

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Maj Smerkol

Digitalni video efekti za After Effects

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR:

Ljubljana, 2017

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Besedilo teme diplomskega dela študent prepíše iz študijskega informacijskega sistema, kamor ga je vnesel mentor. V nekaj stavkih bo opisal, kaj pričakuje od kandidatovega diplomskega dela. Kaj so cilji, kakšne metode uporabiti, morda bo zapisal tudi ključno literaturo.

Na tem mestu zapišite, komu se zahvaljujete za izdelavo diplomske naloge. Pazite, da ne boste koga pozabili. Utegnil vam bo zameriti. Temu se da izogniti tako, da celotno zahvalo izpustite.

Svoji dragi Alenčici.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Vizualni učinki v filmu in videu	3
2.1	Kaj so vizualni učinki in zakaj jih uporabljamo	3
2.2	Primerjava uporabe vizualnih učinkov v filmu in videu	4
2.3	Vpliv videa na alternativno kulturo	5
2.4	Pregled zgodovine vizualnih učinkov	6
3	Uporaba After Effects za vizualne učinke	9
3.1	Hiter pregled After Effects	9
3.2	Kompozicija digitalnega videa	9
3.3	Kompozicija videa v After Effects	10
3.4	Uporaba digitalnih vizualnih učinkov v AE	12
4	Razvoj vtičnikov za After Effects	15
4.1	Delovanje vtičnikov	16
4.2	Metoda razvoja	18
4.3	Vtičnik PixelRain	19
	Literatura	22

Seznam uporabljenih kratic

kratica	angleško	slovensko
VFX	visual effects	vizualni učinki
SFX	special effects	posebni učinki
AEGP	After Effects general purpose plug-in	splošno namenski vtičnik za After Effects
AE	Adobe After Effects	Adobe After Effects
SDK	software development kit	paket za razvoj programske opreme
CGI	computer generated imagery	računalniško generirana gra- fika

Povzetek

Naslov: Digitalni video efekti za After Effects

Avtor: Maj Smerkol

V vzorcu je predstavljen postopek priprave diplomskega dela z uporabo okolja L^AT_EX. Vaš povzetek mora sicer vsebovati približno 100 besed, ta tukaj je odločno prekratek. Dober povzetek vključuje: (1) kratek opis obravnavanega problema, (2) kratek opis vašega pristopa za reševanje tega problema in (3) (najbolj uspešen) rezultat ali prispevek magistrske naloge.

Ključne besede: video, After Effects, vizualni učinki, posebni učinki.

Abstract

Title: Digital video effects for After Effects

Author: Maj Smerkol

This sample document presents an approach to typesetting your BSc thesis using L^AT_EX. A proper abstract should contain around 100 words which makes this one way too short.

Keywords: video, After Effects, visual effects, special effects.

Poglavje 1

Uvod

Cilj diplomske naloge je raziskati metode razvoja vtičnikov za programsko orodje Adobe After Effects (v nadaljevanju AE). AE je programski paket, namenjen montaži videa in izdelavi vizualnih efektov. Ena izmed njegovih dobrih lastnosti je razširljivost s pomočjo vtičnikov, ki omogočajo dodajanje poljubnih funkcionalnosti programskemu paketu. Razširjanje je omogočeno prek programskega vmesnika in prosto dostopne razvojne dokumentacije za uporabo le tega.

Vtičnike za AE lahko delimo v 4 kategorije: splošno namenske vtičnike (AEGP), vizualne učinke (effect), vhodno izhodne vtičnike (IO) in vtičnike Artisan, ki prevzamejo upodabljanje 3D slojev oziroma nadomestijo privzeti algoritem za upodabljanje v AE. Gostitelj še vedno sam skrbi za upodabljanje 2D slojev. Ker sem se odločil za razvoj vizualnih učinkov, je izbira med tipi vtičnikov jasna. Vtičniki za vizualne učinke implementirajo VFX kot operacijo nad posameznimi sličicami v videu. Programski vmesnik AE ponuja nemalo operacij, ki pospešijo izvajanje operacij nad piksli in skrajšajo čas razvoja, saj ponujajo visokonivojski dostop do vgrajenih funkcij AE.

Splošno namenski vtičniki zahtevajo več resursov od sistema, na katerem tečejo, omogočajo pa programerju več možnosti integracije z AE, spreminjanje uporabniškega vmesnika in odzivanje na interno stanje celotnega programskega paketa. Omogočajo mnogo več, kot potrebujemo za implementacijo

tipičnih vizualnih učinkov. Vhodno izhodni vtičniki so namenjeni dodajanju podpore za branje in pisanje novih datotečnih formatov, implementaciji kodekov in podobno.

Pri razvoju VFX je seveda prvi korak ugotavljanje namena vizualnega učinka oziroma definiranje umetniške vizije. Pred tem sem pregledal obstoječe vtičnike. Po namenu sem jih ločil v tri kategorije. Prva kategorija so vtičniki za popraviljanje izvornih posnetkov, kot so odstranjevanje šuma, popraviljanje raznih vizualnih artefaktov in podobno. Namen teh vtičnikov je reševanje slabih posnetkov, kadar ponovni zajem videa oziroma ponovno snemanje ni mogoče, zaradi cene, pomanjkanja časa ali drugih razlogov. Le redko lahko s pomočjo teh vtičnikov dosežemo rezultate enake kvalitete, kot če bi bili posnetki dobri.

V drugo kategorijo štejem efekte, ki so namenjeni dodajanju vrednosti posnetki v post produkciji. Večino vizualnih učinkov pravzaprav lahko štejem v to kategorijo. Sem spadajo tudi vsi tisti efekti, ki vizualno pomagajo pripovedništvu, pomagajo ustvariti fiktiven svet, like in podobno.

V tretjo kategorijo uvrščam destruktivne efekte - to so tisti, ki ne dodajajo informacij sliki, ampak nasprotno zmanjšajo kvaliteto slike za dosego nekega vizualnega sloga ali pa pomagajo ustvarjati vzdušje in gledalcu, če so uspešno uporabljeni, lahko vzbudijo določena čustva. Primere takšnih efektov lahko najdemo marsikje, v popularni kulturi (Mr Robot, 2015), računalniških igrah (SOMA, 2015) in filmih (The Hunger), na različne načine proizvedene vizualne artefakte pa celo v galerijah in knjižnih zbirkah umetniških del (ang. *ASCII Graphic Glitch Art*, Rozita Fogelman).

Poglavje 2

Vizualni učinki v filmu in videu

2.1 Kaj so vizualni učinki in zakaj jih uporabljamo

Vizualni učinki so različne metode, kako sliki (filmu, videu, fotografiji ali drugačnemu vizualnemu gradivu) dodamo elemente oziroma spremenimo ali izboljšamo sliko[9]. Gre za tehnike, ki jih ne moremo posneti na kamero med snemanjem žive akcije. Uporabljamo jih iz različnih razlogov, predvsem pa kadar bi bilo želeno nemogoče posneti, kot zahteva scenarij (primer potovanje na luno v filmu *Apollo 13*, 1995). Če bi bilo snemanje prizora prenevarno za igralce ali druge udeležence na snemanju, se prav tako lahko uporabi vizualne učinke za doseg končnega rezultata brez izpostavljanja ljudi pravi nevarnosti. Tako so uporabljeni vizualni efekti naprimer v akcijskih filmih, pa tudi naprimer v filmu *Fire Brigade*, 1926, v katerem je prikazana dojenčica v goreči hiši. Naslednji razlog za uporabo vizualnih efektov pa je zniževanje cene produkcije. Tako naprimer v filmu *Gospodar prstanov: Stolpa* (ang. *The Lord of the Rings: The Two Towers*, 2002) horde orkov napadajo trdnjavo. Najeti toliko statistov bi bilo drago in nepraktično, saj je vsak moral tudi čez proces maske vsak snemalni dan. Lažje in prav tako dobro je posneti manjšo skupino maskiranih statistov in jih nato v postprodukciji vstaviti v sliko večkrat.

Med vizualne učinke štejemo na vse načine pridobljene grafične elemente, ki jih ne moremo posneti hkrati z akcijo. Ločimo jih od posbnih učinkov oziroma praktičnih učinkov. Poseni učinki pa so tisti, ki jih posnamemo med snemanjem akcije. Primeri takšnih učinkov so streli z orožjem, eksplozije, umeten dež s cevi in podobno. Danes se jih tipično kombinira z vizualnimi učinki za povečan realizem. Primeri kombiniranih efektov so na primer eksplozije v Transformerjih (ang. *Transformers*, 2007).

2.2 Primerjava uporabe vizualnih učinkov v filmu in videu

Ko govorimo o vizualnih učinkih, se je naenostavneje opreti na primere iz sveta filma. Filmi večinoma ciljajo na neko mero realizma. Seveda se razlikujejo po stopnji realizma - od izjemno realističnih filmov Michaela Hanekeja, na primer Sedmi kontinent (nem. *Der Siebente Kontinent*, 1989), hitro pridemo do fantazijskih filmov in celo popolnoma neprepričljivih filmov, kot je na primer japonski eksperimentalni film Tetsuo, železni človek (jap. *Tetsuo*, 1989). V filmu se skoraj vedno trudimo skriti vizualne učinke, jih narediti dovolj prepričljive, da jih gledalec ne opazi. Skozi leta in razvoj tehnologije so seveda vizualni učinki postajali bolj in bolj prepričljivi. Danes jih nepoučen opazovalec le redko opazi, pa še takrat v nizkoprorračunskih produkcijah, ki ponavadi uporabljajo cenejšo in starejšo tehnologijo, poleg tega pa ponavadi tudi nimajo dostopa do izkušenih umetnikov vizualnih učinkov.

Danes, ko je film vedno pogostejše tudi digitalno posnet in tako z vidika medija pravzaprav video, je ta razlika manjša. Z enakimi orodji delajo veliki filmski studiji in alternativni umetniki, ki ustvarjajo umetniški avtorski video. Adobejev programski paket Premiere Pro so uporabili v studiju Marvel za montažo filma Deadpool (ang. *Deadpool*, 2016),

Avtorski video pa je po naravi popolnoma drugačen. V osnovi se ne opira na vizualni realizem, ampak skozi uporabo vizualnega in zvočnega materiala predvsem poskuša na človeka vplivati čustveno[3]. Pascal Bonitzer primerja

video s poezijo, specifično s haikujem. Video se odmakne od realizma prav zaradi enostavnosti digitalne manipulacije. Za razliko od filmskega traku, ki je enkratno zapisan in je za večino obdelave potrebno presnemavanje na drug trak, skeniranje v digitalno obliko in nato optično tiskanje nazaj na trak, je video v osnovi spremenljiv.

V videu so se uporabljali drugačni vizualni učinki kot v filmu. Začetek vizualnih učinkov za video predstavlja uporaba slike v sliki med napovedovanjem novic, maskiranje s pomočjo modrega zaslona pri napovedovanju vremena in podobni vizualni učinki za televizijske oddaje.

2.3 Vpliv videa na alternativno kulturo

Avtorski video je bil zelo popularen med alternativnimi umetniki v 70. in 80. letih 20. stoletja. Takrat so postale video kamere in sistemi za montažo postopoma dostopni tudi za javnost, ne le za velike televizijske in produkcijske hiše [5]. Na uporabo videa v umetnosti, sicer predvsem performančni, je močno vplival pojav Sonyjevega Portapacka leta 1967. To je bil majhen prenosni sistem za urejanje videa. Sestavljen je bil iz majhne črno-bele kamere in sistema za zapis, predvajanje in presnemavanje magnetih trakov. Zaradi možnosti takojšnjega pregleda video in relativno nizke cene je postal zelo popularen med protivojnimi protestniki in umetniki v ZDA, ki so bile takrat v vijni z Vietnamom [7]. Tako se je video utemeljil kot medij alternativne kulture. Seveda se je hkrati uporabljal tudi za televizijske produkcije, kar je prineslo razvoj v drugi smeri. Za namene televizijskih prenosov so bile razvite tehnike za vstavljanje slike v video, različni prehodi in podobno. Tudi takšne tehnike so bile pogosto uporabljene v umetniških produkcijah. V Sloveniji so konec 1970. let in kasneje alternativni umetniki pogosto sodelovali z RTV Ljubljana [2]. Od njih so si tudi izposojali opremo in razvili nove tehnike uporabe le te. Tehnika imenovana feedback loop (vstavljanje zakašnjega videa samega vase) je bila razvita v alternativni sceni, kasneje pa se je uporabljala tudi v komercialnih programih.

2.4 Pregled zgodovine vizualnih učinkov

Filmska montaža je postopek, pri katerem iz posnetkov sestavimo zgodbo. Je ključna dejavnost v postprodukciji. Sergej Mihajlovič Eisenstein, ruski filmar (Oklepnic Potemkin, 1925), celo postavi montažo na prvo mesto v filmskem procesu. Pravi, da je montaža najpomembnejša faza filmske produkcije.

Zgodovinsko se je tehnologija filmske in video montaže začela razvijati že konec 19. stoletja. Leta 1895 je v kratkem filmu Usmrnitev Marije I. Škotske (ang. *The Execution of Mary, queen of Scots*, 1895) režiser Alfred Clark prikazal, kako kraljici odsekajo glavo. Navidezno usmrnitev kraljice je dosegel tako, da je tik pred rabljevim zamahom s sekiro kamero ustavil in v tem času zamenjal igralko z lutko, nato pa nadaljeval snemanje. To je prvi dokumentiran vizualni učinek. Viri navajajo, da je marsikateri gledalec verjel, da so igralko med snemanjem zares usmrtili. Ta film je pokazal, da montaža in vizualni učinki lahko pripomorejo k pripovedni moči filma.

Razvoj vizualnih učinkov gre z roko v roki z novimi montažerskimi prijemi, novo filmsko tehnologijo in razcvetom filma v 20. stoletju. Že nekaj let kasneje je za film Veliki rop vlaka (ang. *The Great Train Robbery*, 1903) studio Thomasa Edisona razvil tehnologijo maskiranja filma (ang. *black matte*) in ponovne ekspozicije maskiranega predela, da je v studiju lahko prikazal pogled skozi okno, v katerega je vstavil razgled. V 1930. letih so začeli z uporabo projekcije ozadij ali drugih elementov na platno, pred katerega so nato postavili scenografijo in igralce. Istočasno se je razvila tudi uporaba pomanjšanih setov in figur oziroma miniaturn, ki so jih lahko animirali na način stop motion in kasneje z uporabo projekcije ali maskiranja kombinirali s posnetki igrane akcije. V tem času se je uporabljala tudi tehnika slikanja na steklo, ki se je postavilo pred lečo kamere za dodajanje vizualnih elementov. Zanimivo je, da so v 30. letih, v času črnobelih filmov, že začeli uporabljati tehniko odstranjevanja ozadja s pomočjo modrega zaslona. Moder zaslon namreč lahko z uporabo modrega filtra na črnobelem posnetku popolnoma izgine in pusti za seboj neizpostavljen film, kamor lahko vstavimo druge slike.

V 50. letih prejšnjega stoletja so v Disneyevih studijih odkrili tehniko,

kako hkrati posneti sliko in potujočo masko (ang. *traveling matte* - način maskiranja, pri katerem se masko posname na trak in se le ta zato popolnoma ujema s sliko) s pomočjo modrega zaslona. S to tehniko so posneli film *Marry Poppins* (195x), v katerem naslovna junakinja leti s pomočjo dežnika, oziroma s pomočjo modrega zaslona in potujoče maske. Tehniko so v 80. letih izpopolnili s pomočjo na UV svetlobo občutljivega filmskega traku in barvanjem elementov maskiranja z UV odbojno barvo za film *Vojna Zvezd* (ang. *Star Wars*).

V naslednjih desetletjih se je razvoj filmskih posebnih učinkov razvijal predvsem v smeri izpopolnjevanja že poznanih tehnik. Tako so v 60. letih razvili novo tehniko projiciranja na zaslon v studiju od spredaj, kar je poskrbelo za bolj jasno sliko ozadij, večji kontrast in bolj pravilne barve. V 70. letih se je začela uporabljati tehnika maskiranja z modrim zaslonom tudi za barvne filme, v 80. letih pa so uspeli doseči maskiranje brez uhajanja modre svetlobe na objekte ¹.

Prvi računalniško ustvarjeni vizualni efekti so se pojavili že v 50. letih 20. stoletja. John Whitney je uporabil odslužen analogni računalnik za vodenje protiletalskih izstrelkov kot pripomoček za animacijo. Na podlagi njegovega dela so kasneje razvili tudi tehniko, s katero so v filmu *Odiseja 2001* posneli sekvenco potovanja skozi črvino.

Prvi pravi digitalni posebni učinki pa so bili uporabljeni v filmu *Westworld* (ang. *Westworld*, 1973). V tem filmu so prikazali človeku podobne robote in v nekaterih prizorih dele igralcev nadomestili s 3D računalniško generirano grafiko. Podjetje Triple-I je kasneje tehniko razvijalo dalje. Nekaj let kasneje so podobne vizualne učinke uporabili v filmu *Futureworld* (ang. *Futureworld*). Leta 1982 pa so za film *Tron* (ang. *Tron*, 1982) s pomočjo računalniške 3D CGI upodobili celoten fikijski svet in objekte v njem. V filmu *Tron* efekti niso bili prepričljivi, kar pa ustvarjalcev ni oviralo, saj film prikazuje digitalen

¹ang. *blue-spill* je odsev modre svetlobe s platna na druge objekte. Izoognemo se mu lahko s pomočjo natančne osvetlitve platna, neodvisne od drugih virov svetlobe ali s pomočjo filtrov v post produkciji.

svet oziroma navidezno resničnost. Leta 1985 je studio Pixar ustvaril prvi 3D CGI animiran lik. To je bil stekleni vitez v filmu Mladi Sherlock Holmes (ang. *Young Sherlock Holmes*). V 90. letih so digitalni vizualni učinki hitro napredovali zaradi razvoja računalniške tehnologije, ki je postala dostopnejša in končno dovolj zmogljiva za resno delo. Leta 1992 so v filmu prvič prikazali prepričljive 3D CGI živali, pingvine in netopirje v filmu Batman se vrača (ang. *Batman Returns*, 1992). Leto kasneje pa je film Jurski Park (ang. *Jurassic Park*, 1993) zelo uspešno kombiniral 3D CGI, lutke, živo igro in posnetke miniaturne.

Danes velika večina filmov in videa uporablja digitalne video učinke. Ti efekti so pogosto popolnoma neopazni, kot na primer dodan sneg v filmu Kradljivka knjig (ang. *The Book Thief*, 2013). V filmski industriji se vizualni efekti večinoma uporabljajo čimbolj neopazno in ne opozarjajo gledalca nase [9]. Izjema so namerni umetniški učinki, ki na primer v filmu Doktor Strange (ang. *Doctor Strange*, 2016) prikazujejo drug svet. Tam jih ne moremo zgrešiti, pa jih kljub temu gledalci lahko ignorirajo, saj so uspešno uporabljeni kot pripovedno orodje.

Poglavje 3

Uporaba After Effects za vizualne učinke

ghdfgh

3.1 Hiter pregled After Effects

After Effects je orodje za video montažo, izdelavo vizualnih učinkov in video kompozicijo.

3.2 Kompozicija digitalnega videa

Primarni namen After Effects je kompozicija videa. Kompozicija je postopek, pri katerem iz več videov, slik in drugih vizualnih elementov sestavimo končno sliko tako, da jih razporedimo enega nad drugega v 2.5D prostoru¹ in nato med seboj kombiniramo[10]. Ko kombiniramo plasti lahko izberemo med več različnimi načini mešanja (ang. *blending mode*) oziroma operacija nad piksli.

Današnja definicija kompozicije videa je bila postavljena leta 1984 v članku Compositing Digital Images[?]. Članek je osnoval nov algoritem za kompozicijo slike s pomočjo uvedbe kanala alpha (ang. *alpha channell*), ki ponazarja pokrivnost piksla. Kanal alpha je definiran kot relativna pokrivnost ele-

operacija	prevod	pomen
clear	prazno	Izhodna slika je popolnoma nepokrivna oziroma prazna.
A	A	Izhodna slika je enaka prvemu operandu.
B	B	Izhodna slika je enaka drugemu operandu.
A over B	A nad B	Izhodna slika je sestavljena iz slike A, postavljene nad B. Kjer ima prvi operand vrednost α 1, je v izhod enak prvemu operandu. Sicer je izhod enak linearni kombinaciji αA

Tabela 3.1: Operacije kompozicijske algebre.

menta, pri kateri 0 pomeni nično pokrivnost, 1 pa polno pokrivnost. Vmesne vrednosti se izračunajo z linearno interpolacijo. Slike, ki vsebujejo podatek o pokrivnosti posameznih pikslov tako brez težav lahko definiramo v prostoru RGBA.

V istem članku je bila predstavljena tudi kompozicijska algebra, kakršno uporabljamo še danes. Definira skupino operacij nad dvema slikovnim elementoma, pri čimer posebno pozornost namenja ohranjanju pravilne vrednosti kanala α v izhodni sliki. S tem je dosežena možnost kombiniranja mnogih slojev videa s kombiniranjem binarnih operacij med sosednjimi sloji. Operacije so našteje in razložene v tabeli.

3.3 Kompozicija videa v After Effects

V After Effects enoto dela, ki ponavadi predstavlja sceno, imenujemo sekvenca[1]. Ko ustvarimo novo sekvenco, vidimo časovnico te sekvence in prazno platno. Platno predstavlja delovno površino, ki tipično predstavlja ciljni format videa, ki določi njeno resolucijo, razmerje med višino in širino in s tem tudi

¹2.5D je izraz, ki pomeni 2 dimenzionalen prostor, v katerem pa lahko posamezne 2D objekte razporedimo enega nad drugega v smeri tretje osi Z.

relativno širino piksla. V časovnico ali pa na platno lahko dodajamo video, slike in druge elemente. Na platnu lahko elemente postavimo v prostoru ekrana, časovnico pa uporabljamo za postavitev v časovnem prostoru in jih razporedimo v smeri Z osi.

V AE se video komponira na podlagi kompozicijske algebre, ki poudarja uporabnost kanala alpha. Poleg tega pa lahko dele videa tudi ločeno maskiramo oziroma izberemo predele slike, ki bodo vidna. Pri tem imamo na izbiro več orodij, kot so:

- pravokotne maske, ponavadi uporabljane za hitro odstranjevanje elementov iz sicer drugače maskiranega videa, na primer odstranjevanje luči iz videa, posnetega pred zelenim zaslonom,
- elipsoidne maske, ki se ponavadi uporabljajo v procesu popravljanja barv² za enostavno ločevanje obrazov od ozadja,
- rotoskopirane maske - animirane maske, ki jih uporabnik naredi s preprostim risanjem maske, AE pa mu pomaga pri ekstrapolaciji animacije na naslednje sličice,
- rotoskopirane maske s pomočjo orodja Mocca for After Effects, komercialnega produkta za rotoskopiranje mask, ki pa je v minimalni verziji že vgrajen tudi v AE,
- efektov, ki ustvarijo masko na podlagi različnih parametrov in slike, in drugih.

Tipično se orodja za hitro maskiranje uporabljajo v kombinaciji z drugimi metodami kompozicije, na primer odstranjevanjem zelenih zaslonov in podobno. Pri vseh orodjih lahko pozicijo, velikost in obliko animiramo. To storimo tako, da na določenih sličicah masko oblikujemo, med temi ključnimi sličicami pa bo AE za nas animacijo interpoliral. Ker je interpolacija lahko nenatančna, imamo na voljo tudi urejevalnik krivulj, s pomočjo katerega lahko natančneje določamo način interpolacije.

Maskam lahko tudi nastavljamo ostrost robov. Ta parameter se v AE imenuje *feather* (slo. *pero*). Nastavimo lahko, koliko pikslov daleč od začrtane maske se bo kanal alpha interpoliral med polnim maskiranjem in odsotnosjo maske. Pri tem parametru je interpolacija linearna, saj se uporablja le kot način, s katerim masko skrijemo pred gledalcem. Ostri robovi pravilnih oblik pritegnejo pozornost, zato jih skoraj vedno vsaj malo zameglimo.

3.4 Uporaba digitalnih vizualnih učinkov v AE

Programski paket Adobe After Effects vsebuje veliko izbiro digitalnih vizualnih učinkov. Vgrajene, lastne ali druge efektne vtičnike lahko uporabimo na več načinov. Najpreprosteje je izbrati plast, ki predstavlja vhodni video, nato pa izbrati željeni učinek z menija „Effects“ (slo. *Učinki*). S tem dodamo efekt na video. V oknu „Effect controls“ se s tem prikaže uporabniški vmesnik vtičnika. Vsak vtičnik ima lasten uporabniški vmesnik. Večinoma so sestavljeni iz vgrajenih klasičnih komponent, kot so drsniki, pojavnna okna, gumbi in podobno, zato so enostavni za razumevanje uporabnikom AE, ki takšne kontrolnike že poznajo iz programa gostitelja.

Na posamezno plast lahko dodamo več učinkov, pri čimer je izhod iz enega uporabljen kot vhod naslednjega. Na ta način je omogočeno sestavljanje kompleksnih novih učinkov. Pri oblikovanju slike je dobro orodje za minimizacijo kompleksnosti predkompozicija (ang. *Precomposing*). S tem orodjem lahko eno ali več plasti izvozimo v novo kompozicijo. Predkomponirane plasti lahko

²Popravljanje barv (ang. *color correction* ali *color grading*) je dvodelen proces spreminjanja barv videa. V prvem koraku se popravi morebitne napake snemalca, kot so nepravilno izpostavljena slika, napačno izbrana nevtralna belina in podobno. Drugi korak je umetniški proces, pri katerem barvni tehnik s pomočjo režiserja prilagodi barve, da z njimi čustveno vpliva na gledalce. Primer močno in stilizirano popravljenih barv je film *Matrica* (ang. *The Matrix*, 1999). V filmu je s pomočjo močne zelene obarvanosti loči virtualno resničnost matrice od bolj nevtralnno obarvanih prozor, ki se dogajajo v resničnem svetu.

uporabljammo enako kot enostavne plasti, nanje dodajamo nove učinke in jih naprej komponiramo po želji. Prekompozicije lahko urejamo neodvisno od celotne kompozicije. Če se predkompozicija spremeni, je seveda potrebno ponovno upodobiti del kompozicije, na katerega ta sprememba vpliva.

Omogočena je tudi uporaba učinkov nad več plastmi hkrati s pomočjo posebnih prilagoditvenih plasti (ang. *Adjustment layer*). Če na prilagoditveno plast dodamo učinek, bo deloval nad vsemi učinki, ki so v skladu plasti pod njo. To omogoča enostavno in nedestruktivno urejanje celotne slike, kar je zelo priročno v določenih procesih post produkcije, naprimer pri popravljanju barv.

After Effects se pogosto uporablja v kombinaciji s Adobe Premiere Pro³. Med programskima paketoma je omogočena enostavna komunikacija. Montažer lahko iz Premiere Pro izvozi sceno ali kader naravnost v AE in jo nadomesti s kompozicijo. Ko v AE shranimo, se kompozicija v Premiere Pro posodobi in montažer lahko uporablja celoten kader z vizualnimi učinki.

³Adobe Premiere Pro je programski paket, namenjen predvsem video montaži. Sicer se zmogljivosti AE in Premiere Pro močno prekrivajo, vendar je vseeno mnogo lažje uporabljati vsak paket za lasten namen.

Poglavje 4

Razvoj vtičnikov za After Effects

Za razvoj vtičnikov potrebujemo razvojno okolje za C++, SDK, ki ga ponuja Adobe na svoji spletni strani, in seveda programski paket After Effects. SDK vsebuje navodila za razvoj, opis delovanja vtičnikov v okolju programa After Effects in seveda opis funkcij, podatkovnih struktur in tipov, ki se uporabljajo v tem okolju. Polet tega je v SDK vključenih tudi veliko primerov vtičnikov. Vidimo lahko njihovo programsko kodo, jo spreminjamo ali uporabimo kot izhodišče za lastne vtičnike.

Adobe močno priporoča razvojno okolje Microsoft Visual Studio, če uporabljamo operacijski sistem Windows in Apple Xcode na operacijskem sistemu Mac OS X. Poskrbeti moramo tudi, da imamo nameščene primerne verzije vseh orodij. Ker pri programiranju vtičnikov izhajamo iz primerov ali predlog, lahko hitro naletimo na težave, če imamo napačno verzijo razvojnega okolja ali SDK, ki ni namenjen razvoju za tisto verzijo AE, ki jo imamo nameščeno za testiranje.

Sam sem se odločil za delo na operacijskem sistemu MS Windows 10 Enterprise z okoljem MS Visual Studio 2015 update 3 za Adobe After Effects CC 2017.1 in Adobe After Effects CC 2017.1 Win SDK.

4.1 Delovanje vtičnikov

Vtičniki so knjižnjice DLL, ki jih After Effects med zagonom naloži. Nato poteka komunikacija med vtičnikom in programom gostiteljem tako, da se vtičnik odziva na klice gostitelja s primerno funkcijo. Vtičnik ima vedno le eno vhodno točko, funkcijo s podpisom:

```
PF_Err plugin_name (  
    PF_Cmd cmd,  
    PF_InData *in_data,  
    PF_OutData *out_data,  
    PF_LayerDef *output,  
    void *extra)
```

After Effects ugotovi, kje se nahaja vhodna točka vtičnika s pomočjo resusra PiPL (ang. *plug-in property list* - seznam lastnosti vtičnika). To je dokument, ki ga napiše programer in katerega vsebina se mora ujemati z implementacijo vtičnika. Moderne verzije AE sicer večino lastnosti pridobijo direktno iz programske kode vtičnika in ne iz datoteke PiPL. Kljub temu se morajo podatki ujemati iz zgodovinskih razlogov in združljivosti s starejšimi verzijami datotek in vtičnikov.

Ko je vhodna funkcija poklicana, se mora na podlagi ukaznega izbirnika `cmd` pravilno odzvati. Vtičnik mora nekatere klice nujno podpreti, nekateri pa so opcijski. Vhodna točka je tako funkcija, ki skrbi le za to, da se bo prava funkcija odzvala na klic programa in pravilno reagira v primeru napak. Vrne številko napake in poskusi brez neželenih stranskih učinkov zalkjučiti izvajanje vtičnika.

Adobe priporoča, da kadar je le možno, uporabimo vgrajene funkcije in programske konstrukte, ki jih implementira AE in lahko do njih dostopamo prek SDK. S tem ne le skrajšamo čas, potreben za razvoj vtičnikov, ampak tudi zmanjšamo verjetnost za vpeljevanje hroščev in napak. Vgrajene

¹Sekvenca v AE predstavlja samostojen projekt z enim ali več sloji videa in zvoka, nad katerimi izvajamo operacije.

ukazni izbirnik	odziv
Ukazni izbirniki, na katere se je potrebno odzvati ne glede na tip vtičnika	
PF_Cmd_ABOUT	Prikaži podatke o vtičniku - gostitelj bo pripravil dialog, vtičnik pa mora pripraviti vsebino.
PF_Cmd_GLOBAL_SETUP	Pripravi globalne podatke o vtičniku in rezerviraj pomnilnik za podatke, ki morajo biti dostopni ves čas delovanja vtičnika.
PF_Cmd_GLOBAL_SETDOWN	Če je v PF_Cmd_GLOBAL_SETUP bil rezerviran pomnilnik, ga sprostí. Sicer se ni potrebno odzvati.
PF_Cmd_PARAM_SETUP	Pripravi uporabniški vmesnik za podajanje parametrov vtičniku.
Za upravljanje s podatki sekvence ¹	
PF_Cmd_SEQUENCE_SETUP	Rezerviraj pomnilnik za podatke, ki niso odvisni od sličice, ki se obdeluje, ampak morajo biti dostopni tekom cele sekvence.
PF_Cmd_SEQUENCE_SETDOWN	Sprostí ves spomin, ki je bil rezerviran v PF_Cmd_SEQUENCE_SETUP
Ukazni izbirniki, ki se pošljejo za vsako sličico videa	
PF_Cmd_RENDER	Upodobi sličico. Klic zahteva, da vtičnik pripravi in v dodeljeni spomin zapiše vrednosti vseh pikslov. klic se uporablja samo za prikazovanje slojev z barvno globino 8 ali 16 bitov na kanal.
PF_Cmd_SMART_PRE_RENDER	Se uporablja pri efektih tipa SmartFX. Vtičnik mora glede na željene pogoje prepoznati dele slike, ki jih bo upodobil.
PF_Cmd_SMART_PRE_RENDER	Klic SmartFX za upodobitev slike. V vnaprej prepoznane ciljne dele slike vpiši nove piksle.

Tabela 4.1: Nekateri izmed pogostejše podprtih ukaznih selektorjev.

funkcije seveda ne bodo delovale, če jih kličemo nepravilno in imajo vgrajene mehanizme za preprečevanje puščanja pomnilnika in sesuvanja. Pomembno je, da tudi, če vtičnik ne deluje pravilno, ne sesuje gostitelja. Če se sesuje gostitelj, lahko namreč končni uporabnik izgubi podatke in s tem svoje delo.

Zato nam naprimer SDK ponudi funkcije za rezervacijo pomnilnika, s katerimi ne tekmujemo z gostiteljem in operacijskim sistemom za delovni pomnilnik, saj rezervacijo za nas opravi gostitelj. S tem se izognemo situaciji, ko bi vtičnik porabil toliko spomina, da bi ga zmanjkalo za AE. S tem bi se sesul AE in z njim seveda tudi vtičnik, ki je odvisen od izvajanja programa gostitelja.

4.1.1 SmartFX

Posebna kategorija so vtičniki SmartFX. Od navadnih efektov se razlikujejo po tem, da omogočajo dvosmerno komunikacijo z AE in podpirajo video z barvno globino 32bitov na kanal. Dvosmerno komunikacijo se doseže z implementacijo dodetnega ukaznega izbirnika `PF_Cmd_SMART_PRE_RENDER`, ki AE sporoči, katere dele slike bo upodobil.

4.2 Metoda razvoja

Pred razvojem željenega vtičnika se je dobro spoznati z uporabo SDK. V navodilih priporočajo, da si programer med primeri prebere in analizira tiste, ki so po načinu delovanja podobni ciljnemu[4]. Tega lahko nato začnemo postopoma spreminjati in testirati, da spoznamo, kako vtičnik komunicira in sodeluje z aplikacijo AE. Marsikateri podatki v navodilih manjkajo ali pa so zastareli, zato je potrebno kar veliko preizkušanja in raziskovanja. Skozi proces spoznavanja interakcije med programom in vtičnikom lahko oblikujemo strukturo vtičnika. Odločiti se moramo, kako razporediti operacije med funkcije in kako strukturirati podatke, da bodo dostopni ob pravem času in ne bodo ostajali v spominu. Glavno pri razvoju vtičnikov je pravzaprav, da se program pravilne odzove na vse klice gosti-

telja posamezno, hkrati pa ostane vtičnik logična enota. Seveda je prav tako pomembno, da je vtičnik uporaben in prijeten za uporabo z vidika končnega uporabnika. Na to moramo paziti pri oblikovanju uporabniškega vmesnika in tudi pri implementaciji učinka, da ostane odziven v različnih primerih uporabe. Med implementacijo si seveda lahko veliko pomagamo s razhroščevalnikom razvojnega okolja, navodili in Adobejevim forumom za razvijalce (http://forums.adobe.com/community/aftereffects_general_discussion/aftereffects_sdk).

4.3 Vtičnik PixelRain

4.3.1 Razvoj ideje

Prvi vtičnik, za katerega sem dobil idejo, sem poimenoval PixelRain. Ideja je zelo preprosta: nekatere piksele bo vtičnik prekopiral navzdol, s čimer se bodo na sliki pojavile črte. Kateri piksli se bodo kopirali in kateri ne, določi uporabnik na podlagi izbire barve, kateri podobne piksele bo vtičnik prekopiral. Poleg izbire barve bo uporabnik lahko določil tudi maksimalno dolžino sledi - kolikokrat se lahko piksel prekopira navzdol, in maksimalno razdaljo med percepcijo izbrane barve in barve piksla, da bo še štel kot podoben.

V originalni ideji smo podobnost barve definirali kot razdaljo med dvema barvama v prilagojenem prostoru RGB. Ker je cilj poiskati podobne barve, smo prostor RGB transformirali oziroma raztegnili, da bi razdalja bila bližja človeški percepciji razlike v barvah. Transformacija RGB prostora izhaja iz tega, da človeško oko naprimer zazna več različnih odtenkov zelene barve kot ostalih - bolj natančno ločimo zelene odtenke kot ostale. To upoštevamo v izračunu vizualne podobnosti barv tako, da razdaljo izračunamo po formuli

$$\sqrt{2R^2 + 4G^2 + 3B^2}[8].$$

Testiranje je to metodo pokazalo kot nerazumljivo in navidez nelogično. Pogosto so bili rezultati drugačni, kot bi jih pričakovali, kar z vidika uporabnika pomeni ni zaželeno[11].

Po premisleku smo se odločili za implementacijo izbire barve s pomočjo treh parametrov v barvnem prostoru HSL[6]. Tako lahko uporabnik ločeno izbira barvni odtenek, nasičenost in svetllost. Ta način izbire barve sicer ne izbere barv, ki so enako oddaljene od ciljne v nobenem barvnem prostoru, je pa razumljiv in producira rezultate, kakršne uporabnik pričakuje. Zaradi ločenja na tri izbirnike smo s tem dosegli tudi večjo fleksibilnost vtičnika.

4.3.2 Implementacija vtičnika

Vtičnik implementira odzive na ukazne izbirnike `PF_Cmd_ABOUT`, `PF_Cmd_GLOBAL_SETUP`, `PF_Cmd_PARAMS_SETUP` in `PF_Cmd_RENDER`.

Literatura

- [1] <http://www.adobe.com/training/books/aftereffects.html> !TODO.
- [2] TODO asdfadf. *TODO to je ena izmed knjig ki mi ji h je dala Helena*. 1999.
- [3] Pascal Binitzer. *Slepo Polje*. ŠKUC, Znanstveni inštitut Filozofke fakultete, Ljubljana, Slovenija, 1985.
- [4] Russell Belfer et al. *After Effects CC 2017.1 SDK Guide, Release 1*. !TODO, 2017.
- [5] Marina Gržinić. *Rekonstruirana fikcija: novi mediji, (video) umetnost, postsocializem in retroavantgarda: teorija, politika, estetika: 1997-1985*. Koda. Študentska organizacija univerze, Študentska založba, Ljubljana, Slovenija, 1997.
- [6] B. Hill, Th. Roger, and F. W. Vorhagen. Comparative analysis of the quantization of color spaces on the basis of the cielab color-difference formula. *ACM Trans. Graph.*, 16(2):109–154, April 1997.
- [7] Nenad Puhovski. Jedan pomalo tužan pogled na američki video 1983. godine. In Žarana Papić, editor, *Videosfera*, pages 42–45. Studentski izdavački centar UKSSO Beograda, Beograd, Srbija, 1986.
- [8] Thiadmer Riemersma. Colour metric, Apr 2012.
- [9] Visual Effects Society. *The VES Handbook of Visual Effects*. Esevier Inc., Oxford, Anglija, 2010.

- [10] Tom Duff Thomas Porter. Compositing digital images. *Computer Graphics SIGGRAPH'84*, 18(3):253–259, 1984.
- [11] TODO. *TODO to je knjiga iz FRI založbe o UI.*