

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1243

Virtualna šminka

Margareta Kušan

Zagreb, lipanj 2016.

Zagreb, 11. ožujka 2016.

DIPLOMSKI ZADATAK br. 1243

Pristupnik: **Margareta Kušan (0036463538)**

Studij: Računarstvo

Profil: Računarska znanost

Zadatak: **Virtualna šminka**

Opis zadatka:

Proširena stvarnost je skup tehnologija kojima se u sliku stvarnog svijeta uključuju virtualni elementi. Pritom je važno da se poravnavanje stvarnih i virtualnih elemenata radi u trodimenzionalnom koordinatnom sustavu te da sustav radi u stvarnom vremenu. Jedan od vidova proširene stvarnosti je iscrtavanje virtualnih elemenata u slici lica, zasnovano na praćenju lica korisnika. Jedna takva primjena je virtualna šminka, što omogućava korisniku ili korisnici virtualno isprobavanje raznih boja i vrsta šminke.

Vaša je zadaća proučiti osnove proširene stvarnosti i sustav praćenja lica dostupan na Zavodu, te na temelju tog sustava implementirati sustav za virtualno isprobavanje šminke.

Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Zadatak uručen pristupniku: 18. ožujka 2016.

Rok za predaju rada: 1. srpnja 2016.

Mentor:

Prof. dr. sc. Igor Sunday Pandžić

Djelovođa:

Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za
diplomski rad profila:

Prof. dr. sc. Siniša Srblijić

Zahvala:

Veliku zahvalu upućujem mentoru prof. dr. sc. Igoru Pandžiću za potporu tijekom razvoja diplomskog rada i cjelokupnog mentorstva tijekom studija. Također zahvaljujem svojoj obitelji te kolegama s FER-a.

Sadržaj

UVOD.....	1
1. Trenutna rješenja na tržištu	2
1.1. Oriflame Makeup Wizard	2
1.2. MakeUp Genius (L'Oreal SA)	2
1.3. YouCam Makeup-Makeover Studio	2
1.4. Modiface Live	3
2. Alati i tehnologije	4
1.1. Visage SDK.....	4
2.2. The Unity Engine (Unity3D).....	5
3. Razvoj i komponente aplikacije	6
3.1. Ruž.....	8
Catmull – Rom algoritam za centripetalnu interpolacijsku krivulju [4].....	9
3.2. Olovka ili tuš za oči (<i>eyeliner</i>).....	13
3.3. Sjenilo	16
3.4. Maskara.....	19
3.5. Rumeno.....	21
3.6. Puder	24
3.7. Grafičko sučelje	25
4. Rezultati.....	26
4.1. Aplikacija.....	26
4.2. Usporedba aplikacije s prethodno opisanim dostupnim rješenjima na tržištu.....	30
4.3. Poboljšanja.....	33
Zaključak	34
Literatura	35
Naslov, sažetak i ključne riječi (HRV i ENG)	36

UVOD

Prvi znaci korištenja šminke isključivo u estetske svrhe datiraju u razdoblje 3000 godina prije nove ere kada su Sumerani koristili drago kamenje te ga usitnjavali i koristili za dekoraciju usana i očiju. Danas je kozmetika prisutna u gotovo svim društvima na Zemlji. Zanimanje vizažista podrazumijeva osobu koja se razumije u alate, boje, teksture i njihove kombinacije kako bi se na svakom licu postigao željeni izgled. Razlog korištenja šminke nije isključivo uljepšavanje, razlozi korištenja šminke mogu biti različiti, od raznih transformacija ili efekata koji se žele postići u kazalištu ili filmskoj industriji do zabave. Ako gledamo šminku kao alat za uljepšavanje, shvaćamo da ljudska lica nisu jednaka stoga ne pristaje svaka šminka svakom licu. Iz tog razloga, ako želimo poboljšati naš izgled te naglasiti ili sakriti određene karakteristike našeg lica, trebamo platiti iskusnog vizažista ili kupiti određeni broj proizvoda te ih isprobavati na vlastitom licu. Zbog financijske nemogućnosti, nedostatka vremena ili iskustva često odustajemo od pokušaja eksperimentiranja s različitim kozmetičkim proizvodima. No što ako umjesto skupih proizvoda možemo koristiti virtualnu šminku te u stvarnom vremenu koristeći mobilnu aplikaciju poput zrcala pronaći naše omiljene kombinacije?

Zbog vrlo visoke potražnje na tržištu te široke interesne skupine koja svakim danom raste, cilj ovog rada je stvoriti aplikaciju koja će omogućiti isprobavanje više komponenata šminke u stvarnom vremenu. Razvoj ovog područja trenutno je u porastu te se sve više različitih besplatnih i komercijalnih rješenja stavlja na tržište aplikacija. U sklopu ovog rada istražuje se tržište takvih aplikacija, dan je pregled tehnologija korištenih prilikom izrade aplikacije te se predstavlja implementirana aplikacija i daje usporedba takve implementacije s ranije opisanim.

1. Trenutna rješenja na tržištu

Aplikacije uljepšavanja trenutno su vrlo popularne na tržištu. Razni filteri i uređivači slika već su dulje vrijeme prisutni na tržištu mobilnih aplikacija. Automatska aplikacija šminke detekcijom lica u slici također je dosta rasprostranjena, ali aplikacije koje koriste praćenje u videu ili u stvarnom vremenu za aplikaciju šminke su relativno nove na tržištu i one su navedene u nastavku.

1.1. Oriflame Makeup Wizard

Oriflame Cosmetics je švedska kompanija za direktnu prodaju kozmetike. Aplikacija je besplatna za korištenje, nudi aplikaciju proizvoda koji se prodaju u Oriflame-ovoj kolekciji The ONE. Nudi aplikaciju sjenila, maskare, tuša, proizvoda za usne i rumenila. Također nudi gotove kombinacije proizvoda (*lookove*) i mogućnost naručivanja proizvoda. Oriflame aplikacija koristi isti Tracker tvrtke Visage Technologies koji je korišten u sklopu ovog rada.

1.2. MakeUp Genius (L'Oreal SA)

Tvrtka koja je izdala ovu aplikaciju je poznata kozmetička kompanija L'Oréal. MakeUp Genius nudi L'Oréalove proizvode od kojih nudi aplikaciju olovke ili tuša za oči, bronzeru, pudera, sjenila, ruža za usne, sjajila za usne i rumenila. Kao i Oriflame Makeup Wizard, nudi gotove kombinacije proizvoda (*lookove*) i mogućnost naručivanja proizvoda.

1.3. YouCam Makeup-Makeover Studio

Dizajnirana od strane tvrtke Perfect Corp. Aplikacija uz šminkanje također nudi niz ostalih opcija poput promjene frizure i boje kose, promjenu boje očiju, aplikaciju raznih filera na sliku, micanje akni, sjaja kože, mijenjanje kontura lica poput nosa i čeljusti, aplikacija *face painta*, dodavanje naočala, ostalih modnih dodataka i slično. Dio alata koji se može koristiti u stvarnom vremenu uz praćenje lica: tekstura maskare ili umjetnih trepavica uz razne oblike i boje, sjenilo uz razne oblike i boje, tuš ili olovka za oči uz razne

oblike, gotove kombinacije šminke (*lookovi*), boja i tekstura usana, oblik i boja rumenila, boja i tekstura šarenica oka.

1.4. Modiface Live

Modiface Live je trenutno dostupna samo za iOS platformu. Modiface je kompanija koja izdaje preko 150 aplikacija za analizu i vizualizaciju karakteristika lica i izgleda. Aplikacije koje izdaju uključuju *makeover* koji sadrži sve komponente, kao i aplikacije za svaku komponentu *makeovera* posebno: promjena boja zjenica, promjena boje kose, virtualna šminka itd.

2. Alati i tehnologije

U sklopu rada koristi se Visage|SDK razvojno okruženje tvrtke Visage Technologies te Unity3D Game Engine uz Monodevelop editor. Teksture su razvijane koristeći Adobe Photoshop CC 2015 i GIMP 2 uređivač slika. Korišten je i Microsoft Visual Studio 2015 za modifikaciju *VisageTrackerUnityPlugina* (C++) te Android Studio 2.0 za izgradnju Android aplikacije. Za manipulaciju 3D modelom lica te pregled teksture 3D modela korišten je Blender. Blender¹ je besplatan i otvoreni kod (*open source*) program za 3D animaciju.

1.1. Visage|SDK

Visage|SDK [1] integrira širi raspon tehnologija računalnog vida za praćenje i analizu lica. Tehnologije su trenutno dostupne u 4 odvojeno licencirana paketa:

1. FaceTrack – 2D i 3D praćenje glave i ključnih točaka lica u videu, uključujući praćenje pogleda očiju (*eye* ili *gaze tracking*)
2. HeadTrack – 2D i 3D praćenje pozicije glave u videu
3. FaceDetect – 2D i 3D detekcija lica i ključnih točaka lica u slikama
4. FaceAnalysis – Detekcija spola i emocija u slikama

Tehnologije podržavaju razne platforme od kojih su za ovaj rad vrlo važne Windows, Android te Unity platforme. Od paketa korišten je FaceTrack paket te *VisageTrackerUnityPlugin wrapper*² koji omogućuje uporabu alata praćenja lica unutar Unity3D editora za jezike Javu i C#. Jezik korišten za izradu aplikacije je C#. Posljednja verzija Visage|SDK koja se koristi u radu je v8.1a2.

Unutar Visage|SDK dostupan je primjer Unity3D scene koja nudi isprobavanje 3D naočala u stvarnom vremenu. Scena se nalazi unutar paketa koji služi kao *demo* verzija za pokretanje Visage *face tracking* softvera unutar Unitya. Koristeći taj paket nadograđuje se projekt virtualne šminke.

¹ <https://www.blender.org/>

² *Wrapper* je 'omotač' koji omogućuje korištenje funkcionalnosti jedne tehnologije unutar druge

2.2. The Unity Engine (Unity3D)

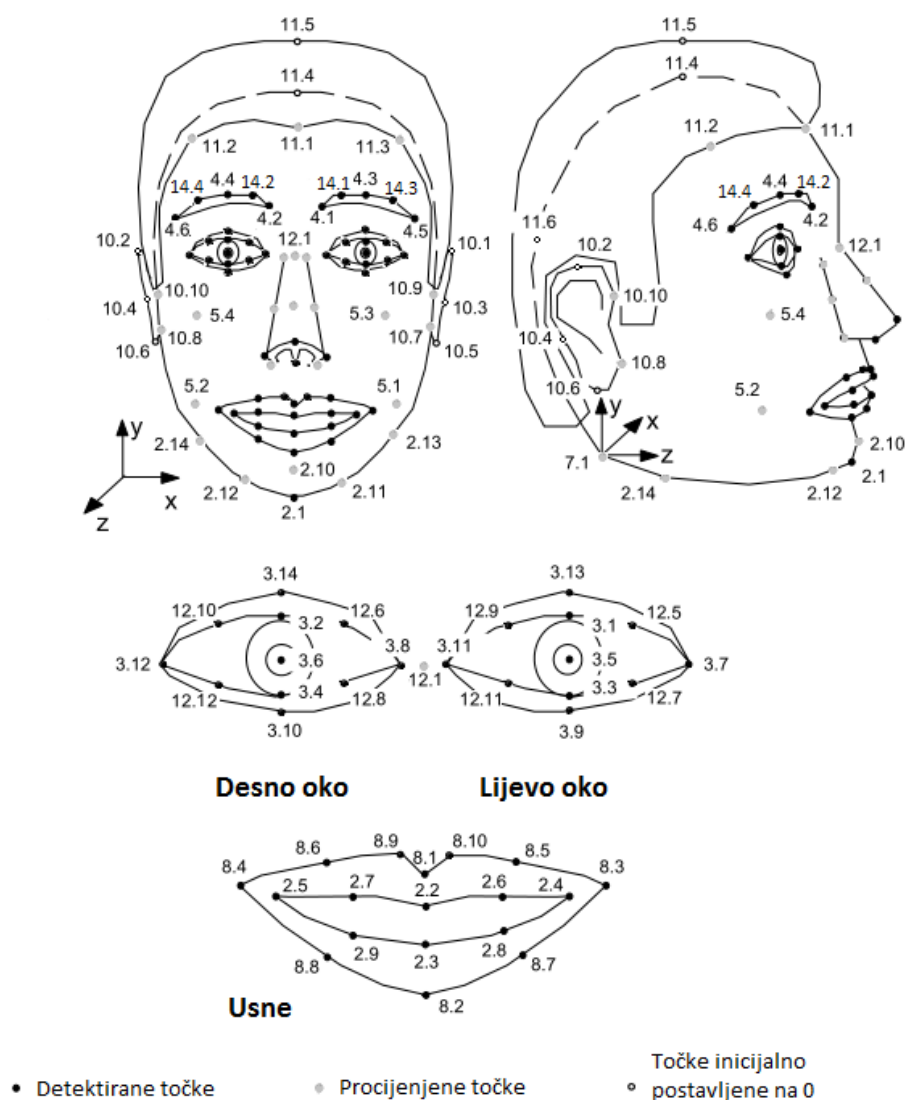
Unity3D je *game engine*³, trenutno najčešće korištena platforma za razvoj igara, interaktivnih 2D i 3D iskustava poput treniranja simulacija medicinskih ili arhitektonskih vizualizacija za stolne, mobilne i web platforme te konzole. Tvrdnja da je trenutno najčešće korištena platforma proizlazi iz činjenice da Unity drži 45% tržišnog udjela za platforme za razvoj igara, a prvi konkurent 17%. Broj razvojnih programera unutar razvoja igara koji koriste Unity iznosi 45%, a broj onih kojima je Unity primaran alat za razvoj igara je 29%. Među korisnicima Unity alata mogu se vidjeti neka poznatija imena poput Cartoon Network, Coca-Cola, Disney, Electronic Arts, LEGO, Microsoft, NASA, Nexon, Nickelodeon, Square, Ubisoft i Warner Bros. [2] Unity nudi opciju izrade 2D i 3D aplikacija, unutar ovog rada koristi se 3D scena.

Korištena verzija Unitya u sklopu ovog rada je 5.3.3.f1.

³ *Game engine* se može prevesti kao pokretač igara.

3. Razvoj i komponente aplikacije

Aplikacija je razvijena za Android i Windows stolne uređaje te omogućuje praćenje i aplikaciju virtualne šminke u stvarnom vremenu. Visage Technologies razvija tehnologiju praćenja lica koja je dostupna za korištenje u sklopu ovog rada. Njihov softver za praćenje lica vraća informacije o karakterističnim točkama lica, rotaciji, translaciji i točkama 3D modela. Te informacije se vraćaju pomoću implementiranih funkcija u *VisageTrackerUnityPlugina* iz paketa *FaceTrack* uz *C#*.



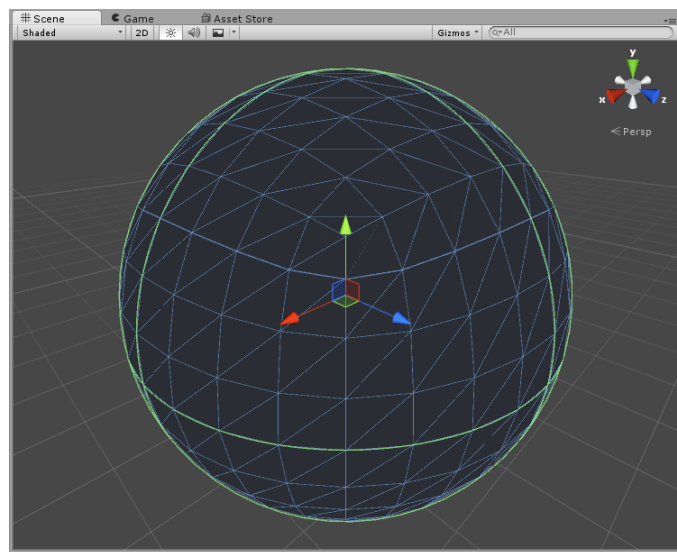
Slika 1. Karakteristične točke lica korištene za apliciranje šminke [3]

Važne funkcije za dohvaćanje informacija:

- `_get3DData` – vraća 3D vrijednosti translacije i rotacije
- `_getFaceModel` – vraća vrhove i trokute, njihov broj i koordinate teksture
- `_getFeaturePoints3DRel` – vraća karakteristične točke lica (slika 1)

Osim navedenih funkcija koje su važne za dohvaćanje informacija o trackeru, iz *VisageTrackerUnityPlugina* se uzimaju ostale funkcije poput funkcije za inicijalizaciju trackera, otvaranje kamere, funkcija za dohvaćanje okvira slike itd.

Aplikacija se sastoji od 6 glavnih komponenti: ruž, olovka/tuš za oči, sjenilo, maskara, rumenilo i puder. Svaka od komponenti je u Unity editoru predstavljena kao jedan *game object*⁴ te sadrži zasebnu istoimenu skriptu unutar koje se provodi manipulacija točkama – dohvaćanje točaka pomoću opisanih funkcija iz *VisageTrackerUnityPlugina*, dodavanje točaka na temelju matematičkih izračuna, promjena lokacije točaka itd. Osim manipulacije točkama, unutar svake skripte izračunava se mreža poligona. Mreža poligona je kombinacija vrhova, rubova i poligona koji definiraju oblik poliedra u 3D grafici i modeliranju.



Slika 2. Mreža poligona jedne komponente koja u Unity sceni stvara sferu

⁴ Svaki objekt u sceni unutar Unity-a je *game object*. Svaki *game object* sadrži specifična svojstva koja ga čine točno određenim objektom – primjerice može reprezentirati drvo, robota ili psa. U daljem tekstu *game object* se referira kao objekt.

Kamere, teksture i svjetla su objekti Unity scene koji su vrlo bitni za svaku aplikaciju. Postavljanje scene u ovoj aplikaciji nije bilo potrebno raditi iz početka zato što Visage Technologies u sklopu svojeg softvera nudi Unity paket koji u Unityu daje scenu virtualnog isprobavanja naočala. S obzirom da je ta scena već podešena za prikaz kamere, tekstura i osvjetljenja, ona je korištena kao podloga za razvoj aplikacije.

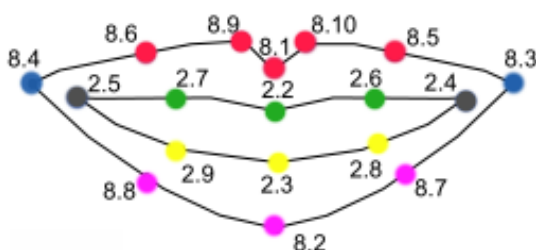
3.1. Ruž

Usne su još u prošlosti bile jedna od najvažnijih elemenata šminke. Danas postoje razne vrste proizvoda za usne koje služe njezi, zaštiti i stvaranju efekta željenog izgleda. Tako danas postoje balzami za njegu, ruževi za usne različitih vrsta, sjajila, olovke za usne itd. Stoga je ruž za usne neizostavna komponenta aplikacije za virtualnu simulaciju šminke.

Korak 1 – Dohvat točaka iz Trackera i priprema varijabli

Karakteristične točke usana dohvaćaju se iz Visage Trackera (*VisageTrackerUnityPlugin*). Takvih točaka ima 18 te se nalaze na slici 1. – 'Usne'. Ove točke dijele se na 4 seta točaka:

1. gornja usna gornji rub: 7 točaka – 8.4, 8.6, 8.9, 8.1, 8.10, 8.5 i 8.3,
2. gornja usna donji rub: 7 točaka – 8.4, 2.5, 2.7, 2.2, 2.6, 2.4, 8.3,
3. donja usna gornji rub: 7 točaka – 8.4, 2.5, 2.9, 2.3, 2.8, 2.4, 8.3,
4. donja usna donji rub: 5 točaka – 8.4, 8.8, 8.2, 8.7, 8.3.



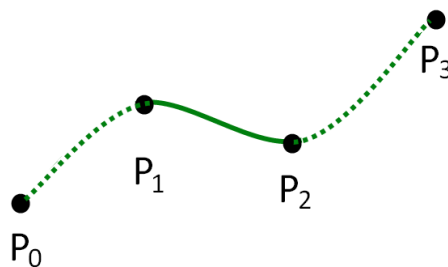
Slika 3. Karakteristične točke usana: 1 – gornja usna gornji rub: 5 crvenih + 2 plave točke, 2 – gornja usna donji rub: 3 zelene + 2 sive + 2 plave točke, 3 – donja usna gornji rub: 3 žute + 2 sive + 2 plave točke, 4 – donja usna donji rub: 3 ljubičaste + 2 plave točke.

Skripta za usne sadrži 4 *float* varijable – pomak debljine gornje i pomak debljine donje usne te dvije vrijednosti *ratio1* i *ratio2* koje predstavljaju broj točaka koje se dodaju prema Catmull – Rom⁵ krivulji između svake dvije karakteristične točke seta. *Ratio1* primjenjuje se na prva tri seta točaka, a *ratio2* na posljednji set točaka (donja usna donji rub). Vrijednosti *ratio1* i *ratio2* odabiru se tako da razlika konačnog broja točaka trećeg i četvrtog seta, kad im se pridodaju dodatne točke krivulje, bude što manja. U finalnoj verziji aplikacije, ove vrijednosti postavljene su na *ratio1* = 3 i *ratio2* = 5. Ova razlika bitna je zbog načina iscrtavanja (popunjavanja) usana. Vrijednosti pomaka debljine eksperimentalno su uvedene te se u konačnoj verziji koristi samo pomak debljine gornje usne koji iznosi 0.0008, dok je pomak debljine donje usne postavljen na 0.

Korak 2 – Računanje okvira usana

Nakon što su poznate 3D koordinate setova točaka koje dobivamo pozivom funkcije u VisageUnityTrackerPlugin-u, šalju se lokacije točaka jednog seta i vrijednosti *ratio1* ili *ratio2*, ovisno o setu, u funkciju *CalcSpline* koja pomoću Catmull – Rom krivulje izračunava zadan broj točaka i vraća sve njihove koordinate.

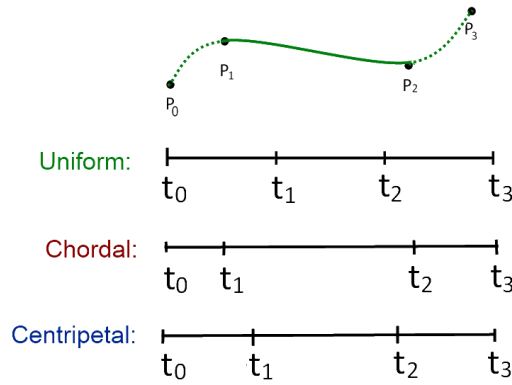
Catmull – Rom algoritam za centripetalnu interpolacijsku krivulju [4]



Slika 4. Catmull-Rom interpolacijska krivulja između 4 kontrolne točke [4]

Catmull-Rom interpolacijska je krivulja koja prolazi kroz 4 kontrolne točke, P0-P3 (slika 4). Krivulja se iscrtava samo između točaka P1 i P2.

⁵ Catmull-Rom krivulja – autori Edwin Catmull i Raphael Rom [4]



Slika 5. Parametrizacija čvorova za Catmull – Rom algoritam

Prema definiciji, C je segment krivulje (na slici 4. segment između P_1 i P_2), $t_0 - t_3$ su sekvence čvorova (slika 5), $\alpha \in [0, 1]$, $i = 0, 1, 2, 3$, $t_0 = 0$.

$$\mathbf{C} = \frac{t_2 - t}{t_2 - t_1} \mathbf{B}_1 + \frac{t - t_1}{t_2 - t_1} \mathbf{B}_2$$

gdje

$$\mathbf{B}_1 = \frac{t_2 - t}{t_2 - t_0} \mathbf{A}_1 + \frac{t - t_0}{t_2 - t_0} \mathbf{A}_2$$

$$\mathbf{B}_2 = \frac{t_3 - t}{t_3 - t_1} \mathbf{A}_2 + \frac{t - t_1}{t_3 - t_1} \mathbf{A}_3$$

$$\mathbf{A}_1 = \frac{t_1 - t}{t_1 - t_0} \mathbf{P}_0 + \frac{t - t_0}{t_1 - t_0} \mathbf{P}_1$$

$$\mathbf{A}_2 = \frac{t_2 - t}{t_2 - t_1} \mathbf{P}_1 + \frac{t - t_1}{t_2 - t_1} \mathbf{P}_2$$

$$\mathbf{A}_3 = \frac{t_3 - t}{t_3 - t_2} \mathbf{P}_2 + \frac{t - t_2}{t_3 - t_2} \mathbf{P}_3$$

$$t_{i+1} = \left[\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \right]^\alpha + t_i$$

Funkcija *CalcSpline* vraća sve točke krivulje – detektirane karakteristične točke i dodatne točke dobivene korištenjem Catmull-Rom krivulje.

Broj točaka n jedne linije (gornja ili donja usna – gornji ili donji rub) iznosi:

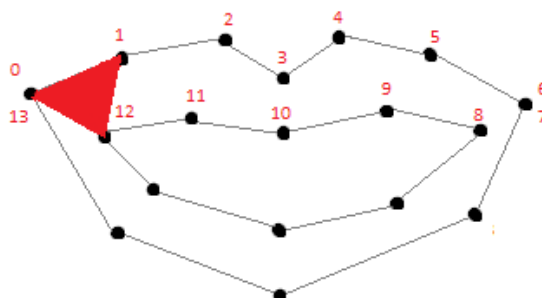
$$n = n_{set} + (n_{set} - 1) * ratiox,$$

gdje je n_{set} broj točaka u setu (7 ili 5), a $ratiox$ je $ratio1$ (set 1, 2, 3) ili $ratio2$ (set 4).

Nakon što su izračunate 4 krivulje – gornji i donji rub gornje i donje usne, koordinate točaka mijenjaju se prema debljini usana na način da se y osi svake točke gornje usne (osim prvoj i zadnjoj točki) dodaje vrijednost *float*a varijable debljine gornje usne. Isti postupak se vrši s donjom usnom, samo što se umjesto dodavanja oduzima vrijednost *float*a varijable debljine donje usne. Sada je poznat okvir usana – vrhovi i njihove koordinate.

Korak 3 – Konstrukcija mreže iz dobivenih točaka

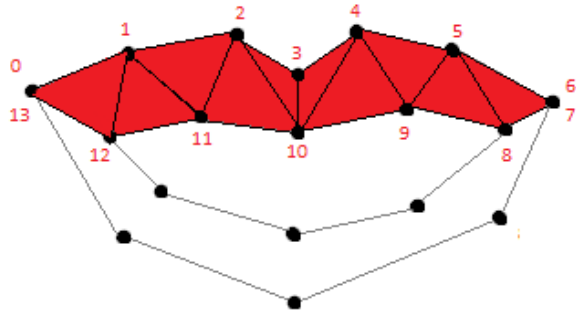
Sljedeći korak je konstrukcija mreže usana. Na svaki objekt u Unityu mogu se primijeniti mreže (*meshes*). Mreža se zadaje iz skripte u kojoj su definirani vrhovi (*vertices*) i trokuti (*triangles*).



Slika 6. Pojednostavljeni primjer mreže gornje usne – vidljivi su vrhovi i trokuti. Ukupan broj točaka je veći, zbog jednostavnosti na slici nisu prikazane točke dobivene Catmull-Rom-ovim algoritmom.

Broj vrhova je zbroj svih točaka u sve 4 linije: gornja i donja usna – gornji i donji rub. Za svaki vrh postoje 3D koordinate koje se predaju mreži kao niz *Vector3*⁶ koordinata. Algoritam koji određuje trokute prolazi kroz sve vrhove uzimajući u obzir broj vrhova u pojedinom setu te vraća niz integera koji označavaju broj vrha. Trokuti moraju biti u smjeru obrnutom od smjera kazaljke na satu, u suprotnom se ne iscrtavaju jer su normale trokuta okrenute u suprotnom smjeru. Za primjer sa slike 6 algoritam bi vratio indekse vrhova trokuta (1,0,12), (12,11,1), itd. te bi se gornja usna popunila kao na slici 7.

⁶ Vector3 je struktura unutar Unity Engine, služi za reprezentaciju 3D vektora i točaka. [5]

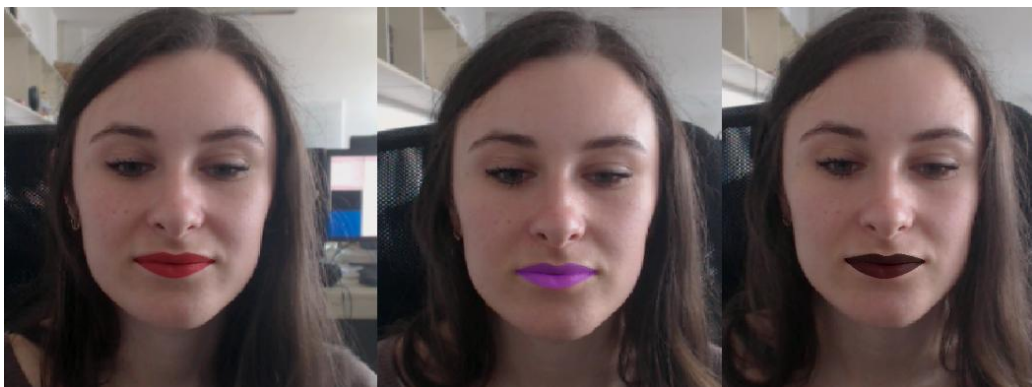


Slika 7. Usne sa slike 6 s popunjenom gornjom usnom

Na isti način se popunjava i donja usna.

Korak 4 – Popunjavanje mreže

Unutar Unitya, svaki objekt koji sadrži mrežu treba imati *mesh renderer* i *mesh filter*. Preko *mesh filter*a se pristupa mreži, a *mesh renderer* služi za njezino iscrtavanje. Unutar *mesh renderer*-a nalazi se polje *Materials*, koje sadrži materijale koji se koriste prilikom popunjavanja mreže. Manipulacijom parametara materijala mogu se dobiti različite teksture i boje. Stvaranjem Unity materijala specificiramo njegov *shader*. Shaderi su male skripte koje koriste matematičke izračune i algoritme za računanje boje svakog piksela uzimajući u obzir osvjetljenje i konfiguraciju materijala. Različiti rezultati stavljanja ruža koji se mogu vidjeti na slici 8 dobiveni su koristeći tri različita materijala.



Slika 8. Rezultat opisanog postupka: crvena, ljubičasta i crna nijansa ruža.

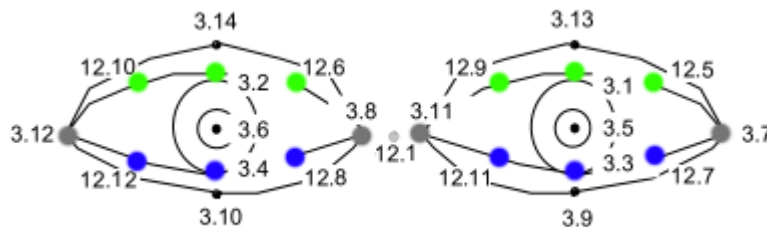
3.2. Olovka ili tuš za oči (*eyeliner*)

Obrubljivanje okvira očiju crnom bojom prvi put se javlja u drevnom Egiptu te u Mezopotamiji. 10 000 godina pr.Kr. Egipćani su nosili crne linije oko očiju ne samo zbog estetike, već su vjerovali da takva praksa pomaže zaštititi od uroka. U prošlosti su korišteni razni materijali poput bakra i antimona. Danas se u svrhu naglašavanja okvira očiju koriste tuševi za oči s raznim aplikatorima te olovke za oči.

Korak 1 – Dohvat točaka iz Trackera i priprema varijabli

Kao i kod aplikacije virtualnog ruža, kod aplikacije *eyeliner* dohvaćaju se karakteristične točke očiju iz *VisageTrackerUnityPlugin*a – slika 1. Traže se 4 seta točaka (slika 9 – lijevo i desno oko):

1. gornja linija lijevog oka: 5 točaka – 3.11, 12.9, 3.1, 12.5, 3.7,
2. donja linija lijevog oka: 5 točaka – 3.11, 12.11, 3.3, 12.7, 3.7,
3. gornja linija desnog oka: 5 točaka – 3.12, 12.10, 3.2, 12.6, 3.8,
4. donja linija desnog oka: 5 točaka – 3.12, 12.12, 3.4, 12.8, 3.8.



Slika 9. Karakteristične točke očiju – gornje linije očiju: 3 zelene točke + 2 sive točke, donje linije očiju: 3 plave točke + 2 sive točke.

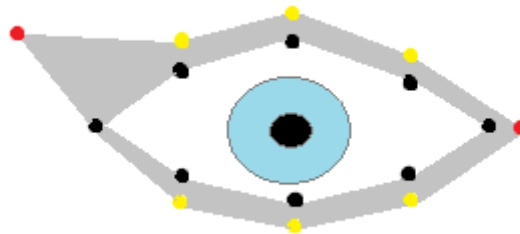
Varijable koje su potrebne pri izračunu su *eyelinerThickness* koji specificira pomak točaka kao debljina linije tipa *float*, *boolean* varijabla *lowereye* koja određuje iscrta li se donja linija oka, *ratio* tipa *integer* koji predstavlja broj točaka koje se dodaju prema Catmull – Rom krivulji između svake dvije karakteristične točke seta. Dvije float varijable *wingUpDown* i *wingLength* određuju duljinu i visinu *cateye looka*. Također postoji *sorting* varijabla tipa *float* koja služi za manipulaciju z-osi te na taj način određuje položaj komponenti maskare, sjenila i *eyeliner*a u prostoru.

Varijabla *ratio* u finalnoj verziji aplikacije postavljena je na 2, a sve ostale varijable mijenjaju se ovisno o željenom obliku *eyelinera*.



Slika 10. Primjer *cateye eyeliner*

Korak 2 – Računanje okvira *eyelinera*

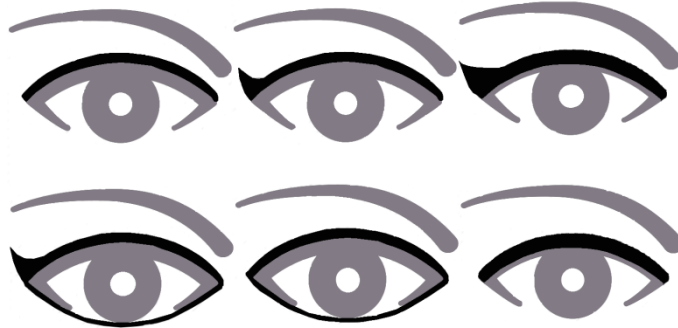


Slika 11. Pojednostavljeni primjer iscrtavanja okvira *eyelinera*. Crnom bojom su označene osnovne karakteristične točke koje su detektirane od strane *Trackera* (također prikazane na slici 1). Žutom i crvenom bojom su označene točke koje su dodane. Područje označeno sivom bojom je ispunjena *eyeliner*. Ukupan broj točaka je veći, zbog jednostavnosti na slici nisu prikazane točke dobivene Catmull-Rom-ovim algoritmom.

Kad su poznate 3D koordinate detektiranih točaka po setovima, potrebno je stvoriti gornju liniju *eyelinera*. To se provodi tako da se kopiraju te točke, zatim se sve kopirane točke osim prve i zadnje točke seta podignu za pomak debljine *eyelinera* (na slici 11 žute točke). Zatim se dodaju dvije dodatne točke (na slici 11 crvene točke). Crvene točke služe za smanjenje oštih rubova te dodavanje *cateye looka*. Ako se želi dobiti donja linija *eyelinera*, prati se ponovo postupak kopiranja točaka i pomicanja, samo se ovaj puta pomicanje vrši u suprotnom smjeru.

Korak 3. Popunjavanje mreže

Okvir *eyelinera* (primjer okvira na slici 11) popunjava se na jednak način kao i okvir ruža.



Slika 12. Neki od oblika *eyelinera*.

Oblik *eyelinera* se može mijenjati pomoću više parametara navedenih u prošlom koraku. Mijenjanjem tih parametara, mogu se dobiti različiti oblici od kojih su neki prikazani na slici 12.

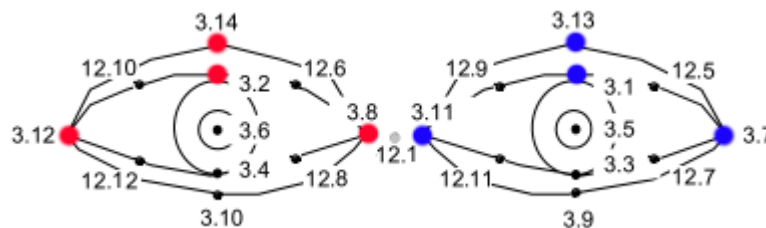
U skripti *eyeliner*, mijenjanje ovih parametara vrši se u ovisnosti o odabiru opcije padajućeg izbornika. Funkcija koja ostvaruje promjenu parametara automatski vrši postavljanje parametara na zadan broj, ovisno o željenoj opciji. Ovo je jedina komponenta u kojoj ova funkcija ne radi promjenu materijala. Trenutno nije moguće mijenjati boju *eyelinera*. Kad bi se ta funkcionalnost implementirala u sklopu poboljšanja implementacije, bilo bi potrebno mijenjati materijale, kao i kod ostalih komponenti.

3.3. Sjenilo

Korak 1 – Dohvat točaka iz Trackera i priprema varijabli

Aplikacija sjenila započinje dohvaćanjem točaka očiju iz *VisageTrackerUnityPlugina* – slika 1. Traže se 2 seta točaka (slika 13. - lijevo i desno oko):

1. lijevi kapak: 4 točke – 3.12, 3.14, 3.8, 3.2,
2. desni kapak: 4 točke – 3.11, 3.13, 3.7, 3.1.

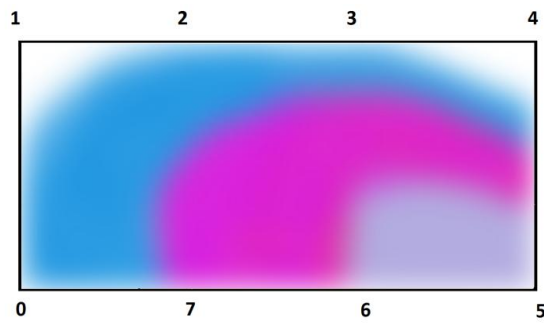


Slika 13. 2 seta točaka: lijevi kapak crvene točke, desni kapak plave točke

Od varijabli, sjenilo koristi *sorting* varijablu tipa *float* koja određuje položaj maskare, sjenila i *eyeliner* u prostoru te *thickness* varijablu također tipa *float* koja određuje širinu područja na koje se aplicira sjenilo.

Korak 2 – Konstrukcija mreže iz dobivenih točaka

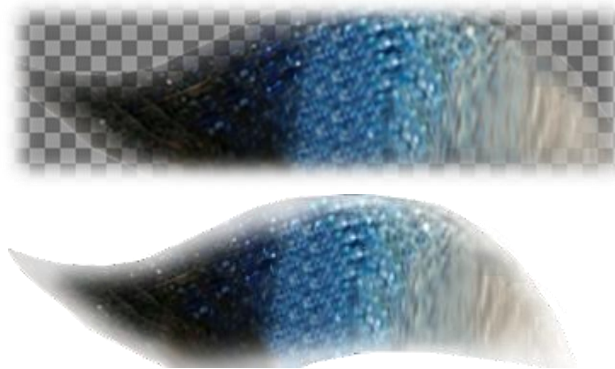
Mreža jednog oka sastoji se od 8 točaka koje čine pravokutnik. Točki 7 sa slike 14. postavljaju se koordinate točke 3.12, točki 6 koordinate točke 3.2, točki 5 koordinate točke 3.8, a točki 3 koordinate točke 3.14. Koordinate y točke 0 postavlja se na vrijednost y točke 3.12, a koordinatu x dobiva se tako da se od x koordinate točke 3.12 oduzme $\frac{1}{2}$ x vrijednosti između točke 3.12 i 3.8. Točkama 1,2 i 4 y koordinata se postavlja na y vrijednost točke 3.14, a x koordinate su iste kao točkama 0,7,5 respektivno. Isti postupak, ali s drugim točkama, se koristi i za drugo oko.



Slika 14. Primjer mreže jednog oka s nalijepljenom teksturom

Korak 3 – Izrada teksture

Za izradu teksture potreban je uređivač slike, primjerice Photoshop ili GIMP. Za realističan izgled sjenila, potrebno je raditi s alpha kanalom teksture kako bi se dobila prozirnost.



Slika 15. Gore – sivi kvadratići predstavljaju prozirnost. Dolje – tekstura se prilagođava boji pozadine.

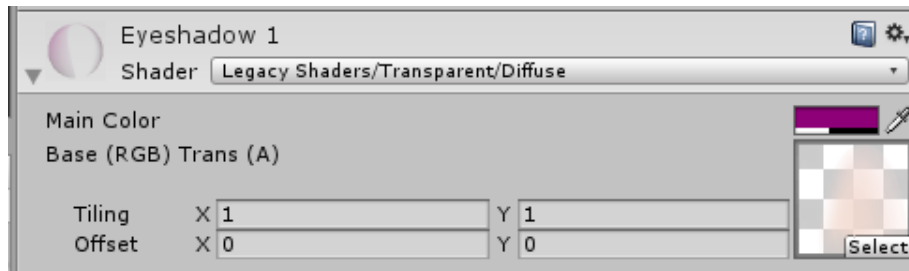
Uz prozirnost, vidljiv je prijelaz sjenila koji se stapa s pozadinom. Na boji lica, ovaj efekt daje dojam realističnosti.

Aplikacija ovakve teksture u Unityu se izvodi pomoću *UV* koordinata. *UV* koordinate su normaliziran⁷ dvodimenzionalni koordinatni sustav gdje se osnovna točka (0,0) nalazi u lijevom donjem kutu prostora. *UV* koordinate su u ovom slučaju sinonim *XY* koordinatama, ali s obzirom da su *XY* koordinate korištene u *XYZ* 3D prostoru, koriste se *UVW* koordinate za teksture kako ne bi došlo do zabune. Prema tome, prostor 0-1 se može gledati kao 0-100% slike. Primjerice, vrh s *UV* koordinatama (0.5, 0.5) će odgovarati sredini teksture.

⁷ [0, 1]

Korak 4 – Popunjavanje mreže

Stvara se novi Unity materijal te se koristi jedan od standardnih shadera Transparent/Diffuse. Promjenom teksture mogu se dobiti različite vrste sjenila.



Slika 16. Materijal „Eyeshadow 1“ koji koristi osnovnu teksturu sjenila.



Slika 17. Rezultat aplikacije teksture sa slike 15.

3.4. Maskara

Maskara je kozmetički preparat kojim se nastoji dobiti privid gušćih i dužih trepavica.

Korak 1 – Dohvat točaka iz Trackera i priprema varijabli

Jednako kao i kod sjenila. Traže se 2 seta točaka (slika 13. - lijevo i desno oko):

1. gornja linija lijevog oka: 5 točaka – 3.11, 12.9, 3.1, 12.5, 3.7,
2. gornja linija desnog oka: 5 točaka – 3.12, 12.10, 3.2, 12.6, 3.8,

Od varijabli, maskara koristi *sorting* i *ratio* varijable koje su već objašnjene u prethodnom tekstu. Također koristi *mascaraLength* tipa *float* koja određuje veličinu maskare (širenje mreže po y osi).

Korak 2 – Konstrukcija mreže iz dobivenih točaka

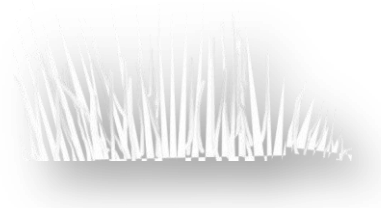
Linija jednog seta čini gornji rub (linija kapka) oka. Aplikira se Catmull-Rom algoritam kao i kod ruža i olovke/tuša za oči što rezultira većim brojem točaka na liniji uz rub oka što daje puno prirodnije prijanjanje liniji oka. Ako se taj rub klonira te se pomakne za određeni iznos prema gore po y koordinatama točaka, dobiva se mreža koja ima oblik koji je prikazan na slici 18.



Slika 18. Izgled mreže na koju se aplicira maskara.

Korak 3 – Izrada teksture

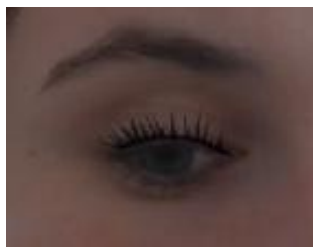
Izrada teksture za mrežu je jednaka kao i kod sjenila. Potreban je uređivač slike te korištenje alpha kanala. Korištenjem UV koordinata definira se željena pozicija teksture na mreži.



Slika 19. Gore - primjer teksture korištene kod aplikacije maskare.

Korak 4 – Popunjavanje mreže

Postupak popunjavanja mreže je jednak kao i kod aplikacije sjenila, ali boja koja se koristi kod primjene teksture je crna.

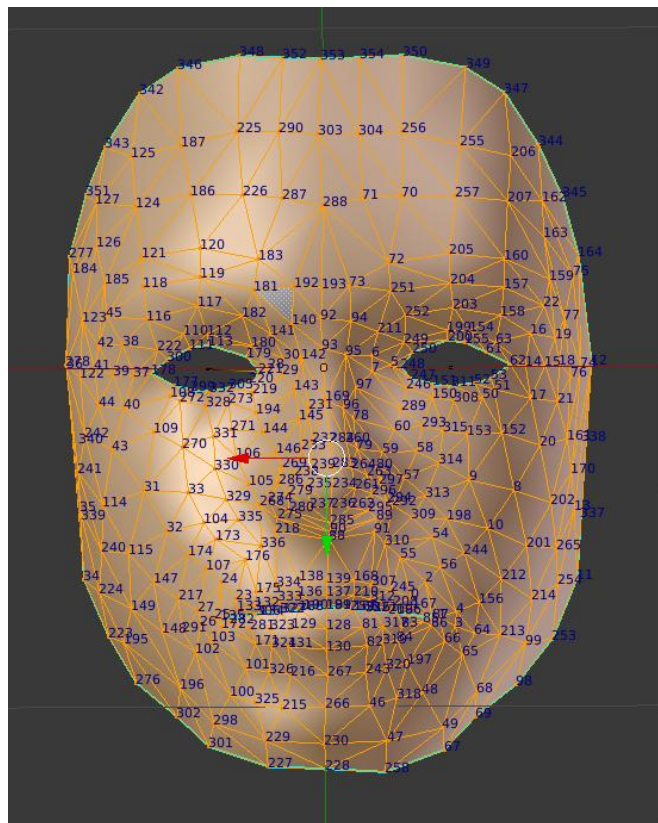


Slika 20. Rezultat primjene teksture maskare.

3.5. Rumenilo

Rumenilo je kozmetički preparat koji se stavlja na obraze kako bi se dodala svježina i boja na lice te kako bi se imitiralo prirodno rumenilo nakon stavljanja pudera i ujednačavanja tena koji oduzima prirodnu boju s lica.

Za razliku od prethodno opisanih elemenata šminke, rumenilo ne koristi karakteristične točke lica prikazane na slici 1. Umjesto toga, koristi se 3D model lica prikazan na slici 21. Zainteresirani čitatelj bi se mogao pitati zašto 3D model nije korišten kod ostalih dijelova šminke. Razlog je taj što je 3D model neprecizniji kod određivanja tipičnih karakterističnih točaka koje se koriste kod aplikacije šminke na oči ili usta.

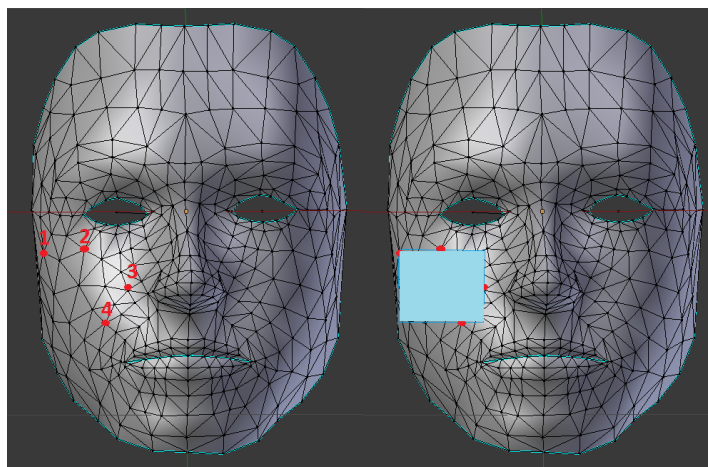


Slika 21. 3D model lica

Korak 1 – Dohvat vrhova i trokuta iz Trackera i priprema varijabli

Pomoću funkcije iz *VisageTrackerUnityPlugin*a dohvaćaju se sve pozicije vrhova i trokuta 3D modela, te njihov ukupan broj. Zatim se odabiru 4 vrha za lijevi i 4 vrha za desni obraz. Vrhovi se odabiru tako da tvore četverokut oko obraza.

Korak 2 – Konstrukcija mreže iz dobivenih vrhova

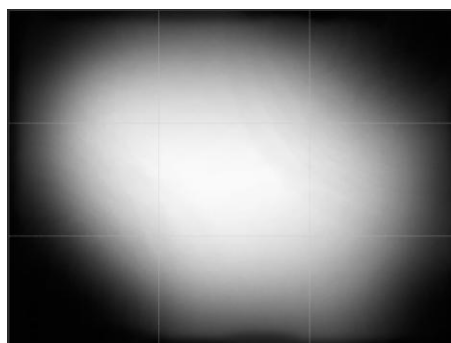


Slika 22. Primjer izrade mreže. Lijevo: 4 odabrana vrha za desni obraz. Desno: Primjer mreže napravljen pomoću ta 4 vrha.

Koristeći 4 odabrane točke stvora se mreža. Slika 22 lijevo prikazuje primjer 4 odabrane točke. Točka 1 određuje lijevu x koordinatu, točka 2 određuje gornju y koordinatu, točka 3 određuje desnu x koordinatu, a točka 4 određuje donju y koordinatu pravokutne mreže koja je prikazana na slici 22 desno obojana plavom bojom.

Korak 3 – Izrada teksture

Crtanje teksture se izvodi pomoću uređivača slika, jednako kao i tekstura za maskaru i sjenilo.

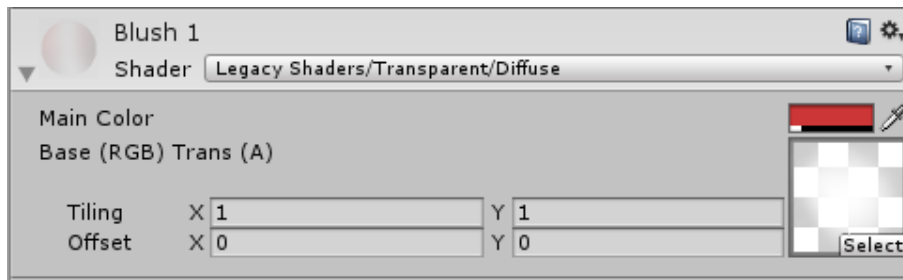


Slika 23. Alpha kanal teksture sjenila koji definira prozirnost slike prema pozadini. Tamo gdje je alpha kanal crn slika je potpuno prozirna, a tamo gdje je bijele boje slika je neprozirna. Između crne i bijele se nalaze razne nijanse sive koje uvjetuju poluprozirnost dijelova slike.

Tekstura se sastoji od bijele boje te alpha kanala prikazanog na slici 23. Kako bi tekstura bila raspoređena po mreži na željeni način, definiraju se UV koordinate koje su objašnjene u trećem koraku poglavlja „Sjenilo“.

Korak 4. Popunjavanje mreže

Kako bi se tekstura mogla aplicirati preko mreže, u Unityu se stvara novi materijal.



Slika 24. Materijal „Blush 1“ koji koristi shader Transparent/Diffuse

Za rumenilo najbolje odgovara shader Transparent/Diffuse zato što omogućuje manipulaciju boje i prozirnosti teksture te odabir same teksture. Za primjenu različitih vrsta sjenila može se mijenjati boja i prozirnost teksture što korisniku daje privid odabira različite nijanse sjenila (odabirom boje), intenzivnosti (odabirom prozirnosti) te stila nanošenja (odabir teksture).

S obzirom na različite rase i boje kože, pojavljuje se potreba za različitim intenzitetom simulacije rumenila. Osim različitih nijansi rumenila, moguće je birati intenzitet sjenila. Intenzitet je implementiran kao klizač na grafičkom sučelju. Mijenjanje vrijednosti klizača rezultira promjenom alpha vrijednosti na materijalu. Promjenom alpha vrijednosti manipulira se prozirnošću teksture.

3.6. Puder

Prikrivanje nedostataka na koži pomoću pudera je vrlo široko korištena metoda uljepšavanja. Puder se može pronaći u mnogim oblicima poput pudera u prahu, tekućeg pudera, raznih BB, CC i toniranih krema koje služe istoj svrsi.

Korak 1 – Dohvat vrhova i trokuta iz Trackera i priprema varijabli

Aplikacija pudera koristi isti 3D model kao i rumenilo. U ovom slučaju koriste se svi vrhove i trokuti 3D modela. Unutar Blendera napravljen je 3D model prikazan na slici 22. Mapiranjem 3D modela, napravljena je datoteka koja mapira vrhove 3D modela što omogućuje aplikaciju pudera kao teksture.

Korak 2 – Konstrukcija mreže iz dobivenih vrhova

Konstrukcija 3D mreže radi se dohvaćanjem podataka iz 3D modela. Podaci koji se dohvaćaju su broj vrhova, broj trokuta, pozicija vrhova, trokuti te vrijednosti mapiranja teksture. Mreža se radi prema dohvaćenom 3D modelu.

Korak 3 – Izrada teksture

Prema načinu mapiranja 3D modela može se napraviti tekstura koja će odgovarati 3D modelu. Kod izrade teksture vrlo je bitno da rubovi imaju zamućeni prijelaz kako simulacija pudera ne bi djelovala neprirodno – s vidljivim rubovima. Osim prijelaza na rubovima 3D modela, bitno je da aplikacija zaobiđe oči i usta s obzirom na to da se puder ne stavlja na ta mjesta.

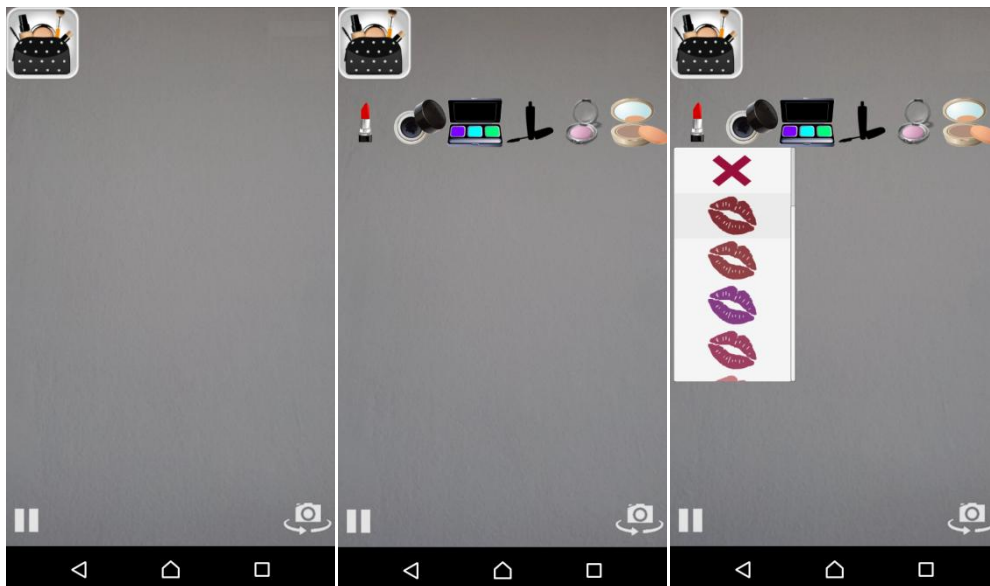
Korak 4. Popunjavanje mreže

Izrada različitih materijala kao i kod prethodno opisanih komponenti daje mogućnost aplikacije raznih boja i tekstura.

3.7. Grafičko sučelje

Grafičko sučelje aplikacije omogućuje korisniku manipulaciju parametrima te isprobavanje različitih kombinacija virtualne šminke koje aplikacija nudi. Sučelje je razvijeno u Unity sceni te se sastoji od više komponenata.

Gledajući sliku 25, tipka u donjem lijevom kutu označava pauzu snimanja kamere čime se cijela simulacija zaustavlja. Tipka u donjem desnom kutu služi za mijenjanje kamere, primjerice promjena između prednje i stražnje kamere mobilnog uređaja. Ove dvije komponente su izvorno komponente iz Visage|SDK paketa za Unity projekt.



Slika 25. Izgled grafičkog sučelja aplikacije na Android uređaju.

Tipka u gornjem lijevom kutu ekrana otvara izbornik. Izbornik se sastoji od komponenata: ruž, tuš, sjenilo, maskara, rumenilo i puder. Ovaj izbornik se vidi na slici 25 u sredini. Klikom na svaku od komponenata otvara se padajući izbornik. Svaki padajući izbornik sadrži različite opcije za tu komponentu. Primjerice na slici 25 desno vidimo mogućnost isključenja virtualne simulacije ruža te različite boje ruževa.

4. Rezultati

4.1. Aplikacija

Aplikacija sadrži 6 elemenata, ruž, tuš/olovka za oči, sjenilo za oči, rumenilo, maskaru i puder koji su opisani u prethodnom poglavlju. Rezultat aplikacije ovih elemenata na različitim osobama se može vidjeti na slici 26.



Slika 26. Lijevo – slika sa simulacijom šminke, desno – originalna slika

Jedan od nedostataka aplikacije uključuje nepreciznost ili pogreške softvera za praćenje lica. Glavni nedostatak softvera za praćenje lica je određena nestabilnost u

praćenju što rezultira drhtavim efektom simulacije šminke. Ovaj efekt se može smanjiti podešavanjem parametara zaglađivanja. Ti parametri stabiliziraju titranje točaka, ali rezultiraju kašnjenjem u praćenju lica. Prema tome, potrebno je pronaći odgovarajući omjer između titranja i kašnjenja tako da ovi efekti budu što manje vidljivi.

Osim navedenog problema, postoji pitanje rezolucije kamere. Što je veća rezolucija, sporiji je rad aplikacije što nas također navodi na traženje odgovarajućeg omjera brzine i kvalitete slike.

Softver za praćenje lica gubi izraze lica kod rotacije glave lijevo-desno. Ovaj problem je riješen tako da je aplikacija šminke ograničena na kuteve rotacije glave. Dok je glava unutar dozvoljenog intervala, renderiranje šminke je aktivno, čim rotacija glave prijeđe dozvoljeni interval, renderiranje se isključuje.

Softver za praćenje lica postaje neprecizan kod velikih odstupanja od opuštene poze lica. Ovaj nedostatak nije primaran problem zato što korisnici prilikom korištenja aplikacije ne trebaju raditi ekstremne izraze lica kako bi dobili dojam o pristajanju određene virtualne kozmetičke komponente.

Još jedan od nedostataka je nepreciznost lociranja određenih karakterističnih točaka, primjerice rubova usana što dalje rezultira lošom simulacijom ruža za usne koji ne prekriva u potpunosti ili prekriva preveliki dio usana. Eksperimentalnim isprobavanjem aplikacije na većem broju primjeraka različitih lica donesen je sljedeći zaključak: ako su usta zatvorena ili razvučena u lagani smiješak bez pokazivanja zubi, softver za praćenje lica uspijeva točno izolirati usne u mnogo većem broju nego što to uspijeva kod otvorenih usana tako da se vide cijeli prednji gornji zubi (slika 28). Ako gledamo podskup zatvorenih usana, softver za praćenje rijetko locira apsolutno točne koordinate točaka na rubovima usana. Stoga koristeći karakteristične točke možemo dobiti što prirodniju liniju rubova ovisno o vrsti krivulje koju koristimo kako bismo dobili veći broj točaka (postupak opisan u poglavlju „Ruž“). Gledajući rezultate koji su vidljivi na slici 27, možemo zaključiti da krivulja Catmull - Rom ne rezultira jednakim pogreškama kod rubova slika – na nekim usnama krivulja prelazi preko linije usana, na nekim slikama se nalazi unutar linije usana, na nekim slikama linije gornje usne odgovara, a linija donje usne ne odgovara i obrnuto. Stoga se može zaključiti da nepravilnosti u poklapanju s točnim koordinatama rubova usana uvelike ovise o samom tipu usana i njihovoj zakrivljenosti koja se razlikuje od osobe do osobe. Na slici 27 prva slika lijevo ima preveliko prekrivanje donje usne, dok prva slika desno ima premalo prekrivanje donje usne.

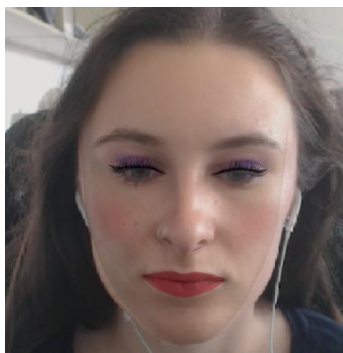


Slika 27. Skup zatvorenih i poluzatvorenih usana.



Slika 28. Skup otvorenih usana

Na slici 28 možemo vidjeti skup otvorenih usana uz problem ranije opisan – softver za praćenje karakterističnih točaka lica ne locira točan položaj usana.



Slika 29. 3D model nije obuhvatio cijelo lice čime postaju vidljive linije simulacije pudera na rubovima 3D modela

Simulacija pudera otkriva novi nedostatak. 3D model prilikom apliciranja na lice ponekad ne obuhvaća čitavo područje lica ili obuhvaća preveliko područje te izlazi iz okvira lica, što dovodi do vidljivih rubova simulacije šminke što se može vidjeti na slici 29. Također, osnovna funkcija pudera je prekrivanje nedostataka na licu koji uključuju crvenilo, akne i ostale diskoloracije. Ovdje razvijen puder jedino može prekriti crvenilo lica i dati dojam izgleda takve nijanse pudera na koži, međutim ne može prekriti nedostatke kao što su mrlje i akne. Manipulacijom slike kamere za zamućivanje određene regije lica taj element bi se mogao riješiti.

Osim nepotpunog prikrivanja usana, postoji jedan nedostatak koji je uočljiv u usporedbi s ostalim rješenjima (slika 30). Zbog načina aplikacije olovke/tuša za oči i ruža za usne rubovi nisu zamućeni i glatki, već oštri. Kad bi se ove dvije komponente aplicirale pomoću konstrukcije mreže uz primjenu koordinata teksture kao ostale komponente, ovaj problem ne bi bio toliko uočljiv.

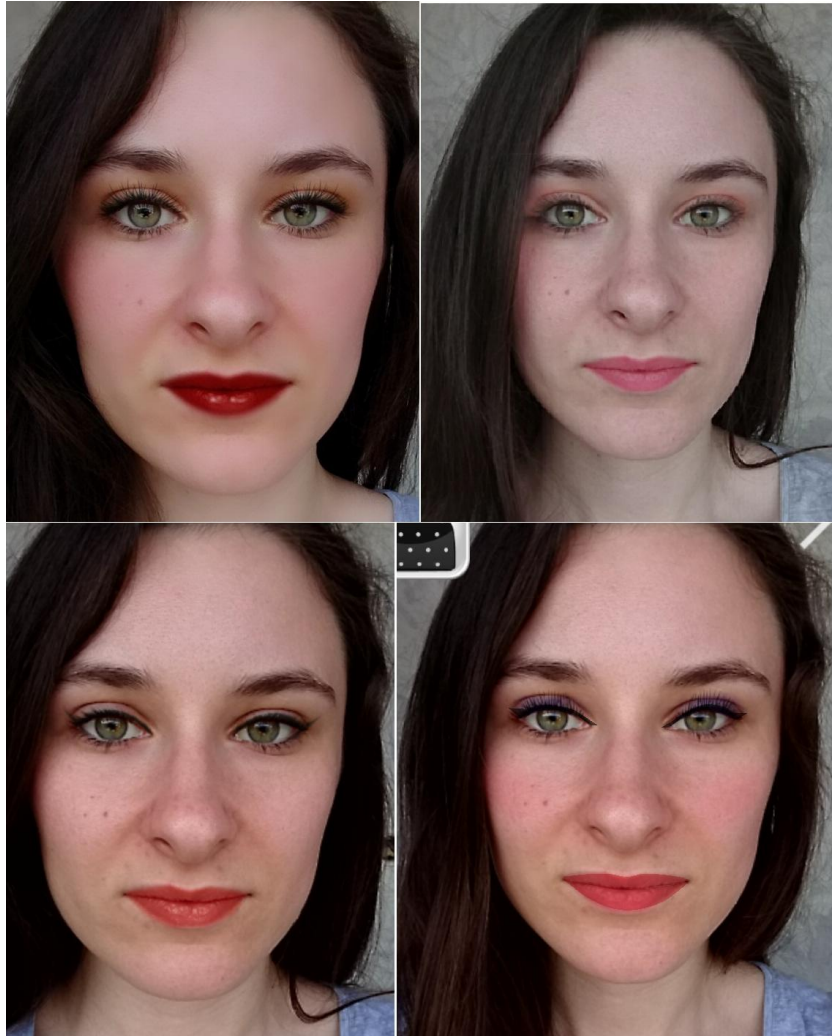
4.2. Usporedba aplikacije s prethodno opisanim dostupnim rješenjima na tržištu

Oriflame Makeup Wizard aplikacija koristi isti softver za praćenje lica razvijen od strane Visage Technologies stoga neke osnovne probleme koji su svojstveni softveru za praćenje lica imaju obje aplikacije. Oriflame nudi iste komponente koje su prisutne u aplikaciji koja je tema ovog rada, te ne nudi opciju pudera. Posljednje ažuriranje u Google trgovini popravilo je mnoge nedostatke aplikacije komponenti koje su izgledale nerealno u usporedbi s ovdje razvijenom aplikacijom. Osim što nudi opciju kupnje proizvoda što nije tema ovog rada, ono što Oriflame nudi dodatno je opcija uzimanja slike ekrana i dijeljenja slike preko različitih aplikacija, ostale razlike su generalno razlike između tekstura i boja te kvalitetnije grafičko sučelje.

L'Orealov MakeUp Genius koristi rumenilo, sjajilo za usne, ruž, puder, bronzer, te tuš i olovke za oči. Sjajilo za usne ili ruž se mogu gledati kao jedna komponenta s obzirom na to da se radi samo o aplikaciji različite teksture. U verziji koja se nalazi u Google trgovini u trenutku pisanja ovog rada postoje mnogi nedostaci. Primjerice uređaj se previše zagrijava tijekom uporabe, određene komponente djeluju nerealno, a puder se aplicira samo na obraze. Prije početka korištenja potrebno je 'skenirati' lice. Kao i Makeup Wizard, MakeUp Genius nudi opciju kupnje proizvoda te sadrži kvalitetnije grafičko sučelje.

YouCam Makeup aplikacija ne nudi prodaju kozmetičkih proizvoda određenog brenda, već stoji kao samostalna aplikacija za zabavu korisnika. Osim što nudi dodatne opcije poput boje kose, zjenica i modnih dodataka, ono što je relevantno za usporedbu s aplikacijom koja je tema ovog rada je rad u stvarnom vremenu. Aplikacija koristi određene filtre manipulacije slike poput zaglađivanja (*blur*) koji se automatski apliciraju na kameru. Zatim postoji opcija aplikacije složenog gotovog izgleda, pa čak i gotovih izgleda slavni osoba. Osim izgleda, mogu se birati zasebne komponente: sjenilo, maskara (ili umjetne trepavice - nalijepljena tekstura), olovka/tuš za oči, ruž (tekstura na usnama), rumenilo i boja zjenice. Dakle, dodatne komponente koje ova aplikacija koristi za praćenje u stvarnom vremenu su promjena boje zjenica i filter zaglađivanja slike.

Modiface Live aplikacija kao i YouCam Makeup ne nudi prodaju kozmetičkih proizvoda. Aplikacija nudi veliku stabilnost praćenja. Ne nudi aplikaciju različitih komponenti odvojeno već nudi odabir boja za različite kombinacije simulacije šminke. Također nudi opciju manipulacije licem (povećanje očiju, stanjivanje nosa itd.) te izmjenu boje zjenice oka. Jedna zanimljiva ideja koju su razvojni programeri aplikacije implementirali je mijenjanje kombinacija šminke pomoću podizanja obrva za šminku očiju te napučivanja usnica za šminku usana. Aplikacija ne nudi kupnju proizvoda.



Slika 30. Usporedba aplikacija: gore lijevo – YouCam MakeUp, gore desno- Oriflame Makeup Wizard, dolje lijevo – L'Oreal Makeup Genius, dolje desno – aplikacija koja je tema ovog rada

Na slici 30 može se vidjeti usporedbu aplikacija. Sve aplikacije su slikane istom kamerom u razmaku par sekundi, te na licu nije bilo šminke prije isprobavanja aplikacije. Kod svake od aplikacija odabran je *look* – gotova kombinacija komponenti dostupna unutar aplikacije, te nije bilo dodatnog mijenjanja komponenti ili editiranja slike osim izrezivanja. Slikano je uz odlično osvjetljenje, što je vrlo važna komponenta za svaku od aplikacija.

Najvažnija razlika između ovdje razvijane aplikacije i komercijalnih rješenja je oštrina rubova koja je vrlo vidljiva kod olovke/tuša za oči i ruža za usne u aplikaciji koja je tema ovog rada. Kod Oriflame Makeup Wizard aplikacije vidljiv je nedostatak prilikom simulacije sjenila – na desnom oku prekriva dio kose, trepavice te staje prenisko kod oka. YouCam aplikacija radi filter zaglađivanja koji se ne može maknuti ili modificirati prilikom simulacije šminke. Zbog toga slika djeluje pomalo nerealno, iako je to dosta subjektivna pretpostavka.

Kada bi se gledale sve performanse aplikacija, bilo bi potrebno uspoređivati softvere za praćenje lica. S obzirom na to da se ovaj nastavlja na već razvijen softver koji također koristi Oriflame Makeup Wizard aplikacija, možda je najbolje uspoređivati ovdje razvijenu aplikaciju s Oriflame aplikacijom. Makeup Wizard aplikacija ima bolje riješen problem titranja slike te se simulacija šminke čini vrlo stabilna. Teksture usana su kod Makeup Wizard aplikacije zamučene na rubovima, što daje bolji dojam. Međutim, kod Oriflame aplikacije vidljivi su neki nedostaci poput prelaska sjenila preko bjeloočnice oka te tekstura prekriva trepavice. Kod ovdje razvijene aplikacije to je riješeno tako da se pomoću varijable *sorting* komponentama na očima daje prioritet po z-osi, s obzirom na to da se scena aplikacije radi u 3D sceni. Stoga se sjenilo stavlja u pozadinu, iznad njega dolazi *eyeliner*, a iznad *eyelinera* maskara (trepavice).

4.3. Poboljšanja

Razvijena funkcionalnost i aplikacija imaju mjesta za poboljšanje. Povećanje opcija i mogućnosti koje aplikacija nudi dijelom ovisi o grafičkom sučelju i dizajnu te izradi tekstura i boja. Teksture se dizajniraju u uređivačima slika, a pri tome postoji beskonačno mnogo različitih boja i oblika. Pomoću tekstura možemo kompletno promijeniti izgled komponenti. Mogućnost i dostupnost različitih tekstura i boja je jedan od prioriteta korisnika prilikom korištenja aplikacije za simulaciju šminke u svrhu zabave, stoga je potrebno pružiti što veći broj kombinacija uz primjereno grafičko sučelje kako bi se time zadržao i/ili proširio broj korisnika.

Problem lociranja točnih koordinata točaka rubova usana, opisan ranije u tekstu, kao i problem različitih vrsti i oblika ljudskih usana te oština rubova bi se mogli pokušati riješiti zamučivanjem rubova usana. S obzirom na to da se ruž ne lijepi teksturom na usne kao sjenilo, maskara ili rumenilo, nije moguće manipulirati teksturom. Jedno od alternativnih načina zamučivanja rubova moglo bi se dobiti pisanjem vlastitog *shadera*, a drugi bi uključivao stvaranje vlastite mreže te njezino mapiranje kao i kod primjene sjenila, maskare i rumenila. Razlog zašto to nije tako napravljeno je taj da sjenilo, maskara i rumenilo nemaju oštre rubove unutar kojih se moraju kretati, dok usne imaju točne koordinate prijelaza.

Jedna od ideja koja se nameće gledanjem ostalih rješenja je aplikacija filtra zaglađivanja na kameru. Tako se oštri rubovi na licu zamučuju, a koža se doima kao da sadrži manje nepravilnosti. Međutim ponekad taj efekt daje neprirodan rezultat. Osim manipulacije kamerom, ovakav filter se može aplicirati isključivo na područje 3D modela lica čime komponenta pudera poprima svoj puni smisao.

Zaključak

Na pitanje zašto bi se razvijala ovakva aplikacija, može se odgovoriti brojem instalacija opisanih aplikacija dostupnih na tržištu. Kupovina preko interneta danas poprima sve veće razmjere, sve više ljudi smatra da je u današnjem užurbanom svijetu jednostavnije izvršiti kupovinu iz udobnosti vlastitog doma. Međutim, takva kupovina ima i svoje mane. Jedna mana takve kupovine je nemogućnost isprobavanja proizvoda. Trgovine sa šminkom obično nude testere pomoću kojih se unutar trgovine može isprobati većina proizvoda. Upravo ovakve aplikacije pomažu premostiti mane internet kupovine. Osim internet kupovine, ovakve aplikacije imaju zabavni element koji također može rezultirati velikim brojem aktivnih korisnika, što dokazuje YouCam Makeup Makeover Studio.

Gledajući usporedbu navedenih rješenja dostupnih na tržištu, može se zaključiti da razvijena aplikacija može konkurirati navedenim aplikacijama. Dvije ključne razlike između ovdje razvijene aplikacije i ostalih aplikacija dostupnih na tržištu: (1) grafičko sučelje te broj dostupnih tekstura, boja i oblika te (2) softver za praćenje lica. Grafičko sučelje te broj dostupnih proizvoda (boja i tekstura) moguće je odrediti prema zahtjevima kompanije koja želi pustiti ovakvu aplikaciju na tržište. Softver za praćenje lica čini temelj aplikacije. Uz kvalitetan softver za praćenje lica koji daje mnogo informacija o položaju točaka na slici moguće je napraviti vrlo kvalitetne i stabilne aplikacije sa simulacijom šminke u stvarnom vremenu. Iako nije poznato pomoću kojih su softvera rađena ostala rješenja, Unity se ovdje pokazao kao vrlo koristan i moćan alat, a korišteni softver za praćenje lica kao odličan alat za dohvat karakterističnih točaka lica u videu ili stvarnom vremenu.

Literatura

[1] Visage Technologies, www.visagetechnologies.com, 2016

[2] Unity3D Game Engine, <https://unity3d.com/>, 2016

[3] Visage documentation, *visage|SDK™ 8.0*, Visage Technologies, 2016

[4] Wikipedia, Centripetal Catmull-Rom Spline,
https://en.wikipedia.org/wiki/Centripetal_Catmull%E2%80%93Rom_spline

[5] Unity Documentation – Scripting API
<http://docs.unity3d.com/ScriptReference/Vector3.html>

Naslov, sažetak i ključne riječi (HRV i ENG)

NASLOV: Virtualna šminka

SAŽETAK:

Cilj ovog rada je razvoj aplikacije koja nudi mogućnost virtualne simulacije šminke u stvarnom vremenu pomoću tehnologije praćenja i analize lica. U sklopu rada dan je pregled trenutnih rješenja dostupnih na tržištu, pregled tehnologija korištenih prilikom izrade aplikacije, objašnjen je postupak razvoja aplikacije po komponentama, prikazani su rezultati aplikacije uz usporedbu s ostalim rješenjima na tržištu te su predložena moguća poboljšanja.

KLJUČNE RIJEČI: šminka, aplikacija šminke u stvarnom vremenu, virtualna šminka, Unity, Visage|SDK

TITLE: Virtual Makeup

ABSTRACT:

Goal of this paper was to develop a virtual makeup app that uses face tracking and analysis technologies to allow users a real time makeup simulation. The paper contains a review of related work available on the mobile app markets and overview of technologies used in development of the app. Also in the paper the process of app development is explained, results are shown with comparison to related work solutions and possible improvements are given.

KEYWORDS: makeup, real-time makeup application, virtual makeup, Unity, Visage|SDK