

KORIŠĆENJE DOP, SOP I VOP OPERACIJA ZA DOBIJANJE EFEKTA RAZARANJA U HOUDINI SOFTVERU**USING DOP, SOP AND VOP OPERATORS FOR DESTRUCTION EFFECT IN HOUDINI**Aleksandra Stolić, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSKA GRAFIKA**

Kratak sadržaj – Vizuelni efekti su neizbežni kada se govori o poboljšanju sekvence u nekom filmu ili seriji. Za razliku od specijalnih efekata koji se fizički dešavaju prilikom snimanja, vizuelni efekti se dodaju kasnije, digitalnom obradom. U ovom radu opisan je proces dobijanja efekta razaranja u Houdini softveru. Iako su i drugi softveri mogu koristiti za dobijanje efekata, ono što izdvaja Houdini je njegovo proceduralno node-based radno okruženje koje nam omogućava da izvršavamo promene u bilo kom delu pipeline-a sa malo ili bez ikakvih posledica. Implementacija efekta razaranja u Houdini-u se u ovom radu sastoji iz dva dela. Prvi deo opisuje proceduralno modelovanje objekta stene pomoću VOP operacija, dok se drugi deo bavi fragmentacijom stene prilikom razaranja, gde su primenjene DOP i SOP. Cilj je bio ispitati prednosti i mane DOP, SOP i VOP operacija prilikom izrade efekta.

Ključne reči: *Houdini, DOP, VOP, SOP*

Abstract – Visual effects is an important term for enhancing the imagery of a film or television series with the aid of computers. While special effects are physical and happen in the real world, visual effects are added afterwards digitally. This paper describes a process of obtaining destruction effect in Houdini. There are many other softwares that could be used, but what sets it apart is node-based environment that allows changes to be made at any time in any area of the pipeline, with little to no consequences. The implementation of destruction effect consists of two parts. The first one describes procedural modeling of stone model, by using VOP operations, and the second one it is about stone fragmentation, by using DOP and SOP operations. The goal is to serach advantages and disadvantages DOP, SOP and VOP operators when creating the effect.

Keywords: *Houdini, DOP, VOP, SOP***1. UVOD**

Houdini je 3d grafički softver koji je razvijen u Torontu od strane *Side Effects Software* kompanije. Prva verzija *Houdini* softvera je izbačena na tržište 1996. U ovom radu je korišćena verzija 17.5 [1]

Houdini se najpre koristio u filmovima i na televiziji.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Ratko Obradović, red. prof.

Veliki broj filmova je u svojoj produkciji koristio *Houdini* softver, neki od primera su filmovi producirani od strane *Marvel's Cinematic Universe* i *Disney*, kao što su *Frozen* i *Moana* [1].

Osim u filmskoj, *Houdini* se sve više koristi i u industriji video igara. Naslovi igara koji su u nekom delu svoje produkcije koristili *Houdini* softver su *Anthem* napravljen od strane *BioWare* studija, *Cry 5* napravljen od strane *Ubisoft* studija, i *Marvel's Spider-Man* iz *Insomniac Games* studija [1].

U ovom radu će biti opisan proces dobijanja efekta razaranja na modelu stene koja je dobijena proceduralnim modelovanjem u *Houdini* softveru. Za potrebe dobijanja efekta korišćeni su DOP, SOP i VOP operatori. Ideja je da imamo dve grupe delova objekta. Prva će predstavljati veće komade objekta na koji će objekat biti podeljen, a druga grupa sadrži komade objekta koji se odvajaju od ivica komada iz prve grupe.

2. SOP (Surface Operators)

U *Houdini*-u svaki operater pripada određenom tipu podataka. SOP je skraćenica za *Surface Operator* i predstavlja tip podataka koji određuje geometriju. SOP opisuje i procesira podatke o geometriji površine, što podrazumeva da sadrži podatke o tačkama, geometrijskim primitivima, ivicama, itd. Postoje dva tipa SOP *node*-ova, a to su SOP *node*-ovi zasnovani na *Python* programskom jeziku i VEX/VOP, gde VEX predstavlja „jezik izražen preko vektora“ koji je najslbližnji C programskom jeziku [2].

Neki od SOP operatora koji su korišćeni u ovom radu su *Iso Offset* i *Voronoi* frakacija.

2.1. Iso Offset [3]

Iso Offset operacija definiše implicitnu funkciju na osnovu ulazne geometrije. Pomoću te implicitne funkcije kreira omotač u odnosu na zadati parametar *offset*-a. Izlazni parametri *IsoOffset*-a mogu biti *IsoSurface*, *FogVolume*, *SDF Volume* i *Tetra Mesh*.

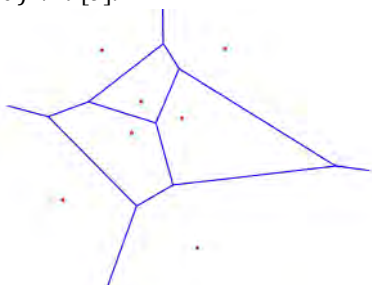
Iso Surface se dobija definisanjem parametra *offset*-a u implicitnoj funkciji i kao rezultat se dobija poligonalni *mesh*.

Fog Volume daje poziciju vokselu koji čine volumen geometrije i koji će imati vrednost 1 ukoliko se nalazi u objektu, u suprotnom će imati vrednost 0. Čelije volumena koje ograničavaju ulaznu geometriju će imati vrednost koja je interpolirana između ove dve vrednosti.

SDF Volume predstavlja volumen geometrije koji je dobijen pomoću SDF funkcije (*signed distance function*). *Tetra Mesh* daje izlaznu geometriju koja je ispunjena tetraedrima. Mesh koji se dobija kao izlazna geometrija se koristi za deformacije i fizičke simulacije deformabilnih tela (*soft body dynamics*).

2.2. Voronoi frakcija

Voronoi dijagrami se koriste u različitim naučnim poljima, kao na primer za prikaz modelovanja dinamike šumskog ekosistema, animaciju toka lave i kreiranje neuralnih mreža. Estetski potencijal Voronoi dijagrama (slik 1) čini ga da bude koristan i za arhitektonske aplikacije [4]. Posmatračemo skup P koji čini n tačaka na površini. Voronoi dijagram skupa P će činiti n ćelija na koje će biti podeljena površina, svaka ćelija će sadržati po jednu tačku iz skupa P sa svojstvom da se tačka q nalazi u ćeliji p_q , ako i samo ako je $dist(q, p_i) < dist(q, p_j)$ za svako $p_j \in P$ gde je $j \neq i$ [5].



Slika 1. Voronoi dijagram [4]

U kompjuterskoj grafici Voronoi frakcija ima široku primenu kod pripreme *rigid body* objekta za simulaciju destrukcije. Njegovo svojstvo da generiše konveksne oblike je naklonjeno algoritmima za simulaciju koji uglavnom imaju brže performanse radeći sa konveksnim oblicima. Proces razaranja objekta pomoću ove metode se može kontrolisati raspodelom izvornih tačaka. Kada simuliramo objekat koji udara o drugi objekat, najčešće su tačke gusto raspoređene u prostoru gde je došlo do kontakta između dva objekta [5].

Voronoi frakcija se najčešće koristi kod simulacija razaranja objekta na delove jer pruža dobar balans između vizuelno kompleksnog doživljaja i kompleksnosti proračuna [5].

3. VOP (Vector Operators)

VOP – *Vector Expression (VEX) Operators* predstavlja programski jezik koji se reprezentuje preko *node*-ova i preko kog se može upravljati različitim atributima objekta, kao što je pozicija, brzina čestica ili boja, odnosno za bilo koji atribut koji postoji u geometrijskoj strukturi. [6]

U narednim poglavljima će biti objašnjeni VOP atributi koji su korišćeni u ovom radu za proceduralno modelovanje.

3.1. Perlin noise

Noise, a naročito *Gradient Noise* je prvi put implementirao Perlin 1985 i kasnije 1989 Perlin i Hoffert. *Perlin noise* se najčešće koristi i u *Houdini*-u je

implementiran kao funkcija koja može da prihvati parametre koji su izraženi pomoću vektora ili preko jednog ili nekoliko decimalnih brojeva. Krajnji rezultat funkcije može biti decimalni broj ili vektor decimalnih brojeva, koji variraju pseudo-nasumično kroz n -dimenzionalan prostor. Algoritam polazi od predračuna za gradijentne vektore kako bi se kreirale pseudo-nasumične vrednosti. Funkcija *noise* koja se poziva u *Houdini*-u i izlazne vrednosti su u opsegu 0-1 i nije periodična. Računske operacije nisu zahtevne u odnosu na druge *noise* funkcije koje su dostupne. *Perlin noise* se može koristiti za kreiranje nasumičnih paterna za dobijanje „realističnih stohastičkih tekstura“ [7].

Noise predstavlja funkciju koju čine skalarni vrednosti i izražene su preko trodimenzionalnog vektora. *Noise* funkcija ima sledeće osobine:

- Nepromenljivost položaja u odnosu na rotaciju. Koliko god da rotiramo neku oblast, položaj *noise*-a ostaje nepromenjen.
- Ima usku granicu opsega frekvencije. Nema većih ili manjih vidljivih karakteristika sa promenom opsega frekvencije.
- Nepromenljivost položaja u odnosu na translaciju. Koliko god da transliramo neku oblast *noise* će imati iste karakteristike [8].

Noise je dobar način za oblikovanje tekstura primitiva jer se na *straightforward* način mogu kreirati željene stohastičke karakteristike u željenim rezolucijama, bez gubljenja kontrole efekta pri translaciji, rotaciji i skaliranju. Ovaj način teksturisavanja se poklapa sa vizuelnim sistemom čoveka koji teži da analizira slike na osnovu nivoa detalja različitih veličina [8].

3.2. Worley noise

Definišaćemo osnovnu ideju koja će se zasnivati na ideji nasumične karakteristične tačke. Za svaku lokaciju koju ćemo obeležiti sa x , postoji neka karakteristična tačka koja je najbliža x od bilo koje druge karakteristične tačke. Definišaćemo i funkciju $F_1(x)$ koja predstavlja distancu od x do najbliže karakteristične tačke. U zavisnosti od promene lokacije x , funkcija $F_1(x)$ će imati blage promene s obzirom da će se distance između uzorkovane lokacije i fiksne lokacije karakteristične tačke, imati blage promene. Ipak, u određenim lokacijama tačka x će biti na jednakoj udaljenosti od dve karakteristične tačke. U ovom slučaju vrednost funkcije $F_1(x)$ je odgovarajuća bez obzira koja tačka je odabrana za računanje distance. Funkcija F_1 će vratiti vrednosti koje su neprekidne, bez obzira na variranje lokacije x . Izvod funkcije F_1 će postati neprekidan kada dođe do „obrta“ proračuna distance između jedne karakteristične tačke i njene okoline [9].

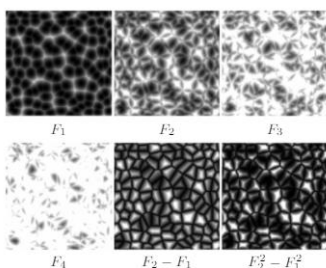
Ovde se može primetiti da se lokacije gde funkcija F_1 „skrene“ od jedne karakteristične tačke na drugu (tamo gde dolazi do neprekidnosti izvoda funkcije) nalaze na površinama koje su jednako udaljene i razdvojene su dvema tačkama u R^3 . Ove površine su upravo površine koje dalje prolaze kroz Voronoi dijagram, koji predstavlja podelu prostora na ćelije, gde su sve tačke svakog regiona bliže tački koja definiše tu tačku, nego bilo kojoj drugoj tački [9].

Definišaćemo funkciju $F_2(x)$ koja predstavlja distancu između lokacije x i druge najbliže karakteristične tačke.

Iz sličnih razloga kao u prethodnom primeru funkcija F_2 je neprekidna, ali njen izvod nije neprekidan, kada dođe do zamene druge najbliže tačke sa prvom ili trećom najbližom tačkom. Slično kao u prethodnom primeru definišaćemo funkciju $F_n(x)$ koja predstavlja distancu između x i n -te najbliže tačke [9].

Funkcija F ima zanimljiva svojstva. F_n je uvek neprekidna. F_n je neopadajuća; $0 \leq F_1(x) \leq F_2(x) \leq F_3(x)$. Uopšteno definicija glasi $F_n(x) \leq F_{n+1}(x)$. Nagib funkcije F_n je jedinični vektor između n -te najbliže tačke i karakteristične tačke x [9].

Linearne ili kombinacije *Worley* funkcija F_n daju veoma interesantne paterne, gde svaki patern ima svoju karakteristiku (slika 2). Kako je F_n zasnovana na distanci, različite metrike se mogu koristiti za dobijanje drugačijih rezultata. Drugi način na koji se može uticati na izmenu funkcije je promena količine *jitter*-a, ili pozicija karakteristične tačke u ćeliji i rešetki [9].



Slika 2. *Worley noise* funkcije kao i kombinacije više *Worley* funkcija daju različite paterne [9]

4. DOP (Dynamic operators)

U *Houdini*-u je implementiran sistem koji se naziva *DOP Network* i služi za simulaciju. *DOP Network* sadrži operacije koje su esencijalne za dinamičke simulacije kao i simulaciju fluida, odeće i interakcije objekata.

U ovom radu je *DOP* sistem za simulaciju korišćen u delu kada stena pod uticajem gravitacije pada na zemlju i dolazi do njenog razaranja.

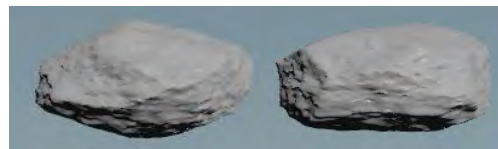
5. UTICAJ DOP, SOP I VOP OPERATORA NA KRAJNI IZGLED EFEKTA

Efekat razaranja koji je odrađen u ovom radu se oslanja na video *Alessandro Nardini*-a u kome objašnjava dinamičku aktivaciju krutih tela (*RBD smart activation*). U ovom poglavlju ćemo testirati neke od *DOP*, *SOP* i *VOP* operatora koji najviše utiču na krajnji rezultat izgleda efekta.

Kao što je već navedeno stena je dobijena proceduralnim modelovanjem pomoću *VOP* operatora. Prilikom modelovanja napravljene su tri promenljive koje kontrolišu izgled stene nakon što su definisani *Worley noise* i *Perlin noise*, a to su sledeći parametri:

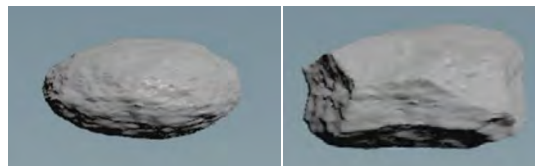
1. *offset* – koji definiše izgled ivica stena
2. *scale* – koji definiše zaobljenost stene
3. *amplitude* – koji definiše veličinu *bump*-ova.

Na slici 3 je prikazan izgled stene usled promene parametra *offset*. Promenom parametra dolazi do pomeranja ivica za vrednost koja je zadata, tako da vrednost parametra zavisi od korisnika odnosno željenog oblika stene.



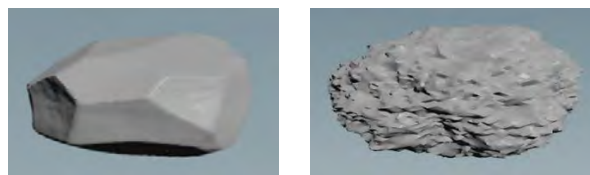
Slika 3. Promena parametra *offset* (levo vrednost iznosi 0,1, dok je desno vrednost 0,8)

Na slici 4 je prikazan uticaj parametra *scale* na izgled stene. Može se videti da za manje vrednosti *scale*-a stena postaje zaobljenija, dok za veće vrednosti ivice na steni su jasnije i oštre.



Slika 4. Uticaj parametra *scale* na izgled stene (levo vrednost je 0,025 dok je desno vrednost 0,8)

Paramter *amplitude* je parametar koji najviše utiče na teksturu stene. Što je ovaj parametar manji stena ima veću uglačanost, i obrnuto (slika 5). Ukoliko ovaj parametar ima preveliku vrednost tekstura na steni postaje neprirrodna i geometrija deluje izlomljeno (slika 5, desno).



Slika 5. Izgled stene za vrednosti *amplitude* 0,025 (levo) i vrednost 1 (desno)

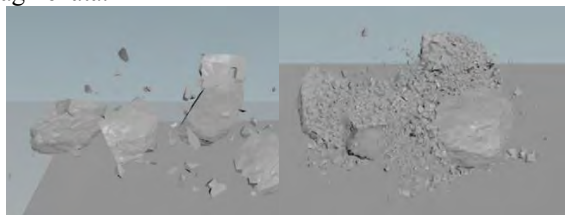
U zavisnosti od korisnikove želje kombinacijom ova tri parametra se mogu kreirati različiti oblici stena. Ova tri parametra su dodata kako bi se na lakši način upravljalo *Worley noise*-om, *Perlin noise*-om i *Displacement*-om. Stoga, važno je napomenuti da ove promene koje su prikazane u prethodnim primerima važe za vrednosti *Worley noise*-a i *Perlin noise*-a, kao i *Displacement*-a koje su zadate prilikom izrade efekta.

Sledeći operator je *Voronoi fracture*, koji kako je već navedeno spada u grupu *SOP* operatora i ima najveći uticaj na izgled stene nakon razaranja. S obzirom da se radi o objektu koji je od čvrstog materijala, nakon razaranja biće podeljen u manje delova nego što bi to bio slučaj sa objektom koji je napravljen na primer od stakla. Na slici 6 su prikazane reference iz gde je došlo do frakcije stakla i stene gde se može uočiti razlika u njihovom izgledu nakon frakcije. Na osnovu posmatranih referenci je odlučeno da u ovom radu stena bude podeljena na četiri dela nakon frakcije.



Slika 6. Razlika frakcije kod različitih materijala

Nakon što smo stenu podelili na četiri dela dolazi do dalje fragmentacije ta četiri dela, kako bi se formirali manji delovi koji predstavljaju čestice prašine da bi efekat bio uverljiviji. U ovom radu broj tih manjih delova iznosi 10000, što je bilo optimalno u skladu sa performansama računara na kom je rađen projekat. Ovaj broj je dao dobar rezultat, gde je vreme za generisanje svih delova bilo optimalno. Može se zaključiti da je za manji broj fragmentisanih delova potrebno manje vremena za procesiranje tih delova, i obrnuto. Zato pri odabiru broja fragmenata treba obratiti pažnju na krajnju svrhu efekta kako se resursi ne bi uzalud trošili. Na slici 7 levo je prikazan izgled razaranja sa manjim brojem fragmenata, dok je sa desne strane izgled efekta sa većim brojem fragmenata.



Slika 7. Efekat razaranja sa manjim i većim brojem fragmenata

Dosadašnji operatori (VOP i SOP) su davali izmene na samoj geometriji objekta, dok su DOP operatori zaduženi za dinamiku kretanja objekta. Pri padu stene deluje samo sila gravitacije. Ukoliko nije dodata još neka sila kao što je na primer sila otpora (*DragForce node*), u *Houdini*-u postoje predefinisani sistemi u sklopu *gravity node*-a koji simuliraju silu gravitacije u vakumu. Podrazumevane vrednosti koje su definisane u sklopu tog sistema su bile odgovarajuće za simulaciju primera koji je prikazan u ovom radu.

Ukoliko bismo imali jasno definisanu sredinu u kojoj se pad dešava, mogli bismo lako prilagoditi dinamiku kretanja stene u skladu sa zadatim uslovima. Takođe, važno je napomenuti još dva parametra koja utiču na kretanje objekta, a to su *Bounce* i *Friction*, koja se nalaze u *Physical menu*-u kada se selektuje ulazna geometrija na kojoj se izvršavaju dinamičke operacije. Parametar *Bounce* nam govori o uticaju podloge na objekat koji pada, dok nam *Friction* govori o podlozi na koju objekat pada, odnosno o sili trenja.

Povećanjem parametra *Bounce* objekat će više odskočiti kada dođe u kontakt sa podlogom, dok ukoliko je vrednost *Bounce* 0 podloga neće imati nikakav uticaj na objekat. *Friction* određuje silu trenja koja deluje na kretanje tela, tako da ukoliko je njena vrednost 0 objekat će nakon pada kliziti po podlozi, što bi moglo da se iskoristi ukoliko telo klizi po ledu, dok za veće vrednosti telo će se teže kretati po podlozi.

6. ZAKLJUČAK

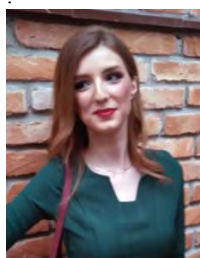
Vizuelni efekti danas imaju veliku primenu u industriji filmova, video igara kao i u reklamama. Oni čine njihov sadržaj interesantnijim i dočaravaju priču koja se priča u datom video sadržaju. Softveri, poput *Houdini*-a, koji omogućavaju njihovo kreiranje, pružaju mnoge benefite ovim industrijama.

Uz pomoć njega se uz manje troškove mogu kreirati raznovrsni vizuelni sadržaji. Mogu se, takođe, otkloniti greške prilikom snimanja. Zatim, alati koji su dostupni u ovim softverima pružaju korisniku kreativnu slobodu tako da je moguće formirati neki jedinstven efekat koji će publici uvek biti prepoznatljiv. Međutim, da bi vizuelni efekti bili uverljiviji i realističniji potrebno je posmatrati svet koji nas okružuje, kao i zakonitosti koja se dešavaju pri njihovim kretanjima. Pomoću DOP, SOP i VOP operatora koji su implementirani u *Houdini*-u je u ovom radu od primitiva sfere najpre izvršeno proceduralno modelovanje i od sfere je dobijen model stene, a zatim pomoću sistema za simulaciju koji su definisani u *Houdini*-u i koji poštuju zakone fizike, izvršena simulacija pada stene i njena fragmentacija nakon kontakta sa podlogom.

7. LITERATURA

- [1] V. Saario, „Visual Effects in SideFX Houdini,“ 2019.
- [2] H. Jia, „THE EXPLOSION EFFECT: A Custom Volume,“ avgust, 2012.
- [3] https://en.wikipedia.org/wiki/Signed_distance_function Pristupljeno: septembar. 2020
- [4] S. Sietzen, „A Method for Automatically Animating the Reassembly of,“ Institute of Computer Graphics and Algorithms, Vienna / Austria, 2017.
- [5] Shao-Xiong Zhang, „Physical Solid Fracture Simulation Based on Random Voronoi Tessellation,“ International Conference on Computer Engineering and Information Systems, China, 2016.
- [6] A. Agrotis, „A Fluid Implicit Particle (FLIP) Solver Built in Houdini,“ Bournemouth University, NCCA, Avgust, 2016.
- [7] E. Goffredo, „A tool for procedural destruction in Houdini,“ N.C.C.A Bournemouth University, 20 Avgust, 2010.
- [8] Ken Perlin, Eric M. Hoffert, „Hypertexture,“ Siggraph, Boston, Jul, 1989.
- [9] S. Worley, „A Cellular Texture Basis Function“.

Kratka biografija:



Aleksandra Stolić rođena je 13.12. 1993. godine u Prištini. Osnovnu školu „Svetozar Marković Toza“ završila je u Novom Sadu. Zatim, je pohađala Gimnaziju „Svetozar Marković“, opšti smer, takođe u Novom Sadu. Osnovne studije Animacija u inženjerstvu na Fakultetu tehničkih nauka završava 2016. godine u Novom Sadu. Od oktobra 2016. god. do decembra 2019. godine je radila u *Eipix* studiju, kao *level designer*, radeći na HOPA igricama. Od decembra 2019. godine počinje da radi kao *Technical artist* u *Playrix.rs* studiju