Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/06SA01Sunjka

PROCEDURALNO KREIRANJE RIG-A REALISTIČNOG LJUDSKOG LICA

PROCEDURAL RIGGING OF A REALISTIC HUMAN FACE

Nenad Šunjka, Lidija Krstanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSKA GRAFIKA

Kratak sadržaj – *U ovom radu opisan je koncept proceduralnog kreiranja rig-a realističnog ljudskog lica koji je u potpunosti automatizovan. Analizirana je i implementacija ovakvog sistema koji je razvijen unutar softvera Maya koristeći Python programski jezik. Rezultat je aplikacija pod nazivom Rigen koja nudi korisniku da u šest koraka samostalno kreira rig proizvoljnog karaktera i time ga pripremi za računarsku animaciju. Koraci su: kreiranje glavnog direktorijuma; definisanje modela karaktera; kreiranje digitalnog skeleta; kreiranje rig-a; korekcija oblika kontrolnih objekata; i zadavanje težinskih koeficijenata koji odgovaraju uticaju određenih kostiju na čvorove. Ti koraci su naizgled jednostavni, ukoliko se posmatraju iz ugla manuelnog kreairanja rig-a. Ipak, kao takvi, oni grade stabilan okvir koji omogućava proceduralnost čitavog procesa. Potpuna automatizacija ovog procesa se ogleda u koraku kreiranja digitalnog skeleta pomoću veštačke inteligencije. Za razliku od svih komercijalnih softvera, gde je neophodno manuelno kreirati digitalni skelet za lice karaktera, u ovom slučaju to se postiže automatski korišćenjem unapred istrenirane neuronske mreže.*

Ključne reči: *Rigging, rigging lica, rigging karaktera, veštačka inteligencija, računarska grafika*

Abstract – *This paper describes the concept of procedural rigging of a realistic human face that is completely automated. The implementation of such a system, which was developed within the Maya software using the Python programming language, was also analyzed. The result is an application called Rigen which offers the user to build a rig for arbitrary character in six steps, and thus prepare it for computer animation. The steps are: creating a directory; defining character models; creating a digital skeleton; build a rig; correction of control shape objects; and assigning weights corresponding to the influence of certain bones on vertices. These steps are seemingly straightforward, if viewed from the angle of manual crew rig. Nevertheless, as such, they build a stable framework that allows the procedurality of the entire process. Complete automation of this process is reflected in the step of creating a digital skeleton using artificial intelligence. Unlike all commercial software, where it is necessary to manually create a digital skeleton for face, in this case, this is achieved automatically by using the pre-trained neural network.*

Keywords: *Rigging, facial rigging, character rigging, artificial intelligence, computer graphics*

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bila dr Lidija Krstanović, docent.

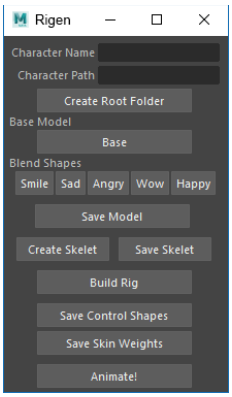
1. UVOD

Kreiranje *rig*-a je proces kreiranja skupa hijerarhijski povezanih kostiju modela koji omogućava dalju animaciju tog modela. Prednosti proceduralnog kreiranja ovakvog sistema su velika ušteda vremena, dobijanje na preciznosti i manja verovatnoća pravljenja grešaka prilikom manuelnog kreiranja *rig*-a. Glavni koraci u postupku su definisanje osnovnog modela, kreiranje digitalnog skeleta i kreiranje hijerarhijske strukture kontrolnih objekata koji će služiti za animaciju. Akcenat je stavljen na rešavanje problema dobijanja digitalnog skeleta.

S obzirom da ovakav sistem treba da kreira *rig* za modele različitih karakteristika, kao što su pol, uzrast, rasa i slično, neophodno je u njega integrisati i algoritme za detekciju obeležja ljudskog lica. Za potrebu detekcije obeležja ljudskog lica je moguće koristiti istreniranu neuronsku mrežu, kao što je to rađeno u [1]. U navedenom radu su koristili *Python* programski jezik, kao i biblioteke *OpenCV* i *dlib* sa istreniranim prediktorom. Detekcija obeležja lica se u ovom radu svodi na tri koraka: renderovanje 3D modela, detekcija obeležja lica na 2D slici, dobijanje odgovarajućih obeležja lica u 3D prostoru na osnovu dobijenih 2D obeležja lica. Korišćenjem tako dobijenih 3D koordinata, kreiraće se kosti digitalnog skeleta.

2. IMPLEMENTACIJA

Na slici 1, prikazan je grafički korisnički prozor (eng. *Graphical User Interface, GUI*) aplikacije *Rigen*. Izbor dugmadi u korisničkom prozoru kreiran je za potrebe ovog rada.



Slika 1. *Rigen* GUI

Ceo razvoj *Rigen* aplikacije rađen je u softveru *Eclipse Oxygen*. *Eclipse* je integrisano razvojno okruženje – IDE (eng. *Integrated Development Environment*) koje se koristi u programiranju i najčešće se koristi Java IDE. Sadrži osnovni radni prostor i proširivi sistem za

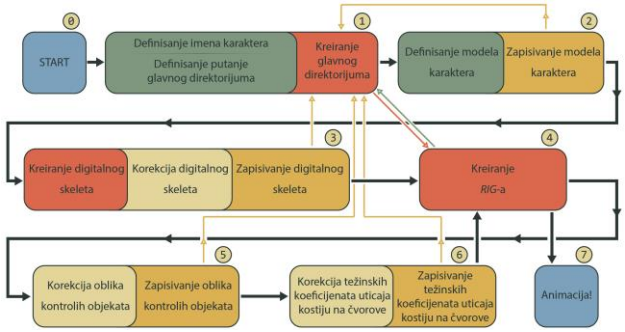
uključivanje dodatnih komponenti integrisanog radnog okruženja. U ovom radu, okruženje je prilagođeno za programski jezik *Python*. U tebali 1 je prikazana struktura `rigen` paketa i modula koji se nalaze u njemu. Paket je podeljen u podpakete koji nose nazive: `anim`, `character`, `data`, `rig`, `temp`, `tools` i `ui`.

rigen						
anim	character	data	rig	temp	tools	ui
__init__ animtools	__init__ model namepath	__init__ rigenData	__init__ buildRig	tempScript tempTools	__init__ conrols deform joint root skelet transform vector	__init__ makeGui

Tabela 1. Struktura paketa `rigen`

3. KONCEPT

Na slici 2, nalazi se grafički prikaz potrebnih koraka kao i njihov redosled, kroz koje korisnik treba da prođe koristeći *Rigen* aplikaciju, kako bi samostalno kreirao *rig* za svog karaktera. Glavni koraci su prikazani u celinama, koje se sastoje od jednog ili više koraka i numerisani su brojevima od 0 do 7. Nulti i sedmi korak su prikazani samo iz razloga kako bi se obuhvatila šira slika čitavog procesa, odnosno početak i kraj. Oni se svode na pokretanje *Rigen* aplikacije i prekid njenog rada. Stoga, primećuje se da se ceo postupak može izvesti u svega 6 koraka.



Slika 2. Grafički prikaz koraka potrebnih za kreiranje *rig*-a

Korak ⓪ predstavlja pokretanje aplikacije, odnosno *Rigen* korisničkog prozora.

U koraku ① korisnik definiše ime karaktera i apsolutnu putanju kako bi se na osnovu tih podataka kreirao glavni direktorijum. U njemu će se zapisivati svi potrebni elementi za *rig*.

Korak ② predstavlja definisanje modela karaktera u sceni gde korisnik bira odgovarajući 3D model, koji će predstavljati digitalnog karaktera, a nosiće ime koje je prethodno definisano. Zatim, definisan model karaktera se zapisuje u odgovarajući folder koji se nalazi u glavnom direktorijumu.

U koraku ③ se pomoću mašinskog učenja, koristeći *dlib* biblioteku, detektuju ključne tačke lica na osnovu kojih se kreira digitalni skelet za zadati model. Ukoliko ima potrebe, korisnik može da koriguje poziciju digitalnih kostiju. Najčešće je to slučaj sa vilicom i očima, odnosno kapcima. Nakon toga, vrši se zapisivanje digitalnog skeleta u odgovarajući folder koji se nalazi u glavnom direktorijumu.

To nas dovodi do najznačajnijeg koraka - kreiranje *rig*-a. Korak ④ podrazumeva preuzimanje svih zapisanih elemenata iz glavnog direktorijuma i građenje hijerarhijske strukture kontrolnih objekata koji su u relaciji sa digitalnim skeletom.

Korak ⑤ namenjen je za korekciju oblika kontrolnih objekata s obzirom da se oni generišu na osnovu pozicije i veličine kostiju digitalnog skeleta u prethodnom koraku. U nekim slučajevima, potrebno ih je prilagoditi modelu karaktera, nakon čega se oni zapisuju u odgovarajući folder u glavnom direktorijumu.

Korak ⑥ odnosi se na korekciju težinskih koeficijenata koji odgovaraju uticaju određene kosti na čvorove. U koraku ④ će se iskoristiti dokument koji je dobijen u koraku ③ u kom se nalaze informacije o težinskim koeficijentima koji odgovaraju uticaju određenih kostiju digitalnog skeleta na čvorove modela odgovarajuće topologije. Ukoliko korisnik želi da kreira *rig* karaktera drugačije topologije, moraće manuelno da koriguje pomenute težinske koeficijente. Nakon zapisivanja te korekcije u pomenuti dokument, potrebno je ponovo pokrenuti četvrti korak da bi se kreirao novi *rig*. Upravo u ovome se ogleda proceduralnost čitavog procesa.

Korak ⑦ prekida rad *Rigen* aplikacije i time se završava proces kreiranja *rig*-a što znači da je karakter spreman za animaciju.

4. KORAK ①: KREIRANJE GLAVNOG DIREKTORIJUMA

Glavni direktorijum predstavlja lokalno mesto u memoriji gde će se zapisivati svi elementi koji su potrebni za kreiranje *rig*-a. Kada se kaže elementi, to se odnosi na model karaktera, digitalni skelet, oblik kontrolnih objekata, težinske koeficijente koji odgovaraju uticaju određenih kostiju na čvorove, teksture itd. Sam naziv glavnog direktorijuma će biti isti kao i dodeljeno ime karaktera i nalaziće se na određenoj putanji. Njegova struktura je prikazana na slici 3.

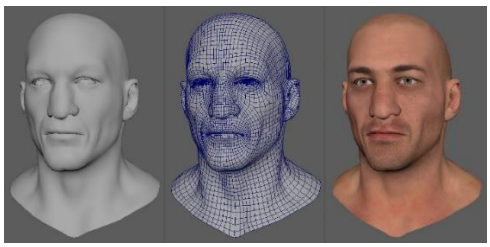
Slika 3. Kreiran glavni direktorijum

5. KORAK ②: DEFINISANJE MODELA KARAKTERA

Nakon kreiranja glavnog direktorijuma, potrebno je definisati model karaktera. Model karaktera mogu da kreiraju tehnički umetnici ili da se dobije fotogrametrijom[1].

Na slici 4, prikazan je model karaktera koji će se koristiti kao primer u daljem tekstu.

[1] Fotogrametrija je veština, nauka i tehnologija dobijanja pouzdanih kvantitativnih informacija o fizičkim objektima procesom beleženja, merenja, analiziranja i interpretacije fotografskih snimaka.

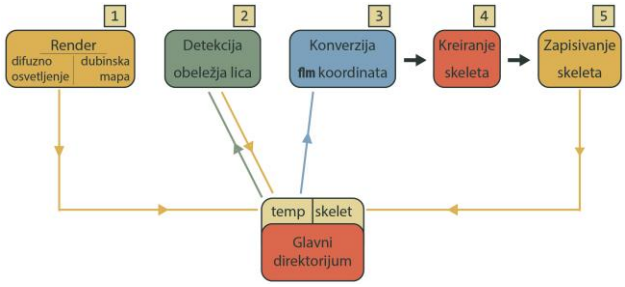


Slika 4. Model karaktera [4]

Definisanje i zapisivanje modela karaktera je postignuto korišćenjem metoda `setModel()` i `exportModel()`. Obe metode se nalaze u modulu rigen.character.model.

6. KORAK ③: KREIRANJE DIGITALNOG SKELETA

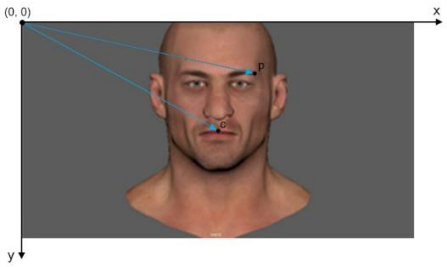
Za potrebe korisnika radi formiranja digitalnog skeleta, korisnički prozor sadrži dugmad *Create Skelet* i *Save Skelet* (slika 1). Ove opcije se sastoje iz pet koraka koji su šematski prikazani na slici 5.



Slika 5. Kreiranje i zapisivanje digitalnog skeleta

Korak 1 podrazumeva formiranje slike koja se dobije kada se koristi samo difuzno osvetljenje, kao i dubinske mape [2] za dati render. Ove dve slike se zapisuju u privremeni folder (eng. *temp folder*). Zatim se u koraku 2 vrši detekcija (x, y) obeležja lica na slici koja se dobije kada se koristi samo difuzno osvetljenje. Tako dobijene (x, y) koordinate, tzv. `flm` koordinate, (eng. *Facial LandMarks coordinates*) se zapisuju u privremeni folder. Za pronalaženje obeležja modela našeg karaktera u 3D prostoru, neophodno je da se izračuna i z koordinata pronađenih obeležja. Ona se dobija kao vrednost piksela na dubinskoj mapi sa pozicije (x, y) prethodno pronađenih obeležja. Koordinate (x, y) pronađenih obeležja, kao i z koordinata, dobijaju se korišćenjem *OpenCV* i *dlib* biblioteke. Problem je što se koordinati sistem *OpenCV*-a razlikuje od koordinatnog sistema *Maya*-e u kom se nalazi model našeg karaktera. Zato se u koraku 3 radi konverzija ovako dobijenih koordinata. Naime, *OpenCV* i *Maya* imaju drugačiju orijentaciju koordinatnih osa. Iako su oba koordinatna sistema Dekartova, y osa u *Maya*-i je usmerena prema gore, dok je y osa u *OpenCV*-u usmerena prema dole. Pored toga, ova dva koordinatna sistema imaju drugačije dužine jediničnih duži. Iz tih razloga je potrebno uraditi konverziju koordinata iz jednog koordinatnog sistema u drugi kao što je to objašnjeno u nastavku teksta.

[2] Slika koja se sastoji od piksela definisanih vrednostima od `0` do `255`. Vrednost `0` označava najudaljenije mesto u 3D sceni, dok vrednost `255` označava najbliže mesto u 3D sceni.



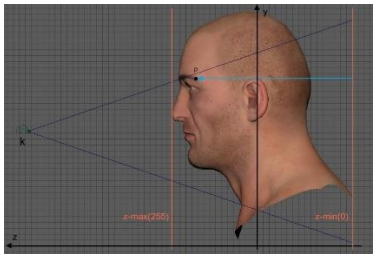
Slika 6. *OpenCV* koordinatni sistem slike

Na osnovu prethodne slike, `flm` koordinate za obeležje p se preračunavaju sledećom formulom:

$$p_x = (flm_x * ratio - c_x) * p_f,$$
$$p_y = (c_y - flm_y * ratio) * p_f + k_y.$$

gde su:

$$c_x = \frac{width}{2},$$
$$c_y = \frac{height}{2},$$
$$p_f = 0.05$$
$$k_y = \frac{modelHeight}{2}$$ (pozicija kamere na y osi).



Slika 7. Preračunavanje `z` koordinate

S obzirom da je z koordinata dobijena sa dubinske mape, ona se preračunava formulom:

$$p_z = flm_z * z_f + m_z,$$

gde je

$$z_f = \frac{modelZdepth}{255.0}.$$

Kada su `flm` koordinate preračunate, u koraku 4 je moguće kreirati kosti digitalnog skeleta. Nakon toga, korisnik može u sceni da proveri da li digitalni skelet odgovara modelu karaktera i da, ukoliko je potrebno, izvrši korekcije. U koraku 5 se skelet zapisuje u folder *skelet*.

7. KORAK ④: AUTOMATSKO KREIRANJE *RIG*-a

Nakon što su zapisani model karaktera i digitalni skelet, moguće je kreirati *rig*. Ovo predstavlja najkompleksniji korak u čitavom procesu. Iz tog razloga, za potrebe ovog rada, napisan je novi modul `buildRig.py`. Ovaj modul sadrži 11 klasa i desetine metoda koje služe za preuzimanje svih zapisanih dokumenata i formiranje hijerarhijske strukture koja sadrži model karaktera, digitalni skelet, kontrolne objekte i težinske koeficijente koji odgovaraju uticaju određenih kostiju na čvorove. Modul `buildRig.py` se nalazi u `rigen.rig` modulu. Klase koje se nalaze u modulu `buildRig.py` su:

- `NewScene`
  - Kreira se nova scena u softveru *Maya*. Učitavaju se model karaktera i digitalni skelet. Takođe, izvršava se provera da li postoje zapisani dokumenti sa podacima o težinskim

koeficijentima koji odgovaraju uticaju određenih kostiju na čvorove i obliku kontrolnih objekata. Kreiranje nove scene, vrši se metodom `buildRigen`.

- `GlobalControl`
  - Kreira se globalna kontrola. Ona je najuočljivija kontrola u *rig*-u s obzirom da je odgovorna za transliranje, rotiranje i skaliranje čitavog karaktera tokom animacije.
- `Neck`
  - Kreiraju se tri kontrole za vrat. One služe za rotiranje vrata. Prva kontrola se vezuje za globalnu kontrolu.
- `Head`
  - Kreira se kontrola za glavu. Ova kontrola služi za rotaciju glave. Kontrola glave se vezuje za poslednju kontrolu vrata.
- `Ears`
  - Kreiraju se kontrole za uši. Njihova funkcionalnost je ograničena translacija i rotacija ušiju. One se vezuju za kontrolu glave.
- `Jaw`
  - Kreira se kontrola vilice. Služi za otvaranje usta, odnosno rotaciju i translaciju vilične kosti.
- `Lips`
  - Kreiraju se četiri kontrole usana. Dve se nalaze u uglovima usana, dok su druge dve na sredini gornje i donje usne. Vezuju se za kontrolu vilice i kontrolu glave.
- `Cheeks`
  - Kreiraju se kontrole obraza. Služe za korekciju facijalnih ekspresija prilikom otvaranja usta, osmeha i slično. Vezuju se za kontrolu vilice i kontrolu glave.
- `Eyes`
  - Kreiraju se kontrole očiju i kapaka. Njima se kontroliše pravac posmatranja očiju, otvaranje i zatvaranje kapaka - treptanje. Ove kontrole se vezuju za globalnu kontrolu i kontrolu glave.
- `Eyebrows`
  - Kreira se kontrole obrva. Služe za podizanje i spuštanje obrva. Vezuju se za kontrolu glave.
- `Nose`
  - Kreiraju se kontrole nosa i nozdrva. Njima se postiže ograničena rotacija i translacija segmenata nosa kao što su koren nosa, vrh nosa i nozdrve. Kontrole nosa se vezuju za kontrolu glave.

8. KORAK ⑤: KOREKCIJA OBLIKA KONTROLNIH OBJEKATA

Nakon što se *rig* kreira, u sceni će biti prikazan kompletan sistem za animaciju. U tom trenutku je moguće korigovati izgled kontrolnih objekata. Napisana je metoda koja zapisuje izgled kontrolnih objekata u `.csv` document – `cs` dokument (`cs` - eng. *Curve Shapes*).
Klikom na dugme *Save Control Shapes*, poziva se metoda `exportCurveShapes()`.

9. KORAK ⑥: KOREKCIJA TEŽINSKIH KOEFICIJENATA KOJI ODGOVARAJU UTICAJU ODREĐENIH KOSTIJU NA ČVOROVE

Metoda `buildRigen()` svakim pokretanjem izvršava proveru da li postoji dokument sa zapisanim težinskim koeficijentima koji odgovaraju uticaju određenih kostiju na čvorove – `sw` dokument (`sw` – eng. *Skin Weights*). Ukoliko takav dokument ne postoji u trenutku kreiranja *rig*-a, modelu karaktera neće biti dodeljen *SkinCluster*[3]. Ovo bi značilo da, nakon kreiranja *rig*-a, korisnik mora manuelno da zadaje težinske koeficijente koji odgovaraju uticaju određene kosti na čvorove. U slučaju da korisnik manuelno definiše ovu deformaciju, ili je samo koriguje, nakon toga će koristiti metodu `exportSkinWeights()` koja kreira `sw` document sa `.csv` ekstenzijom.

[3] *SkinCluster* je element (eng. *node*) koji povezuje temena (eng. *vertex*) 3D modela sa digitalnim skeletom.

10. ZAKLjUČAK

Aplikacija *Rigen* se pokazala kao veoma stabilno okruženje koje na intuitivan način vodi korisnika do krajnjeg proizvoda, a to je kreiran *rig* koji omogućava dalju računarsku animaciju. Implementacija ovakvog sistema je rezultat dvogodišnjeg izučavanja objektno orijentisanog programiranja u *Python* programskom jeziku i upoznavanja sa `maya.cmds` modulom.
Ipak, postoji i slaba strana pomenutog rešenja. Radi se o metodi `importSkinWeights` koja učitava težinske koeficijente koji odgovaraju uticaju određene kosti na čvorove.

11. PRAVCI DALjIH ISTRAŽIVANjA

Naredni korak bi mogla da bude optimizacija metode `importSkinWeights()` kako bi se smanjilo utrošeno vreme koje je potrebno da se pojedinačno učitaju težinski koeficijenti koji odgovaraju uticaju određenih kostiju na čvorove. Trenutno, ovo vreme iznosi približno 15 minuta, što ujedno predstavlja i najduži vremenski period u okviru čitavog procesa. Razlog za tako nešto je to što je metoda napisana *Brute Force* algoritmom. Sa druge strane, pokazano je da je moguće kreirati aplikaciju koja služi za automatsko kreiranje *rig*-a realističnog ljudskog lica, ali to ne mora da bude jedini slučaj. Ovakav sistem bi mogao da se implementira za kreiranje *rig*-a kompletnog ljudskog tela koristeći druge biblioteke koje sadrže unapred istrenirane neuronske mreže za estimaciju poza.

12. LITERATURA

[1] V. Kazemi, J. Sulivan, *One Millisecond Face Alignment with an Ensemble of Regression Trees*, Stockholm, 2015.
[2] L. Dutreve, A. Meyer, V. Orvalho, S. Bouakaz, *Easy Rigging of Face by Automatic Registration and Transfer of Skinning Parameters*, HAL, 2017.
[3] P. Viola, M. Jones, *Robust Real-Time Face Detection*, Cambridge, 2003.
[4] Besplatan model karaktera preuzet sa adrese: www.turbosquid.com, pristupljeno: 3.7.2018.

Kratka biografija:

Nenad Šunjka je rođen 25.9.1993. godine u Novom Sadu. Osnovnu školu „Danilo Zelenović" završio je u Sirigu, gde je i odrastao. Završio je Gimnaziju „Jovan Jovanović Zmaj" prirodno-matematičkog smera u Novom Sadu. Osnovne studije Animacije u inženjerstvu na Fakultetu tehničkih nauka u Novom Sadu završio je 2016. godine.
kontakt: sunjkanenad@gmail.com

Lidija Krstanović završila je osnovne i master studije na Prirodno-matematičkom fakultetu u Novom Sadu, smer profesor matematike. Doktorirala je 2017. godine na Fakultetu tehničkih nauka, smer Matematika u tehnici. Godine 2018. izabrana je u zvanje docenta na istom fakultetu na katedri za Animaciju u inženjerstvu.