnim objektom (žuti vektori na slici). Reflektovani i prelomljeni zraci određuju se u odnosu na vektor normale na površ objekta u tački preseka, predstavljene crvenim vektorima, pri čemu se prelomljeni zrak određuje ukoliko je dati

<sup>1</sup> Tarner Vitid, američki informatičar.

objekat providan. U suprotnom se određuje samo reflektovani zrak. Rekurzija rejtrensing algoritma se prekida ili kada doprinos trenutnog zraka konačnom rešenju postane manji od unapred zadate vrednosti (tj. kada konačni izgled slike prestane da zavisi od daljih rekurzivnih poziva) ili kada se dostigne određena dubina rekurzije. U praksi se pokazuje da se za većinu scena dobri rezultati ostvaruju koristeći dubine rekurzije 2 ili 3. Jednostavna implementacija rekurzivnog rejtrensing algoritma data je na slici 17, u vidu pseudokoda.

Ukoliko se pretpostavi da su BRDF-ovi objekata u sceni predstavljeni Fongovim modelom, tada jednačina globalne iluminacije koja se izračunava rejtrensing algoritmom postaje:

$$I_{\lambda} = k_{\alpha}(I_{a\lambda}k_{\alpha}O_{d\lambda} + \sum_{1 \leq i \leq m} S_i f_{att} I_{\lambda}(k_d O_{d\lambda} \cos \theta + k_s O_{s\lambda} \cos^n \alpha) + I_r) + k_t I_t$$

Jednačina 7: formula globalne iluminacije sa Fongovim BRDF modelom.

U navedenoj jednačini  $k_{\alpha}$  je koeficijent refleksije površi,  $k_t$  koeficijent transmisije,  $I_r$  ukupna jačina reflektovane svetlosti, a  $I_t$  ukupna jačina prelomljene svetlosti. Ostali simboli su identični odgovarajućim simbolima iz jednačine 4. Shodno rekurzivnoj prirodi algoritma, intenziteti reflektovane odnosno refraktovane svetlosti  $I_r$  i  $I_t$  se računaju koristeći navedenu jednačinu.

**Boja** *rejtrensing*( **Zrak** primarni\_zrak )

```
{
    Boja direktna_boja, reflektovana_boja, prelomljena_boja
    Objekat o

    o = nađi najbliži presek( primarni_zrak )
    direktna_boja = nađi direktnu boju( o )

    ako ( o reflektuje svetlost )
        reflektovana_boja = rejtrensing( nađi reflektovani zrak( primarni_zrak, o ) )
    ako ( o propušta svetlost )
        prelomljena_boja = rejtrensing( nađi prelomljeni zrak( primarni_zrak, o ) )

    vrati ( kombinuj boje( direktna_boja, reflektovana_boja, prelomljena_boja ) )
}
```

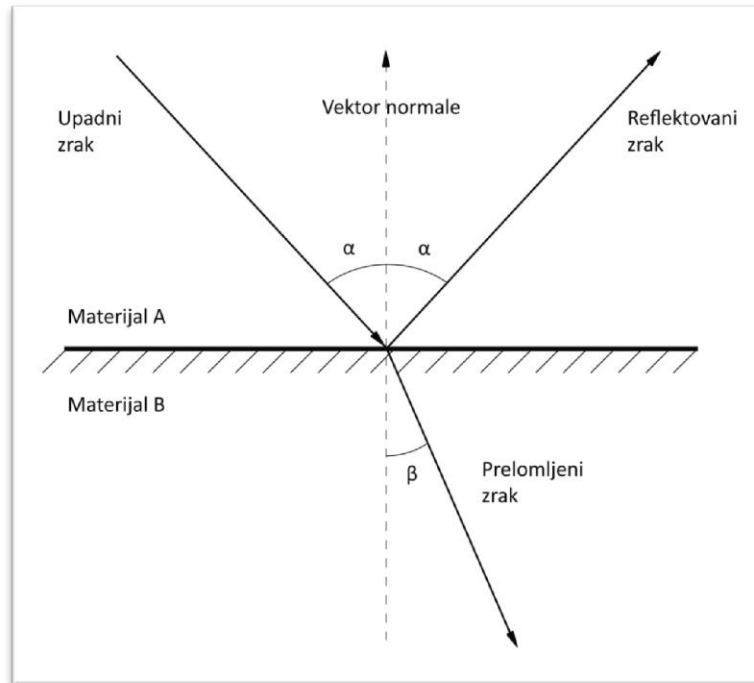
Slika 17: pseudokod rekurzivnog rejtrensing algoritma.

Već je napomenuto da se reflektovani i prelomljeni zrak određuju pomoću vektora normale na površ u tački preseka. Preciznije rečeno, reflektovani zrak se određuje pomoću pravila ogledala, a prelomljeni zrak pomoću *Snelovog*<sup>1</sup> zakona. Pravilo ogledala kaže da će se zrak u sudaru sa objektom odbiti tako da je ugao koji upadni zrak zahvata sa normalom na površ u tački preseka jednak uglu koji reflektovani zrak zahvata sa tom normalom; Snelov zakon prelamanja tvrdi da je odnos sinusa uglova upadnog i prelomljenog zraka, datih u odnosu na normalu na objekat u tački preseka, jednak odnosu *koeficijenata prelamanja*<sup>2</sup> materijala od kojih se sastoje objekat i prostor oko njega (slika 18). Važno

<sup>1</sup> Vilebrord Snel van Rojen, holandski matematičar; 1580 – 1626.

<sup>2</sup> Koeficijent prelamanja svetlosti materijala zavisi od brzine prostiranja svetlosti kroz navedeni materijal.

je napomenuti da će upadni zrak svetlosti, počevši od nekog ugla  $\alpha_{max}$ , biti u potpunosti reflektovan. Ta pojava naziva se *totalna unutrašnja refleksija* i u skladu je sa Snelovim zakonom. Interesanto je primetiti da je zakon refrakcije svetlosti pronađen mnogo vekova pre Snela: Ibn Sal<sup>1</sup> je definisao taj zakon u X veku nove ere, a do istog rešenja su došli i drugi naučnici, poput Dekarta.



Slika 18: ilustracija zakona refleksije i prelamanja svetlosti.

Primetimo da ovako uvedeni rejtrensing algoritam nije u stanju da realistično prikaže difuznu komponentu osvetljenja. S obzirom na to da se prate samo reflektovani i prelomljeni zraci svetlosti, jasno je da je time prioritet dat spekularnoj komponenti osvetljenja. Ukoliko se želi postići veća realističnost krajnjeg rezultata rejtrensing algoritam se mora dopuniti. Stoga se, u skladu sa metodologijom algoritama klase Monte Karlo, u tački preseka primarnog zraka sa objektom definišu i takozvani *zraci senčenja*, tj. bira se nasumično određeni broj zraka koji se prate od tačke preseka do raznih delova scene, koji pripadaju istom poluprostoru kao i primarni zrak (ne uzimaju se u obzir zraci senčenja koji prolaze kroz objekat). Time se pokušava aproksimirati difuzno osvetljenje u sceni. U poglavlju 4.3 diskutovano je o osobinama koje takva distribucija slučajne veličine mora imati da bi rešenje bilo statistički nepristrasno. U praksi se pokazuje da je praćenje dodatnih 1000 zraka obično dovoljno da se dobro aproksimira difuzna komponenta globalne iluminacije.

### 5.3 Tehnike ubrzavanja

Mana rekursivnog rejtrensing algoritma je njegova kompleksnost: za svaki zrak u sceni koji se prati mora se odrediti presek sa objektom koji je najbliži početnoj tački datog zraka. Uzimajući u obzir

<sup>1</sup> Abu Sad al-Ala ibn Sal, arapski matematičar; oko 940 – 1000.

rekurzivnu prirodu algoritma, kao i činjenicu da simuliranje difuzne komponente zahteva kreiranje velikog broja dodatnih zraka, jasno je da ukupan broj zraka koje je potrebno pratiti raste eksponencijalno sa dubinom rekurzije. Za svaki od tako kreiranih zraka mora se odrediti presek sa objektom najbližem početnoj tački datog zraka, kao i senčenje koristeći lokalni model osvetljenja. Stoga su razvijene brojne tehnike čiji je krajnji cilj ubrzavanje izvršavanja algoritma.

### 5.3.1 Paralelizacija rejtrensing algoritma

Razmatrajući način na koji je rejtrensing algoritam definisan može se primetiti da se on može primeniti na svaki piksel slike nezavisno, s obzirom na to da su zruci koje je potrebno pratiti za bilo koja dva piksela slike nezavisni. To je podloga za *paralelizaciju* algoritma, koja je i najčešće korišćen metod ubrzavanja rejtrensing algoritma (Reinhard i Jansen, 1997), (Samardžić, 1998). Pri tome, koriste se tri osnovna platformska rešenja: implementacije za centralne procesore sa više jezgara, implementacije za grafičke procesore i kreiranje specijalnog rejtrensing hardvera.

Najjednostavniji model paralelizacije podrazumeva da su procesorske jedinice organizovane u formi *klastera sa nezavisnim čvorovima*, tj. da svaka jedinica poseduje odvojene resurse neophodne za izračunavanje. Kao najbitniji resurs koji te jedinice poseduju, izdvaja se glavna memorija u kojoj se čuvaju podaci o sceni neophodni za izvršavanje algoritma. Prilikom renderovanja scene na klasteru računara centralni računar snosi odgovornost da rasporedi izvršavanje rejtrensing algoritma čvorovima klastera na optimalan način. Obično se ta raspodela vrši deljenjem mreže piksela u sceni na podmreže manje rezolucije i renderovanje tako pojednostavljene scene na čvorovima. Na kraju centralni računar mora i da sprovede kombinovanje rezultata renderovanja dobijenih od svakog od čvorova u konačnu sliku. U teoriji, ovakav pristup omogućava ubrzavanje izvršavanja rejtrensing algoritma koja su istog reda veličine kao i broj čvorova klastera.

Postoji nekoliko problema prilikom ovakve naivne implementacije paralelizacije. Osnovni problem koji se javlja je mali broj klastera računara čije su individualne procesorske jedinice u potpunosti nezavisne, što kao posledicu ima neophodnost za pažljivijim dizajniranjem načina na koji svaki čvor može da koristi resurse koji su mu na raspolaganju. Pored toga, u određenim situacijama geometrijska kompleksnost scene može biti prostorno neravnomerna i ukoliko se ta činjenica ne uzme u obzir prilikom partitionisanja mreže piksela, dobitak u performansama može biti drastično smanjen – najveći deo posla ponovo obavlja samo deo raspoloživih procesorskih jedinica.

Realističniji scenario od klastera nezavisnih procesorskih jedinica je klaster međusobno zavisnih procesorskih jedinica, kod kojeg je određeni broj raspoloživih procesorskih jedinica primoran da koristi deljene resurse. Primer takvog scenarija su višeprocessorske radne stanice i personalni računari, kao i grafičke kartice novije generacije sa *unificiranim šejderima*<sup>1</sup>. U ovom slučaju velika pažnja se posvećuje korišćenju deljene memorije, s obzirom na to da zruci u rejtrensing algoritmu obično nisu *koherentni*. Drugim rečima, mora se voditi računa o tome koji zruci poseduju slične osobine prilikom prolaska kroz scenu (seku se sa istim objektima, fizički su bliski u sceni, itd). Takođe, model korišćenja zajedničke memorije dodatno se komplikuje ukoliko se uvedu brojne optimizacije

---

<sup>1</sup> Počevši od NVidia GeForce 8 generacije kartica.



na nivou praćenja jednog zraka svetlosti – paralelizacija u tom slučaju postaje daleko kompleksnije pitanje, a ponekad je nemoguće takav algoritam paralelizovati.

Rejtrejsing na centralnim procesorima sa više jezgara postaje sve popularnije rešenje, s obzirom na to da se broj jezgara u standardnim procesorima konstantno povećava (standardno su dostupna četvorोजezgrena rešenja). Osim toga, centralni procesori omogućavaju pojednostavljeno manipulisanje memorijom, kao i brz pristup relativno nekoherentnim podacima, što ih čini izuzetno pogodnim za rejtrejsing. Kompanija Intel vrši aktivno istraživanje mogućnosti korišćenja sopstvenih procesora u svrhe rejtrejsinga i njihovi poslednji rezultati su veoma impresivni – Daniel Pohl i saradnici kreirali su interaktivnu verziju računarske igrice *Enemy Territory: Quake Wars* koja se u potpunosti izvršava na 16 jezgara procesora klase *Nehalem* kompanije Intel. Korišćena je 720p HD rezolucija i postignuto je vreme izvršavanja od 16 slika u sekundi. Intel takođe aktivno radi na kreiranju hibridne procesorske jedinice koja objedinjuje napredni memorijski model i programske instrukcije poput centralnih procesora sa visokim nivoom paralelizacije grafičkih procesora. U pitanju je projekat *Larrabee*, a kao jedna od osnovnih aplikacija tog hibridnog procesora navodi se upravo rejtrejsing algoritam (Seiler et al., 2008).

Grafički procesori su izuzetno uspešni i u takozvanom „lomljenju brojeva“, tj. veoma brzom izračunavanju. Za razliku od centralnih procesorskih jedinica koje su veoma moćne i sposobne da izvrše veoma komplikovane serijalizovane instrukcije, jezgra grafičkih procesora su pojednostavljena i nisu u stanju da efikasno izvršavaju kompleksne instrukcije. Pored toga, u trenutnim verzijama grafičkih procesora ona imaju i limitiran pristup memoriji. Snaga grafičkih procesora međutim leži u paralelizaciji – iako svako od njegovih jezgara pojedinačno nije izuzetno moćno, njihov veliki broj omogućava paralelno izvršavanje velikog broja instrukcija, što ove uređaje čini idealnom platformom za dobro formulisane paralelne algoritme. Rejtrejsing međutim nije jednostavno implementirati na grafičkim karticama, jer on zahteva komplikovan pristup memoriji, kao i dodatne memorijske strukture radi optimizacije izvršavanja. I pored toga brojni istraživači bave se problemom kreiranja rejtrejsing algoritama za grafičke procesore i u tom polju postignuti su dobri rezultati (Purcell, 2004), (NVIDIA, OptiX, 2009). Buduće verzije grafičkih procesora kompanije NVIDIA (sa kodnom oznakom Fermi) posedovaće poboljšanu preciznost izračunavanja u pokretnom zarezu, kao i hijerarhiju keš memorije, poput centralnih procesora, što bi trebalo da olakša kreiranje rejtrejsing aplikacija na grafičkim procesorima, ali i da dodatno ubrza njihovo izvršavanje (NVIDIA, Next Generation CUDA Architecture, 2009).

Pored ovih standardnih računarskih komponenti, tokom vremena bilo je pokušaja da se kreira specijalizovani hardver čija bi osnovna namena bila hardverska podrška za rejtrejsing odgovarajućim specijalnim operacijama. Analizom rejtrejsing algoritma mogu se pronaći osnovne primitive koje su neophodne za njegovo izvršavanje i direktno se implementirati u hardveru. Takvi uređaji bili bi višestruko efikasniji od postojećih rešenja, kada je u pitanju izvršavanje rejtrejsing koda. Međutim, njihova primena bi bila limitirana obzirom na njihovu specijalizaciju – većina drugih izračunavanja bila bi otežana ili nemoguća na takvim arhitekturama. Primer takvog pokušaja je *RPU*<sup>1</sup> arhitektura kreirana na Univerzitetu u Saarlandu (Woop et al., 2005).

---

<sup>1</sup> RPU je skraćenica od *Ray Processing Unit*, tj. jedinica za procesiranje zraka.

### 5.3.2 Hijerarhije graničnih tela

Već je napomenuto da broj zraka koji se kreiraju tokom rada rejtrensing algoritma u tipičnom scenariju može biti veoma veliki: u slučaju kreiranja slika čiji je kvalitet usporediv sa fotografijama broj zraka prelazi i nekoliko milijardi. Iako je paralelizacijom moguće postići odlične rezultate, oni ipak zavise od broja procesorskih jedinica koji su na raspolaganju rejtrensing algoritmu. Kako mreže računara ogromne veličine, koje bi omogućile veoma brzo izvršavanje algoritma, obično nisu dostupne, mora se pristupiti optimizaciji samog algoritma, tj. onih njegovih delova koji oduzimaju najviše procesorskog vremena. Analizom raznih implementacija rejtrensing algoritma došlo se do zaključka da se najveći deo vremena tokom njegovog izvršavanja troši na izračunavanje preseka između zraka svetlosti i objekata u sceni. Logično je dakle da je mnogo truda i vremena uloženo u nalaženje optimalnijih metoda za rešavanje problema preseka zraka sa objektom (Ericson, 2005), (Moore, 1991).

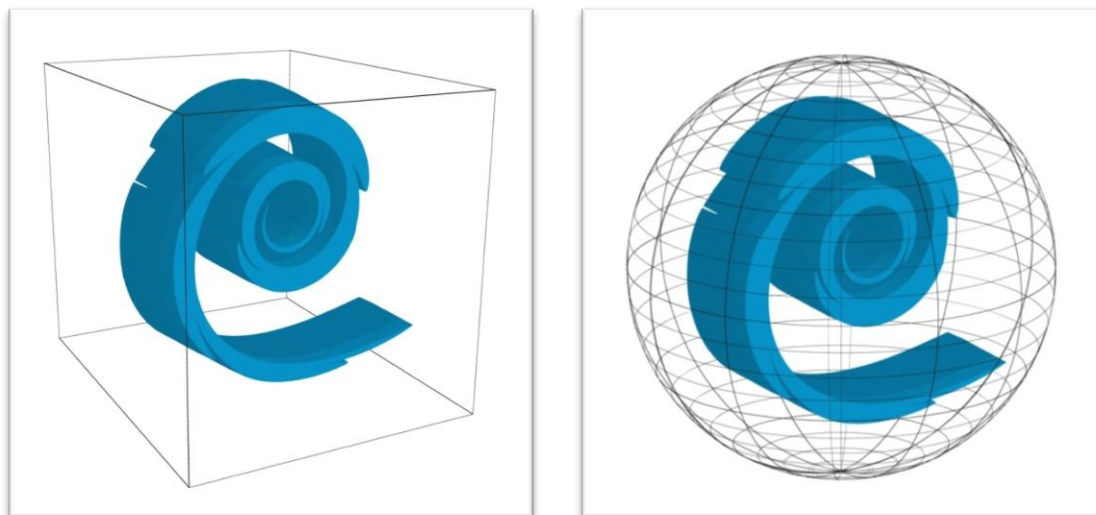
Razmotrimo naivnu implementaciju problema preseka. Za dati primarni zrak i skup objekata u sceni izračunava se presek datog zraka sa svakim od objekata, ukoliko on postoji, i od svih preseka bira onaj koji je najbliži tački pogleda. Pri tome ignoriše se činjenica da su objekti u sceni verovatno raštrkani, te da je verovatnoća da svaki zrak prolazi kroz svaki objekat veoma mala. Pored toga, mnogi objekti u sceni su sastavljeni od kompleksnih mreža poligona i računanje tačke preseka takvih objekata sa zrakom svetlosti nije jednostavno. Kada su takvi objekti u pitanju poželjno je vršiti računanje preseka samo onda kada je vrlo verovatno da taj presek postoji. Da bi se poboljšale performanse algoritma vidljivosti definišu se takozvana *granična tela* i njihove hijerarhije, kao i *kd-stabla*. Njihova osnovna namena je da particionisanjem prostora omogućuje jednostavnije određivanje verovatnoće da dati zrak svetlosti i dati objekat u sceni imaju zajedničkih tačaka.

Granično telo, poznatije i kao *granična površ*, je nevidljivi objekat u sceni u čijoj unutrašnjosti se nalazi jedan ili više vidljivih objekata scene. Kada je granično telo za dati skup objekata definisano, tada se računanje preseka primarnog zraka sa objektima tog skupa može započeti računanjem preseka datog zraka i graničnog tela. Ukoliko presek ne postoji može se izvući zaključak da ne postoji ni presek sa objektima unutar graničnog tela. Ukoliko presek postoji, tada se može pristupiti računanju preseka datog zraka i svih objekata unutar graničnog tela ponaosob. Jasno je da uspeh optimizacije kreiranjem graničnih površi zavisi prevashodno od odnosa kompleksnosti same površi i skupa objekata scene koji se nalaze unutar te površi. Pored toga, granična tela u idealnom slučaju zadovoljavaju i dodatne zahteve:

- Efikasno računanje preseka sa zracima svetlosti
- Minimizovanje praznog prostora između objekta i graničnog tela
- Jednostavno kreiranje samog tela
- Memorijska efikasnost
- Efikasno izračunavanje izometrija tela (za dinamičke scene).

Pošto je većina algoritama trodimenzine računarske grafike bazirana na korišćenju poligonalnih modela geometrije objekata u sceni, kao granična tela najčešće se koriste kvadri, koji predstavljaju veoma jednostavne poliedarske površi. Pored toga, ravni kojima pripadaju strane kvadara kao graničnih tela obično se biraju tako da budu paralelne sa koordinatnim ravnima, što dalje pojednostavljuje detekciju kolizije sa zracima svetlosti, no tada je moguće povećanje praznog

prostora između objekta i tela što dovodi do povećanja broja pogrešnih pozitivnih testova preseka. Ukoliko rejtrensing sistem podržava kreiranje objekata korišćenjem analitičkih formula, onda se kao granično telo može koristiti i sfera, s obzirom na to da se izračunavanje preseka prave i sfere može efikasno izvršiti (slika 19).

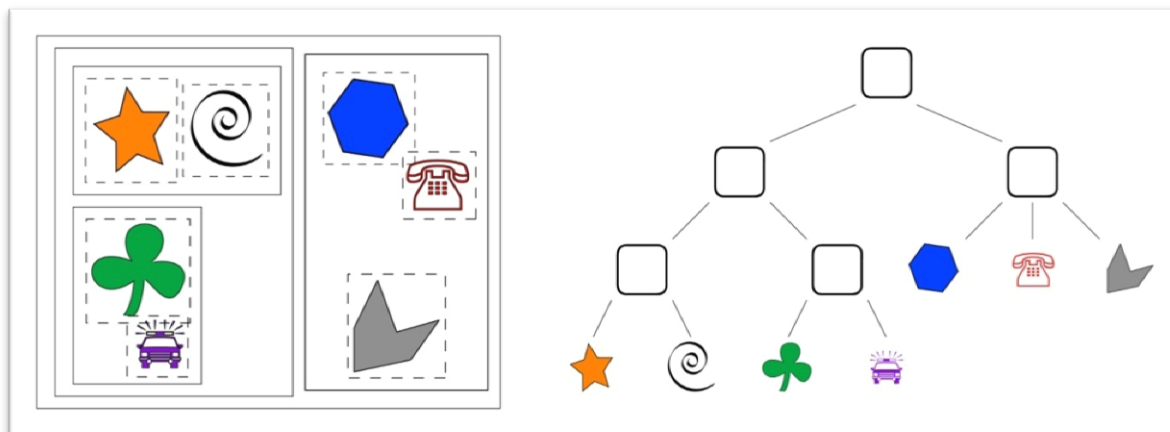


Slika 19: koncept graničnih tela – sferno granično telo je prikazano kao mrežni poligonski model u ilustrativne svrhe.

Razmatranjem prostornog rasporeda objekata u scenama može se doći do zaključka da je veliki broj objekata poziciono grupisan. Takvi objekti se, pored njihovih individualnih graničnih tela, prirodno mogu ograditi i graničnim telom grupe. Sada se proverava presek zraka sa objektima u sceni definiše kao pretraga takve hijerarhije graničnih tela – prvo se testira presek zraka i graničnog tela cele grupe, pa ukoliko taj presek postoji može se preći i na granična tela svakog od objekata (slika 20).

Prilikom dizajniranja hijerarhije graničnih tela neophodno je voditi računa o određenim karakteristikama stabla hijerarhije i njegovih čvorova, u cilju postizanja optimalnih rezultata. Na primer, poželjno je kreirati granična tela minimalne veličine na svakom nivou hijerarhije, kao i minimizovati potencijalne preseke između graničnih tela; više pažnje prilikom kreiranja strategije podele prostora na granična tela treba posvetiti višim čvorovima u hijerarhiji – eliminisanje višeg čvora iz daljeg razmatranja donosi veći benefit pošto se eliminiše veći broj objekata scene. Na posletku, veoma važno je kreirati stablo balansirano u određenom smislu, pošto se kod nebalansiranog stabla sečenje grana ne izvršava optimalno. Primer jednostavnog metoda koji je moguće koristiti je metod *binarnog particionisanja*. Kod ovog metoda se, polazeći od korena stabla (tj. graničnog tela koje sadrži sve objekte scene), podela na čvorove-naslednike pravi tako da je zbir površina<sup>1</sup> objekata unutar njihovih respektivnih graničnih tela jednak. Ponavljajući ovu proceduru za svaki od čvorova naslednika dobijamo stablo koje je balansirano u smislu površine objekata scene.

<sup>1</sup> Razlog zbog kojeg se tretira površina objekata, umesto npr. zapremine, je taj da povećanje površine takođe povećava verovatnoću postojanja preseka objekta sa zrakom svetlosti.



Slika 20: ilustracija hijerarhije graničnih tela u dve dimenzije i odgovarajućeg stabla hijerarhije; čvor stabla predstavlja sve objekte u sceni i ako zrak svetlosti ne seče čvor stabla onda nema preseka ni sa elementima scene. Ukoliko presek postoji tada se proveravaju čvorovi potomci. Na kraju se potencijalno testira presek sa samim objektima scene.

Jedan nedostatak hijerarhija graničnih tela leži u tome što se njima eksplicitno definišu i granice koje se implicitno podrazumevaju – time se komplikuje provera preseka tog segmenta prostora sa datim zrakom, čuva se više informacija nego što je to neophodno i povećava rizik od pogrešnih pozitivnih rezultata testiranja preseka. Stoga se u implementacijama rejtresing algoritma danas najčešće koriste takozvana kd-stabla<sup>1</sup>, koja predstavljaju varijaciju *BSP stabala*. BSP stabla definišu hijerarhiju binarnih particija prostora ravnima, pri čemu je izbor ravni podele proizvoljan. Kod kd-stabala se prostor ne deli na proizvoljne ravni, već se uvek bira neka od ravni paralelna sa koordinatnim osama, u cikličnom maniru – obično se prvo koristi ravan paralelna *x*-osi, zatim *y*-osi, *z*-osi i potom se sekvenca ponavlja. Pored toga, sledeća ravan podele bira se tako da je njena koordinata na osi koja joj je paralelna jednaka srednjoj vrednosti koordinata svih objekata tog segmenta prostora na toj istoj osi. Time se postiže balansiranost stabla.

Pretraga prostora koji je particionisan kd-stablama definiše se veoma jednostavno – prvo se testira presek zraka sa korenom stabla, tj. sa jednom ravni. Ukoliko presek postoji tada se rekurzivno proveravaju potomci tog čvora u stablu, jer je moguće da zrak seče neki od objekata scene u oba segmenta prostora koji je podeljen tim čvorom. Na kraju se, prateći preseke hijerarhije, dolazi do testiranja preseka sa objektima scene koji se nalaze u segmentu prostora definisanim ravnima sa kojima je pronađen presek. Ova pretraga stabla može se izvršiti veoma brzo, međutim problem nastaje kod njihovog kreiranja, s obzirom na to da ono oduzima veliku količinu vremena. Stoga je primena kd-stabala u dinamičkim scenama još uvek predmet rada mnogih istraživača (uopšteno govoreći, još uvek se traga za strukturom podataka koja ubrzava pretragu preseka zraka svetlosti i objekata u sceni, ali omogućava i brzo kreiranje ili ažuriranje u slučaju dinamičkih scena).

<sup>1</sup> kd je skraćenica za „k-dimenziono“.

## 6. Hiperbolički rejtresing

U ovoj glavi biće predstavljena verzija rejtresing algoritma prilagođena Poenkareovom sfernom modelu hiperboličkog prostora. Najpre će biti izloženi ciljevi projekta, a potom prezentovane modifikacije načinjene nad euklidskim rejtresing algoritmom – sam algoritam detaljno je predstavljen u glavi 5. Na kraju će biti razmotren problem preslikavanja hiperboličke ravni na euklidsku ravan, korak koji je neophodan da bi se rezultujuća slika prikazala na ekranu računara.

### 6.1 Postavljeni ciljevi

Motivacija za kreiranje hiperboličkog rejtresing algoritma bila je želja da se definiše novi alat za proučavanje hiperboličkog prostora. U cilju ostvarivanja te ideje kreirana je programska biblioteka koja pruža funkcionalnost neophodnu za kreiranje Poenkareovog sfernog modela na računaru.

Najvažniji uslov koji biblioteka mora da zadovolji je da se sva izračunavanja dosledno sprovede u Poenkareovom modelu i bez geometrijskih aproksimacija. Svi objekti koji su uključeni u definisanje scene moraju biti definisani u Poenkareovom modelu kao i interakcije među objektima. Insistiranje na tom uslovu omogućuje da se svim korišćenim metodama, ali i dobijenim rezultatima, da stroga geometrijska interpretacija.

Biblioteka (HyperGeomLib) je kreirana tako da, kao osnovne operacije, sadrži primitive za kreiranje osnovnih geometrijskih objekata i primitive za određivanje preseka, normala, refleksija i sl. Nakon što je izgrađena bazična funkcionalnost Poenkareovog sfernog modela, nad njom kao osnovom je implementiran hiperbolički rejtresing sistem. U vezi sa tim postavljaju se dva pitanja: kako definisati hiperbolički rejtresing algoritam i na koji način definisati prelaz iz Poenkareovog sfernog modela u euklidski prostor<sup>1</sup>? Uslov koji taj prelaz treba da zadovolji jeste da se on primenjuje kao poslednji korak generisanja vizualizacije 3D scene u Poenkareovom sfernom modelu. Drugim rečima, prelaz iz hiperboličkog u euklidski prostor se odlaže što je više moguće, čime se garantuje da su operacije korišćene tokom izvršavanja rejtresing algoritma korektne u hiperboličkom smislu.

U biblioteci HyperGeomLib se ne koriste hijerarhije graničnih tela niti kd-stabla za ubrzanje izvršavanja programa (poglavlje 5.3). Razlog za donošenje ovakve odluke leži u kompleksnosti Poenkareovog modela hiperboličkog prostora, tj. korišćenja zakrivljenih pravih i površi (u euklidskom smislu) za njegovo definisanje. Pored toga, kao lokalni model refleksije svetlosti koristi se BRDF model, koji ne predstavlja realističan model interakcije svetlosti sa fizičkim objektima (poglavlje 4.1). Ovaj rad nema za cilj kreiranje apsolutnog ekvivalenta klasičnog rejtresing algoritma u hiperboličkom prostoru, već da ilustruje mogućnost njegovog korišćenja kao alata vizualizacije hiperboličke geometrije.

---

<sup>1</sup> Taj prelaz se neminovno mora izvršiti jer se mi nalazimo u prostoru koji se može smatrati euklidskim i uređaji za prikaz slika na računaru su takođe euklidske prirode.

## 6.2 Osnovni elementi hiperboličkog rejtrejsinga

Analizom euklidskog rejtrejsing algoritma uvedenog u poglavlju 5.2 mogu se identifikovati i izdvojiti elementi algoritma koji zahtevaju prilagođavanje hiperboličkom prostoru. Neki elementi rejtrejsing algoritma su nezavisni od geometrije prostora u kojem se algoritam definiše (tj. oni su definisani u terminima apsolutne geometrije) i stoga je prilagođavanje tih elemenata trivijalno. Postoje i elementi koji se menjaju sa promenom geometrijskog prostora, ali koji imaju precizno definisane ekvivalente u drugom prostoru. Najveći problem prilagođavanju algoritma Poenkareovom sfernom modelu predstavlja činjenica da se neki njegovi njegovi elementi oslanjaju na osobine euklidske geometrije za koje u hiperboličkoj geometriji ne postoje odgovarajući ekvivalenti. U tom slučaju neophodno je pronaći reprezentaciju takvih elemenata koja odgovara „hiperboličkom posmatraču“ prisutnom u datoj sceni. U nastavku teksta biće opisane sve uvedene promene pojedinačno i, ukoliko je neophodno, izneti razlozi zbog kojih su one uvedene:

1. Osnovni elementi algoritma opisuju se korišćenjem osnovnih geometrijskih pojmova tačke, prave i ravni. Na primer, virtuelni piksel u vidnoj ravni predstavljen je tačkom, zrak svetlosti predstavljen je pravom (ili polu-pravom), a objekti scene poligonima, koji su segmenti ravni. Prelaz u hiperbolički prostor se trivijalno definiše korišćenjem  $h$ -tačaka,  $h$ -pravih i  $h$ -ravni umesto euklidskih tačaka, pravih i ravni. Time se kao rezultat rada hiperboličkog rejtrejsing algoritma dobija centralna projekcija hiperboličkog prostora na  $h$ -ravan.
2. U poglavlju 5.2 opisan je način određivanja zraka refleksije i prelamanja u odnosu na neki objekat u prostoru, pri čemu su korišćeni pojednostavljeni fizički zakoni refleksije i prelamanja svetlosti. U euklidskom prostoru se ti zakoni svode na razmatranje vektora pravca prostiranja zraka i vektora normale na površ u tački preseka zraka sa datom površi. U hiperboličkom prostoru se ne mogu definisati vektori pravca i stoga se refleksija i prelamanje u Poenkareovom sfernom modelu definišu korišćenjem svojstva konformnosti modela koje je opisano u poglavlju 2.3. U poglavlju 6.3 opisan je metod za računanje zraka refleksije u Poenkareovom sfernom modelu.
3. Intenzitet osvetljenja u sceni zavisi od udaljenosti razmatranog objekta scene do svetlosnog izvora. U poglavlju 4.2 opisano je ponašanje svetlosnog talasa u vakuumu i efekat slabljenja intenziteta svetlosti. U hiperboličkoj geometriji važi isti princip širenja svetlosti u obliku sfernog talasnog fronta. Razlika je u zavisnosti površine talasnog fronta od dužine puta koju je svetlost prešla, tj. od poluprečnika sfere talasnog fronta (u hiperboličkoj metrici). Površina sfere poluprečnika  $r$  se u hiperboličkoj geometriji računa po formuli (Munzner, 1997)

$$P = 4 \pi \sinh^2 r$$

**Jednačina 8: površina sfere u hiperboličkom prostoru.**

Odatle se koeficijent slabljenja intenziteta svetlosti može izračunati kombinovanjem jednačine 5 i jednačine 8:

$$f_{att} = \frac{1}{4\pi \sinh^2 r}$$

Jednačina 9: formula slabljenja intenziteta svetlosti u hiperboličkom prostoru.

U praksi, poluprečnik sfernog talasnog fronta nije dat eksplicitno, već implicitno kao rastojanje od tačke izvora svetlosti do tačke preseka zraka svetlost i objekta u sceni. Da bi se izračunala eksplicitna vrednost prečnika koristi se metrika data jednačinom 1.

- Virtualni pikseli u vidnoj ravni euklidskog rejtrensing algoritma formiraju pravougaonu mrežu. Problem definisanja adekvatne mreže piksela u vidnoj  $h$ -ravni je istovremeno i najteži problem za rešavanje prilikom kreiranja hiperboličkog rejtrensing algoritma. Pikseli u vidnoj ravni euklidskog rejtrensing algoritma pripadaju dvama pramenovima paralelnih pravih koji su međusobno ortogonalni. Međutim, u hiperboličkom prostoru takav pramen ne može se konstruisati, tj. ne može se konstruisati pravougaonik, jer je zbir uglova četrougla strogo manji od 360 stepeni. U poglavlju 6.4 biće izložena moguća rešenja ovog problema.

### 6.3 Osnovne geometrijske primitive

Osnovni problem koji se javlja prilikom kreiranja geometrijskih vizualizacija je neophodnost kreiranja osnovnih primitiva koje omogućavaju opisivanje datog prostora na prirodan geometrijski način. U ovom poglavlju su definisane operacije nad osnovnim elementima Poenkareovog sfernog modela,  $h$ -tačkama,  $h$ -pravama i  $h$ -ravnima. Pored toga, opisane su i primitive za izvršavanje rejtrensing algoritma, koje se prevashodno odnose na kreiranje kompleksnijih objekata u modelu, njihove transformacije i izračunavanje preseka  $h$ -pravih i objekata u modelu.

**$H$ -refleksija  $h$ -tačke u odnosu na  $h$ -ravan** – neka su date  $h$ -ravan  $\alpha$  i  $h$ -tačka  $A$  koja joj ne pripada. Ukoliko je  $h$ -ravan  $\alpha$  deo euklidske ravni tada se  $h$ -refleksija  $h$ -tačke  $A$  u odnosu na  $\alpha$  definiše pomoću odgovarajuće euklidske refleksije tačke u odnosu na euklidsku ravan. U suprotnom  $h$ -ravan  $\alpha$  je deo euklidske sfere  $s$  poluprečnika  $r$  čiji je centar tačka  $C$ . Euklidske koordinate  $h$ -tačke  $B$ , određene  $h$ -refleksijom  $h$ -tačke  $A$  u odnosu na  $h$ -ravan  $\alpha$ , dobijaju se inverzijom euklidske tačke  $A$  u odnosu na sferu  $s$ . Drugim rečima, euklidska tačka  $B$  zadovoljava sledeće jednačine:

$$\vec{B} = \vec{A} + t(\vec{C} - \vec{A}),$$

$$|\vec{CA}||\vec{CB}| = r^2.$$

Rešavanjem sistema jednapo parametru  $t$  dobijaju se koordinate tražene  $h$ -tačke.

**Simetralna  $h$ -ravan  $h$ -duži** – neka je  $h$ -duž određena dvema  $h$ -tačkama  $A$  i  $B$  sa euklidskim koordinatama  $(a_x, a_y, a_z)$  i  $(b_x, b_y, b_z)$ . Ukoliko je euklidsko rastojanje tačaka  $A$  i  $B$  od središta apsolute jednako, tada je simetralna  $h$ -ravan date  $h$ -duži deo euklidske ravni koja sadrži središte apsolute, čiji je vektor normale  $\vec{n}$  određen sa  $(b_x - a_x, b_y - a_y, b_z - a_z)$ . U suprotnom, tražena  $h$ -ravan je deo sfere koja je upravna na apsolutu i čiji centar se nalazi na pravoj  $AB$  (jer se inverzijom u

odnosu na traženu sferu tačka  $A$  preslikava u tačku  $B$ ). Prava  $AB$  može se predstaviti parametarskom jednačinom

$$\frac{x - a_x}{n_x} = \frac{x - a_y}{n_y} = \frac{x - a_z}{n_z} (= t),$$

pri čemu su  $n_x, n_y$  i  $n_z$  koordinate vektora  $\vec{n}$ , a  $t$  parametar koji je potrebno izračunati. Ako je centar tražene sfere tačka  $S$  sa koordinatama  $(s_x, s_y, s_z)$ , a poluprečnik sfere  $r$ , tada je uslov ortogonalnosti sfere i apsolute dat jednačinom

$$1 + r^2 = \sqrt{s_x^2 + s_y^2 + s_z^2}.$$

Kombinovanjem prethodne dve formule dobija se tražena vrednost parametra  $t$ :

$$t = \frac{1 - \sqrt{a_x^2 + a_y^2 + a_z^2}}{n_x(a_x + b_x) + n_y(a_y + b_y) + n_z(a_z + b_z)}.$$

Zamenom izračunate vrednosti parametra  $t$  u jednačinu prave dobijaju se koordinate tačke  $S$  kojom je jednoznačno određena tražena sfera.

**H-ravan ortogonalna na h-duž u temenu date h-duži** – neka je, kao i u prethodnom razmatranju,  $h$ -duž data  $h$ -tačkama  $A$  i  $B$ , čije su euklidske koordinate  $(a_x, a_y, a_z)$  i  $(b_x, b_y, b_z)$  (može se razmatrati i  $h$ -poluprava sa temenom  $A$  i jednom  $h$ -tačkom  $B$ ). Razlikuju se dva slučaja: data  $h$ -duž je u jednom slučaju segment euklidske prave, a u drugom segment euklidskog kruga. Razmotrimo slučaj  $h$ -duži koja je segment euklidske prave

Ukoliko je  $h$ -tačka  $A$  ujedno i središte apsolute, tada je tražena  $h$ -ravan deo euklidske ravni koja sadrži središte apsolute i čiji je vektor normale jednak vektoru položaja  $h$ -tačke  $B$  u modelu:  $\vec{n} = (b_x, b_y, b_z)$ . Ako  $h$ -tačka  $A$  nije središte apsolute, tada tražena  $h$ -ravan pripada euklidskoj sferi ortogonalnoj na apsolutu, čiji centar  $S$  pripada pravoj  $AB$ . Vektor položaja tačke  $S$  može se parametarski predstaviti pomoću vektora položaja  $h$ -tačke  $A$  kao  $\vec{S} = t\vec{A}$ . Sada se iz uslova ortogonalnosti tražene sfere i apsolute dobija jednačina po nepoznatom parametru  $t$ :

$$t = \frac{1 + |\vec{A}|^2}{2|\vec{A}|^2}.$$

Zamenom parametra  $t$  u jednačini kojom je izražen vektor položaja centra tražene sfere dobija se tačka  $S$ .

**Presek h-poluprave i h-ravni** – kako  $h$ -poluprava može biti segment euklidske prave ili kruga, a  $h$ -ravan deo euklidske ravni ili sfere, računanje traženog preseka deli se na četiri slučaja. Ovde će biti prezentovane formule za slučaj  $h$ -poluprave koja je segment kruga i  $h$ -ravni koja je deo euklidske ravni.

Data  $h$ -poluprava zadata je dvema  $h$ -tačkama  $A$  i  $B$ , pri čemu je  $h$ -tačka  $A$  teme  $h$ -poluprave, i određena je presekom dve  $h$ -ravni: jedne koja je deo euklidske ravni  $\alpha$  čiji je vektor normale  $\vec{n}$



određen koordinatama  $(n_x, n_y, n_z)$  i druge koja je deo sfere  $\rho$  poluprečnika  $r$  čiji centar  $S$  pripada ravni  $\alpha$  i određen je koordinatama  $(s_x, s_y, s_z)$ . Data  $h$ -ravan je deo euklidske ravni  $\beta$  i određena je vektorom normale  $\vec{m}$  čije su koordinate  $(m_x, m_y, m_z)$ . Tražena  $h$ -tačka nalazi se u preseku ravni  $\alpha, \beta$  i sfere  $\rho$ . Odredimo prvo presečnu pravu ravni  $\alpha$  i  $\beta$ . Vektor pravca  $\vec{k}$  date prave određen je vektorskim proizvodom vektora  $\vec{n}$  i  $\vec{m}$  i njegove koordinate su  $(m_y n_z - m_z n_y, m_z n_x - m_x n_z, m_x n_y - m_y n_x)$ . Kako ravni  $\alpha$  i  $\beta$  sadrže središte apsolute, presečna prava takođe sadrži središte apsolute. Sada je tražena  $h$ -tačka presek sfere  $\rho$  i dobijene presečne prave. Parametarska jednačina presečne prave je:

$$\frac{x}{k_x} = \frac{y}{k_y} = \frac{z}{k_z} (= t).$$

Ako se gornja jednačina po parametru  $t$  uvrsti u jednačinu sfere  $\rho$  dobija se kvadratna jednačina po parametru  $t$  koja može imati nijedno, jedno ili dva rešenja, u zavisnosti od vrednosti diskriminante jednačine:

$$D = (k_x s_x + k_y s_y + k_z s_z)^2 - |\vec{k}| \left( \sqrt{s_x^2 + s_y^2 + s_z^2} - r^2 \right).$$

Ako je  $D \leq 0$  tada nema rešenja, odnosno  $h$ -prava i  $h$ -ravan se ne seku<sup>1</sup>. Ako je  $D > 0$  tada se jedna vrednost parametra  $t$  računa po formuli:

$$t = \frac{k_x s_x + k_y s_y + k_z s_z + \sqrt{D}}{|\vec{k}|}.$$

Uvrštavanjem vrednosti  $t$  u jednačinu presečne prave dobijaju se koordinate presečne tačke  $K$ . Ukoliko je rastojanje date tačke od središta apsolute manje od 1 tada je to tražena presečna  $h$ -tačka. U suprotnom se  $t$  računa po formuli:

$$t = \frac{k_x s_x + k_y s_y + k_z s_z - \sqrt{D}}{|\vec{k}|}.$$

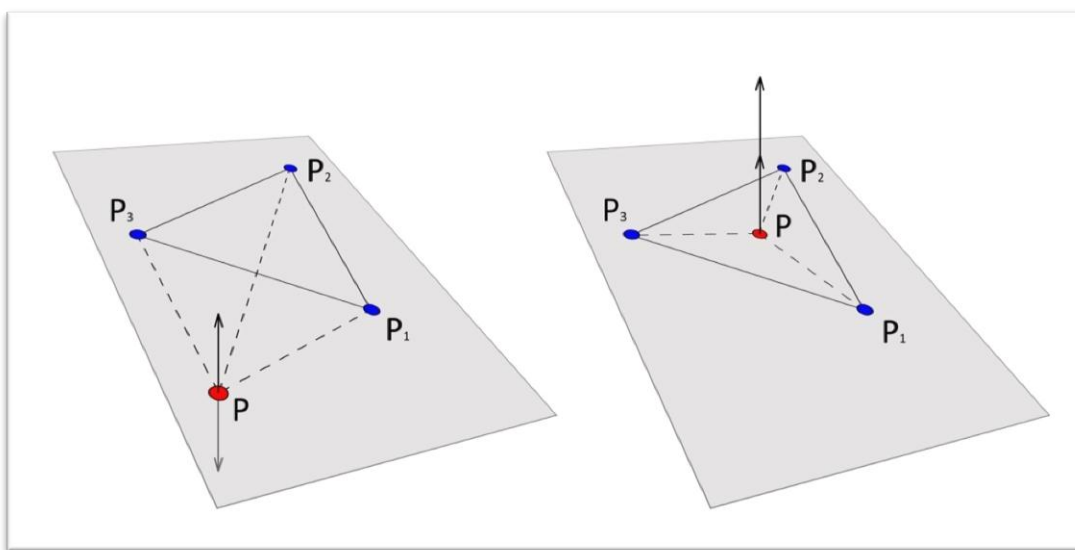
Uvrštanjem parametra  $t$  u jednačinu presečne prave dobijaju se koordinate tačke  $K$  čije je rastojanje od središta apsolute sigurno manje od 1, tj.  $K$  je tražena  $h$ -tačka. Ostaje još da se proveriti da li je dobijena  $h$ -tačka sa iste strane temena date  $h$ -poluprave kao i  $h$ -tačka  $B$ , koja je element date  $h$ -poluprave. Ukoliko je  $K = A$  ili  $K = B$  tada je  $K$  tražena presečna  $h$ -tačka. U suprotnom,  $h$ -tačke  $A, B$  i  $K$  obrazuju trougao u prostoru, a jedini raspored temena trougla koji ne zadovoljava navedeni uslov je raspored u kojem je najduža stranica trougla stranica  $BK$ . U preostala dva slučaja može se konstatovati da je  $h$ -tačka  $K$  traženi presek date  $h$ -poluprave i  $h$ -ravni.

**Ispitivanje pripadnosti  $h$ -tačke unutrašnjosti  $h$ -poligona** – pretpostavimo da  $h$ -tačka  $P$  pripada  $h$ -ravni  $h$ -poligona čija su temena  $P_1, P_2, \dots, P_n$ <sup>2</sup>. Ukoliko  $h$ -tačka  $P$  nije središte apsolute, tada se sva temena  $h$ -poligona preslikavaju  $h$ -refleksijom u odnosu na  $h$ -ravan kojom se  $P$  preslikava u središte apsolute. Sada se za tako dobijeni niz temena  $h$ -poligona vrši provera da li se središte apsolute nalazi

<sup>1</sup> Ako je  $D = 0$  tada postoji tačno jedno rešenje kvadratne jednačine, no njime je određena tačka na apsoluti koja ne pripada datom  $h$ -prostoru.

<sup>2</sup> Definisani su samo prosti konveksni planarni  $h$ -poligoni, poglavlje 7.1.2.

unutar tog  $h$ -poligona. Za taj slučaj razvijena je varijanta algoritma *broja navoja* koji, uslovno govoreći, broji koliko puta kriva rotira oko kandidat  $h$ -tačke. Ukoliko je taj broj 0, tada je  $h$ -tačka van date krive. U suprotnom  $h$ -tačka pripada unutrašnjosti krive. U slučaju  $h$ -poligona proverava se svodi na testiranje da li vektorski proizvod vektora položaja dvaju uzastopnih transformisanih temena  $h$ -poligona menja svoj smer, pri čemu se temena  $h$ -poligona obilaze u jednom unapred izabranom smeru (slika 21). Promena smera vektora jednaka je promeni znaka njegovih koordinata, što se jednostavno proverava. Stoga je proveru promene smera vektora dovoljno izvršiti samo za jednu proizvoljno izabranu koordinatu datih vektora (određenosti radi, pretpostavimo da se radi o  $x$  koordinati unutar Poenkareovog modela). U slučaju da ravan  $h$ -reflektovanog  $h$ -poligona sadrži  $x$  osu koordinatnog sistema i svi vektorski proizvodi imaju kao svoju  $x$  koordinatu vrednost 0, tada se može probati sa  $y$  koordinatom. Ako su sve  $y$  koordinate takođe 0 tada se prelazi na  $z$  koordinatu. Na taj način se izbegava potreba da se vektorski proizvod računa u potpunosti.



Slika 21: ilustracija algoritma za proveru da li tačka pripada unutrašnjosti poligona. U prvom slučaju vektorski proizvodi  $\vec{PP_2} \times \vec{PP_3}$  i  $\vec{PP_3} \times \vec{PP_1}$  su suprotnog smera. Na slici desno svi vektorski proizvodi su istog smera.

## 6.4 Mreža piksela i tipovi projektovanja

U ovom poglavlju biće razmotreni detalji problema projektovanja objekata  $h$ -prostora na  $h$ -ravan, mreže piksela u hiperboličkom prostoru kao i problem predstavljanja mreže piksela u euklidskoj ravni u cilju njenog prikaza na ekranu računara. Sama mreža piksela je u potpunosti određena svojim tipom, dimenzijama i rezolucijom (tj. brojem piksela po jedinici dužine ili površine). S obzirom na to da tip projektovanja mreže piksela iz hiperboličkog prostora u euklidski direktno zavisi od tipa mreže, tipovi mreža biće identifikovani pomoću njima odgovarajućih tipova projekcija. Ukoliko nije drugačije naglašeno, podrazumevaće se da se tačka pogleda nalazi u središtu apsolute modela. Razlog za ovu pretpostavku leži u činjenici da su prave koje prolaze kroz središte apsolute euklidske duži (poglavlje 2.3), što uveliko olakšava manipulisanje mrežom piksela.

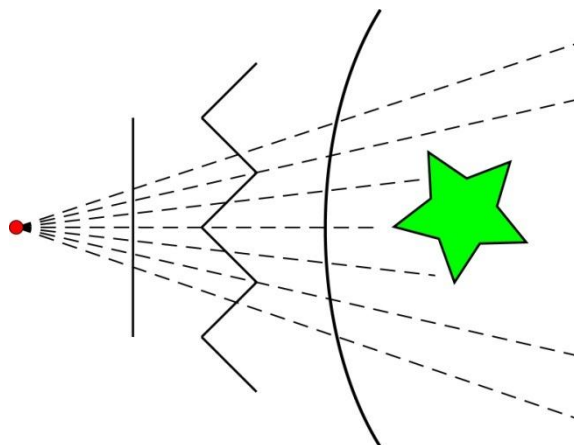
Dati problem projekcije mreže piksela može se razmatrati sa dva međusobno nezavisna stanovišta. Prvi cilj je kreiranje vizualizacije koja će posmatraču verno predstaviti izgled scene, tj. da izgled slike

na ekranu u potpunosti odgovara onome što bi posmatrač mogao videti da se nalazi u hiperboličkom prostoru i posmatra zadatu scenu (drugim rečima cilj je što realističniji prikaz). S druge strane, pošto je jedna od osnovnih namena ovog projekta primena u edukativne svrhe, drugi cilj je da se struktura prostora što deskriptivnije prikaže korisniku (što ne mora istovremeno da podrazumeva i realističnost prikaza). Dobar primer takvog prikaza je *ortografska projekcija* u geometriji, koja odstupa od korektna perspektivne projekcije, ali u određenim situacijama pruža jasniju geometrijsku sliku objekata u sceni.

#### 6.4.1 Projektovanje objekata $h$ -prostora na $h$ -ravan

U poglavlju 5.1 definisani su *object-precision* algoritmi vidljivosti. Kod tih algoritama se vidljivost objekata ne određuje za svaki piksel slike u vidnoj ravni posebno, već se kompletna trodimenziona scena projektuje na vidnu ravan analiziranjem odnosa između celih objekata ili njihovih delova. Zatim se u vidnoj ravni može uzorkovanjem odrediti za svaki piksel slike koji je najbliži projektovani objekat. S obzirom na to da ovaj princip ne zavisi od toga u kojoj geometriji se vrši projektovanje, on se može primeniti i na hiperboličku geometriju. U euklidskoj geometriji se projektovanje objekata prostora na ravan, posmatrano iz date tačke pogleda koja ne pripada datoj ravni, vrši određivanjem geometrijskog mesta tačaka preseka date ravni i svih polupravih čije je teme data tačka pogleda i koje seku neki od objekata prostora. Ukoliko se radi o hiperboličkoj geometriji, tada se opis projekcije dobija prostom zamenom reči prostor, ravan, prava i tačka rečima  $h$ -prostor,  $h$ -ravan,  $h$ -prava i  $h$ -tačka. Drugim rečima, iz date  $h$ -tačke pogleda se  $h$ -prostor projektuje na datu  $h$ -ravan pomoću preseka svih  $h$ -polupravih sa temenom u datoj  $h$ -tački koje seku objekte  $h$ -prostora.

U specifičnom slučaju Poenkareovog sfernog modela, projekcija prostora na  $h$ -ravan, posmatrano iz središta apsolute, svodi se na projekciju unutrašnjosti apsolute na deo euklidske sfere kojoj data vidna  $h$ -ravan pripada. Međutim, sa stanovišta posmatrača koji se nalazi u središtu apsolute, sasvim je svejedno da li se prostor projektuje na  $h$ -ravan ili na neku drugu neprekidnu površ koja seče svaki od zraka vidljivosti (slika 22). Ovo proizilazi iz činjenice da posmatrač koji scenu gleda iz samo jedne tačke pogleda nije u stanju da odredi rastojanje vidne površi od tačke pogleda, već samo da li se u specifičnom pravcu pogleda nalazi presek zraka definisanog tim pravcem i nekog objekta scene.



Slika 22: ilustracija projekcije 2D objekta na jednodimenzionu liniju - posmatrač u sva tri slučaja vidi istu sliku.

Razmotrimo proizvoljnu  $h$ -pravu u modelu koja ne sadrži središte apsolute, tj.  $h$ -pravu koja je predstavljena segmentom euklidskog kruga. Data  $h$ -prava je jednoznačno određena pomoću bilo koje dve razne  $h$ -tačke koje ta  $h$ -prava sadrži. Međutim, ona se može zadati i kao presek dve proizvoljne  $h$ -ravni koje sadrže date  $h$ -tačke. Ukoliko jedna takva  $h$ -ravan sadrži središte apsolute, tada je ona deo euklidske ravni, tj. i sama  $h$ -prava pripada toj istoj euklidskoj ravni. S obzirom na to da i tačka pogleda, koja se nalazi u središtu apsolute, pripada toj euklidskoj ravni, sledi da je projekcija  $h$ -prave na euklidsku vidnu ravan uronjenu u model, posmatrano iz središta apsolute, deo euklidske prave. Drugim rečima, datu  $h$ -pravu posmatrač lociran u središtu apsolute doživeće kao euklidsku duž. Na sličan način, s obzirom na to da projekcija ne zavisi od površi na koju se projektuje, data  $h$ -duž projektovana na vidnu  $h$ -ravan će posmatraču lociranom u središtu apsolute takođe izgledati kao deo euklidske prave.

#### 6.4.2 Mreže piksela u Poenkareovom sfernom modelu

U prethodnom poglavlju dat je opis centralnog projektovanja  $h$ -prostora na  $h$ -ravan, što predstavlja krajnji rezultat hiperboličkog rejtrensing algoritma. Da bi se kreirala vizualizacija dobijenog rezultata neophodno je dobijenu  $h$ -ravan prikazati na ekranu računara. To je moguće učiniti na dva principijelno različita načina: dobijena  $h$ -ravan se može u određenom smislu direktno prikazati na ekranu računara (u duhu euklidskog rejtrensing algoritma), ili se dobijena  $h$ -ravan može  $h$ -izometrijom preslikati u  $h$ -ravan koja pripada velikom krugu apsolute<sup>1</sup> i prikazati na ekranu računara. U oba slučaja neophodno je definisati odgovarajuću mrežu piksela u modelu koji će biti prikazani na ekranu računara. Opišimo detaljnije navedene pristupe.

**Direktan prikaz vidne  $h$ -ravni** – ovaj pristup prikazu vidne  $h$ -ravni zasniva se na ideji da se svakom pikselu mreže tačaka definisane u toj  $h$ -ravni pridruži bijekcijom tačno jedan piksel rezultujuće slike koja se prikazuje na ekranu računara. Intuitivno ovakav prikaz predstavlja „uranjanje“ ravni ekrana u model i centralnu projekciju vidne  $h$ -ravni na ekran. Ovim prikazom definisane su sledeće mreže tačaka u vidnoj  $h$ -ravni:

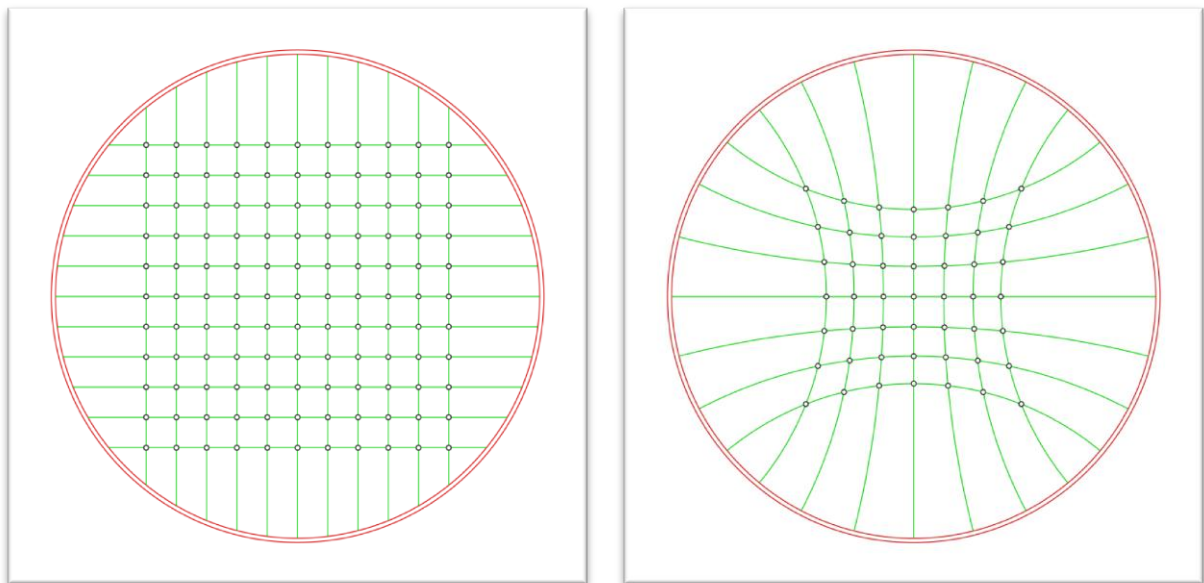
- **Euklidska mreža** – ovaj tip mreže odstupa od principa koji su izloženi u poglavlju 6.1, s obzirom na to da se projekcija vrši na površ koja nema jednostavnu geometrijsku interpretaciju u modelu. Pored toga, za izračunavanje rastojanja između piksela u vidnoj ravni umesto hiperboličke metrike koristi se euklidska metrika. Pikseli u vidnoj ravni su raspoređeni na identičan način kao i u standardnim implementacijama rejtrensing algoritma. Drugim rečima, projekcija na vidnu ravan koja koristi ovu mrežu piksela predstavlja klasično euklidsko projektovanje „uronjeno“ u Poenkareov sferni model (slika 23). Mrežu piksela kreiramo u ravni kojoj pripada veliki krug apsolute i potom euklidskom izometrijom transformišemo u željenu poziciju u sceni. Napomenimo da se ovom mrežom mogu definisati i pikseli koji ne pripadaju modelu, tj. pikseli koji se nalaze izvan apsolute, čime je omogućeno prikazivanje čitave vidne  $h$ -ravni na ekranu računara. Ovi pikselima se automatski dodeljuje crna boja.

---

<sup>1</sup> Primitimo da je takva  $h$ -ravan nužno deo euklidske ravni u modelu i predstavlja Poenkareov disk model hiperboličke geometrije.

- **Hiperbolička mreža hiperparalelnih  $h$ -pravih** – ovaj tip mreže poštuje principe navedene u poglavlju 6.1. Projekcija se vrši na mrežu piksela koja je definisana u  $h$ -ravni. Sama mreža dobija se kao presek dva pramena  $h$ -ekvidistantnih  $h$ -pravih ortogonalnih na dve međusobno ortogonalne  $h$ -prave. Pikseli se prvo definišu u  $xOy$  ravni Poenkareovog sfernog modela (tj. u euklidskoj ravni), a zatim se odgovarajućom  $h$ -translacijom mreža piksela postavlja u željenu vidnu  $h$ -ravan. U ravni  $xOy$  navedeni pramenovi  $h$ -pravih definišu se kao pramenovi ortogonalni na  $h$ -prave koje su delovi euklidskih pravih kojima pripadaju ose  $x$  i  $y$  koordinatnog sistema (slika 23).

U euklidskoj geometriji mreža uvedena na gore navedeni način bila bi nužno i pravougaona ekvidistantna, tj. bila bi ekvivalentna mreži tačaka koja se standardno koristi u euklidskom rejtrensing algoritmu – u hiperboličkoj geometriji to nije slučaj. Primetimo da mreža definisana na ovaj način ne može na prirodan način pokriti čitavu  $h$ -ravan kojoj pripada, s obzirom na to da se  $h$ -prave pramenova kojima je mreža definisana neće seći počev od neke  $h$ -prave (neformalno – kada se te  $h$ -prave približe apsoluti). Ovakva mreža se na ekran računara može preslikati bijektivno, tj. tako da se  $h$ -tačka mreže i piksel slike sa koordinatama<sup>1</sup>  $(i, j)$  preslikavaju jedno u drugo.



Slika 23: ilustracija euklidske mreže i hiperboličke mreže piksela u  $h$ -ravni.

- **Euklidska pravougaona neekvidistantna mreža** – kako se, posmatrano iz središta apsolute, sve  $h$ -prave u modelu vide kao delovi euklidske prave (poglavlje 6.4.1), bez obzira na tip projekcije površi koju koristimo, sledi zaključak da se može koristiti mreža tačaka u vidnoj ravni koja je deo euklidske ravni uronjene u model. Iako ovaj pristup odstupa od principa iznetih u poglavlju 6.1, s obzirom na to da se kao projektivna površ ne koristi  $h$ -ravan, rezultat koji bi se dobili korišćenjem bilo koje od te dve površi je isti. Stoga je ispunjen

<sup>1</sup> U oba slučaja radi se o koordinatama unutar mreže, a ne o geometrijskim koordinatama u modelu ili na ekranu.

najvažniji uslov koji kreirana mreža mora da zadovolji, a to je izračunavanje bez geometrijskih aproksimacija.

Imajući u vidu osobine hiperboličkog i euklidskog centralnog projektovanja iznete u prethodnim odeljcima, može se zaključiti da se hiperboličko centralno projektovanje na  $h$ -ravan može predstaviti odgovarajućim modifikovanim euklidskim centralnim projektovanjem. Dobijena euklidska pravougaona mreža tačaka se od odgovarajuće euklidske ekvidistantne pravougaone mreže razlikuje po funkciji kojom se definiše ekvidistantnost: u jednom slučaju koristi se standardna metrika euklidskog prostora; u drugom slučaju traženu funkciju nije jednostavno iskazati analitičkom formulom<sup>1</sup>. Stoga ovaj tip mreže nije implementiran već se koristi kao ilustracija mogućnosti da je, koristeći mrežu piksela definisanu unutar modela ili mrežu koja u modelu nema jasnu interpretaciju, moguće dobiti identičan konačni izgled scene.

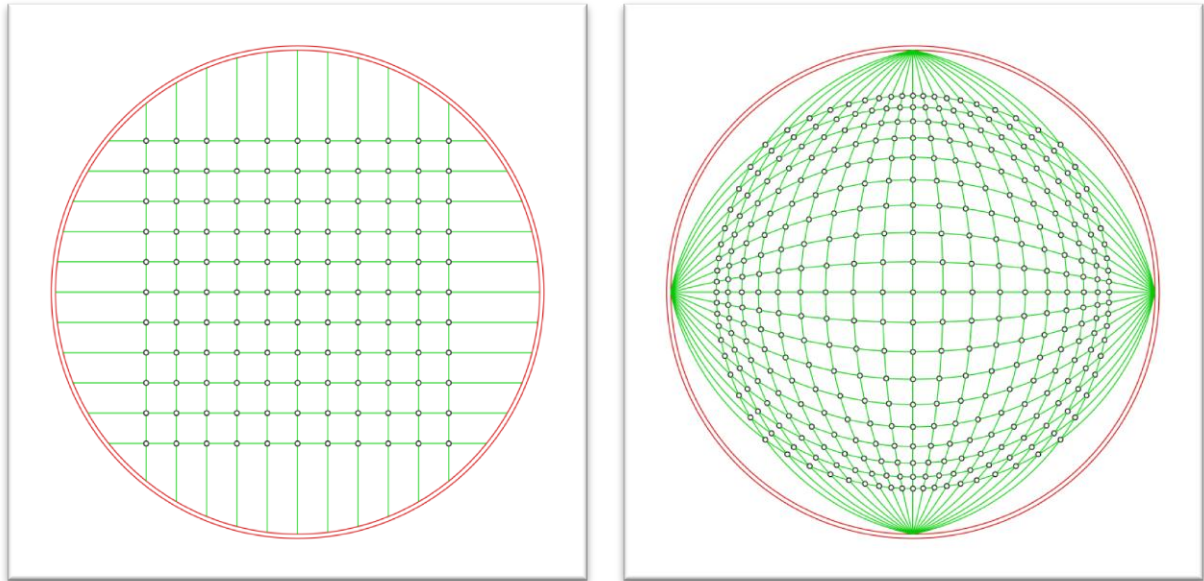
**Prikaz projekcijom na Poenkareov disk model** –  $h$ -izometrijom se vidna  $h$ -ravan može preslikati na veliki krug apsolute, tj. vidna ravan se preslikava u Poenkareov disk model (poglavlje 2.2). Poenkareov disk model defisan je u euklidskoj ravni i može se bez daljih projekcija prikazati na ekranu računara. Odgovarajuće mreže tačaka kojima se može kreirati prikaz projekcijom na disk model su:

- **Euklidska mreža velikog  $h$ -kruga apsolute** – ovaj tip mreže predstavlja najprirodniji način uzorkovanja vidne  $h$ -ravni projektovane na veliki krug apsolute (u euklidskom smislu): mreža se kreira u  $h$ -ravni koja je deo euklidske ravni i potom se  $h$ -izometrijom ta  $h$ -ravan preslikava u željenu poziciju u sceni. Nakon završetka rada hiperboličkog rejtrensing algoritma takva mreža može se direktno prikazati na ekranu računara. Primetimo da, slično kao i u slučaju euklidske pravougaone ekvidistantne mreže, ova mreža može sadržati piksele koji se nalaze van apsolute, te se ti pikseli ne uzimaju u razmatranje tokom rada hiperboličkog rejtrensing algoritma.
- **Hiperbolička mreža  $h$ -ekvidistantnih krivih** – ovaj tip mreže inspirisan je definicijom  $h$ -cilindra u hiperboličkoj ravni datoj u poglavlju 2.2. Pošto pramen hiperparalelnih  $h$ -pravih nije istovremeno i pramen  $h$ -ekvidistantnih pravih, ukoliko se želi održati pojam  $h$ -ekvidistantnosti čvorova u mreži mora se posegnuti za drugačijim rešenjem. Ovaj tip mreže kreira se pomoću dva skupa  $h$ -ekvidistanti, ortogonalnih na ose  $x$  i  $y$  koordinatnog sistema u kojem je definisan Poenkareov sferni model (slika 24). Ovako kreirana mreža ima neke pogodne osobine. Pre svega, njome je moguće pokriti čitavu vidnu  $h$ -ravan, s obzirom na to da se svake dve krive koje pripadaju datim dvama skupovima  $h$ -ekvidistanti seku. Pored toga, ovako uvedena mreža  $h$ -tačaka pokriva vidnu  $h$ -ravan na ravnomeran način, čuvajući rastojanja između čvorova mreže u  $h$ -ravni u hiperboličkom smislu.

Ovakav pristup zahteva interpolaciju izračunatih boja u određenim  $h$ -tačkama mreže, s obzirom na to da preslikavanje mreže na piksele ekrana nije bijekcija (Meyer et al., 2002). Pored toga, implementiran je i prikaz u kojem svakom pikselu generisane slike odgovara tačno jedna  $h$ -tačka mreže, slično prikazu hiperboličke mreže hiperparalelnih  $h$ -pravih.

---

<sup>1</sup> Pogodnim izborom funkcije rastojanja dobija se mreža koja iz centra apsolute izgleda identično hiperboličkoj mreži hiperparalelnih  $h$ -pravih.



Slika 24: ilustracija kreiranja mreže  $h$ -tačaka pomoću preseka  $h$ -ekvidistantnih krivih. Ova mreža predstavlja drugačije uzorkovanje velikog kruga apsolute od euklidske mreže i zahteva interpolaciju radi prikaza na ekranu računara.

U kreiranom programskom kodu podržano je kreiranje euklidske mreže, hiperboličke mreže hiperparalelnih  $h$ -pravih i hiperboličke mreže  $h$ -ekvidistantnih krivih sa ili bez interpolacije. Ovih pet mreža označavaćemo njihovim indeksom koji je korišćen i u programskom kodu: euklidska mreža je mreža 0, hiperbolička mreža hiperparalelnih  $h$ -pravih je mreža 1, euklidska mreža velikog  $h$ -kruga apsolute je mreža 2, hiperbolička mreža  $h$ -ekvidistantnih krivih bez interpolacije je mreža 3, a sa interpolacijom mreža 4.

## 7. Implementacija

U narednim poglavljima biće izneti detalji o implementaciji hiperboličkog rejtrensing algoritma i pratećeg modula za opisivanje scena u modelu, kao i primeri scena koji se njome mogu kreirati.

### 7.1 Opis biblioteke HyperGeomLib

U ovom poglavlju biće opisana biblioteka *HyperGeomLib* (Hyperbolic Geometry Library – Biblioteka Hiperboličke Geometrije) koja sadrži primitive za definisanje Poenkareovog sfernog modela, objekata u modelu i hiperboličkog rejtrensing algoritma.

#### 7.1.1 Izbor programskog jezika i korišćenih biblioteka

Kao platforma za biblioteku HyperGeomLib koristi se C++ programski jezik u Microsoft® Visual Studio® 2008 okruženju. C++ programski jezik je široko rasprostranjeni predstavnik objektno-orijentisane programske paradigme koja preovlađuje na tržištu kreiranja aplikativnih programa. Pored toga, C++ sadrži brojne gotove biblioteke koje olakšavaju i pojednostavljaju proces programiranja. Iako se kao razvojno okruženje koristi Visual Studio, želja je bila da se kreira biblioteka nezavisna od platforme na kojoj se program izvršava. Stoga su korišćene isključivo standardne C++ biblioteke ili biblioteke za koje postoji verzija za sve važnije platforme.

Kao kontejner klasa koristi se klasa **vector** iz standardne biblioteke šablona *STL*, u kojoj se čuvaju svi skupovi elemenata u modelu. Pored toga, koriste se i standardne matematičke rutine definisane u svim verzijama C++ programskog jezika, dok se kreirane slike čuvaju u TIFF datoteci uz pomoć biblioteke *libtiff*, raspoložive za većinu programskih platformi.

#### 7.1.2 Dizajn biblioteke i hijerarhija klasa

HyperGeomLib biblioteka sadrži klase neophodne za definisanje elemenata Poenkareovog sfernog modela, kao i hiperboličkog rejtrensing algoritma. Prateći osnovne aksiomatske principe zasnivanja geometrije bazirane na tačkama, pravama i ravnima kao objektima koji se ne definišu, definisane su osnovne klase u biblioteci za reprezentovanje datih osnovnih objekata. Potom se tako definisane klase prirodno proširuju novim klasama koje obuhvataju izvedene geometrijske objekte. Osnovne geometrijske klase biblioteke HyperGeomLib su:

- **HPoint** – definiše *h*-tačku. **HPoint** je osnovna klasa hijerarhije i omogućava definisanje *h*-tačaka u *h*-prostoru, kao i primitive za njihovu transformaciju: *h*-refleksiju, *h*-rotaciju i *h*-translaciju. U ovoj klasi je implementirana i metrika u prostoru (poglavlje 2.3) kao funkcija rastojanja između dve *h*-tačke.



- **HLine** – definiše  $h$ -pravu. Klasa **HLine** omogućava definisanje  $h$ -prave u prostoru, definiše primitive za njihovu transformaciju, kao i dodatne primitive koje su od značaja za ostatak biblioteke. Implicitno su klasom **HLine** definisane i  $h$ -poluprava i  $h$ -duž. Sama klasa kao podatke sadrži dva **HPoint** objekta, koji predstavljaju dve razne  $h$ -tačke koje jednoznačno određuju datu  $h$ -pravu. Te  $h$ -tačke se koriste i za transformaciju **HLine** objekata, pošto se novi položaj  $h$ -prave može odrediti pomoću novog položaja bilo koje dve  $h$ -račke koje joj pripadaju. **HLine** klasa sadrži i primitive za određivanje  $h$ -ravni ortogonalne na  $h$ -duž u temenu date  $h$ -duži, simetralne  $h$ -ravni date  $h$ -duži, kao i dve  $h$ -ravni koje sadrže datu  $h$ -pravu i zahvataju dati ugao.

S obzirom na to da se u rejtresing algoritmu najčešće koristi operacija računanja preseka prave i objekata prostora, klasa **HLine** je definisana tako da izvršavanje operacije traženja preseka bude što je brže moguće. Stoga je kreirana reprezentacija  $h$ -prave koja je deo euklidskog kruga<sup>1</sup> u njenom sopstvenom koordinatnom sistemu. Pogodnim izborom osa koordinatnog sistema  $h$ -prave, čiji se koordinatni početak nalazi u centru kruga koji sadrži datu  $h$ -pravu, problem traženja preseka  $h$ -prave i objekta scene svodi se na primenu dve linearne transformacije koordinata, kojima se prelazi iz svetskog koordinatnog sistema u koordinatni sistem  $h$ -prave i nazad i traženja preseka u novom koordinatnom sistemu, što je u većini slučajeva znatno pojednostavljeno. U određenim situacijama koristi se i pristup bez transformacije koordinata, pošto novi pristup ne dovodi do poboljšanja performansi ili je isuviše komplikovan za implementaciju.

- **HPlane** – definiše  $h$ -ravan. Klasa **HPlane** kao podatke sadrži tri **HPoint** objekta, koji predstavljaju tri nekolinearne  $h$ -tačke (koje ne pripadaju jednoj  $h$ -pravoj) kojima je data  $h$ -ravan jednoznačno određena. Transformacija **HPlane** objekta se, slično kao i u slučaju klase **HLine**, svodi na transformaciju tri **HPoint** objekata koji pripadaju klasi. Od važnijih primitiva klase treba izdvojiti primitive za računanje preseka date  $h$ -ravni sa  $h$ -pravom, kao i primitive za računanje refleksije i refrakcije zraka svetlosti predstavljenog  $h$ -pravom od  $h$ -ravni.

Kao što je već napomenuto, koristeći navedene tri osnovne klase biblioteke, izvedene su i brojne druge klase. Njihov primarni cilj je definisanje objekata unutar Poenkareovog sfernog modela – objekata koji se žele renderovati kreiranim algoritmom i prikazati na ekranu. Sve klase koje definišu objekte imaju zajedničku roditeljsku klasu, koja je čista virtuelna klasa. Razlog ovakve strukture leži u lakšem baratanju objektima u sceni, pošto se na ovaj način svi objekti mogu smestiti u jedan vektor pokazivača na virtuelnu roditeljsku klasu. Klase objekata u biblioteci su:

- **HObject** – bazna klasa hijerarhije klasa kojima su definisani objekti u  $h$ -prostoru. Definisana kao čisto virtuelna klasa, ona ne dozvoljava definisanje objekata klase. Međutim, koristeći pokazivače na klasu **HObject** mogu se reprezentovati svi ostali objekti ove pod-hijerarhije klase u biblioteci. U klasi je kao podatak klase definisan BRDF objekta, dok su virtuelni metodi klase koji moraju biti implementirani u klasama naslednicama metodi za računanje preseka zraka svetlosti i objekta, metodi koji implementiraju lokalni model osvetljenja, kao i metodi za traženje refleksije i refrakcije. Pored ovih metoda definisani su i standardni metodi transformacije objekta klase, od kojih su neki čisto virtuelni i moraju se implementirati u izvedenim klasama, dok su neki zajednički implementirani u **HObject** klasi.

<sup>1</sup> Slučaj kada je  $h$ -prava deo euklidske prave je računski jednostavan.

- **HPlaneObject** – klasa koja definiše  $h$ -ravan kao objekat u modelu. Definisana kao klasa naslednik klasa **HObject** i **HPlane**, ona objedinjuje funkcionalnost ove dve klase i omogućava prikazivanje čitavih  $h$ -ravni na ekranu. Iako se čini kao neprirodna potreba za definisanjem dve različite klase koje predstavljaju  $h$ -ravan, primetimo da je **HPlaneObject** klasa specijalizacija klase **HPlane** i sadrži elemente koji nisu neophodni u grupi klasa koje definišu osnovne elemente  $h$ -prostora.
- **HSphereObject** – klasa koja definiše  $h$ -sfere u modelu. Svaka  $h$ -sfera jednoznačno je u modelu definisana  $h$ -tačkom koja je centar  $h$ -sfere i poluprečnikom zadatim u hiperboličkoj metriki. Međutim, za potrebe lakšeg i bržeg manipulisanja  $h$ -sferama i njihovih transformacija, kao podaci klase čuvaju se i njen euklidski centar i poluprečnik (podsetimo se da je  $h$ -sfera u modelu i u euklidskom smislu sfera, ali im centri nisu nužno u istoj euklidskoj tački – poglavlje 2.3), kao i četiri nekoplanarne  $h$ -tačke koje joj pripadaju. Razlog zbog kojeg se čuvaju date četiri  $h$ -tačke leži u pojednostavljenim transformacijama objekata klase **HSphereObject**. Bilo koja transformacija  $h$ -sfere može se predstaviti transformisanjem četiri nekoplanarne  $h$ -tačke, kojima je  $h$ -sfera jednoznačno određena. Nakon što se primene odgovarajuće transformacije na date  $h$ -tačke lako se mogu odrediti njeni centri, u euklidskom i hiperboličkom smislu.
- **HCylinderObject** – klasa koja definiše  $h$ -cilindar i  $h$ -valjak u hiperboličkom prostoru. U poglavlju 2.3 definisan je  $h$ -cilindar kao skup  $h$ -tačaka hiperboličkog prostora podjednako udaljenih od date  $h$ -prave koja se naziva osom  $h$ -cilindra. Ukoliko osa  $h$ -cilindra sadrži središte apsolute tada je  $h$ -cilindar površ nastala revolucijom oko ose  $h$ -cilindra kružnog luka koji sadrži presečne tačke ose  $h$ -cilindra i apsolute. Ukoliko je osa  $h$ -cilindra  $y$  osa koordinatnog sistema kojim je definisan model, tada je jednačina  $h$ -cilindra data sa:

$$y^2 + (\sqrt{x^2 + z^2} + d)^2 = r^2,$$

Jednačina 10: jednačina  $h$ -cilindra oko  $y$  ose.

pri čemu je sa  $d$  dato rastojanje centra datog kružnog luka od koordinatnog početka, a sa  $r$  poluprečnik kružnog luka (i važi  $d^2 + 1 = r^2$ ).

Problem koji se javlja prilikom korišćenja  $h$ -cilindra u Poenkareovom sfernom modelu je izračunavanje preseka  $h$ -cilindra i  $h$ -prave. Ukoliko se osa  $h$ -cilindra ne poklapa sa  $y$  osom koordinatnog sistema prvo se odgovarajućim transformacijama  $h$ -prava i dati  $h$ -cilindar preslikaju u odgovarajuću poziciju. Potom se računa presek i na kraju se inverznim transformacijama dobijena presečna  $h$ -tačka preslikava u finalnu poziciju. Međutim, računanje preseka u slučaju  $h$ -prave koja je deo euklidskog kruga nije jednostavno – potrebno je naći presek površi date jednačinom 10, euklidske ravni i euklidske sfere. Jednačina preseka tada postaje jednačina 4. stepena sa kvadratnim korenom koja se ne može rešiti. Stoga se u slučaju  $h$ -cilindra presek sa  $h$ -pravom računa numerički modifikovanom metodom polovljenja intervala<sup>1</sup>. Ukoliko **HCylinderObject** objekat definiše  $h$ -valjak, tada se prvo izračuna presek  $h$ -prave i  $h$ -cilindra čija je osa  $h$ -prava koja sadrži osu datog  $h$ -valjka, a potom se proverava da li je rastojanje datog preseka od  $h$ -tačaka  $h$ -duži koja je osa  $h$ -valjka manje od poluprečnika  $h$ -valjka i ako jeste dobijeni presek tada

<sup>1</sup> Pošto sistem jednačina ima u opštem slučaju dva rešenja, tokom pretrage intervala u kojem se nalaze moguća rešenja ponekad je neophodno nastaviti sa rekursivnom pretragom u oba dela datog intervala.

predstavlja traženu  $h$ -tačku. Ukoliko je rastojanje veće, tada presek  $h$ -prave i  $h$ -valjka ne postoji.

- **HPolygonObject** – definiše  $h$ -poligon u modelu.  $H$ -poligon se definiše pomoću zatvorenog skupa  $h$ -duži, uz uslove da dati skup mora biti bez samo-presecanja (tj. da bilo koje dve  $h$ -duži datog skupa mogu imati kao zajedničku  $h$ -tačku samo teme), mora biti  $h$ -koplanaran i mora biti konveksan. U samoj klasi se dati skup duži predstavlja pomoću vektora njihovih temena, kao i  $h$ -ravni kojoj dati  $h$ -poligon pripada. Razlog zbog kojih se uvode navedena ograničenja je povećana jednostavnost, kao i efikasnost izvršavanja koda, bez gubitka mogućnosti da se prikažu i kompleksni  $h$ -poligoni<sup>1</sup>.  $H$ -ravan kojoj dati  $h$ -poligoni pripadaju koristi se prilikom izračunavanja preseka zraka svetlosti i  $h$ -poligona, kao i njegove refleksije/refrakcije.

**HPolygonObject** klasa je proširena i mogućnošću definisanja takozvanih *žičanih  $h$ -poligona*, tj. žičanih poligonalnih modela (žičani poligoni su zapravo sinonim, često korišćen u računarskoj grafici, za poligonalnu liniju). Ukoliko se želi definisati žičani  $h$ -poligon, tada je potrebno u klasi **HPolygonObject** podesiti parametar debljine  $h$ -linije u modelu, koji je definisan u hiperboličkoj metrici. Sada se presek zraka svetlosti i  $h$ -poligona računa tako što se proverava rastojanje zraka svetlosti od svake  $h$ -duži koja je ivica datog  $h$ -poligona – ukoliko je rastojanje manje ili jednako od definisane debljine  $h$ -poligona, smatra se da presek postoji. U suprotnom se zrak svetlosti i  $h$ -poligon ne seku. Stoga se žičani  $h$ -poligon definiše nizom  $h$ -valjaka koji predstavljaju ivice datog  $h$ -poligona.

- **HPolyhedronObject** – definiše  $h$ -poliedarske površi u modelu.  $H$ -poliedarska površ predstavljena je u klasi unijom  $h$ -poligonskih površi, pri čemu se ne postavljaju nikakvi zahtevi pred tu uniju. Drugim rečima, dozvoljeno je samopresecanje, a  $h$ -poliedarska površ ne mora biti ni zatvorena. Razlozi za kreiranje objekata klase **HPolyhedronObject** na ovaj način leže u kompleksnosti Poenkareovog sfernog modela. Umesto da se  $h$ -poliedarska površ definiše pomoću liste njenih temena, ivica i stranica, kao što je to slučaj u euklidskim implementacijama rendering algoritama, u Poenkareovom modelu nije trivijalno izračunati sve parametre  $h$ -poligonskih i  $h$ -poliedarskih površi samo na osnovu njihovih temena i ivica. U slučaju žičanog  $h$ -poliedra koristi se niz  $h$ -valjaka koji predstavljaju ivice datog  $h$ -poliedra, na sličan način kako je to učinjeno u klasi **HPolygonObject**.

Da bi se pojednostavilo kreiranje scena koje sadrže poznate geometrijske objekte, u biblioteci HyperGeomLib je kreirana lista predefinisanih objekata klase **HObject** i omogućeno korisnicima da date objekte dodaju u scenu uz minimalnu upotrebu parametara za njihovo kreiranje<sup>2</sup>. Predefinisani objekti kreirani su koristeći svojstvo konformnosti Poenkareovog sfernog modela, kao i činjenicu da

<sup>1</sup> Može se pokazati da se neplanarni, konkavni ili  $h$ -poligoni sa samo-presecanjem mogu predstaviti kao unija dva ili više planarna, konveksna prosta  $h$ -poligona.

<sup>2</sup> Mnogi poligonski i poliedarski objekti su zanimljivi, kako sa geometrijskog tako i sa estetskog aspekta. Neki od njih su predstavljali i osnovni cilj antičke geometrije – u Elementima Euklid započinje izvođenje geometrije aksiomama i postulatima, a završava *Platonovim poliedarskim površima* (Platon, starogrčki filozof i matematičar; oko 428 p.n.e. – oko 348 p.n.e.), koje su antički Grci smatrali savršenim objektima prirode zbog njihove topološke pravilnosti. Neki od objekata imaju i poseban značaj za hiperboličku geometriju – Sakerijev i Lambertov četvorougao (slika 4) korišćeni su kroz istoriju kao polazni elementi u dokazima protivrečnosti hiperboličke geometrije.

su dve  $h$ -tačke u modelu podjednako udaljene od centra apsolute ako i samo ako su njima odgovarajuće dekartovske tačke podjednako udaljene od centra koordinatnog sistema. Koristeći navedena svojstva Poenkareovog sfernog modela može se dokazati da temena pravilnih poligona (Platonovih poliedara) čiji je centar opisanog kruga (sfere) ujedno i centar koordinatnog sistema određuju temena pravilnih  $h$ -poligona ( $h$ -poliedara) u modelu<sup>1</sup>.

Pored navedenih klasa koje omogućavaju definisanje osnovnih objekata hiperboličke geometrije u Poenkareovom sfernom modelu, kao i komplikovanijih dvodimenzionih i trodimenzionih površi i tela u modelu, da bi se kreirao rendering algoritam potrebne su i dodatne klase koje nisu vezane za geometrijsku prirodu modela u kojem se rendering obavlja. To su klase koje omogućavaju izvršavanje senčenja i rekurzivnog rejtrensing algoritma u izabranom prostoru.

- **Color** – klasa koja definiše boju. Koristi se standardni trokomponentni RGB kolor model, tj. definisane su crvena, zelena i plava vrednost boje kao podaci klase. Svaka od komponenata boje predstavljena je kao 32-bitni broj sa pokretnim zarezom (`float` u C++ programskom jeziku). Nakon toga se dobijene vrednosti mapiraju u 8-bitni raspon, što predstavlja standard kada su u pitanju izlazni uređaji na računaru.
- **BRDF** – klasa koja definiše parametre Fongovog BRDF modela lokalnog osvetljenja. Podaci klase su koeficijenti ambijentalne, difuzne i spekularne refleksije za svaku od komponenti boje pojedinačno, kao i generalni koeficijenti reflektivnosti, transparentnosti, sjajnosti i indeks prelamanja. Dalje se ovi koeficijenti koriste u rejtrensing metodu na način kako je to opisano jednačinom 7 u poglavlju 5.2.
- **HLight** – definiše tačkasti izvor svetlosti u modelu. Podaci klase **HLight** su intenziteti difuzne i spekularne komponente svetlosti, kao i  $h$ -tačka izvora svetlosti.
- **HScene** – definiše 3D scenu i rejtrensing algoritam. Klasa **HScene** sadrži vektor svih objekata definisanih u sceni (vektor pokazivača na objekte abstraktne klase **HObject**), vektor svih izvora svetlosti u sceni kao vektor objekata klase **HLight**, poziciju posmatrača u sceni, mrežu  $h$ -tačaka kojom je definisana projekcija  $h$ -prostora na  $h$ -površ i sam hiperbolički rejtrensing algoritam, kao i rutine za kreiranje mreže  $h$ -tačaka<sup>2</sup>. Klasa **HScene** podržava promenu pozicije posmatrača i transformacije objekata i izvora svetlosti u sceni. Time je omogućeno animiranje scene animiranjem individualnih objekata ili izvora svetlosti.

## 7.2 Opisivanje scene

Kao što je već navedeno, rejtrensing algoritam kreira realistične slike na osnovu datog opisa trodimenzionalne scene. Da bi se olakšalo kreiranje scena koje će biti prosleđene rejtrensing algoritmu definisan je format ulaznih datoteka kojima se jednostavno mogu zadati scene hiperboličkog prostora. Takve ulazne datoteke omogućavaju zadavanje parametara scene, parametara rejtrensing algoritma, objekata scene i njihovih transformacija. Gramatika kojom su data pravila opisivanja scene kreirana je imajući u vidu osobine Poenkareovog sfernog modela hiperboličkog prostora koje su intuitivno razumljive korisnicima. Moduli za leksičku i sintaksnu analizu, kao i generisanje odgovarajućeg koda prilikom obrade scene kreirani su pomoću

<sup>1</sup> Uz uslov da sva temena moraju pripadati unutrašnjosti apsolute.

<sup>2</sup> Podržano je kreiranje euklidske pravougaone ekvidistantne mreže, hiperboličke mreže hiperparalelnih  $h$ -pravih i hiperboličke mreže  $h$ -ekvidistantnih krivih sa ili bez interpolacije.

programskih paketa *flex* i *bison*. Dati programi omogućavaju jednostavno kreiranje leksičkog analizatora i parsera na osnovu opisa željene gramatike.

### 7.2.1 Tokeni gramatike ulazne datoteke

Leksički analizador učitava ulaznu datoteku element po element, vrši prepoznavanje tokena i prosleđuje ih parseru koji vrši sintaksnu i semantičku analizu. Izrada leksičkog analizatora izvršena je pomoću programa *flex*. Osnovni token koji leksički analizador prepoznaje je decimalni broj, tj. broj sa pokretnom tačkom. Ovaj token koristi se i za cele brojeve, u cilju smanjivanja kompleksnosti – sintaksni analizador može naknadno da odredi koja od te dve reprezentacije je neophodna. Ostali tokeni uključuju nazive sekcija u ulaznoj datoteci, nazive njenih parametara, nazive objekata, transformacija itd. Praznine se ignorišu, ali se prate pojavljivanja znaka za novi red čime je omogućeno adekvatno informisanje korisnika o lokaciji na kojoj se pojavljuju eventualne greške.

Podržano je i komentarisanje unutar datoteke, na identičan način kako je to učinjeno u C++ programskom jeziku – ukoliko se prepozna sekvenca `'/*'` tada leksički analizador ulazi u stanje komentar i sve dok se ne naiđe na sekvencu `'*/'` koja označava kraj komentara ignorišu se znakovi ulazne datoteke. Pored toga, mogu se komentarisati i individualni redovi datoteke korišćenjem sekvence `'//'` na početku reda koji želimo da komentarišemo. Na ovaj način će svi karakteri između date sekvence i prvog pojavljivanja karaktera za novi red biti ignorisani.

### 7.2.2 Gramatika ulazne datoteke

Kada leksički analizador odredi koji tip tokena se nalazi na trenutnoj poziciji u ulaznoj datoteci, taj token prosleđuje se sintaksno-semantičkom analizatoru, kreiranom pomoću programa *bison*. Taj program zatim proverava da li se dati token uklapa u definiciju gramatike jezika koji se kompajlira i u skladu sa tim generiše odgovarajući izlaz: u slučaju da je token ispravan, generiše se izlazni kod, a u slučaju da je token pogrešan generiše se izveštaj o grešci.

Gramatikom ulazne datoteke opisivanje scene podeljeno je u segmente, svaki od kojih definiše određeni aspekt scene. Segmenti počinju ključnom reči koja definiše koji segment je u pitanju, nakon čega sledi telo segmenta, unutar vitičastih zagrada, u kojem se definišu njegovi elementi. U nastavku će biti izložen kraći opis svakog od segmenata. Primer ulazne datoteke dat je u Dodatak A.

**Globals** – osnovni segment kojim se podešavaju parametri rejtrensing algoritma. U ovom segmentu definišu se rezolucija mreže piksela, kao i njen tip i fizička veličina u modelu, dubina rekurzije rejtrensing algoritma, itd.

**BRDFs** – segment u kojem se definišu osobine materijala koji se koriste prilikom definisanja objekata scene. Neophodno je definisati koeficijente ambijentalne, difuzne i spekularne refleksije za svaku komponentu RGB kolor modela pojedinačno, zatim koeficijente reflektivnosti, transparentnosti i refrakcije, kao i sjajnost objekta.

**Points** – segment u kojem se definišu sve  $h$ -tačke koje su neophodne za kreiranje objekata scene i njihovih transformacija. Svaka  $h$ -tačka zadaje se kao trojka njenih koordinata unutar absolute i proverava se da li je  $h$ -tačka ispravno zadata (da li se nalazi unutar absolute).

**Objects** – segment u kojem su definisani svi objekti koji se pojavljuju u sceni. Svaki objekat počinje ključnom reči kojom se definiše tip objekta, nakon čega sledi odgovarajuća lista parametara kojima je specifični objekat određen. Podrazumeva se kreiranje objekata čije se središte nalazi u koordinatnom početku – time se koristi osobina Poenkareovog sfernog modela da se hiperbolički i euklidski centar sfere u koordinatnom početku poklapaju, tj. da su ekvidistantne tačke unutar absolute istovremeno i  $h$ -ekvidistantne  $h$ -tačke u modelu.

**Lights** – segment u kojem su definisani svi izvori svetlosti u sceni. Izvor svetlosti definiše se svojim centrom i jačinom difuznog i spekularnog zračenja, u skladu sa Fongovim modelom lokalnog osvetljenja (poglavlje 4.2).

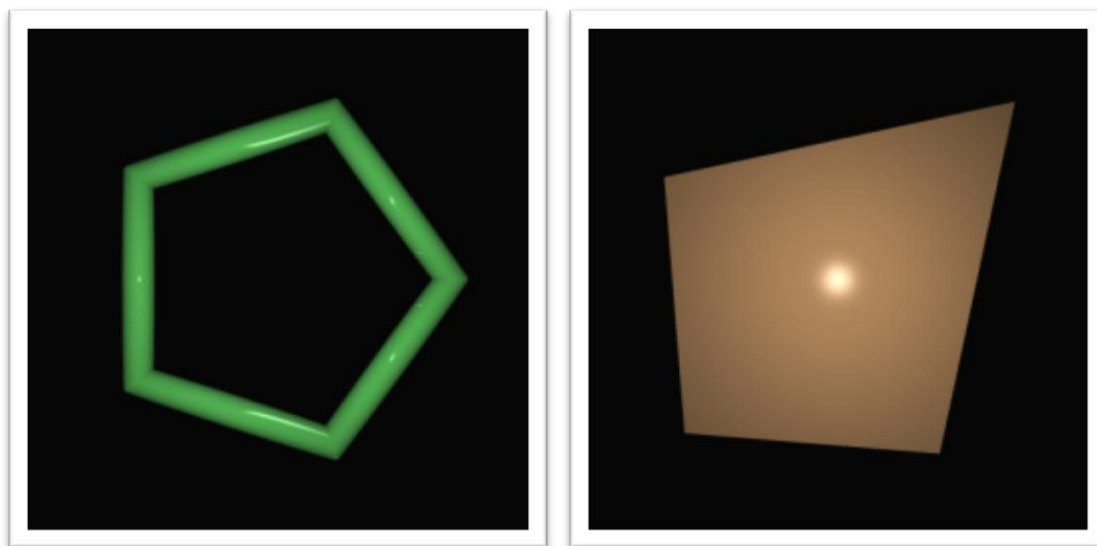
**Transformations** – u ovom segmentu definišu se željene transformacije nad objektima scene definisanih u segmentu **Objects**. Omogućeno je zadavanje  $h$ -pravih i  $h$ -ravni kojima su te transformacije određene, kao i primena transformacija  $h$ -refleksije,  $h$ -translacije i  $h$ -rotacije nad zadatim objektima.

## 8. Primeri hiperboličkih scena

U ovoj glavi su izneseni primeri kreiranih scena u Poenkareovom sfernom modelu. Dati primeri ilustruju mogućnosti razvijenog algoritma, ali i neke interesantne osobine hiperboličkog prostora.

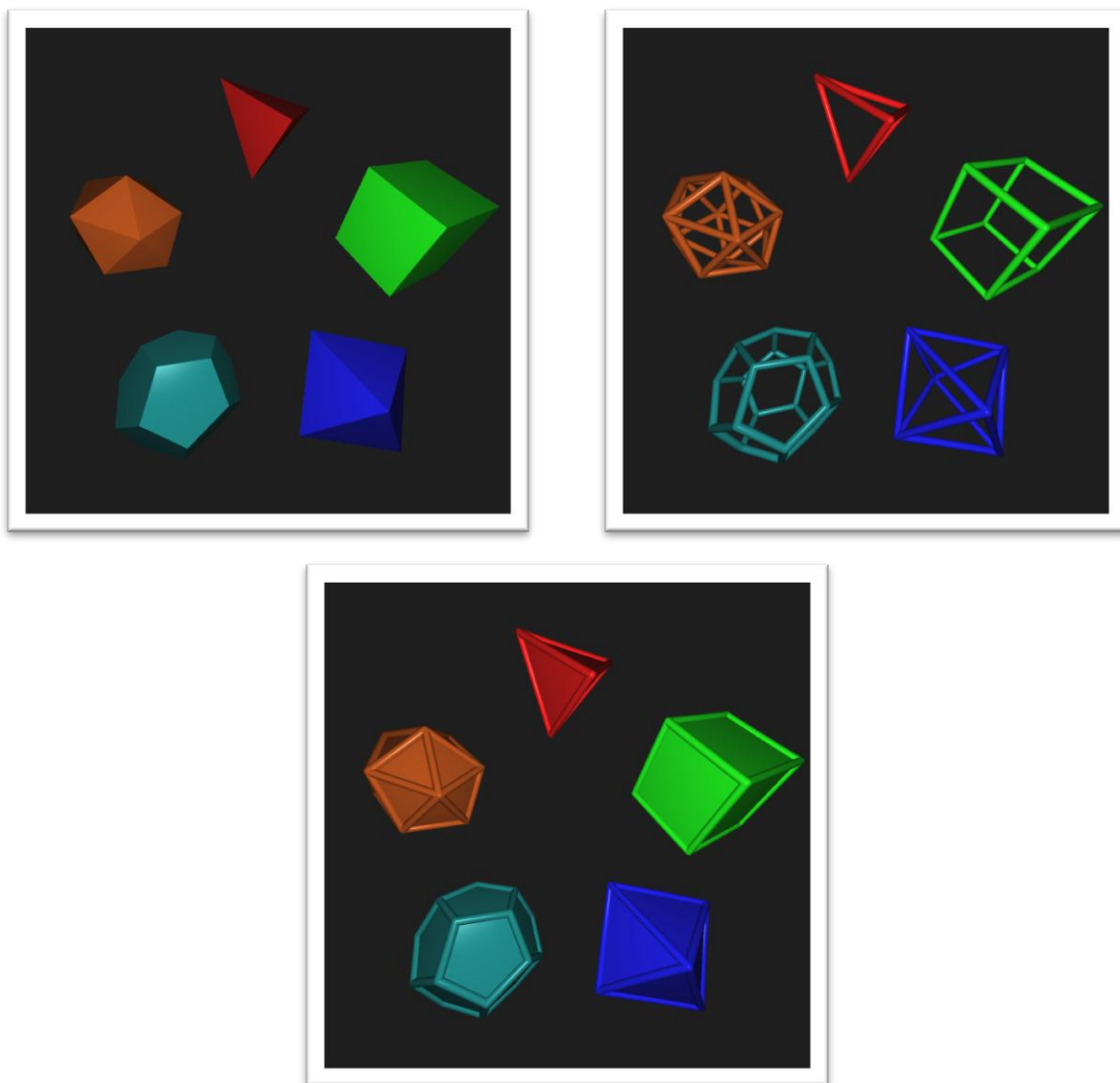
### 8.1 Kreiranje geometrijskih objekata

Najjednostavniji primeri renderovanih scena sadrže  $h$ -poligone. Pomoću kolekcije predefinisanih objekata HyperGeomLib biblioteke moguće je kreirati pravilne  $h$ -poligone sa 3, 4, 5 ili 6 temena, a lako se mogu kreirati i Lambertov i Sakerijev četvorougao. Svi  $h$ -poligoni mogu se kreirati kao planarni modeli i žičani modeli.



Slika 25: ilustracija  $h$ -poligona u modelu: žičani model pravilnog petougla i Lambertov četvorougao, prikaz 0.

Kompleksnije vizualizacije dobijaju se pomoću trodimenzionalnih objekata hiperboličke geometrije. Biblioteka HyperGeomLib podržava kreiranje  $h$ -poligona,  $h$ -sfera i  $h$ -cilindara, čijim se kombinovanjem mogu dobiti kompleksniji objekti. Koristeći biblioteku predefinisanih objekata lako se mogu kreirati kompleksni planarni i žičani  $h$ -poliedarski objekti u sceni. Kao demonstraciju te mogućnosti biblioteke kreirane su slike Platonovih poliedara, jedinih topološki pravilnih prostornih objekata apsolutne geometrije (poglavlje 7.1.2). Dati  $h$ -poliedri kreirani su kao planarni modeli i žičani modeli. Takođe se mogu kreirati i kompleksnije vizualizacije  $h$ -poliedara pomoću kombinacija planarnih i žičanih modela (slika 26). Podržana su i neka od nestandardnih tela, a ipak zastupljenih u literaturi (slika 27).



Slika 26: Platonovi  $h$ -poliedri kao planarni, žičani i kombinovani modeli, prikaz 3.

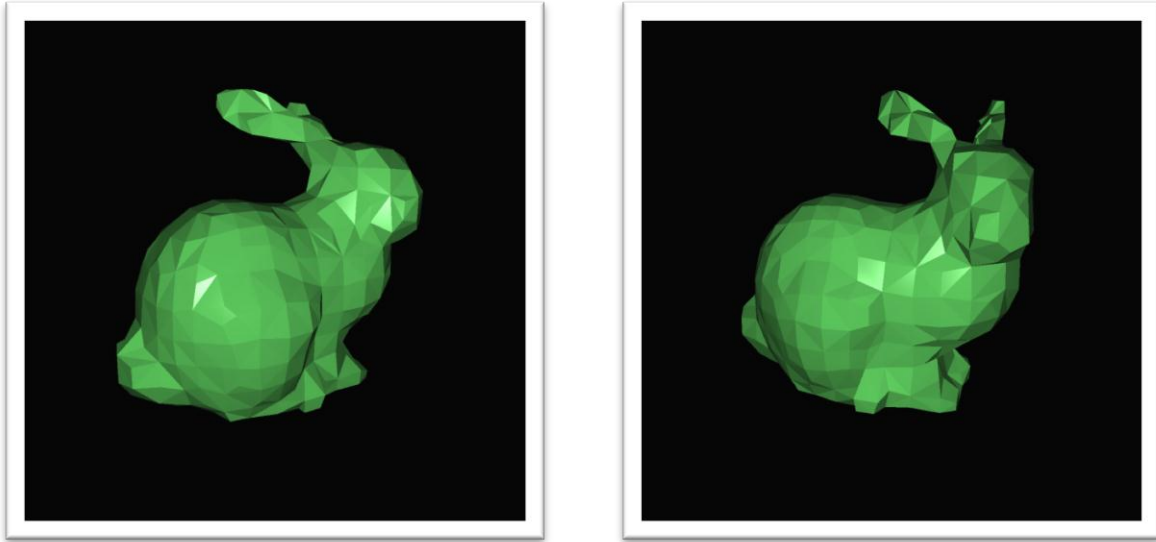


Slika 27:  $h$ -poliedri sa Vitofovom<sup>1</sup> simbolima  $2|3\ 4$  i  $5/2$  i zlatni pravougaonici ikosaedra, prikaz 0.

<sup>1</sup> Viljem Abraham Vitof, holandski matematičar; 1865 – 1939.



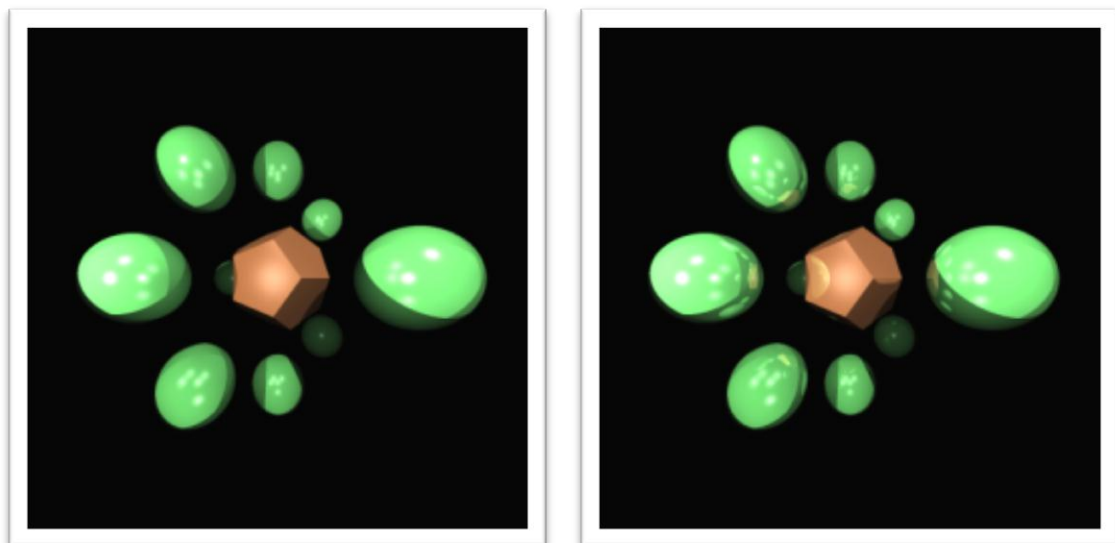
Razvijeni sistem podržava i kreiranje daleko komplikovanijih modela, sa više pojedinačnih objekata koje je neophodno renderovati. Kao ilustracija dat je rezultat renderovanja „zeca sa Stenforda“, čuvenog 3D modela objekta koji je nastao skeniranjem realnog objekta (slika 28). Naravno, vreme izvršavanja hiperboličkog rejtreesing algoritma neuporedivo je sa postojećim euklidskim verzijama.



Slika 28: Stanford Bunny - renderovanje kompleksnijeg 3D modela, prikaz 0.

## 8.2 Ilustracija rejtreesing algoritma – refleksija i refrakcija

Osnovna odlika rejtreesing algoritma je jednostavna podrška za komplikovane svetlosne interakcije poput refleksije svetlosti. Slika 29 ilustruje razliku između renderovanja bez računanja refleksije i renderovanja u kojem se računa refleksija svetlost: u slučaju renderovanja sa refleksijom na  $h$ -sferama i  $h$ -dodekaedru mogu se primetiti refleksije objekata prisutnih u sceni.



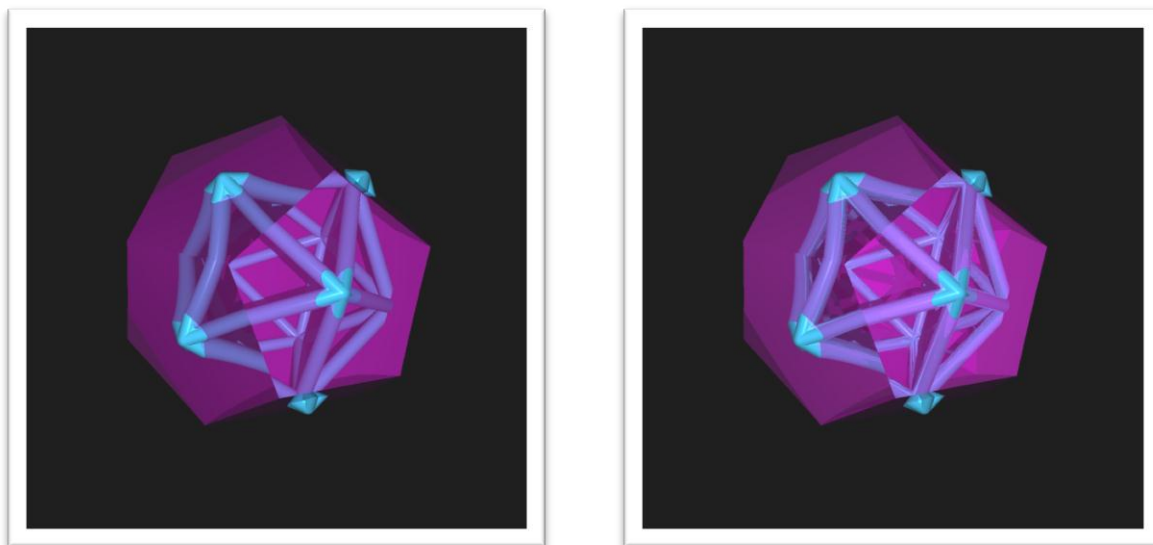
Slika 29: renderovanje bez refleksije (slika levo) poređeno sa renderovanjem sa refleksijama dubine 1, prikaz 0.

Pored refleksije svetlosti neophodno je simulirati i refrakciju, odnosno prelamanje svetlosti, pojavu koja opisuje interakciju svetlosti sa semi-transparentnim objektima. Prelamanje zraka svetlosti koji prelazi iz jednog medijuma u drugi, u odnosu na datu površ, je u potpunosti određeno upadnim uglom svetlosti na datu površ i odnosom koeficijenata prelamanja svetlosti datih medijuma (poglavlje 5.2). Ukoliko su koeficijenti medijuma kroz koje prolazi zrak svetlosti identični tada se pravac prostiranja svetlosti ne menja:



Slika 30: refleksija svetlosti i transparentcija (bez prelamanja), dubina rekurzije 0, 1 i 2, prikaz 3.

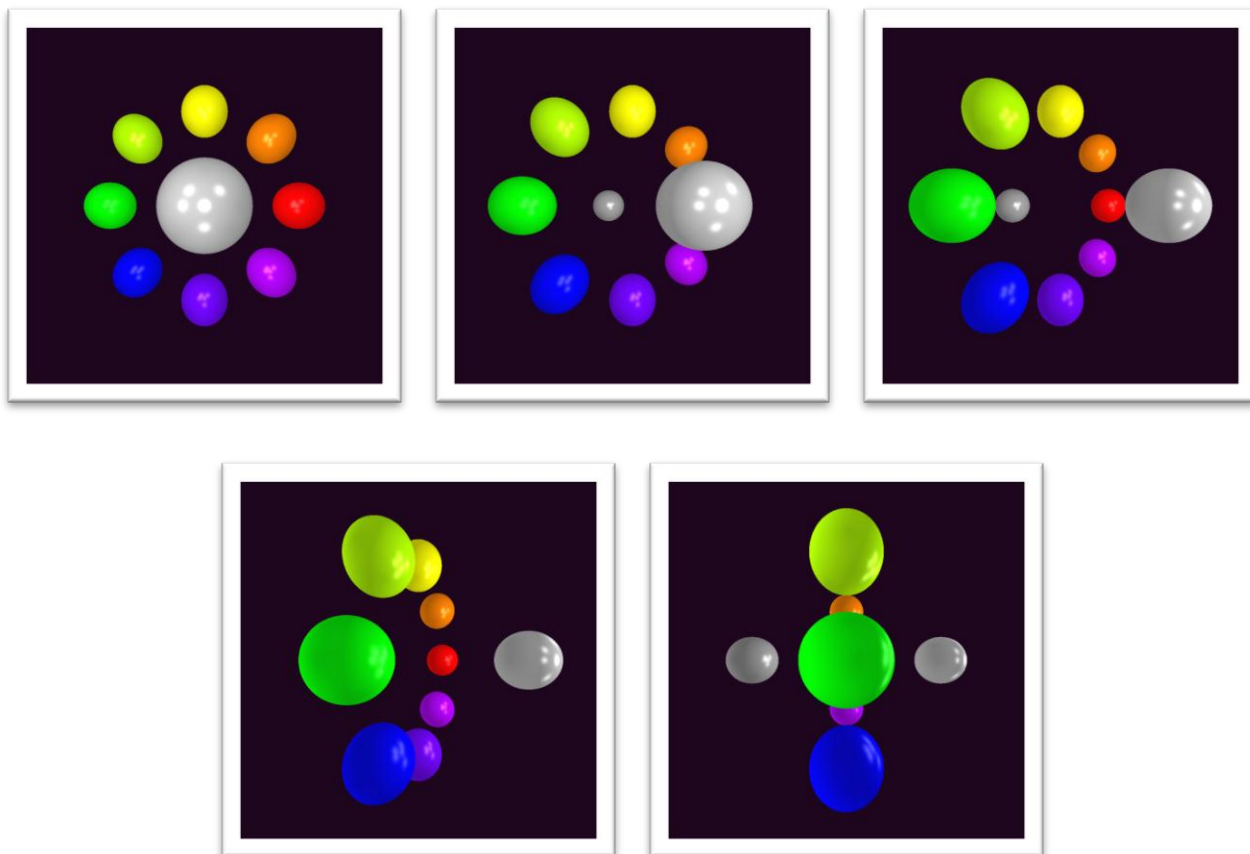
Ukoliko su koeficijenti prelamanja medijuma kroz koje zrak svetlosti prolazi različiti, tada dolazi do prelamanja svetlosti po Snelovom zakonu – zrak svetlosti menja svoj pravac kretanja smanjivanjem ugla između zraka i normale na površ u tački interakcije zraka sa površi (slika 18). Može se primetiti i efekat totalne unutrašnje refleksije koji opisuje pojavu da se zrak svetlosti u potpunosti reflektuje od semi-transparentne površi – teme žičanog modela *h*-ikosaedra u desnom delu slike 31 nije vidljiv iako je *h*-dodekaedar semi-transparentan:



Slika 31: prelamanje svetlosti (indeks refrakcije 1.15), dubina rekurzije 1 i 2, prikaz 3.

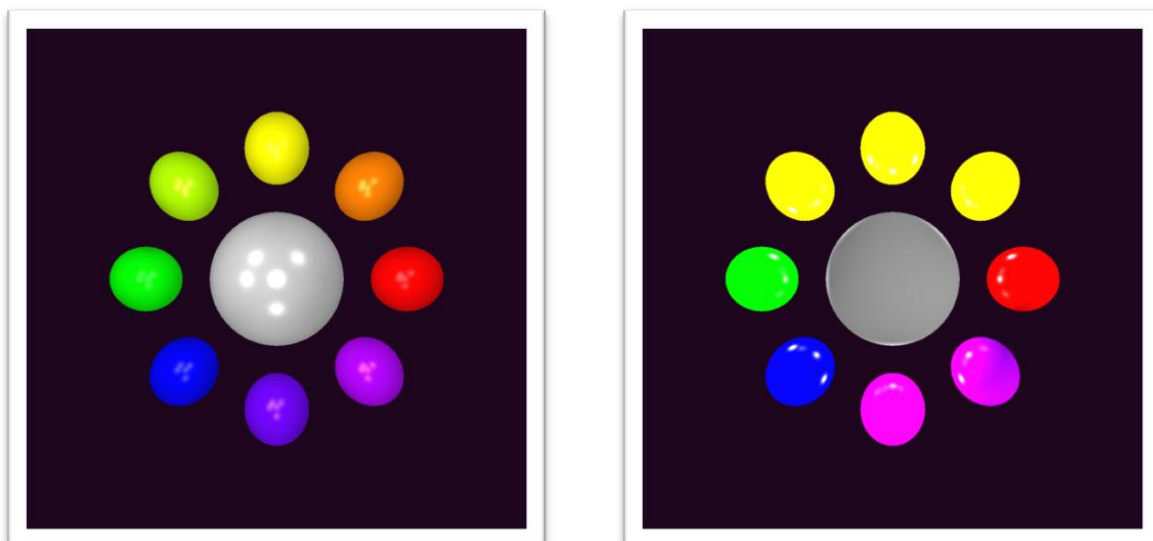
### 8.3 Transformacije scene i pozicije posmatrača

Do sada su sve navedene scene kreirane smeštanjem posmatrača u modelu u centar absolute. Time se olakšava kreiranje scene i njeno intuitivno razumevanje, kao i kreiranje objekata u sceni (poglavlje 7.1.2). Međutim, u sklopu biblioteke HyperGeomLib podržane su i transformacije pozicije posmatrača u modelu u proizvoljnu  $h$ -tačku modela.



Slika 32: pomeranje pozicije posmatrača u modelu  $h$ -rotacijom oko središta scene, prikaz 3.

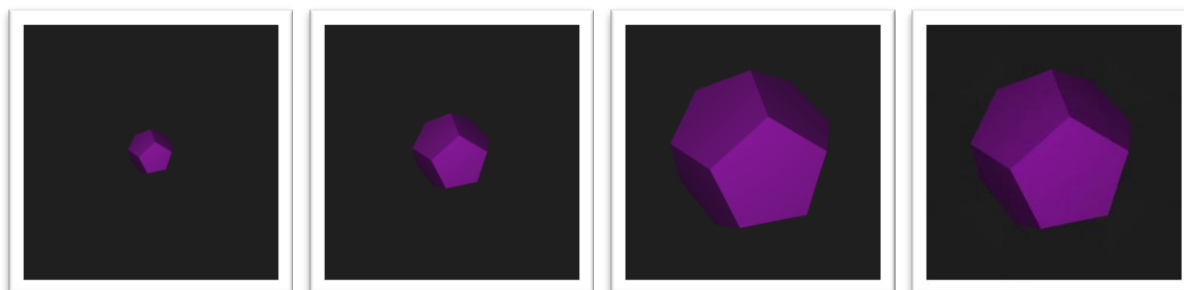
Pomeranje pozicije posmatrača može se iskoristiti i kao demonstracija  $h$ -izometrija u Poenkareovom sfernom modelu. Ukoliko se posmatrač i svi objekti u sceni preslikaju istom  $h$ -izometrijom tada se konačni izgled scene ne menja (slika 33). Primitimo da u navedenom primeru scena položaj izvora svetlosti nije transformisan, te osvetljenje scena nije identično.



Slika 33:  $h$ -translacija scene i posmatrača u sceni, prikaz 2.

## 8.4 Ilustracije osobina hiperboličke geometrije

U poglavlju 6.4 iznesene su osobine mreža piksela koji se koriste u svrhu projektovanja  $h$ -prostora na euklidsku ravan kojoj pripada ekran računara. Ukoliko se prilikom kreiranja scene koriste različite mreže piksela, ne menjajući ostale parametre scene, dobijaju se drugačiji prikazi.

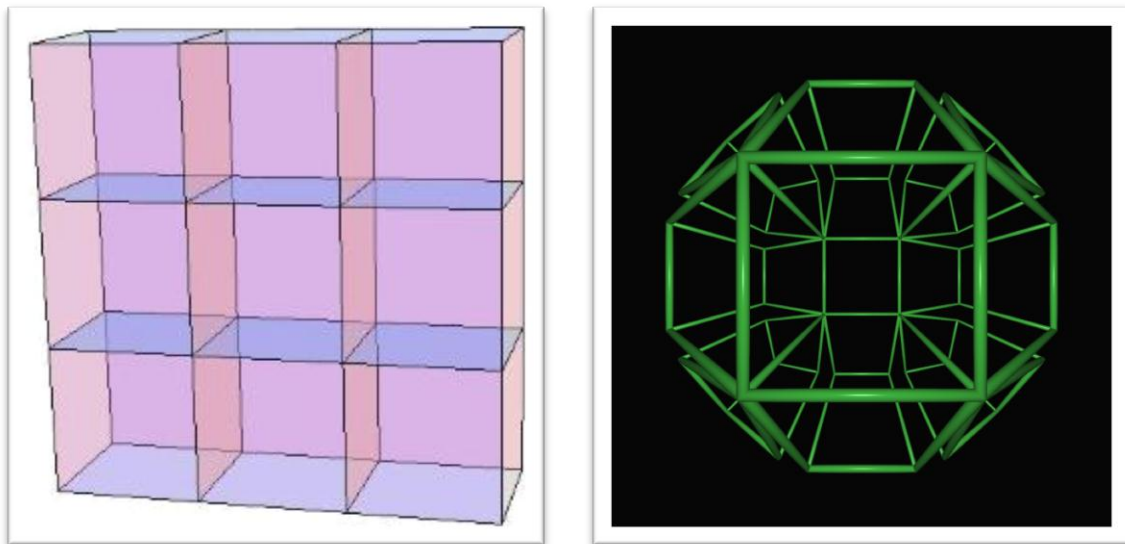


Slika 34: dodekaedar u modelu, kreiran korišćenjem prikazi 0, 1, 3 i 4.

Prisetimo se da se  $h$ -prave u modelu „vide“ kao delovi euklidskih pravih (poglavlje 6.4.1). Stoga se korišćenjem mreža 0 i 1 za prikaz scene kao rezultat dobija slika koja izgleda euklidski – nije prisutno zakrivljenje objekata scene. Razlika u prikazu proističe iz drugačijih dimenzija mreža što rezultuje skaliranjem krajnjeg rezultata. Pored toga, mreža 1 ne pokriva čitavu vidnu  $h$ -ravan (poglavlje 6.4.2). Time je onemogućeno kreiranje prikaza scena koje nisu lokalnog karaktera koristeći sve raspoložive mreže  $h$ -tačaka, pa se korišćenjem i prikaza 2 i 3 dobijaju slike koje izgledaju euklidski.

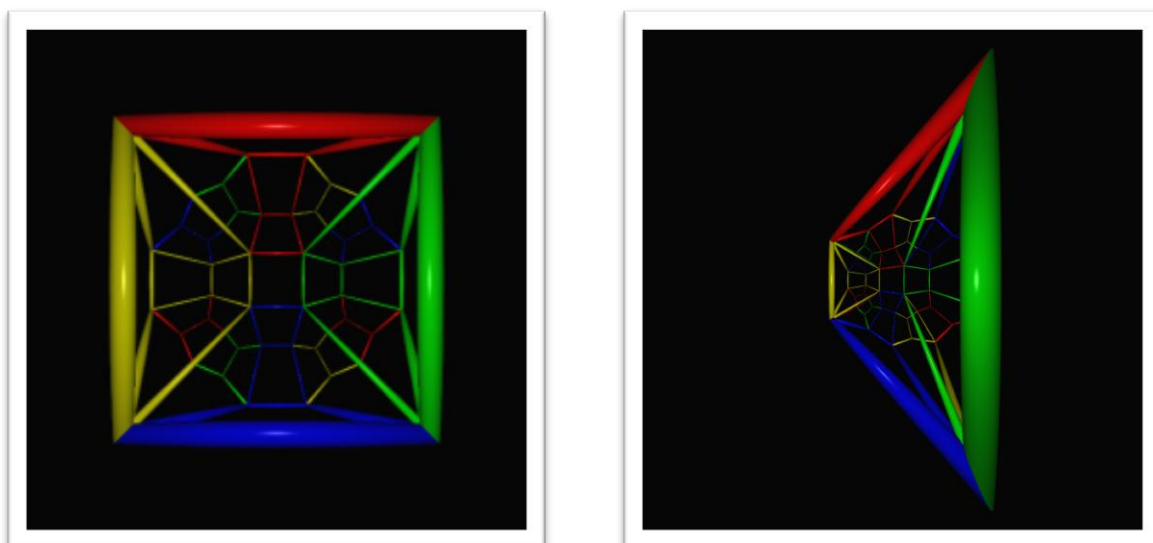
Jedna od osnovnih osobina hiperboličke geometrije, koja je direktna posledica aksiome paralelnosti Boljaj-Gaus-Lobačevskog, je da zbir uglova u  $h$ -trouglu mora biti manji od opruženog ugla (u

euklidskoj geometriji zbir uglova trougla je uvek jednak opruženom uglu). Slični odnosi važe i u slučaju pravilnih poliedara u prostoru: u euklidskoj geometriji se prostor može popuniti bez praznina pravilnim heksaedrima čiji je ugao između svake dve susedne stranice 90 stepeni; s druge strane, u hiperboličkom prostoru ugao između stranica pravilnog  $h$ -heksaedra je uvek manji od 90 stepeni, te se  $h$ -prostor ne može popuniti bez praznina na identičan način kao i u euklidskom prostoru.



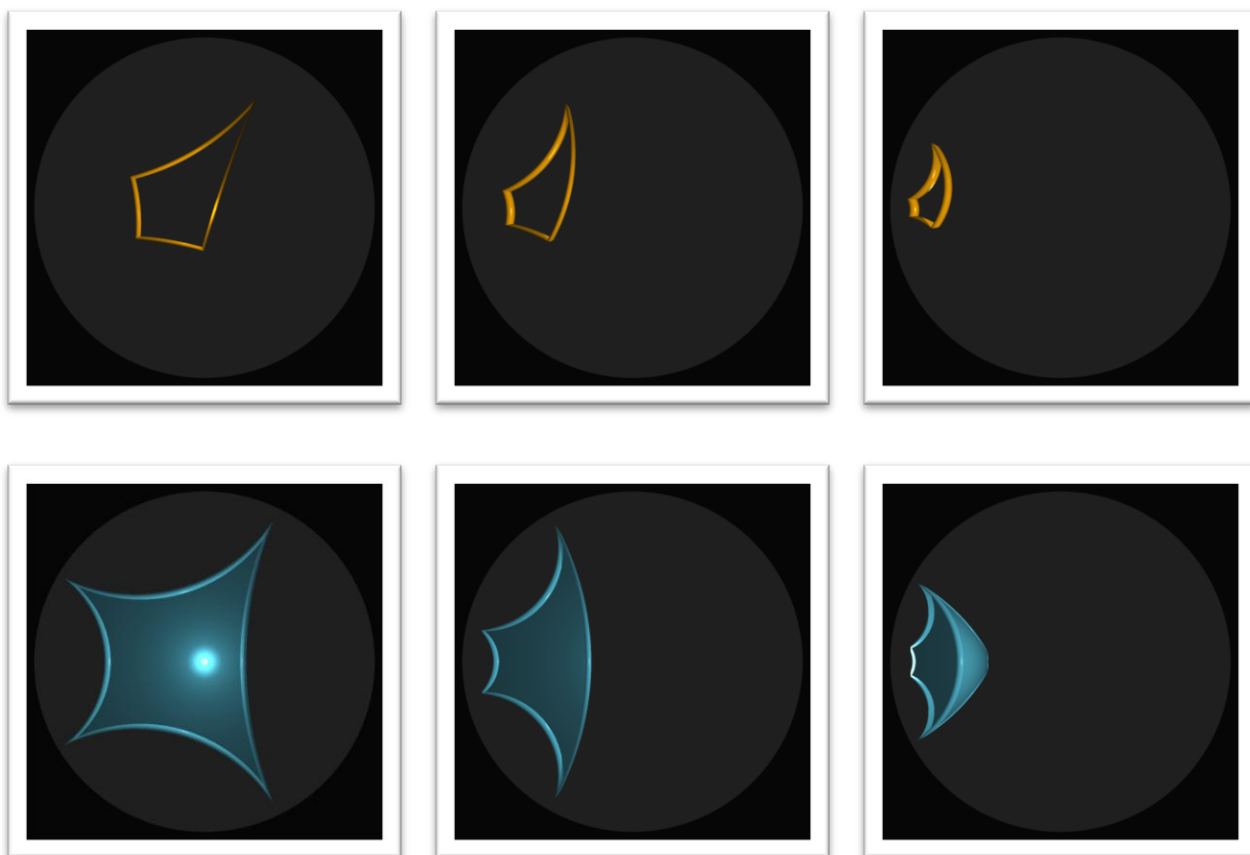
Slika 35: teselacija euklidskog prostora kockama i neuspešan pokušaj ekvivalentne teselacije u hiperboličkom prostoru.

Međutim, objekti hiperboličkog prostora iskazuju interesantnu osobinu prilikom skaliranja – uglovi između njihovih ivica i stranica se menjaju. Povećavanjem dimenzija objekta smanjuju se odgovarajući uglovi, a smanjivanjem dimenzija objekta ti uglovi se povećavaju. Možemo da odredimo dimenzije pravilnog  $h$ -heksaedra u kojem je ugao između dve susedne stranice 72 stepena, te pet takvih  $h$ -heksaedara koji imaju zajedničku tačno jednu ivicu formiraju pun ugao. Ovakvim rasporedom pravilnih  $h$ -heksaedara moguće je popuniti hiperbolički prostor bez praznina.



Slika 36: teselacija prostora pravilnim  $h$ -heksaedrima, prikaz 0.

Može se primetiti da „zakrivljenost“ ivica  $h$ -poligona i  $h$ -poliedara zavisnosi od pozicije objekta unutar apsolute – što je objekat bliže centru apsolute to će percepivano zakrivljenje biti manje. Slika 37 ilustruje ovu osobinu hiperboličkog prostora: Lambertov četvorougao i pravilni  $h$ -heksaedar se  $h$ -translacijom postepeno udaljavaju od centra apsolute, čime se euklidska dužina njihovih ivica smanjuje, a zakrivljenost povećava. Primetimo da se povećava i debljina ivica, što je posledica izduživanja objekata prilikom centralne projekcije (slika 15).



Slika 37: ilustracija  $h$ -translacije Lambertovog četvorougla i pravilnog  $h$ -heksaedra, prikaz 4.

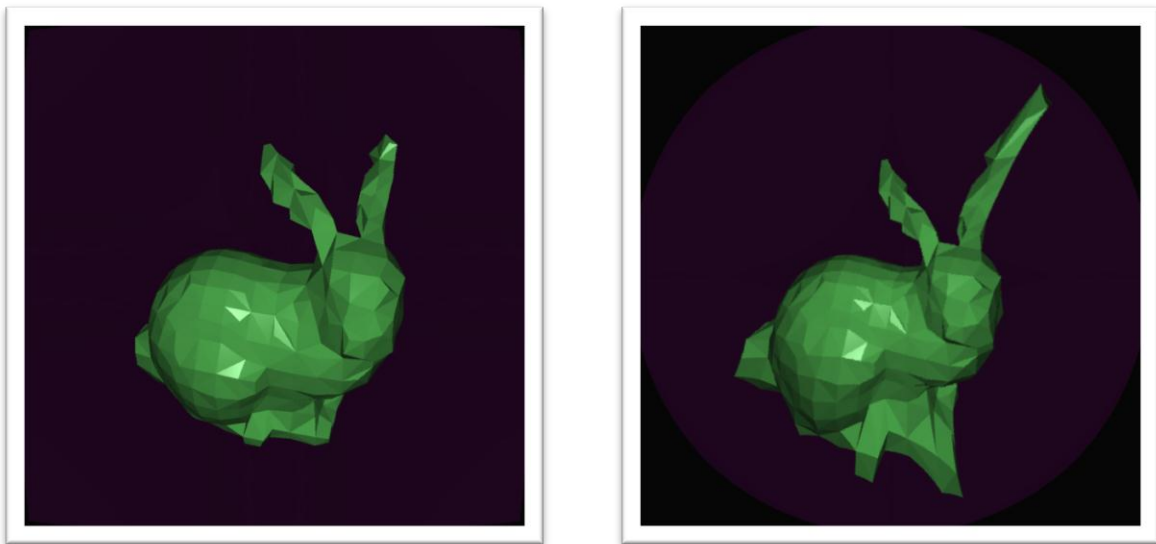
Specijalan slučaj predstavljaju temena  $h$ -poligona ( $h$ -poliedara) koja se nalaze na apsoluti – takva temena nazivaju se *nesvojstvena temena*  $h$ -poligona ( $h$ -poliedra).  $H$ -poligone ( $h$ -poliedre) čija su sva temena nesvojstvena nazivamo *idealnim* (slika 38). Mera ugla čije teme je tačka na apsoluti je 0, a zbir unutrašnjih uglova idealnog  $h$ -poligona ( $h$ -poliedra) je takođe 0.

Primetimo da se hiperboličkom mrežom  $h$ -ekvidistanti tačke koje se nalaze bliže apsoluti neravnomerno uzorkuju u euklidskom smislu – tačke koje leže na  $x$  i  $y$  osama Dekartovskog koordinatnog sistema u kojem je definisan Poenkareov sferni model su favorizovane u odnosu na ostale  $h$ -tačke u modelu. (slika 38 levo, desno teme  $h$ -petougla). Taj problem rešava se interpolacijom koja je implementirana u prikazu 4. Obzirom da se mreža 2 suštinski od mreže 4 razlikuje samo po uzorkovanju vidne  $h$ -ravni, korišćenjem ova dva prikaza dobijaju se isti rezultati (do na kvalitet interpolacije korišćene za prikaz mreže 4).



Slika 38: idealni hiperbolički petougao, prikazi 3, 4 i 2.

Za kraj, ilustrujmo kroz kakvu vizuelnu transformaciju bi morao proći hiperbolički zec da bi se mogao zvati „nesvojstvenim“.



Slika 39: "zec sa Stenforda" pre i nakon skaliranja, prikaz 2.

## 8.5 Efikasnost izračunavanja

Kreirana implementacija rejtrensing algoritma u Poenkareovom sfernom modelu nije efikasna u praksi (poređeno sa klasičnim euklidskim rejtrensing algoritmima). Razlozi za to prevashodno leže u kompleksnoj strukturi Poenkareovog sfernog modela. Kako Poenkareov sferni model nije projektivni model hiperboličkog prostora, ne mogu se koristiti homogene koordinate niti zapisivanje izometrija prostora matricama transformacija, što predstavlja efikasan način za manipulaciju objektima u programima za 3D vizualizaciju. U poglavlju 5.3 opisane su određene tehnike ubrzavanja izvršavanja rejtrensing algoritma, ali i naglašeno da su one veoma teško primenjive u slučaju Poenkareovog

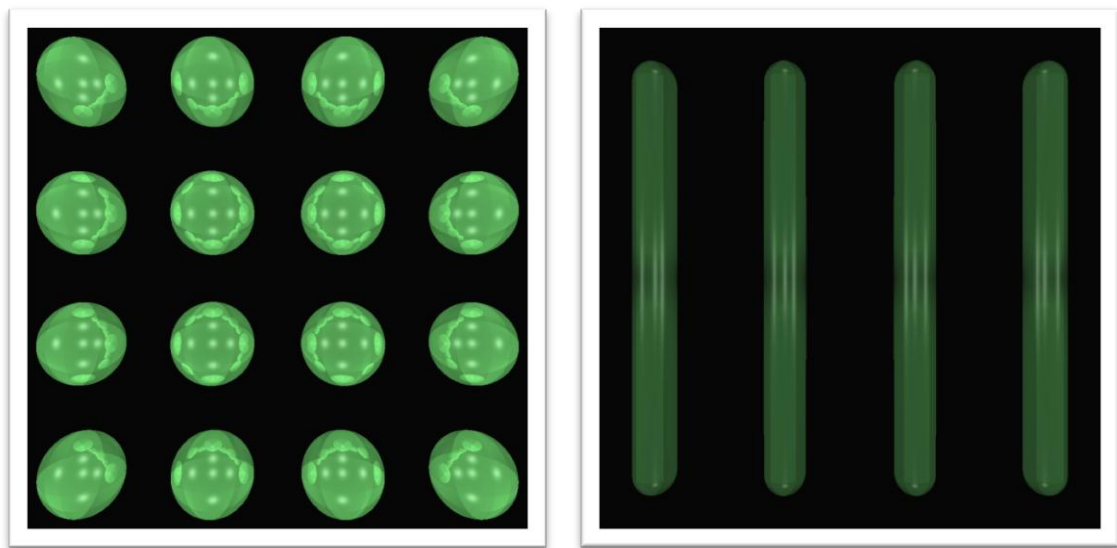
sfernog modela, što je takođe posledica njegove kompleksne strukture. U tabeli 2 i tabeli 3 date su brzine izvršavanja programa za test scene „*h*-sfere“ i „*h*-valjci“ (slika 40). Testiranje je izvršeno na Intel® Core2™ Duo procesoru koji radi na 2.2 GHz, u rezoluciji od 1000x1000 piksela. Sva vremena izvršavanja data su u sekundama. Dobijeni rezultati neuporedivi su sa postojećim implementacijama rejtrejsing algoritma koje koriste projektivna svojstva euklidskog prostora.

<i>h</i> -sfere	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16
Svetla	1	1	1	1	1	5	5	5	5	5	5	5	5	5	5
Rekurzija	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Vreme	2.3	3.4	5.8	10.8	22	3.3	5.5	10.3	20.6	45.6	4.4	7.7	15.4	32	75.7

Tabela 2: vremena izvršavanja za test scenu „*h*-sfere“.

<i>h</i> -valjci	1	2	4	8	1	2	4	8
Svetla	1	1	1	1	5	5	5	5
Vreme	29.1	67.2	146.3	333.1	37.7	101.6	266.9	736.5

Tabela 3: vremena izvršavanja za test scenu „*h*-valjci“.



Slika 40: test-scene „*h*-sfere“ i „*h*-valjci“, mreža 0.



## 9. Pregled relevantnih radova

U ovoj glavi biće dat opis prethodnih radova koji se bave vizualizacijom hiperboličkog prostora, kao i njihov odnos sa ovim radom. Iako postoje brojni sistemi za prikazivanje hiperboličkog prostora na računaru, oni su obično kreirani u nameri da se prikažu osnovne geometrijske strukture prostora korišćenjem jednostavnih metoda senčenja. Ovaj rad je jedan od prvih koji se bavi realističnim vizualizacijama hiperboličkog prostora.

### 9.1 Prve vizualizacije hiperboličkog prostora

Krajem '80-ih i početkom '90-ih godina XX veka Čarls Gan, Mark Filips i Dele Maksvel, tada istraživači Geometrijskog Centra Univerziteta u Minesoti, kreirali su pionirske sisteme za vizualizaciju trodimenzionog hiperboličkog prostora (Phillips i Gunn, 1992). Radeći prevashodno sa Klajnovim sfernim modelom, mada se isti principi mogu primeniti na sve projektivne modele hiperboličke geometrije, pokazali su da se u tim modelima mogu kreirati sistemi za vizualizaciju u realnom vremenu koji se od odgovarajućih sistema u euklidskoj geometriji razlikuju jedino po matricama transformacije. Ako se klasične matrice transformacije euklidske geometrije zamene odgovarajućim *Lorencovim*<sup>1</sup> matricama, dobijaju se transformacije projektivnog hiperboličkog prostora (npr. Klajnovog sfernog modela) ekvivalentne transformacijama euklidskog prostora (Gunn, 2004).

U euklidskom trodimenzionom prostoru koordinate tačaka koje mu pripadaju obično se zadaju koristeći Dekartov koordinatni sistem, kojim se svakoj tački prostora jednoznačno dodeljuje vektor koordinata iz prostora  $\mathbb{R}^3$ . Koordinate tog vektora se obično zapisuju kao uređena trojka  $(x, y, z)$  i nazivaju *afine koordinate*. U računarskoj grafici se, međutim, obično koriste takozvane *homogene koordinate* koje pripadaju prostoru  $\mathbb{R}^4$ . Transformacija između afinih i homogenih koordinata je veoma jednostavna: homogene koordinate  $(x, y, z, 1)$  odgovaraju afinim koordinatama  $(x, y, z)$ , a affine koordinate  $(x/u, y/u, z/u)$  odgovaraju homogenim koordinatama  $(x, y, z, u)$ , ukoliko važi da je  $u \neq 0$ . Prednost homogenih koordinata je da se izometrije i projekcije euklidskog prostora mogu u homogenim koordinatama predstaviti množenjem  $4 \times 4$  matrica.

U trodimenzionom euklidskom prostoru skalarni proizvod dva vektora  $\langle \vec{a}, \vec{b} \rangle$  računa se na osnovu formule  $\langle \vec{a}, \vec{b} \rangle = a_x b_x + a_y b_y + a_z b_z$ , pri čemu su sa  $a_i$  i  $b_i$  date koordinate vektora  $\vec{a}$  i  $\vec{b}$ . U homogenim koordinatama se u formulu skalarnog proizvoda mora uključiti i dodatna koordinata, što je moguće učiniti na dva načina:

$$\langle \vec{a}, \vec{b} \rangle_s = a_x b_x + a_y b_y + a_z b_z + a_u b_u$$

$$\langle \vec{a}, \vec{b} \rangle_h = a_x b_x + a_y b_y + a_z b_z - a_u b_u$$

**Jednačina 11: skalarni proizvodi u homogenim koordinatama kojima su definisani eliptički i hiperbolički prostor.**

---

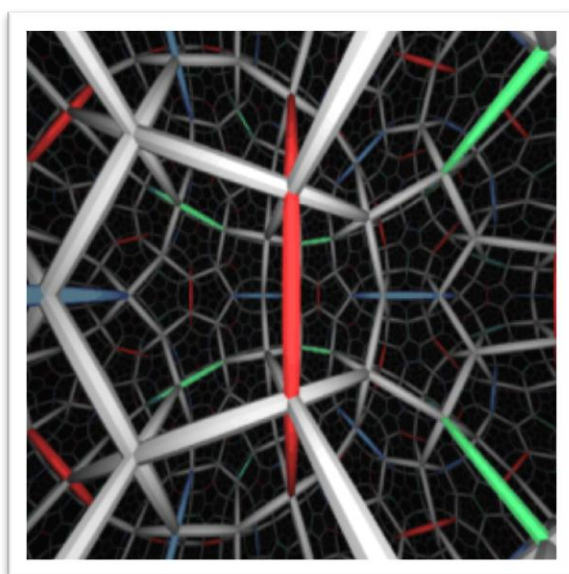
<sup>1</sup> Hendrik Anton Lorenc, holandski fizičar; 1853 – 1928.

Prvi skalarni proizvod odgovara eliptičkoj geometriji, dok drugi skalarni proizvod odgovara hiperboličkoj geometriji (on se još naziva i *skalarnim proizvodom Minkovskog*<sup>1</sup>). Razmotrimo sada skup tačaka  $V_- = \{a \in \mathbb{R}^4, \langle a, a \rangle_h < 0\}$ . Taj skup tačaka obrazuje konus oko  $u$ -ose koordinatnog sistema sa temenom u koordinatnom početku, pri čemu su  $u$  koordinate svih elemenata ovog skupa različite od 0. Sada se normalizacijom homogenih koordinata, slično kao u euklidskom slučaju, dobijaju koordinate trodimenzionog hiperboličkog prostora koji se obično označava kao  $\mathbb{H}^3$ .  $\mathbb{H}^3$  je zapravo predstavljen svim elementima skupa  $V_-$  kod kojih je četvrta koordinata 1. U trodimenzionom euklidskom prostoru taj skup predstavlja tačke koje se nalaze unutar sfere poluprečnika 1, odnosno tačke unutar apsolute, kako se ta sfera obično naziva (poglavlje 2.3).

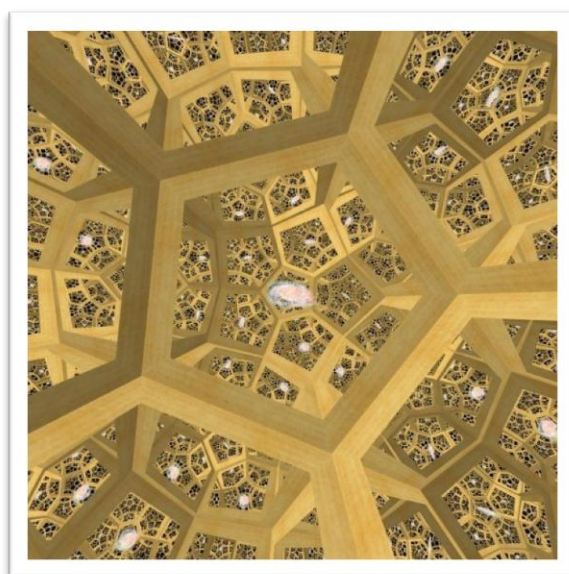
Mogu se definisati matrice transformacije ovog prostora, koje pripadaju podskupu skupa svih matrica klase  $4 \times 4$ , koje predstavljaju izometrije prostora  $\mathbb{H}^3$ . Primera radi, matrica kojom se koordinate tačaka prostora  $\mathbb{H}^3$  rotiraju za hiperbolički ugao  $\alpha$  oko  $x$ -ose je matrica:

$$\begin{bmatrix} \cosh \alpha & 0 & 0 & -\sinh \alpha \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sinh \alpha & 0 & 0 & \cosh \alpha \end{bmatrix}$$

Pomoću ovako izgrađenog sistema kreirane su brojne slike i video zapisi kojima su po prvi put ilustrovane osobine hiperboličkog prostora u tri dimenzije (slika 41). Radi se o isečku iz videa *Not Knot*, koji su kreirali Čarls Gan i Dele Maksvel, kao i slici renderovanoj programom *Curved Spaces* autora Džefri Viksa (Weeks, 2002).



(a)



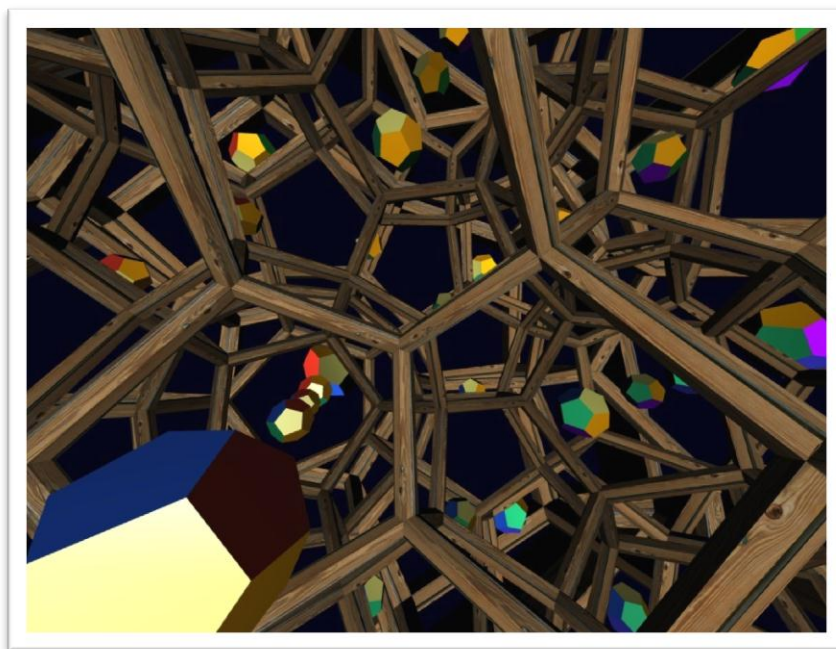
(b)

Slika 41: Vizualizacija Klajnovog sfernog modela – prostorna teselacija pravilnim dodekaedrima iz videa *Not Knot* (a) i programa *Curved Spaces* (b).

<sup>1</sup> Herman Minkovski, nemački matematičar; 1864 – 1909.

## 9.2 Napredne vizualizacije

U nastavku svog rada na hiperboličkim vizualizacijama Čarls Gan posvetio se kreiranju odgovarajuće biblioteke za renderovanje u hiperboličkom prostoru koja bi bila slična postojećoj OpenGL biblioteci koja služi za renderovanje u euklidskom prostoru. Primer rezultata koje je moguće dobiti ovim pristupom dat je na slici 42. Dodata je podrška za senčenje i teksturiranje, međutim rezultati još uvek dosta odstupaju od foto-realističnih rezultata koje je moguće dobiti klasičnim renderovanjem u euklidskom prostoru.



Slika 42: implementacija senčenja i teksturiranja u hiperboličkom prostoru.

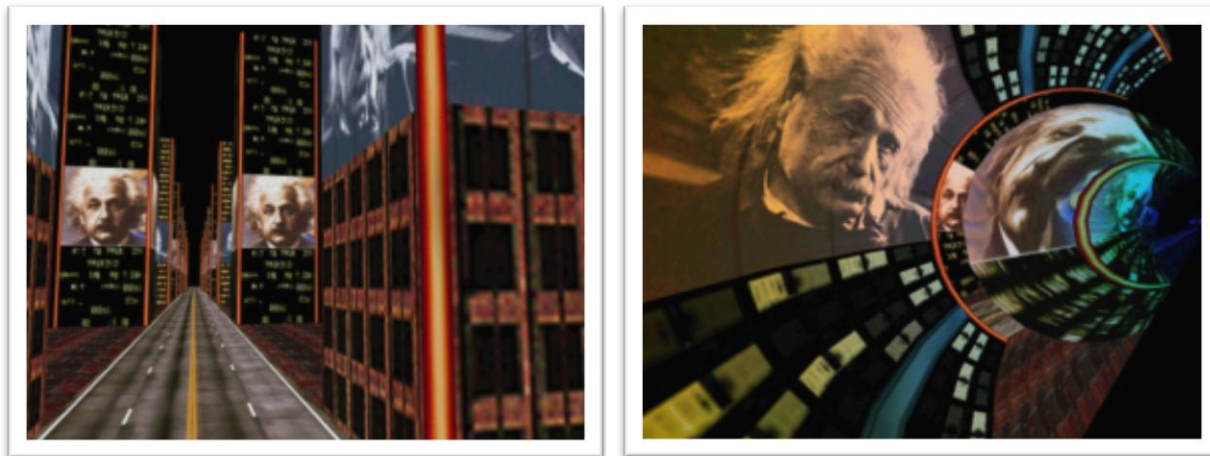
Zanimljivi rezultati postignuti su i tehnikom koja se naziva *relativistički rendering*, koja prilikom kreiranja finalne slike uzima u obzir činjenicu da brzina svetlosti nije beskonačno velika (Howard et al., 1995). Ukoliko se posmatrač u sceni kreće dovoljno velikom brzinom doći će do pojave relativističkih efekata poput *Doplerovog<sup>1</sup> efekta* i *Lorencovih kontrakcija<sup>2</sup>*. Relativistički rendering omogućen je i u brojnim komercijalnim rendering sistemima, poput RayVIS, Pov-Ray i Backlight. Ilustracija relativističkog renderinga data je na slici 43 (Zhang, 1995).

Pored specijalizovanih sistema za vizualizaciju hiperboličke geometrije postoje i brojni sistemi koji omogućavaju proizvoljne matematičke vizualizacije. Program *GeomView* omogućava vizualizovanje objekata euklidske, hiperboličke i eliptičke geometrije, kao i izometrijske transformacije objekata radi boljeg razumevanja njihove strukture. Aplikacije poput programskih paketa *Mathematica* i *JavaView* omogućavaju analitičko definisanje objekata u prostoru, kao i njihovih osobina, te prikaz

<sup>1</sup> Kristijan Johan Dopler, austrijski fizičar i matematičar; 1809 – 1853.

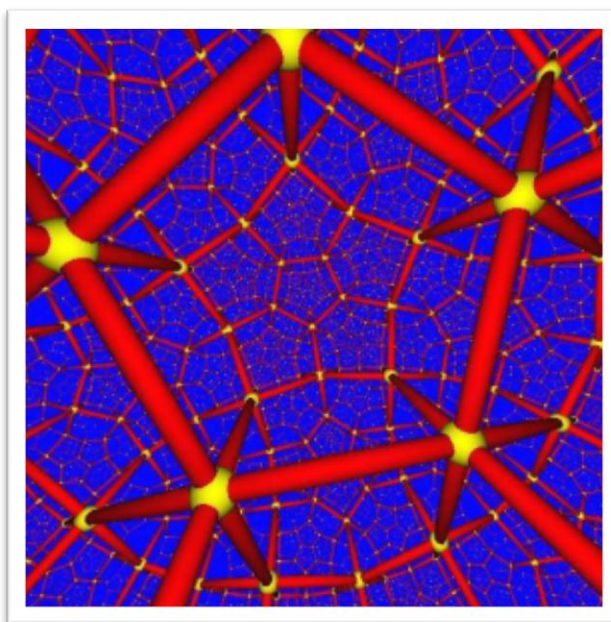
<sup>2</sup> Ovim problemom bave se Ajnštajnovne *opšta i specijalna teorija relativnosti*, sa kojima je hiperbolička geometrija povezana pitanjem na koje još uvek nismo u stanju da damo odgovor: da li je univerzum model euklidske, hiperboličke ili eliptičke geometrije?

na ekranu. Međutim, korišćenje takvih programa zahteva veliku količinu znanja korisnika, kao i veliki napor, s obzirom na to da je neophodno kreirati biblioteke kojima se definišu osobine prostora koji želimo da vizualizujemo. Dodatno, te aplikacije nisu predviđene za realistične vizualizacije i napredni efekti poput realističnog senčenja i teksturiranja nisu nativno podržani.



Slika 43: relativistički rendering - izgled scene u kojoj posmatrač stoji i scene u kojoj se posmatrač kreće brzinom  $0.99c$ . Primetimo zakrivljenje prostora, kao i pojavu Doplerovog efekta (pomeranje spektra vidljive svetlosti).

U vremenu od objavljivanja našeg prvog rada na temu hiperboličkog rejtrejsinga pojavila se još jedna implementacija tog algoritma, kreirana u programskom jeziku Java, autora Hendrika Rupinga. Podržano je kreiranje složenih teselacija prostora (slika 44). Međutim, u vreme pisanja ovog rada implementacija ovog programa, data na sajtu autora, nije pravilno radila, te nismo bili u mogućnosti da kreiramo više rezultata radi poređenja sa rezultatima prezentovanim u ovom radu.



Slika 44: rejtrejsing vizualizacija teselacije hiperboličkog prostora pravilnim dodekaedrima.

## 10. Zaključci i dalji rad

### 10.1 Zaključci

Osnovni motiv za kreiranje hiperboličkog rejtreesing algoritma bila je želja da se kreira alat za proučavanje hiperboličkog prostora, odnosno njegovog Poenkareovog sfernog modela. Proučavanje hiperboličkog prostora do sada je bilo ograničeno na proučavanje njegove matematičke strukture. Naša želja je bila da, koristeći rejtreesing algoritam, omogućimo kreiranje realističnih slika hiperboličkog prostora. Pored toga, umesto projektivnih modela hiperboličkog prostora koji se koriste u drugim sistemima za vizualizaciju izabrali smo Poenkareov sferni model kao podlogu za rejtreesing algoritam. Razlog za ovakvu odluku leži u činjenici da je Poenkareov model konforman, što ga čini intuitivno jednostavnijim od odgovarajućih projektivnih modela, što je od velike pomoći u nastavi geometrije. U ovom radu su takođe predstavljene i neki od osnovnih koncepata računarske grafike koji se standardno koriste za kreiranje realističnih prikaza trodimenzionalnih scena, poput BRDF-ova, jednačine renderovanja i rejtreesing algoritma.

Cilj projekta, kreiranje alata za vizualizaciju Poenkareovog sfernog modela rejtreesing algoritmom koji pruža mogućnost davanja jasne geometrijske interpretacije dobijenih slika, je ostvaren. Obezbeđena je kompletna funkcionalnost Poenkareovog modela, tj. mogućnost kreiranja osnovnih objekata modela i interakcija između njih. Takođe, implementirana je osnova rejtreesing sistema, što uključuje implementaciju izvora svetlosti, osobine materijala od kojih su sačinjeni objekti i sam rejtreesing algoritam, a obezbeđeno je i automatsko učitavanje scene na osnovu opisa iz ulazne datoteke. Krajnji produkt je Windows aplikacija koja objedinjuje opisivanje scene, HyperGeomLib biblioteku i njen rejtreesing sistem, kao i zapis rezultujuće slike u TIFF formatu.

Rejtreesing algoritam se može primeniti koristeći nekoliko metoda kreiranja slike u modelu i projektovanja na euklidsku ravan što omogućuje poređenje dobijenih rezultata. Biblioteka HyperGeomLib detaljno je opisana u kreiranoj dokumentaciji i time je potencijalnim korisnicima olakšano korišćenje biblioteke. Opisivanje scena, renderovanje slika i njihovo čuvanje olakšano je kreiranjem modula za učitavanje scene iz ulaznog fajla i Windows aplikacije koja objedinjuje sve komponente projekta. Sve navedene komponente projekta javno su dostupne u cilju daljeg razvijanja hiperboličkog rejtreesing algoritma.

Nadamo se da se rezultati ostvareni u vidu hiperboličkog rejtreesing algoritma mogu iskoristiti u popularizaciji matematike i problema geometrijskih vizualizacija. Takođe, smatramo da HyperGeomLib biblioteka i prateći programi mogu poslužiti i u obrazovanju kao alat za približavanje hiperboličke geometrije đacima i studentima i pomaže njenom boljem razumevanju. Dobijeni alat može se iskoristiti i u digitalnoj umetnosti – zanimljivost hiperboličke geometrije, pomognuta realističnim izračunom osvetljenja korišćenjem rejtreesing algoritma omogućava novi vid umetničkog izražavanja.

Struktura klasa biblioteke HyperGeomLib, kao i klasnih metoda i podataka, napravljena je sa ciljem da bude lako proširiva, tj. da se omogući jednostavno dodavanje nove funkcionalnosti biblioteke. Pored toga, moguće je iz biblioteke izdvojiti određene grupe klasa i pomoću njih kreirati jednostavnije biblioteke. Time je omogućeno korišćenje cele HyperGeomLib biblioteke ili njenih

delova i u drugim aplikacijama koje kao podlogu koriste Poenkareov sferni model hiperboličkog prostora.

## 10.2 Dalji rad

Do sada kreirana podrška za elemente rejtreesing algoritma i vizualizacije 3D scena na računaru je dovoljna za ilustrovanje mogućnosti korišćenja rejtreesing algoritma i u hiperboličkoj geometriji. Pored toga, dobijene vizualizacije pokazuju da se mogu kreirati i realistične slike 3D scena neeuklidskih prostora. Međutim, još uvek hiperbolički rejtreesing algoritam nije kompletiran, a dalji razvoj algoritma uključuje dodavanje podrške za sledeće elemente:

- Anti-aliasing (eliminisanje nazubljenih ivica poligona)
- Podrška za teksturiranje objekata
- Podrška za animacije
- Podrška za proširene izvore svetlosti
- Podrška za renderovanje visokog dinamičkog raspona.

Dodavanjem navedenih elemenata mogućnosti hiperboličkog rejtreesing algoritma postale bi jednake mogućnostima koje poseduju napredne verzije euklidskog rejtreesing algoritma. Najpoznatije aplikacije za vizualizaciju rejtreesing algoritmom u euklidskom prostoru (3D Studio, Maya, Lightwave, POV-Ray, ...) bazirane su upravo na tim elementima za definisanje 3D scene i rejtreesing algoritma.

U poglavlju 5.3.1 predstavljene su neke ideje za paralelizaciju rejtreesing algoritma u euklidskom prostoru. Na sličan način moguće je kreirati i paralelnu implementaciju hiperboličkog rejtreesing algoritma. Cilj paralelizacije je ubrzavanje rada algoritma koristeći činjenicu da je praćenje raznih primarnih zraka u algoritmu nezavisno, a s obzirom na to da je prezentovana implementacija hiperboličkog rejtreesinga relativno računski neefikasna, ubrzavanje paralelizacijom bi drastično smanjilo vreme neophodno za dizajniranje novih scena i renderovanje odgovarajućih slika.

Za uspešno prikazivanje sveta koji nas okružuje na računaru neophodno je ovladati i fizičkim osobinama kretanja objekata u prostoru i prostiranja talasa, naročito prostiranja zvuka, a slično važi i u hiperboličkoj geometriji. U budućnosti pažnju treba posvetiti kreiranju okruženja koje omogućava precizno definisanje prostornih odnosa u sceni, kao i kretanja objekata i prostiranja zvuka koji poštuju zakone hiperboličkog prostora, što kao krajnji rezultat daje animacije visokog kvaliteta slike i zvuka. Nije teško zamisliti i filmske sekvence ili računarske igre u kojima se kompletna radnja odvija u hiperboličkom prostoru.

Inherentna kompleksnost Poenkareovog sfernog modela i neprilagođenost tog modela kreiranju vizualizacija na računaru značajna je prepreka daljem proširivanju postojeće verzije hiperboličkog rejtreesing algoritma. Stoga se pažnja može posvetiti i kreiranju hiperboličkih vizualizacija u nekom od projektivnih modela hiperboličkog prostora (npr. u Klajnovom sfernom modelu). U glavi 9 ukratko su opisani rezultati na polju vizualizacije hiperboličke geometrije u projektivnim modelima. Ti rezultati mogu se iskoristiti za kreiranje verzija rejtreesing algoritma koji bi se po brzini rada izjednačili sa najbržim verzijama euklidskog rejtreesing algoritma. S obzirom na to da modifikacije

koje je neophodno izvršiti nad euklidskim rejtrejsing algoritmom, u cilju prilagođavanja projektivnom modelu hiperboličke ili eliptičke geometrije, nisu obimne i komplikovane, može se zamisliti rejtrejsing sistem u kojem je moguće koristiti proizvoljnu geometriju, uz sve napredne efekte koje moderni rejtrejsing algoritmi uključuju prilikom kreiranja scene.

## Dodatak A. Primer ulazne datoteke

```
Globals {
    recursion_depth = 1;
    resolution = (1200,1200);
    window_size = (0.15,0.15);
    net_type = 0;
    to = (0,0,0.1);
    background = (0,0,0);
    ambient = (0.1,0.1,0.1);
}

BRDFs {
    ambient = (0.4,0.8,0.4),
    diffuse = (0.4,0.8,0.4),
    specular = (0.7,0.7,0.7),
    alpha = 0.6,
    trans = 0.3,
    shininess = 50,
    refraction = 1;

    ambient = (0.8,0.5,0.3),
    diffuse = (0.8,0.5,0.3),
    specular = (0.7,0.7,0.7),
    alpha = 0.6,
    trans = 0.3,
    shininess = 50,
    refraction = 2.5;
}

Points {
    (0,0,0);
    (0,0,0.16);
    (0.1,0,0);
    (0.07071067811865,0.07071067811865,0);
    (0,0.1,0);
    (-0.07071067811865,0.07071067811865,0);
    (-0.1,0,0);
    (-0.07071067811865,-0.07071067811865,0);
    (0,-0.1,0);
    (0.07071067811865,-0.07071067811865,0);
    (0,0,-0.1);
    (0,0,0.1);
}

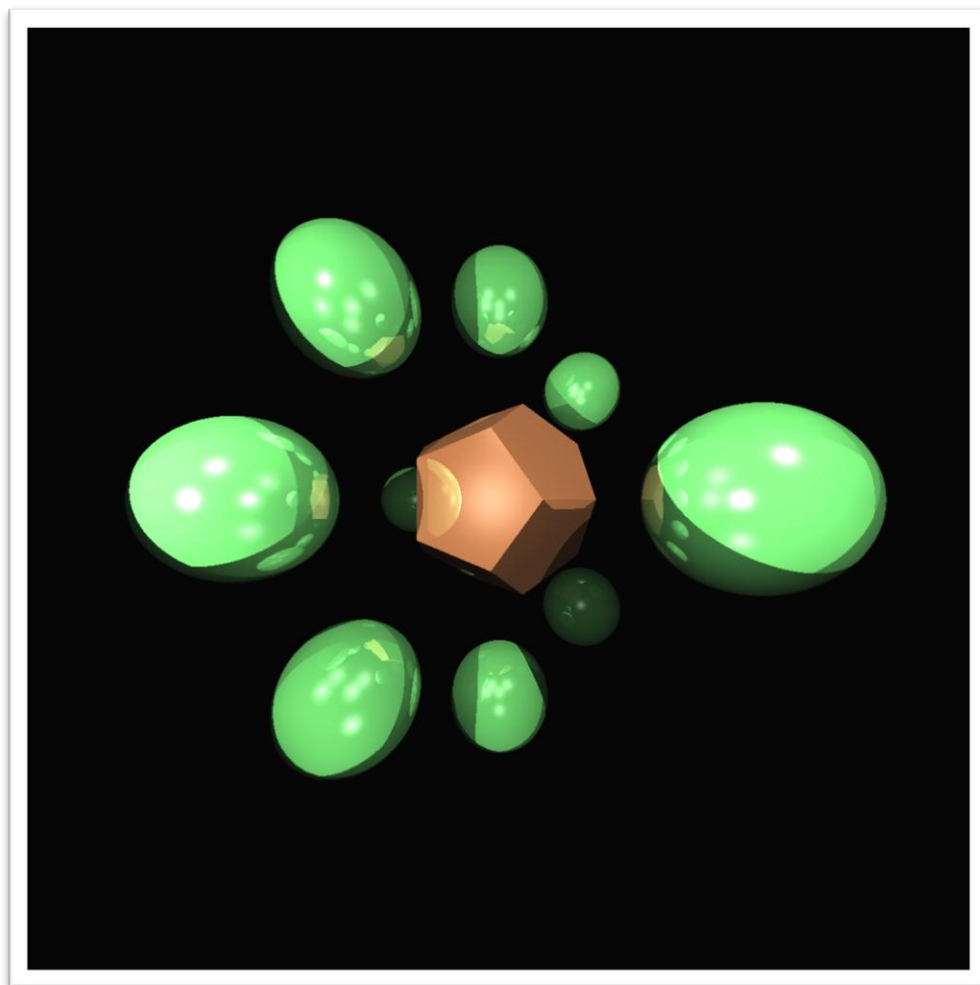
Lights {
    %1, (0.35,0.35,0.35), (0.7,0.7,0.7);
    %4, (0.35,0.35,0.35), (0.7,0.7,0.7);
    %6, (0.35,0.35,0.35), (0.7,0.7,0.7);
}
```



```
%7, (0.35,0.35,0.35), (0.7,0.7,0.7);  
%9, (0.35,0.35,0.35), (0.7,0.7,0.7);  
}
```

```
Objects {  
  Sphere %3, 0.05, #1;  
  Sphere %4, 0.05, #1;  
  Sphere %5, 0.05, #1;  
  Sphere %6, 0.05, #1;  
  Sphere %7, 0.05, #1;  
  Sphere %8, 0.05, #1;  
  Sphere %9, 0.05, #1;  
  Sphere %10, 0.05, #1;  
  Sphere %11, 0.05, #1;  
  Sphere %12, 0.05, #1;  
  Dodecahedron 0.045, #2;  
}
```

```
Transformations {  
  hline (%1,%2);  
  hline (%1,%5);  
  
  %1->Rotate(%2,-0.7);  
  %2->Rotate(%2,-0.7);  
  %3->Rotate(%2,-0.7);  
  %4->Rotate(%2,-0.7);  
  %5->Rotate(%2,-0.7);  
  %6->Rotate(%2,-0.7);  
  %7->Rotate(%2,-0.7);  
  %8->Rotate(%2,-0.7);  
  %9->Rotate(%2,-0.7);  
  %10->Rotate(%2,-0.7);  
  
  %1->Translate(%1);  
  %2->Translate(%1);  
  %3->Translate(%1);  
  %4->Translate(%1);  
  %5->Translate(%1);  
  %6->Translate(%1);  
  %7->Translate(%1);  
  %8->Translate(%1);  
  %9->Translate(%1);  
  %10->Translate(%1);  
  %11->Translate(%1);  
}
```



Slika 45: izgled scene opisane datom ulaznom datotekom

## Bibliografija

- Ajdin, B., Novičić, J., Stamenčić, R., & Janičić, P. (2006). Ray Tracing in Poincaré ball model of hyperbolic space. *DAAD Spring School "Visualization and Geometry"*. Berlin.
- Appel, A. (1968). Some techniques for shading machine renderings of solids. *Proceedings of the Spring Joint Computer Conference 32*, (pp. 37–49).
- Cotton, A., & Freeman, D. (2002). The Double Bubble Problem in Spherical and Hyperbolic Space. *International Journal of Mathematics and Mathematical Sciences*, 641-699.
- Coxeter, S. M. (1965). *Non-Euclidean Geometry*.
- Crowell, B. (2008). *Optics*. Fullerton: www.lightandmatter.com.
- Dana, K. J., van Ginneken, B., Nayar, S. K., & Koenderink, J. J. (1996). *Reflectance and Texture of Real World Surfaces*. Department of Computer Science, Columbia University.
- Ericson, C. (2005). *Real-Time Collision Detection*. Morgan Kaufmann.
- Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1990). *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company.
- Gunn, C. (2004). Visualizing Non-Euclidean Geometry. *Proceedings of Bolyai Bicentennial Conference*.
- Howard, A., Dance, S., & Kitchen, L. (1995). *Relativistic ray-tracing : simulating the visual appearance of rapidly moving objects*. University of Melbourne, Dept. of Computer Science.
- Jensen, H. W., Marchner, S. R., Levoy, M., & Hanrahan, P. (2001). A Practical Model for Subsurface Light Transport. *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (pp. 511-518). ACM.
- Kajiya, J. T. (1986). The Rendering Equation. *In SIGGRAPH 1986: Proceedings of the 13th annual conference on Computer graphics and interactive techniques. Vol. 20* (pp. 143-150). New York: ACM Press.
- Lučić, Z. (1997). *Euklidska i hiperbolička geometrija*. Beograd: Total design i Matematički fakultet.
- Marshall, T., & G.J., M. (2005). Cylinder and Horoball Packing in Hyperbolic Space. *Annales Academiae Scientiarum Fennicae Mathematica*, 3-48.
- Matusik, W., Ajdin, B., Gu, J., Lawrence, J., Lensch, H. P., Pellacini, F., et al. (2009). Printing Spatially-Varying Reflectance. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*. Yokohama.
- Meyer, M., Barr, A., Lee, H., & Desbrun, M. (2002). Generalized Barycentric Coordinates on Irregular Polygons. *Journal on Graphics Tool*, 13-22.
- Molnár, E. (1997). The projective interpretation of the eight 3-dimensional homogeneous geometries. *Beiträge zur Algebra und Geometrie 38(2)*.

- Moore, W. A. (1991). Kd-trees for Cheap Learning. *Efficient Memory-based Learning for Robot Control - PhD Thesis*. Computer Laboratory, University of Cambridge.
- Munzner, T. (1997). H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. *IEEE Symp. Information Visualization*. IEEE CS Press.
- Nicodemus, F. E. (1965). Directional Reflectance and Emissivity of an Opaque Surface. *Applied Optics* 4(7), 767-773.
- NVIDIA. (2009). *Next Generation CUDA Architecture*. Preuzeto sa NVidia: [http://www.nvidia.com/object/fermi\\_architecture.html](http://www.nvidia.com/object/fermi_architecture.html)
- NVIDIA. (2009). *OptiX*. Preuzeto sa NVidia: <http://www.nvidia.com/object/optix.html>
- Phillips, M., & Gunn, C. (1992). Visualizing Hyperbolic Space: Unusual Uses of 4x4 Matrices. *SI3D '92: Proceedings of the 1992 symposium on Interactive 3D graphics* (pp. 209-214). Cambridge: ACM.
- Phong, B. T. (1975). Illumination for Computer Generated Picture. *Communications of the ACM* 18(6), 311-317.
- Purcell, T. J. (2004). Ray Tracing on a Stream Processor. *PhD thesis*. Stanford University.
- Reinhard, E., & Jansen, F. W. (1997). Rendering Large Scenes Using Parallel Ray Tracing. *Parallel Computing*, 873--885.
- Samardžić, A. (1998). Paralelna Implementacija Ray Tracing Algoritma. *magistarski rad*. Matematički Fakultet.
- Seiler, L., Carmean, D., Sprangle, E., Forsyth, T., Abrash, M., Dubey, P., et al. (2008). Larrabee: a many-core x86 architecture for visual computing. *SIGGRAPH '08: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*.
- Weeks, J. (2002). Real Time Rendering in Curved Spaces. *IEEE Computer Graphics and Applications* 22(6).
- Weyrich, T., Lawrence, J., Lensch, H. P., Rusinkiewicz, S., & Zickler, T. (2009). Principles of Appearance Acquisition and Representation. *Foundations and Trends in Computer Graphics and Vision* 4(2), 75-191.
- Whitted, T. (1980). An improved illumination model for shaded display. *Communications of the ACM* 23(6), 343-349.
- Woop, S., Schmittler, J., & Slusallek, P. (2005). RPU: A Programmable Ray Processing Unit for Realtime Ray Tracing. *SIGGRAPH '05: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*.
- Zhang, H. (1995). *Real-time Relativistic Rendering (Special Relativity)*. Retrieved from <http://www.cs.unc.edu/~zhangh/einstein.html>

