

# PRIMJENA FUNKCIJA ORIJENTIRANE UDALJENOSTI I RAY MARCHING TEHNIKE PRILIKOM OBLIKOVANJA DIGITALNIH SCENA

## CREATING DIGITAL SCENES USING RAY MARCHING AND SIGNED DISTANCE FUNCTIONS

Patrik Slade<sup>1</sup>, Ivan Rajković<sup>2</sup>, Dinka Radonić<sup>2</sup>

<sup>1</sup>Tehničko veleučilište u Zagrebu, Vrbik 8, 10000 Zagreb, Hrvatska, Student

<sup>2</sup>Tehničko veleučilište u Zagrebu, Vrbik 8, 10000 Zagreb, Hrvatska

### SAŽETAK

U ovom radu istražuju se osnove prikazivanja digitalnih scena korištenjem ray marching tehnike i funkcija orijentirane udaljenosti. Funkcije orijentirane udaljenosti su matematički izrazi kojima se u bilo kojoj točki u prostoru dobiva najmanja udaljenost do oblika u sceni. Korištenjem ray marching tehnike te funkcije se mogu na razne načine i vizualizirati. Proces ray marchinga bazira se na postepenom „koračanju“ virtualne zrake kroz scenu, te uzorkovanjem udaljenosti do najbližeg objekta za određivanje veličine sljedećeg koraka.

Modifikacijom funkcija udaljenosti jednostavno se ostvaruju raznorazne deformacije objekata, kao što su izduljivanje, glatke binarne funkcije ili prikazivanje beskonačnih kopija proizvoljnog objekta.

Sav kod pisan je u besplatnom online programu Shadertoy, koji koristi programski jezik GLSL i namijenjen je za stvaranje i jednostavnih i kompleksnih shadera. Kao dio ovog rada napravljene su i potpuno renderirane dvije demoscene.

**Ključne riječi:** marširanje zraka, funkcije orijentirane udaljenosti, renderiranje, Shadertoy, računalna grafika

### ABSTRACT

This paper explores the basic usage of ray marching and signed distance functions in the rendering of digital scenes. Signed distance functions are mathematical expressions which—given any point in space—will output the shortest distance to an object in the scene. Using ray marching, these functions can also be visualized in various ways. The process of ray marching is based on a step-by-step marching of a virtual ray through the scene, repeatedly sampling the distance function to determine the size of the next step.

By modifying the distance functions, numerous object deformations can be achieved quite simply, such as stretching, smooth binary functions, or rendering infinite copies of a given object.

All the code was written in the free online program Shadertoy, which itself uses the GLSL programming language and is intended for the creation of both simple and complex shaders. As part of this paper, two demo scenes were created and rendered.

**Keywords:** ray marching, signed distance functions, rendering, Shadertoy, computer graphics

## 1. UVOD

### 1. INTRODUCTION

Većina metoda renderiranja virtualnih scena koristi tzv. „mesheve“; skupine poligona koji stvaraju površinu bilo kojeg željenog objekta. Što je oblik objekta kompleksniji i detaljniji, to je mesh komponenti potrebno više poligona kako bi se objekt točno prikazao.

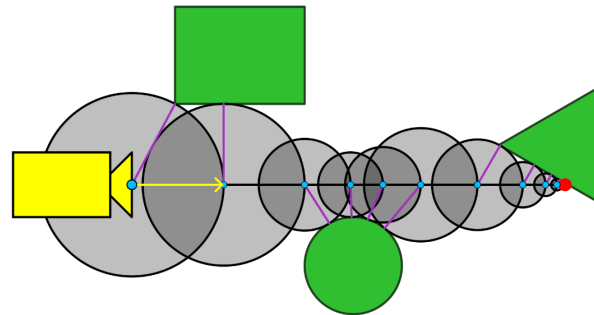
Ray marching je tehnika renderiranja koja se često koristi zajedno s drugim, češće korištenim tehnikama, s ciljem postizanja specifičnih vizualnih efekata kao npr. volumetrijska magla [1]. Međutim, ray marching tehnika se može koristiti i samostalno za stvaranje cijelih scena bez potrebe za meshevima. U tom kontekstu ray marching se odnosi specifično na tzv. sphere tracing [2], koji se snažno oslanja na funkcije orijentirane udaljenosti — koje na način predstavljaju same objekte u sceni — i funkcije deformacije prostora, kojima se objekti manipuliraju [2,3]. To ovu tehniku čini vrlo memorijski efikasnim jer nije potrebno spremati mesheve objekata i njihove brojne poligone.

Cilj ovoga rada je istražiti i objasniti ray marching tehniku, osnove funkcija orijentirane udaljenosti i načine na koje se one mogu modificirati. U svrhu rada, napravljene su dvije virtualne scene kojima će se demonstrirati primjena relevantnih tehnika. Sav kod i renderi ostvareni su u programskom jeziku GLSL na internet stranici shadertoy.com uz osobit doprinos objašnjenja s web stranice iquilezles.com i rada njihovog vlasnika Iniga Quileza.

Funkcije orijentirane udaljenosti (eng. signed distance functions) — tzv. „SDF“-ovi — konceptualno funkcioniraju vrlo jednostavno: od danih ulaznih varijabli koje predstavljaju neku točku u prostoru, funkcija orijentirane udaljenosti vraća vrijednost najmanje udaljenost te točke od objekta kojeg funkcija definira [2,4].

Na sličan način kao i npr. ray tracing, ray marching se temelji na osnovi virtualnih zraka — tj. matematičkih pravaca — uperenih iz smjera virtualne kamere prema sceni. U oba slučaja, svaka od zraka prilikom „pogotka“ ili „sudara“ s objektom u sceni daje povratnu informaciju na temelju koje se određuje boja jednog od piksela

na ekranu. Razlika između ovih dviju tehnika nalazi se u načinu na koji program dolazi do mjesta pogotka: ray tracing direktno izračunava intersekciju zrake s poligonima mesh objekata, dok ray marching „korača“ ili „maršira“ po pravcu zrake približavajući se površini objekta, sve dok nije dovoljno blizu kako bi se lokacija zrake mogla smatrati mjestom pogotka.



**Slika 1** Dijagram ray marchinga. Žuto: virtualna kamera i smjer kretanja zrake. Sivi krugovi: unbounding krugovi/kugle. Plavo: koraci na putanji zrake i mjesta uzorkovanja SDF-ova scene. Ljubičasto: vrijednosti SDF-a u pridruženoj plavoj točki i najmanja udaljenost do objekta u sceni. Zeleno: objekti u sceni. Crveno: točka pogotka.

**Figure 1** Ray marching diagram. Yellow: virtual camera & the ray's direction. Grey circles: unbounding circles/spheres. Blue: steps along the line's trajectory and SDF sampling points. Purple: SDF values in the corresponding blue points, and the smallest distance to an object. Green: objects in the scene. Red: hit point.

Način na koji ray marching to postiže je sljedeći: usmjerena zraka se stvara na lokaciji virtualne kamere. Kako bi zraka mogla napraviti korak naprijed, treba znati koliko veliki korak može napraviti tako da ne pređe rub niti jednog objekta u sceni. Kako bi se to saznalo, poziva se SDF objekta u sceni i prosljeđuju mu se koordinate zrake. SDF vraća najmanju udaljenost zrake od objekta. Ta najmanja udaljenost se u grafičkim dijagramima kao u Slici 1 često vizualizira kao radijus tzv. „unbounding“ kruga ili kugle; radijalnog prostora sa središtem u trenutnoj lokaciji zrake, za kojega se može biti siguran da ne sadrži niti jedan dio objekta jer mu je radijus upravo najmanja udaljenost do objekta. To zatim znači da tada zraka može napraviti korak veličine vrijednosti SDF-a u toj točki i biti sigurna da neće prijeći preko ruba objekta [3]. Nakon tog prvog koraka, zraka se nalazi na rubu prvog unbounding kruga, te ponovno poziva SDF kako bi odredila veličinu sljedećeg koraka. Taj proces koračanja se nastavlja sve dok se zraka ne približi toliko

blizu objektu da se njena lokacija smatra mjestom pogotka. Također je zanimljivo napomenuti da u niti jednom trenutku tijekom tog procesa zraka ne koristi samu lokaciju objekta, već samo udaljenosti do njega.

```
rayInfo rayMarch(vec3 originPoint, vec3 direction) {
    rayInfo ray;
    ray.startPoint = originPoint;
    ray.pos = originPoint;
    ray.dir = direction;
    ray.marchStep = 0;
    ray.dfo = 0.01; //dfo == Distance From Origin

    vec2 mapData;

    for(ray.marchStep; ray.marchStep<MAX_STEPS; ray.marchStep++) {
        //Update the position of the ray;
        ray.pos = ray.startPoint + ray.dir*ray.dfo;
        //Udaljenost do najbližeg oblika & njegov MaterialId:
        mapData = map(ray.pos);
        //Provjeri treba li zraka stati; ako da, prekini petlju
        if(mapData.x<HIT_DIST || ray.dfo>DFO_MAX) {break;}
        //Ako zraka još putuje:
        //Povećaj njenu udaljenost od početne točke
        ray.dfo += mapData.x;
    }
    ray.pos = ray.startPoint + ray.dfo*ray.dir - HIT_DIST*2.0;
    ray.materialId = int(mapData.y);
    return ray;
}
```

*Slika 2 Kod jedne implementacije ray marchinga*

*Figure 2 Code of one implementation of ray marching*

## 2. REALIZACIJA FUNKCIJA ORIJENTIRANE UDALJENOSTI 2. REALIZATION OF SIGNED DISTANCE FUNCTIONS

Za svaki različiti oblik unutar scene postoji posebna SDF funkcija koja ga predstavlja. Mnogi od jednostavnijih oblika (tzv. „primitiva“) zapravo su samo modificirane i deformirane verzije drugih oblika, a mnogi SDF-ovi također su praktički jednaki u 2D i 3D scenama [5,6].

U većini scena bilo koji proizvoljni SDF će se koristiti više puta i na više lokacija, pa se zbog toga aspekti geometrijskih oblika—kao npr. dimenzije kvadra, krajnje točke linije ili radijus kugle—najčešće implementiraju kao varijabilne vrijednosti SDF-a, kako bi se objekti u sceni mogli lako mijenjati bez mijenjanja same originalne funkcije tog SDF-a. Rezultat toga je da gotovo svaki primitiv ima dvije vrste ulaznih varijabli: one koje definiraju njegovu lokaciju i one koje definiraju aspekte samog oblika.

Pomicanje objekata ostvaruje se dodavanjem „offset“ pomaka koordinatama koje predstavljaju lokaciju zrake, čime se uz koordinate same zrake efektivno pomiče cijeli koordinatni sustav u kojemu se SDF objekt nalazi, te se tako dobiva

objekt pomaknut za vrijednost offseta, ali u suprotnome smjeru. To se jednostavno ispravlja inverzijom predznaka offseta.

### 2.1. PRIMJERI PRIMITIVA

#### 2.1. EXAMPLES OF PRIMITIVES

SDF točke jedan je od najjednostavnijih SDF-ova, a ostvaruje se jednostavnim korištenjem Pitagorinog poučka na ulazne vrijednosti.

```
float sdfPoint(vec2 p) {
    return length(p);
}
```

*Slika 3 SDF točke*

*Figure 3 SDF of a point*

SDF-ovi kruga i kugle mogu se smatrati kao deformacije SDF-a točke; još uvijek primjenjuju Pitagorin poučak kako bi izračunali udaljenost dane točke od središta objekta, no na kraju tog izračuna još i dodatno oduzimaju neku proizvoljnu vrijednost. Oduzimanjem te vrijednosti točka se „proširuje“ radijalno; ta vrijednost je dakle radijus kruga ili kugle [2,4,5,6].

```
float sdfCircle(vec2 p, float rad) {
    return length(p)-rad;
}
```

*Slika 4 SDF kruga*

*Figure 4 SDF of a circle*

SDF linije kompleksniji je od SDF-a točke, no i od njega se može stvoriti dodatni primitiv kapsule, koja se također može ostvariti u 2D ili 3D prostoru tehnikom proširivanja—na isti način kao što se od točke stvara krug ili kugla. Unutar SDF-a linije pojavljuje se i varijabla  $h$ , koja se kreće u nekom zadanom intervalu. Krajnje točke linije predstavljaju i krajnje vrijednosti intervala varijable  $h$ ; koristeći tu varijablu moguće je proizvoljne segmente linije „proširivati“ raznim matematičkim funkcijama, kao npr. funkcijom kosinusa ili eksponencijalnom funkcijom [5,6,7].

```
float sdfLine(vec2 p, vec2 a, vec2 b) {
    vec2 pa = p-a;
    vec2 ba = b-a;
    float h = clamp( dot(pa, ba) / dot(ba, ba), 0.0, 1.0 );
    return length( pa - ba * h );
}
```

*Slika 5 SDF linije*

*Figure 5 SDF of a line*

SDF pravokutnika nešto je kompleksniji ali se očito još uvijek vrlo često koristi. Na Slici 6 nalaze se dvije varijable koje predstavljaju dijelove od kojih se pravokutnik sastavlja: `outerDist` i `innerDist`; varijabla `outerDist` stvara SDF pravokutnika s poljem nula u mjestu negativnih vrijednosti. Pošto kod SDF-ova negativne vrijednosti znače da se uzorkovana točka nalazi unutar objekta, a nula predstavlja rub objekta (ili površinu u 3D-u), `outerDist` predstavlja pravokutnik bez „unutrašnjosti“, već s rubom oblika površine. Varijabla `innerDist` stvara suprotno: pravokutnik kojemu je cijeli koordinatni sustav „izvan“ unutrašnjosti zapravo rub. Zbrajanjem ovih dvaju varijabli dobiva se točan SDF pravokutnika [5,6,8].

```
float sdfBox(vec3 p, vec2 dim){
    dim /= 2.0;
    vec2 toPoint = abs(p) - dim;

    float outer1 = max(toPoint.x, 0.0);
    float outer2 = max(toPoint.y, 0.0);
    float outerDist = length(vec2(outer1, outer2));

    float innerDist = min(max(toPoint.x, toPoint.y), 0.0);
    return outerDist + innerDist;
}
```

Slika 6 SDF pravokutnika

Figure 6 SDF of a rectangle



Slika 7 Od lijevo prema desno, SDF polja: `outerDist`, `innerDist`, pravokutnika

Figure 7 From left to right, the SDF fields of: `outerDist`, `innerDist`, rectangle

SDF torusa zanimljiv je zbog činjenice da je zapravo „proširena“ verzija 2D oblika—kruga—u 3D prostoru [5].

```
float sdTorus(vec3 p, float r, float th){
    vec2 q = vec2(length(p.xz)-r,p.y);
    return length(q)-th;
}
```

Slika 8 SDF torusa

Figure 8 SDF of a torus

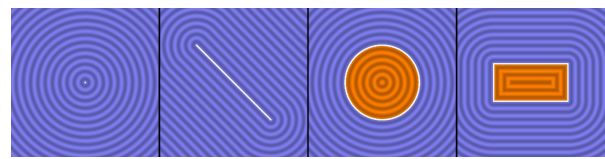
SDF ravnine vrlo je jednostavne građe; on je rezultat množenja koordinate zrake s vektorom smjera koji predstavlja normalni vektor ravnine. Ovaj SDF efektivno odvaja sav prostor u dva

dijela, u kojemu jedan uvijek daje pozitivne, a drugi uvijek negativne rezultate, što u praktičnom smislu znači da je negativni dio prostora „solidan“ tj. „unutar“ objekta. Kako bi se umjesto toga ravnina pretvorila u 2D plohu, njen rezultat se prije izlaza može jednostavno provesti kroz funkciju apsolutne vrijednosti [5].

```
float sdPlane2D(vec3 p, vec3 n) {
    //n treba biti normaliziran
    return abs(dot(p, n));
}
```

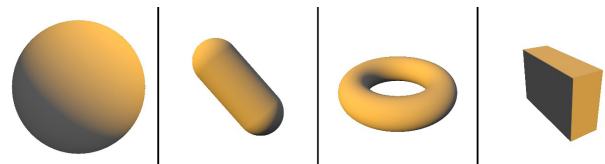
Slika 9 SDF ravnine

Figure 9 SDF of a surface



Slika 10 Polja orijentirane udaljenosti vizualizirana na 2D SDF-ovima. Plavo: pozitivne vrijednosti SDF-a. Bijelo: vrijednosti SDF-a bliže nuli. Narančasto: negativne vrijednosti SDF-a. S lijeva prema desno; polje orijentirane udaljenosti: točke, linije, kruga, pravokutnika.

Figure 10 Signed distance fields visualized with 2D SDFs. Blue: positive SDF values. White: SDF values close to zero. Orange: negative SDF values. Shown from left to right, the SDFs of: a point, a line, a circle, a rectangle.



Slika 11 Trodimenzionalni SDF-ovi vizualizirani ray marchingom. S lijeva prema desno: kugla, kapsula, torus, kvadar.

Figure 11 3D SDFs visualized with ray marching. Shown left to right: a sphere, a capsule, a torus, a cuboid.

## 3. PRIMJERI MODIFIKACIJE OBJEKATA

### 3. EXAMPLES OF OBJECT MODIFICATION

Zbog činjenice da su objekti zapravo samo prikaz svih nultočki SDF-a, oni se mogu dodatnim funkcijama i vrlo lagano modificirati. Neke dosad-spomenute operacije već se mogu brojati kao modifikacije, kao npr. proširivanje oblika oduzimanjem neke vrijednosti od rezultata SDF-a i pomicanje oblika modifikacijom samog prostora

u kojem se on nalazi. Također je dobro zapamtiti da ne postoji limit na to koliko se transformacija i modifikacija može aplicirati na isti objekt.

### 3.1. TRANSFORMACIJA PROSTORA

#### 3.1. SPATIAL TRANSFORMATIONS

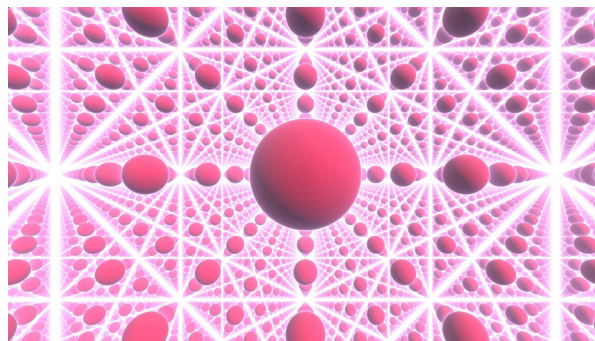
Tehnika anuliranja vrlo je jednostavna i samo zahtijeva da se od izlazne vrijednosti SDF-a uzme apsolutna vrijednost. Tom operacijom bilo koja negativna vrijednost postaje pozitivna, pa efektivno sva solidna unutrašnjost oblika postaje šuplja i površina tog oblika postaje praktički dvodimenzionalna. Proširenjem oblika nakon šupljenja ponovno mu se pridodaje volumen, ali do određene granice on i dalje ostaje iznutra šupalj [5].

Sljedeća najjednostavnija modifikacija oblika bila bi jednostavno izduljenje oblika: na isti način kao što se oblik pomiče pomicanjem koordinatnog sustava, tako se on također može i izduljivati, skraćivati, povećavati i smanjivati. Kao što se funkcija na nekom 2D grafu može pomicati zbrajanjem ili oduzimanjem, ona se može i izduljivati, smanjivati i povećavati množenjem ili dijeljenjem x, y i z osi proizvoljnim vrijednostima. Na vrlo sličan način objekti se mogu i rotirati matičnim operacijama. Pažljivim poretkom apliciranja ovih funkcija mogu se postići rotacije oko bilo koje proizvoljne točke i osi u prostoru, uključujući naravno i originalnu os samog objekta.

Kompleksnijom modifikacijom SDF-ova postižu se još i razni kompleksniji efekti temeljeni na deformaciji koordinatnog sustava: on se može izduljivati „umetanjem“ prostora umjesto množenjem, može se savijati, može se uvijati (kao navoji vijka), a može se i zrcaliti preko bilo koje željene osi [5].

Tehnika repetitije domene specijalna je među drugim tehnikama transformacije po tome što samostalno omogućuje prikazivanje praktički beskonačnog broja proizvoljnih oblika. Repetitija domene funkcionira tako da se prostor— tj. koordinate zrake—modificiraju modulo operatorom, čime se prostor oko oblika efektivno duplicira bez pozivanja dodatnih SDF-ova za svaku kopiju oblika [9,10]. Ova tehnika jedna

je od najutjecajnijih na općeniti potencijal ray marchinga i estetiku njenih scena.



*Slika 12 Demonstracijska scena "Spinning malina grid". Scena koristi samo jedan SDF kugle.*

*Figure 12 Demonstration scene „Spinning malina grid“. The scene uses only one SDF of a sphere.*

<https://www.shadertoy.com/view/ctGBRh>

### 3.2. BINARNE FUNKCIJE

#### 3.2. BINARY FUNCTIONS

Binarne funkcije su ključne ray marchingu; kako bi se uopće u nekoj sceni prikazalo više od jednog objekta, rezultat svih SDF-ova u sceni spaja se operacijom unije kako bi se na kraju operacije dobila samo najmanja vrijednost svih SDF-ova u sceni, tj. najmanja udaljenost do bilo kojeg objekta u sceni. Upravo se ta najmanja udaljenost zatim koristi u samom ray march algoritmu.

```
float sdfUnion(vec3 pos){
    return min(
        sdBox(pos, vec3(0.2, 0.2, 0.8)),
        sdSphere(pos, 0.4)
    );
}
```

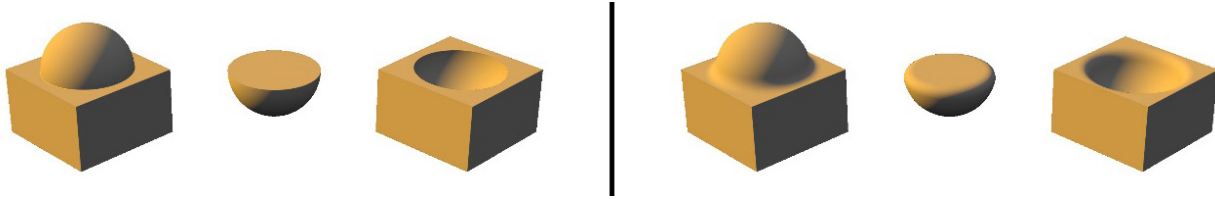
*Slika 13 Osnovna verzija unije dvaju SDF-a*

*Figure 13 Basic version of a union of two SDFs*

Korištenjem drugih binarnih funkcija dobivaju se različiti rezultati:

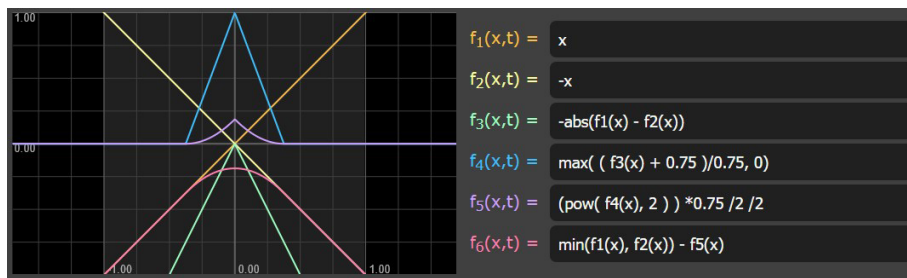
Binarna intersekcija postiže se tako što se od vrijednosti dvaju SDF-a uzima ona veća, a manja odbacuje. Time se oblici tih dvaju SDF-a spajaju u jedan oblik koji će jedino biti „solidan“ u prostoru gdje se ta dva SDF-a preklapaju [4,10].

Funkcijom oduzimanja se od jednog oblika oduzima drugi. Ona se postiže korištenjem binarne intersekcije, no s time da je predznak jedne od vrijednosti okrenut [4,10].



**Slika 14** Primjena binarnih operatora na kuglu i kvadar. S lijeva prema desno: direktna unija, direktna intersekcija, direktno oduzimanje, glatka unija, glatka intersekcija, glatko oduzimanje.

**Figure 14** Application of binary operators onto a sphere and cuboid. Shown from left to right: direct union, direct intersection, direct subtraction, smooth union, smooth intersection, smooth subtraction.



**Slika 15** Postupak stvaranja jedne verzije glatke unije u online programu graphtoy.com

**Figure 15** The process of creating one of the smooth union functions in the online program graphtoy.com

Funkcijom isključivosti postiže se suprotno od funkcije intersekcije: dva preklapajuća SDF-a ostaju solidni samo tamo gdje se ne preklapaju. Zbog toga što se u 3D-u ovaj efekt često ne može najbolje prikazati, ova funkcija se najčešće koristi ili u 2D-u ili uz dodatne transformacijske funkcije [10].

Zanimljivost binarnih funkcija u ray marchingu je mogućnost stvaranja tzv. „glatkih“ binarnih funkcija. Kao što se max() funkcija može postići kombinacijom min() funkcije i okretanjem predznaka, tako se i funkcijom glatke unije mogu postići glatke verzije intersekcije, oduzimanja i isključivosti.

Na slici 13 prikazan je postupak jedne od mnogih verzija funkcije glatke unije. Prevedeno u funkcije udaljenosti, u ovome primjeru funkcije f1 i f2 predstavljaju rezultate dvaju SDF-a, a funkcija f6 predstavlja krajnji rezultat glatke unije [4,11].

## 4. PRIMJENA NA PRIMJERIMA

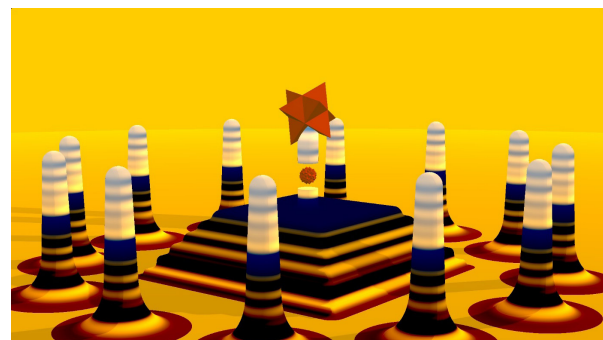
### 4. APPLIED TO EXAMPLES

Zbog svoje memorijske efikasnosti, ray marching je postala jednom od najkorištenijih tehnika demoscene; online zajednice s ciljem stvaranja demonstracijskih scena, tzv. „demoa“, kojima je često cilj prikazati što zanimljiviji rezultat sa što manje memorije.

Kao dio ovog rada, napravljena su dva demoa, koja se za vrijeme pisanja ovog članka još mogu gledati i animirane na shadertoy.com stranici putem linkova u opisima slika.

#### 4.1. DEMO „PIRAMIDA“

##### 4.1. DEMO „PYRAMID“



**Slika 16** Demo „Piramida“

**Figure 16** Demo „Pyramid“

<https://www.shadertoy.com/view/McSXWd>

Svi objektni elementi ove scene osim gornje zvijezde napravljeni su od već spomenutih osnovnih oblika.

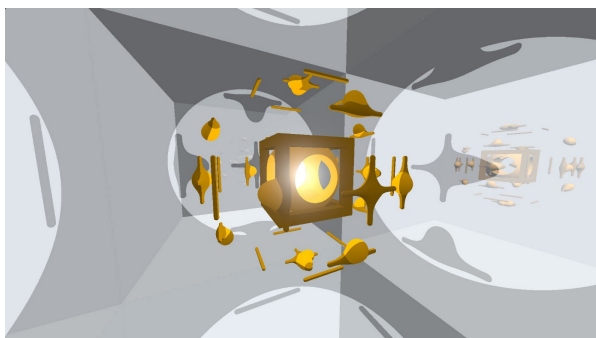
Sama „piramida“ napravljena je kvadara spojenih glatkom unijom i međusobno i s ravninom na kojoj se scena nalazi. Na vrhu piramide nalazi se stup s dvije crvene „zvijezde“. Sam stup je također glatko spojen s piramidom i „odrezan“

binarnim oduzimanjem kvadra, kako bi se između njegovih dijelova mogla nalaziti jedna od zvijezda. Obje zvijezde sastavljene su od jednostavnijih dijelova: donja od kocaka, a gornja od dva tetraedra. Stupovi koji okružuju piramidu jednostavne su izdužene kapsule stvorene kružnom repetitijom domene i glatko binarno spojene s ravninom poda.

Tekstura piramide i stupova ručno je napravljena uz pomoć trigonometrijskih funkcija i visinske (y) komponente točke pogotka. Atmosferski shading postignut je na temelju udaljenosti koju je zraka prošla prije sudara sa scenom.

#### 4.2. DEMO „LIGHTPLAY“

#### 4.2. DEMO „LIGHTPLAY“



*Slika 17 Demo „Lightplay“*

*Figure 17 Demo „Lightplay“*

<https://www.shadertoy.com/view/XXX3R7>

Demo „Lightplay“ (hrv. „Igra svjetla“) tako je nazvan zbog centralnog točkastog svjetlosnog izvora, koje stvara svijetlo i sjene na objekte u sceni.

Cijela scena nalazi se unutar invertirane kocke koja u isto vrijeme prikazuje i sjene drugih objekata i njihove refleksije. Oko centralnog svijetla nalaze se kugla i kocka koji su oboje binarnim oduzimanjem izdubljeni kako bi svjetlost mogla kroz njih prolaziti. Oko tih izdubljenih objekata nalazi se 36 stupova (kapsula), kroz koje se kreće upola toliko kugli. I stupovi i kugle su međusobno glatko binarno spojeni, tako da kugle deformiraju stupove kada god prođu kroz njih.

U animiranoj verziji, kamera kruži oko centra kao i u demou „Piramida“, no prilikom rotacije se po

petlji također pomiče bliže-pa-dalje od središnjeg objekta.

## 5. DISKUSIJA

### 5. DISCUSSION

Kroz ovaj rad predstavljene su i objašnjene osnovni mehanizmi tehnika ray marchinga i SDF-ova, te napravljen par jednostavnih demonstracijskih scena kako bi se na njima kao konkretnim primjerima mogla prikazati funkcionalnost tih tehnika.

Uz kompleksnije programske adaptacije, ray marching i SDF-ovi mogu se primijeniti (i u nekim slučajevima se već primjenjuju) u raznim područjima digitalnih grafike, kao što su renderiranje realističnih poluprozirnih oblika, materijala varijabilne gustoće, kompleksnih fraktala i raznih vrste shadinga i screen-space efekata.

Unatoč korisnim i elegantnim primjenama ove tehnike, njeni najveći problemi su to što ne postoji lako dostupan program koji se njom bavi (Shadertoy je po tom pitanju trenutno najprilagođeniji program za ray marching) i to što je toliko nepoznat da čak i mnogi ljudi iz područja digitalne grafike ne znaju da postoji, a kamoli generalna javnost. Iako je pisan s već postojećim interesom o temi, nada i cilj ovoga rada je da svoje koncepte učini barem malo pristupačnijima i proširi ih barem jednoj novozainteresiranoj osobi.

## 6. REFERENCE

### 6. REFERENCES

- [1.] Unity, „Volumetrics: Introduction to ray marching | Tutorial“, s interneta, <https://www.youtube.com/watch?v=hXYOIXVRRL8>, pristupljeno 2024.07.06.
- [2.] John C. Hart, „Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces“, The Visual Computer, 1995.
- [3.] John C. Hart, Daniel J. Sandin, Louis H. Kauffman, „Ray Tracing Deterministic 3-D Fractals“, Computer Graphics, 1989.
- [4.] Sebastian Lague, „Coding Adventure: Ray Marching“, <https://www.youtube.com/watch?v=Cp5WWtMoeKg>, pristupljeno 2024.01.30.
- [5.] Inigo Quilez, „3D SDFs“, s interneta, <https://iquilezles.org/articles/distfunctions/>, pristupljeno 2024.01.30.
- [6.] Inigo Quilez, „2D SDFs“, s interneta, <https://iquilezles.org/articles/distfunctions2d/>, pristupljeno 2024.01.30.
- [7.] Inigo Quilez, „The SDF of a Line Segment“, s interneta <https://www.youtube.com/watch?v=PMltMdi1Wzg>, pristupljeno 2024.01.30.
- [8.] Inigo Quilez, „The SDF of a Box“, s interneta, <https://www.youtube.com/watch?v=62-pRVZuS5c>, pristupljeno 2024.01.30.
- [9.] Inigo Quilez, „Domain Repetition“, s interneta, <https://iquilezles.org/articles/sdfrepetition/>, pristupljeno 2024.01.30.
- [10.] Benjamin Huang, Jake Waksbaum, Anvay Grover, „COS 426 Final Project Writeup Raymarching with Signed Distance Fields“, 2020.
- [11.] Inigo Quilez, „Smooth minimum for SDFs“, s interneta, <https://iquilezles.org/articles/smin/>, pristupljeno 2024.01.30.

## AUTORI · AUTHORS



• **Ivan Rajković** - (Zagreb, 1978.) diplomirao je 2002. na Akademiji dramske umjetnosti na smjeru Filmska i TV montaža. 2003. godine završava Tehničko Veleučilište u Zagrebu, smjer Elektroničko poslovanje.

Executive Master of Business Administration (Cotrugli MBA) završava 2012. godine, te zatim nastavlja obrazovanje na Carnetovoj E-learning akademiji na smjeru E-learning management. Doktorirao je na smjeru grafičke tehnologije s radom "Digitalna video-reprodukcija u vizualnom i infracrvenom spektru" na Grafičkom fakultetu u Zagrebu, 2018. godine. Viši je predavač na kolegijima multimedije na Tehničkom veleučilištu u Zagrebu. Područje interesa su mu: InfrareDesign tehnologija i daljnje istraživanje unutar područja multimedijiskog sadržaja.

### Korespondencija · Correspondence

ivan.rajkovic@tvz.hr



• **Dinka Radonić** - (Zagreb, 1984.) diplomirala je Filmsko i TV snimanje na Akademiji dramske umjetnosti, Sveučilišta u Zagrebu. Od 2008. radi kao direktorica fotografije i snimateljica na komercijalnim

i umjetničkim projektima. Edukacijom se bavi od 2007. kao voditeljica raznih filmskih škola i radionica. Viša je predavačica na kolegijima multimedije na Tehničkom veleučilištu u Zagrebu. Područje interesa su joj nove filmske forme, kao i znanstvena istraživanja unutar multimedijiskog područja.

### Korespondencija · Correspondence

dinka.radonic@tvz.hr