VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
PROGRAMŲ SISTEMŲ BAKALAURO STUDIJŲ PROGRAMA

Neuroniniais tinklais paremtas vaizdų stiliaus perkėlimas realiu laiku žaidimuose

Real-Time Neural Style Transfer in Video Games

Bakalauro baigiamasis darbas

Atliko:	Augustas Masolas	(parašas)
Darbo vadovas:	asist. dr. Vytautas Valaitis	(parašas)
Darbo recenzentas:	j. asist. Boleslovas Dapkūnas	(parašas)

Vilnius – 2022

Santrauka

Naudojantis konvoliuciniais neuroniniais tinklais atlikti eksperimentai ir remiantis eksperimentų rezultatais pateikiamos naujos stiliaus perkėlimo pritaikymo sritys žaidimų kūrime, naudojant stiliaus perkėlimą realiu laiku. Šiame darbe iškeliamos stiliaus perkėlimo taikymo problemos ir pateikiamos rekomendacijos. Eksperimentų metu rastos naujos panaudojimo sritys, tokios kaip stiliaus perkėlimo panaudojimas prototipų kūrimo metu ir unikalių animuotų tekstūrų kūrime. Naudojantis stiliaus perkėlimo modeliu žaidimų kūrėjai gali poros mygtukų paspaudimu realiu laiku pakeisti vaizdo ar objekto tematiką ir taip susidaryti galutinį žaidimo aplinkos ar objekto vaizdą. Eksperimento metu taip pat nustatyta kitų autorių aprašyta stiliaus perkėlimo problema - stiliaus perkeltų vaizdų kadrų nepastovumas laike (angl. *temporal consistency*). Šiai problemai spręsti rekomendacijų dalyje pateikiami sprendimo būdai, o trečiojo eksperimento metu aprašomas būdas išnaudoti šią problemą kuriant unikalias animuotas tekstūras. Toks animuotų tekstūrų kūrimas nereikalauja papildomo kodo rašymo, skirtingai nei įprasti metodai. Rezultatuose pateikiami palyginimai tarp dviejų skirtingų nagrinėtų konvoliucinių neuroninių tinklų, skirtų stiliaus perkėlimui. Darbe atliekama neuroniniais tinklais paremto stiliaus perkėlimo tinkamumo naudojimui analizė ir pateikiamos naujos stiliaus perkėlimo panaudojimo sritys žaidimuose. Taip pat analizuojami greitojo stiliaus perkėlimo ir interaktyvaus stiliaus perkėlimo konvoliuciniai neuroniniai tinklai - jų privalumai ir trūkumai, pateikiamos rekomendacijos, kurioms užduotims šie tinklai yra efektyvesni.

Raktiniai žodžiai: konvoliuciniai neuroniniai tinklai, stiliaus perkėlimas, mašininis mokymas, neuroniniai tinklai, kompiuterinių žaidimų kūrimas

Summary

Experiments using convolutional neural networks are carried out and new applications of style transfer in real-time game development are presented based on results from the experiments. This paper raises problems of applying style transfer real-time in video games and makes recommendations for solving the resulting problems. The experiments lead to new areas of application, such as the use of style transfer in prototyping and in the creation of unique animated textures. Using style transfer models, game developers can change the theme of an image or an object with a couple of clicks to help imagine the final look of the game environment or object. The experiment also identified a problem with style transfer described by other notable authors, which is that different frames of the style-transferred images from real-time view are temporally inconsistent. The Recommendations section provides solutions to this problem and the third experiment describes a way to exploit this problem by using it to create unique animated textures. This way of creating animated textures does not require writing additional code as conventional methods do. The results show comparisons between two different convolutional neural networks for style transfer. This paper highlights the advantages and disadvantages of these convolutional neural networks for style transfer and highlights which tasks these networks are more effective at.

Keywords: convolutional neural networks, style transfer, machine learning, neural networks, computer game development

TURINYS

ĮVADAS 5

1. KONVOLIUCINIS NEURONINIS TINKLAS 7
   1.1. Naudotas neuroninis tinklas paveikslėlių kokybei vertinti 7
   1.2. Naudotas konvoliucinis neuroninis tinklas paveikslėliams generuoti 7
   1.3. Tiesioginis normalizavimas 9
   1.4. Antras naudotas konvoliucinis neuroninis tinklas vaizdų stiliaus perkėlimui 10

2. NEURONINIAIS TINKLAIS PAREMTAS VAIZDŲ STILIAUS PERKĖLIMAS ŽAIDIMUOSE REALIU LAIKU 12
   2.1. Greitojo stiliaus perkėlimo DNT modelio apmokymas 12
   2.2. Interaktyvaus stiliaus perkėlimo DNT modelio apmokymas 13
   2.3. Žaidimo paruošimas neuroniniui tinklui 14
   2.4. Pirmasis eksperimentas 15
   2.5. Antrasis eksperimentas 16
   2.6. Trečiasis eksperimentas 16

3. EKSPERIMENTŲ REZULTATAI 19
   3.1. Pirmojo eksperimento rezultatai 19
   3.2. Antrojo eksperimento rezultatai 23
   3.3. Trečiojo eksperimento rezultatai 25
   3.4. Rezultatų palyginimas 26

4. REKOMENDACIJOS 28
   4.1. Mirgėjimo arba laikino nepastovumo mažinimas DNT 28
   4.2. Žaidimo optimizavimas 28
   4.3. DNT modelio optimizavimas 28

REZULTATAI 29

IŠVADOS 30

LITERATŪRA 31

PRIEDAI 32
   1 priedas. Žaidimų kodai su stiliaus perkėlimu 33
   2 priedas. Stiliaus perkėlimo modelio apmokymo kodas ir modeliai 34

Įvadas

Pastaraisiais metais mašininio mokymosi sistemos vis dažniau taikomos kūrybinėms užduotims atlikti. Mašininio mokymo išpopuliarėjimo vizualiųjų menų srityje pagrindas, tai 2015 metais Leon A. Gatys pasiūlytas konvoliucinių neuroninių tinklų DNT taikymas transformavimo uždaviniuose ir šių tinklų požymių aktyvacijų (angl. *feature activations*) panaudojimas. Straipsnyje [GEB15] aprašyti eksperimentų rezultatai parodė, kad konvoliuciniai neuroniniai tinklai gali išgauti informaciją apie gerai žinomo meno kūrinio turinį ir stiliaus informaciją, tai leido pradėti taikyti konvoliucinius neuroninius tinklus vaizdų transformavimo užduotyse. Vaizdų transformacijos užduotys apima paveikslėlių spalvinimą [ISI16; ZIE16], rezoliucijos didinimą [DLH+16; JAF16], vaizdų segmentavimą [LSD14], vaizdų stiliaus perkėlimą [JAF16; ULV+16] ir kt. Konvoliucinių tinklų naudojimo pasekmė - daug naujų darbų ir tyrimų, kuriais buvo siekiama tobulinti ir plėtoti stiliaus perkėlimo metodus [JYF+20], taip pat naujų įrankių, kaip Dmitry Ulyanov aprašytas greito neuroninio piešimo įrankis (angl. *fast neural doodle*) ir „NVIDIA Canvas“ atsiradimas [Uly16]. Toks naujų įrankių atsiradimas leido naudotojams sąveikauti ir eksperimentuoti su naujais metodais.

Nauji kuriami paveikslėlių transformavimo DNT modeliai, kaip O.Texler 2020m. aprašytas interaktyvaus stiliaus perkėlimo DNT, realiu laiku grąžina tikslesnius ir žmonėms patinkančius rezultatus [TFK+20]. Tačiau nors kuriami vis tikslesni DNT vis dar bandoma atrasti naujų sričių kur galima būtų juos pritaikyti. Viena iš tokių sričių - kompiuteriniai žaidimai. Prieš metus Ş.Deniz et al. straipsnyje aprašė stiliaus perkėlimo panaudojimą kuriant naujus pilnai procedūriškai generuojamus (angl. *proceduraly generated*) 2D žaidimus [ŞKS21].

Stiliaus perkėlimas atveria galimybes kurti dizaino, turinio kūrimo ir kitus kūrybinius įrankius [FYL20]. Neuroniniais tinklais paremtas vaizdų stiliaus perkėlimas yra technika kuri leidžia sugeneruoti naujus vaizdus su tuo pačiu turiniu, tačiau skirtingu stiliumi, išmoktu iš pasirinkto stiliaus vaizdo [GEB15]. Stiliaus perkėlimas jau dabar naudojamas kompiuteriniuose žaidimuose. Tokie įrankiai kaip Unity ArtEngine, sukurtas 2020 metų gale, palengvina naujų žaidimo tekstūrų kūrimo procesą. Šis įrankis leidžia transformuoti tekstūrų, medžiagų bibliotekas pasitelkdamas rezoliucijos didinimo ir stiliaus perkėlimo dirbtinius neuroninius tinklus. Nvidia 2018 metais pristatė vieną iš savo stiliaus perkėlimo panaudojimų, kuriame žaidėjams leidžiama fotografuoti įsimintinas žaidimo akimirkas ir nuotraukai uždėti norimą stilių naudojantis stiliaus perkėlimo dirbtiniu neuroniniu tinklu.

Tačiau pagal [SC20] straipsnį stiliaus perkėlimas nėra taip plačiai taikomas žaidimuose ir jų kūrime todėl šio bakalauro darbo metu buvo atlikta neuroniniais tinklais paremto stiliaus perkėlimo žaidimuose taikymo analizė, siekiant atrasti naujas stiliaus perkėlimo taikymo sritis. Remiantis analizės rezultatais šiame darbe pateikiami eksperimentai parodantys esamas ir naujas stiliaus perkėlimo pritaikymo sritis. Rezultatų dalyje pateikiamos skirtingų naudotų stiliaus perkėlimo DNT modelių greitaveikos rezultatai, naudotų DNT modelių privalumai ir jų trūkumai.

Šio bakalauro darbo tikslas - atlikti neuroniniais tinklais paremto stiliaus perkėlimo tinkamumo naudoti žaidimuose ir jų kūrime analizę ir atlikus eksperimentus pateikti naujas stiliaus perkėlimo panaudojimo sritis žaidimuose.

Darbo Uždaviniai:

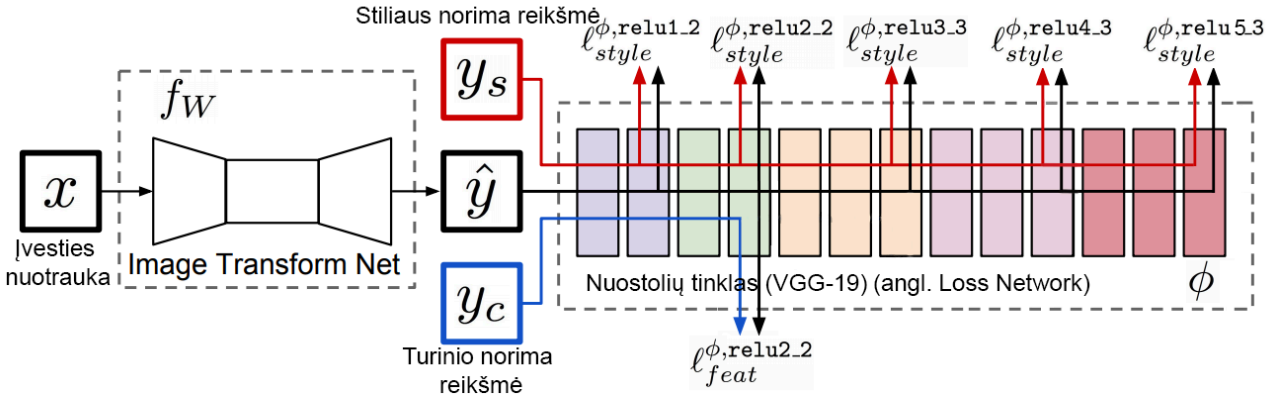
1. Apmokyti stiliaus perkėlimo DNT modelius ir atlikti kodo pakeitimus žaidime, siekiant panaudoti stiliaus perkėlimo modelius žaidime realiu laiku.
2. Pritaikyti stiliaus perkėlimo DNT modelius žaidimo vaizdų transformavimui realiu laiku, siekiant kuo geresnio optimizavimo tarp DNT modelio greitaveikos ir jo kokybės.
3. Atlikti eksperimentus parodančius žaidimų kūrėjams naujas, aktualias stiliaus perkėlimo panaudojimo sritis.
4. Išanalizuoti skirtingų stiliaus perkėlimo DNT modelių privalumus ir trūkumus taikant stiliaus perkėlimą žaidimuose.
5. Pamatuotai palyginti skirtingų modelių rezultatus panaudojimo srityse, bei pateikti rekomendacijas kilusioms problemoms spręsti.

1. Konvoliucinis neuroninis tinklas

Konvoliuciniai neuroniniai tinklai - neuroninių tinklų tipas, kuriame neuroniniai tinklai naudoja konvoliucijos operaciją. Šie specializuoti DNT modeliai dažniausiai naudojami dirbant su paveikslėliais arba garsais. Konvoliucinis neuroninis tinklas susideda iš įvesties, išvesties ir paslėptų sluoksnių. Būtent paslėpti tinklo sluoksniai ir susideda iš keleto serijų konvoliucinių sluoksnių. Pabaigoje, kaip aktyvavimo funkcija įprastai naudojama keletas „ReLU" sluoksnių, kurie padeda išvengti eksponentinio skaičiavimo reikalavimų augimo [AAA17]. Kiekvieną konvoliucinį sluoksnį galima suprasti kaip vaizdo filtrų rinkinį, kuriame kiekvienas filtras iš įvesties išgauna tam tikrą vaizdo požymį. Taigi tam tikro sluoksnio išvestį sudaro vadinamieji požymių žemėlapiai (angl. *feature maps*), kurie yra skirtingai filtruotos įvesties vaizdo versijos.

1.1. Naudotas neuroninis tinklas paveikslėlių kokybei vertinti

Šis DNT modelis naudojamas vertinti išeities paveikslėlius iš stiliaus perkėlimo DNT modelio ir skaičiuoti DNT modelio nuostolius apmokymo metu. Šiai užduočiai skirtingai negu straipsnyje, buvo naudojamas jau apmokytas VGG-19, o ne VGG-16 DNT modelis. VGG-19 geba atpažinti didelį kiekį skirtingų požymių ir yra apmokytas ant nuotraukų iš ImageNet nuotraukų rinkinio, kurį sudaro net 14 milijonų skirtingų nuotraukų [SZ15].



1 pav. Naudota sistema. Apmokomas vaizdų transformavimo tinklas skirtas transformuoti įvesties vaizdus į išvestį. Šiam tikslui naudojamas nuostolių DNT modelis (angl. *loss network*) VGG-19, apmokytas paveikslėlių klasifikavimui, ir sukuriamos suvokimo nuostolių funkcijos (angl. *perceptual loss functions*), kurios matuos suvokiamus skirtumus turinio ir stiliaus paveikslėliuose. Apmokymo metu VGG-19 tinklas nesimoko.

1.2. Naudotas konvoliucinis neuroninis tinklas paveikslėliams generuoti

Stiliaus perdavimui naudojame DNT modelį, kuris naudoja 1 lentelėje nurodytą architektūrą [JAF16]. Šioje lentelėje „C x H x W konv" reiškia konvoliucinį sluoksnį su C filtrų dydžiu H x W, po kurio iškarto seka tiesioginis normalizavimas, o ne rinkinio normalizavimas, kuris pagal [UVL16] straipsnį grąžina geresnius DNT modelio rezultatus ir netiesinis „ReLU". Kiekvienas

liekanos blokas (angl. *residual block*) turi po du 3 x 3 konvoliucinius sluoksnius su vienodais filtrų skaičiais kiekvienam sluoksniui. Šis DNT modelis naudoja liekanų blokų dizainą aprašytą S. Gross ir M. Wilber [GW16].

1 lentelė. Konvoliucinis neuroninis tinklas aprašytas [JAF16]

Sluoksnis	Aktyvacijos dydis
Įvestis	3 x 256 x 256
32 x 9 x 9 konv, žingsnis 1	32 x 256 x 256
64 x 3 x 3 konv, žingsnis 1	64 x 128 x 128
128 x 3 x 3 konv, žingsnis 1	128 x 64 x 64
Liekanų blokas(angl. *residual block*), 128 filtrai	128 x 64 x 64
Liekanų blokas(angl. *residual block*), 128 filtrai	128 x 64 x 64
Liekanų blokas(angl. *residual block*), 128 filtrai	128 x 64 x 64
Liekanų blokas(angl. *residual block*), 128 filtrai	128 x 64 x 64
Liekanų blokas(angl. *residual block*), 128 filtrai	128 x 64 x 64
64 x 3 x 3 konv, žingsnis 1/2	64 x 128 x 128
32 x 3 x 3 konv, žingsnis 1/2	32 x 256 x 256
3 x 9 x 9 konv, žingsnis 1	3 x 256 x 256

Sumažinus filtrų dydžius nuo 128,64,32 į 32,16,8 gauname 2 lentelės DNT modelį, kuris žymiai pagreitina DNT modelio veikimą, prarasdamas santykinai mažai stiliaus perdavimo tikslumo.

2 lentelė. Naudotas konvoliucinis neuroninis tinklas

Sluoksnis	Aktyvacijos dydis
Įvestis	3 x 256 x 256
8 x 9 x 9 konv, žingsnis 1	8 x 256 x 256
16 x 3 x 3 konv, žingsnis 1	16 x 128 x 128
32 x 3 x 3 konv, žingsnis 1	32 x 64 x 64
Liekanų blokas(angl. *residual block*), 32 filtrai	32 x 64 x 64
Liekanų blokas(angl. *residual block*), 32 filtrai	32 x 64 x 64
Liekanų blokas(angl. *residual block*), 32 filtrai	32 x 64 x 64
Liekanų blokas(angl. *residual block*), 32 filtrai	32 x 64 x 64
Liekanų blokas(angl. *residual block*), 32 filtrai	32 x 64 x 64
32 x 3 x 3 konv, žingsnis 1/2	16 x 128 x 128
16 x 3 x 3 konv, žingsnis 1/2	8 x 256 x 256
3 x 9 x 9 konv, žingsnis 1	3 x 256 x 256

Abiejų DNT modelių bandymų rezultatai pateikiami 2 pav. Bandymo metu stiliaus perkėlimas buvo vykdomas ant 800x600 rezoliucijos dydžio žaidimo vaizdų, naudojantis NVIDIA GeForce GTX 1060 Max-Q. DNT modelis, naudojantis 128,64,32 filtro dydžius, vidutiniškai veikė 4,3 kadrų per sekundę greičiu, o turintis 32,16,8 filtro dydžius vidutiniškai veikė 17,2 kadrų per

sekundę greičiu. Taigi buvo nustatyta, kad filtrų sumažinimas pagreitino DNT modelio veikimą 4 kartus ir neturėjo išreikšto poveikio galutiniam vaizdiniam rezultatui, kaip matoma iš žaidimo vaizdų.



(a) DNT modelio filtrų dydžiai 128,64,32 (b) DNT modelio filtrų dydžiai 32,16,8

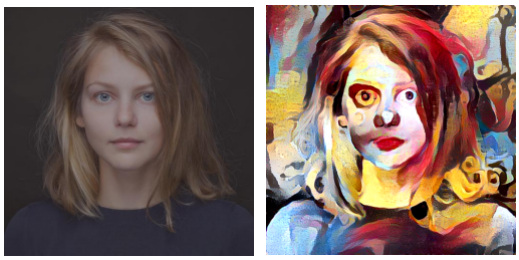
2 pav. DNT modelių stiliaus perkėlimas ir jų veikimo greitis, naudojantis skirtingais filtrų dydžiais

1.3. Tiesioginis normalizavimas

Naudotas stiliaus perkėlimo DNT modelis naudoja tiesioginį normalizavimą kuris pagal [UVL16] straipsnį stipriai optimizuoja naujų paveikslėlių generavimą palyginus su rinkinio normalizacija 3 pav. Taip yra todėl, kad naudojantis tiesioginį normalizavimą, normalizavimo procesas leidžia pašalinti egzemplioriui specifinę paveikslėlių kontrasto informaciją iš turinio paveikslėlio. Rezultate gauname geresnius paveikslėlius 4 pav. per trumpesnį laiko tarpą.



(a) Turinio pav. (b) Stilizuotas pav.



(c) Rurinio pav. (d) Stilizuotas pav.

3 pav. Mažo kontrasto stilizavimo rezultatas, naudojant rinkinio normalizaciją. Rezultatai paimti iš [UVL16] straipsnio

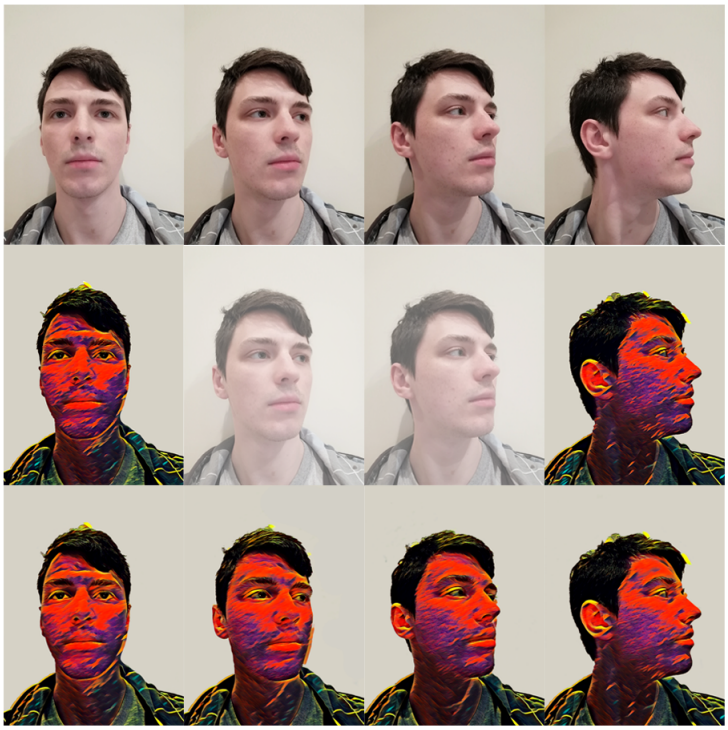


(a) Turinio pav. (b) Stilizuotas pav.

4 pav. Stilizavimo rezultatas, naudojant tiesioginį normalizavimą. Rezultatai paimti iš [UVL16] straipsnio

1.4. Antras naudotas konvoliucinis neuroninis tinklas vaizdų stiliaus perkėlimui

Kaip papildomas konvoliucinis neuroninis tinklas buvo paimtas O.Texler interaktyvus stiliaus perkėlimo modelis naudojantis kelių kadrų iškarpomis pagrįstą mokymą (angl. *interactive video stylization using few-shot patch-based training*), kurį apmokiau iš žaidimo gautais vaizdų kadrais. Savo straipsnyje O.Texler aprašė naują stiliaus perkėlimo sistemą, paremtą vaizdo įrašų kadrų stilizavimu, skirtą bet kokioms vaizdo įrašų sekoms [TFK+20]. Šis DNT sėkmingai perkelia stilių iš vieno vaizdo į kitą, išlaikydamas specifinių judančių objektų stilizavimo detales. O.Texler su savo komanda pristatė keliais kadrais pagrįstą mokymo mechanizmą, kuris pagerino vaizdo vertimo tinklo gebėjimą išskirti objektus aplinkoje neturint didelės mokymo duomenų aibės. Naudojantis jų metodu, net po kelių sekundžių mokymo DNT gali stilizuoti visą lygiagrečiai rodomą seką arba tiesioginį vaizdo įrašų srautą realiu laiku.



5 pav. Vaizdo stilizacijos nustatymas naudojant raktinius kadrus. Pirma eilė rodo įvesties vaizdo įrašo seką. Stilizavimui apmokyti naudojami du nupiešti raktiniai kadrai pateikti antroje eilėje. Šiame darbe naudoti raktiniai kadrai stilizuoti naudojantis Greito neuroninio stiliaus DNT. Užduotyje stilizuojami visi keturi įvesties sekos kadrai, tačiau naudojamas stilius iš dviejų raktinių kadrų. Paskutinė eilutė parodo šio DNT stiliaus perkėlimo metodo rezultatą

Apmokytas DNT gali greitai pateikti laike nuoseklius stilizuotus vaizdo įrašus, kurių vaizdo kokybė prilygsta vienam iš dabartinių pažangiausių vaizdo įrašų stilizavimo pagal raktinius kadrus metodų, kuris naudoja orientuotą, kadrais pagrįstą vaizdo stilizavimą [JST+19]. Tinklo stilizavimo procesas ir jo rezultatai pavaizduoti 5 pav. Pagrindinis šio tinklo privalumas yra tas, kad jis gali veikti nepriklausomai nuo kadrų, o tai labai naudinga dabartinėms profesionalioms vaizdo redagavimo priemonėms, kurios stipriai priklauso nuo atsitiktinės prieigos (angl. *random access*) ir lygiagretaus vaizdų apdorojimo.

2. Neuroniniais tinklais paremtas vaizdų stiliaus perkėlimas žaidimuose realiu laiku

Eksperimentų metu buvo taikomas stiliaus perkėlimas žaidime naudojantis dviem skirtingais stiliaus perkėlimo DNT ir analizuojamos stiliaus perkėlimo panaudojimo sritys žaidimuose. Gauti eksperimentų rezultatai rezultatų dalyje yra palyginami su įprastu tekstūrų pakeitimu ir tarpusavyje tarp skirtingų naudotų DNT.

2.1. Greitojo stiliaus perkėlimo DNT modelio apmokymas

Greitojo stiliaus perkėlimo modelis apmokomas naudojantis COCO 2014m. duomenų rinkiniu, kurį sudaro 83 tūkstančiai įvairių nuotraukų vaizdų. Šio modelio apmokymas naudojantis Kaggle platforma trunka 3h. Prieš paduodant vaizdų rinkinį į modelį nuotraukos yra sumažinamos iki 256x256 dydžio naudojant sumažinimo ir iškirpimo duomenų augmentacijas. DNT modelio nuostoliams skaičiuoti naudojama vidutinė kvadratinė paklaida (angl. *mean squared error*), o nuotraukų požymių ištraukimui naudojamas jau apmokytas VGG-19 DNT modelis. Didinant epochų (angl. *epoch*) skaičių buvo aiškiau matomi objektų kontūrai, stilizavimas tapo intensyvesnis, bet prarado pastovumą tarp kadrų. Didesnis epochų skaičius taip pat sukėlė artefaktų atsiradimą, todėl šio darbo metu buvo naudotas modelis apmokytas per 3 epochas 6 pav. Šio DNT modelio straipsnio autoriai savo darbe naudojo DNT modelį apmokytą per 2 epochas [JAF16].



(a) Turinys (b) 1 epocha (c) 5 epochos

6 pav. Stiliaus perkėlimo rezultatų skirtumas keičiant DNT modelio apmokymo epochų skaičių. Naudojant didesnį epochų skaičių matomi aiškesni objekto detalių kontūrai ir atsiranda daugiau stilizavimo detalių (5 epochos vaizdas apačioje, dešinėje pusėje). Po 5 epochų matomi smulkūs kelių pikselių artefaktai atsirandantys aplink viso vaizdo kraštus (mėlynai pažymėta ir priartinta sritis)

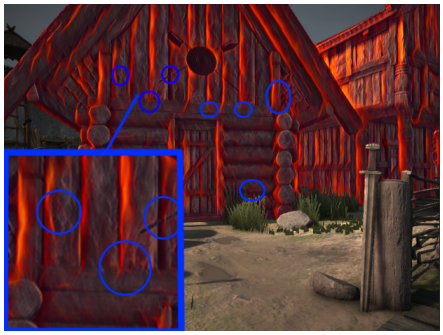
Taip pat buvo atliktas eksperimentas paduodant modeliui 2 tūkstančių stilizuojamo žaidimo vaizdų mokymosi metu, tačiau tai neturėjo matomo teigiamo poveikio šio DNT modelio stilizavimo rezultatui, kadangi 2 tūkstančiai žaidimo vaizdų yra labai maža dalis lyginant su COCO duomenų rinkiniu.

2.2. Interaktyvaus stiliaus perkėlimo DNT modelio apmokymas

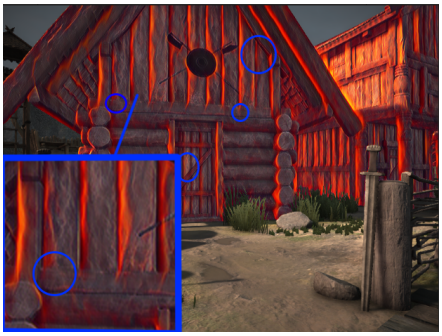
Interaktyvaus stiliaus perkėlimo DNT modelis nereikalauja didelių duomenų rinkinių priešingai negu greito stiliaus perkėlimo DNT. Šiam tinklui apmokyti buvo naudota 200 žaidimo vaizdo nuotraukų su stiliaus perkėlimu, nes Kaggle platforma dėl mažos skiriamos atminties nepalaikė didesnio apmokymo duomenų skaičiaus, o naudotos žaidimo nuotraukos prieš paduodant jas į DNT buvo sumažinamos iki 540xY, kai Y<540, išlaikant santyki tarp ilgio ir pločio. Šio modelio apmokymo metu svarbu, kad vaizdai paduodami į modelį apmokymo metu būtų panašaus dydžio, kaip ir būsimų stilizuojamų kadrų, nes paduodamų vaizdų sumažinimas pakeičia 32x32 dydžio iškarpų turimą informaciją. Šių nuotraukų stilizavimui buvo naudotas [JAF16] aprašytas tinklas, be sumažintų filtrų dydžių. Vietoj to, kad modeliui būtų paduodamas visas apmokymo vaizdas, šis modelis priimtus vaizdus išskaido į mažas kvadratines 32x32 dydžio iškarpas ir mokosi nustatyti šių iškarpų stilizavimą. Dėl šios priežasties DNT autoriai nerekomenduoja naudoti didelių duomenų rinkinių, nes modelis gali turėti per daug skirtingų būdų stilizuoti panašias arba vienodas 32x32 iškarpas iš vaizdų, o tai pagal autorius sukelia daugiau laikinio vaizdų nepastovumo - suintensyvina mirgėjimą tarp kadrų. Vis dėlto pasibandžius 20 ir 200 duomenų rinkinius yra pastebėta, kad didesnis duomenų rinkinys grąžina rezultatus su šiek tiek geresniu laikiniu vaizdų pastovumu (angl. *temporal consistency*) 7 pav.



(a) Pirmas kadras. Modelis apmokytas naudojantis 20 žaidimo vaizdų



(b) Antras kadras. Modelis apmokytas naudojantis 20 žaidimo vaizdų



(c) Pirmas kadras. Modelis apmokytas naudojantis 200 žaidimo vaizdų



(d) Antras kadras. Modelis apmokytas naudojantis 200 žaidimo vaizdų

7 pav. Paimti 2 žaidimo kadrai minimaliai pajudinus kameros objektą žaidime. Paveikslėlyje matomi interaktyvaus stiliaus perkėlimo DNT modelio skirtumai naudojant skirtingus duomenų rinkinio dydžius. DNT modelio apmokyto naudojantis 200 žaidimo vaizdų stilizavimo rezultatas mažiau pakito pakeitus kameros poziciją, negu apmokyto naudojantis 20 vaizdų. Vaizduose apskritimais pažymėtos vietos su staigiai pasikeitusiu stilizavimu

Spalvų skirtumai tarp apmokytų šio stiliaus DNT perkėlimo modelių atsiranda dėl skirtingų mokymosi metu paimtų nuotraukų iškarpų.

2.3. Žaidimo paruošimas neuroniniui tinklui

Neuroninio tinklo paleidimui žaidime, naudojama „ML Agents" biblioteka. Svarbu paminėti, kad toliau esančios žaidimo paruošimo instrukcijos skirtos žaidimui nenaudojančiam naujųjų „Universal Render pipeline" arba „High Definition Render pipeline".

1. Per „Unity package manager" pridedama „ML Agents" biblioteka.
2. Naudojantis stiliaus perkėlimo DNT modelio apmokymo kodu apmokomas norimo stiliaus perkėlimo DNT modelis. Šis DNT modelis po apmokymo eksportuojamas į ONNX formatą. ONNX tai atviras mašininio mokymosi modelių formatas, leidžiantis naudoti įvairius mašininio mokymo modelius tarp skirtingų sistemų ir įrankių.
3. Apmokytas DNT modelis importuojamas į Unity.
4. Sukuriama apskaičiuojama šešėliuoklė (angl. *compute shader*), ir aprašomos ProcessInput, ProcessOutput funkcijos, kurios pakeičia Unity RGB kanalo diapazoną [0, 1] į stiliaus perkėlimo DNT modelio įvesties reikšmes, kurios greitojo stiliaus perkėlimo DNT yra [0, 1] diapazone.
5. Tada sukuriamas C# kodas, kuris atliks atvaizdų apdorojimą - perduos atvaizdus į ir iš DNT modelio. Kadangi DNT modelis bus naudojamas žaidimo metu, sukuriamas kintamasis targetHeight kuris leidžia reguliuoti vaizdo, paduodamo į DNT modelį dydį. Paduodant mažesnio dydžio vaizdus atsakymas iš DNT modelio bus grąžintas daug greičiau.
6. C# kodas prikabinamas prie kameros komponento pasirinkus norimą DNT modelį ir uždedama sukurta apskaičiuotoji šešėliuoklė.

Naudojamas stiliaus perkėlimo DNT modelis neatsižvelgia į nuoseklumą tarp keleto iš paskos einančių atvaizdų ir tai sukelia mirgėjimą. Šis mirgėjimas gali būti nepageidaujamas reiškinys ir jį galima sumažinti naudojant interaktyvųjį stiliaus perkėlimo DNT modelį arba nenaudojant papildomų vaizdo apdorojimo sluoksnių (angl. *post-processing*), tokių kaip Glodinimas (angl. *anti-aliasing*). Papildomai, jeigu norima stiliaus perkėlimą taikyti tik specifiniams objektams/sluoksniams, o ne visam vaizdui:

1. Sukuriamos gylio kameros: viena - turiniui, kita - stiliui. Šios kameros prikabinamos prie pagrindinės kameros, kaip pagrindinės kameros vaikai.
2. Stiliaus gylio kamerai nustatoma sluoksnio kaukė (angl. *culling mask*) į naujai sukurto stiliaus sluoksnį.
3. Papildomas sukurtas šešėliuoklės kodas su Merge funkcija, kuri apjungia originalų vaizdą su vaizdu, kuriam buvo taikomas stiliaus perkėlimas.
4. Atnaujinamas C# kodas, kuris naudodamasis šešėliuoklės Merge metodu apjungia originalų ir stilizuotą kadrą.

Norint naudoti interaktyvų stiliaus perkėlimo modelį reikalingi šie pakeitimai:

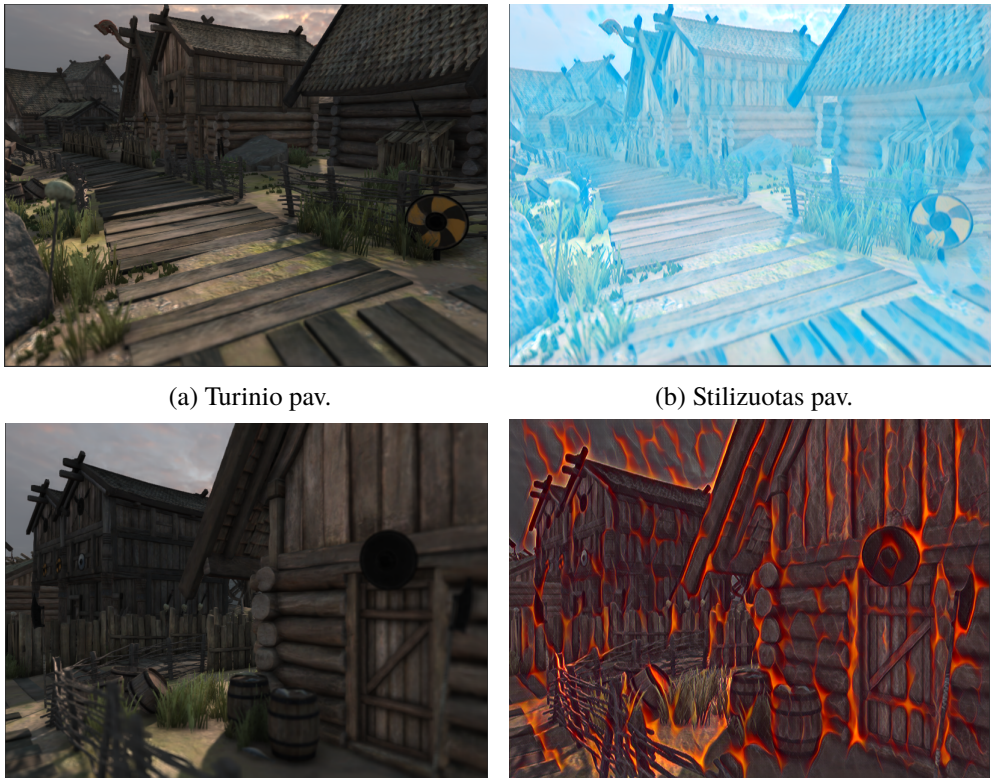
1. Pakoreguojamas šešėliuoklės kodas, kad RGB kanalo diapazonas būtų ne [0,1], o [-1,1] skirtingai negu [0,255], kai naudojamas greito neuroninio stiliaus perkėlimo dirbtinis neuroninis

tinklas.

Visas aprašytas kodas pateikiamas šio darbo prieduose. Visos šiame darbe pateikiamos stiliaus perkėlimo nuotraukos žaidime gautos iš mano pasirinktų žaidimų, juos paruošus šiuo būdu.

2.4. Pirmasis eksperimentas

Pirmasis eksperimentas buvo skirtas patikrinti ar stiliaus perkėlimas visam vaizdui gali būti naudojamas žaidimų aplinkos prototipų kūrimui, tiksliau - aplinkos temos pakeitimui. Šio eksperimento metu pasinaudodamas apmokytais stiliaus perkėlimo DNT modeliais visam ekrano vaizdui taikiau stiliaus perkėlimą. Toks stiliaus perkėlimas realiu laiku leidžia žaidimų kūrėjams testavimo metu vaikščioti po pasirinktą aplinką ir realiu laiku gauti rezultatą, kaip ši aplinka atrodo pritaikius pasirinktą stilių. Viso vaizdo stiliaus perkėlimą taip pat būtų galima panaudoti žaidimų scenų iškarpų arba vaizdo įrašų stilizavimui. Šiuo metu jau yra keletas realių pavyzdžių, kur šiai užduočiau taikomi stiliaus perkėlimo DNT.



(a) Turinio pav. (b) Stilizuotas pav.

(c) Turinio pav. (d) Stilizuotas pav.

8 pav. Miestelio temos pakeitimas, naudojant stiliaus perkėlimą

Tačiau ne visada norime, kad stiliaus perkėlimas perkeltų stilių visam atvaizdui. Prototipų kūrimo metu dažnai siekiama stilizuoti tik, pavyzdžiui, namus, arba visą vaizdą, išskyrus dangų, ir realiu laiku patikrinti, kaip viskas susideda ir atrodo. Tam galime naudoti stiliaus perkėlimą specifiniams objektams, kurį analizuoju antrame eksperimente.

2.5. Antrasis eksperimentas

Prototipuojant dažnai kyla poreikis greitai išbandyti skirtingus stilius skirtingiems objektams. Stiliaus perkėlimas specifiniam objektui sprendžia šią problemą. Žaidimų kūrėjas, turėdamas ne vieną jau apmokintą DNT modelį gali juo pasinaudoti ir pritaikyti stiliaus perkėlimą pasirinktam objektui ar objektams, bei testuodamas pamatyti, kaip jie atrodys 3D erdvėje ar pasirinktoje aplinkoje. Taigi, antrojo eksperimento metu pasinaudodamas tais pačiais apmokytais stiliaus perkėlimo DNT modeliais analizavau stiliaus perkėlimą pasirinktiems objektams ir palyginau rezultatus tarp objekto, kuriam buvo taikytas stiliaus perkėlimas ir objekto, kuriam taikytas tradicinis objekto tekstūros pakeitimas 9 pav.



(a) Turinio pav. su pakeistomis tekstūromis



(b) Stilizuotas pav.



(c) Turinio pav. su pakeistomis tekstūromis

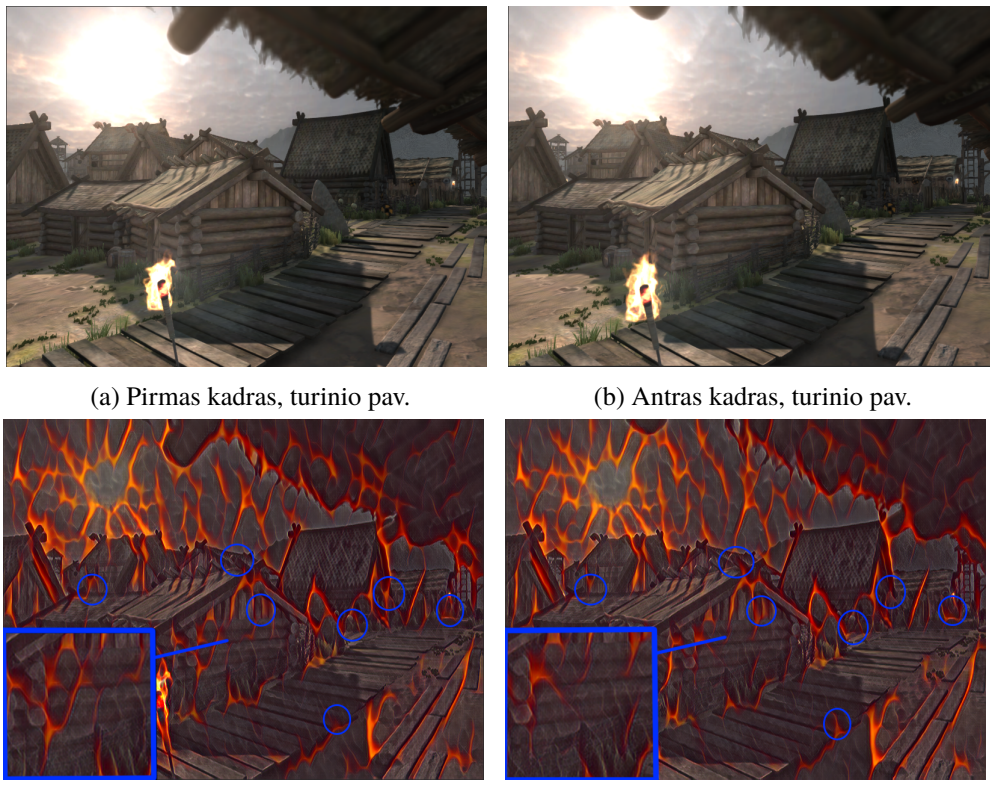


(d) Stilizuotas pav.

9 pav. Palyginimas tarp stiliaus perkėlimo ir tekstūrų pakeitimo

2.6. Trečiasis eksperimentas

Atlikus pirmą ir antrą eksperimentus, buvo pastebėta stiliaus perkėlimo DNT modelių problema - laikinas nepastovumas tarp žaidimo kadrų, matomas 10 pav. Taikant stiliaus perkėlimą žaidime realiu laiku stiliaus perkėlimo rezultatai skiriasi tarp vienas po kito einančių žaidimo vaizdų kadrų ir gyvai žiūrint į rezultatus matomas mirgėjimas ir detalių stilizavimo nepastovumas.



(a) Pirmas kadras, turinio pav.

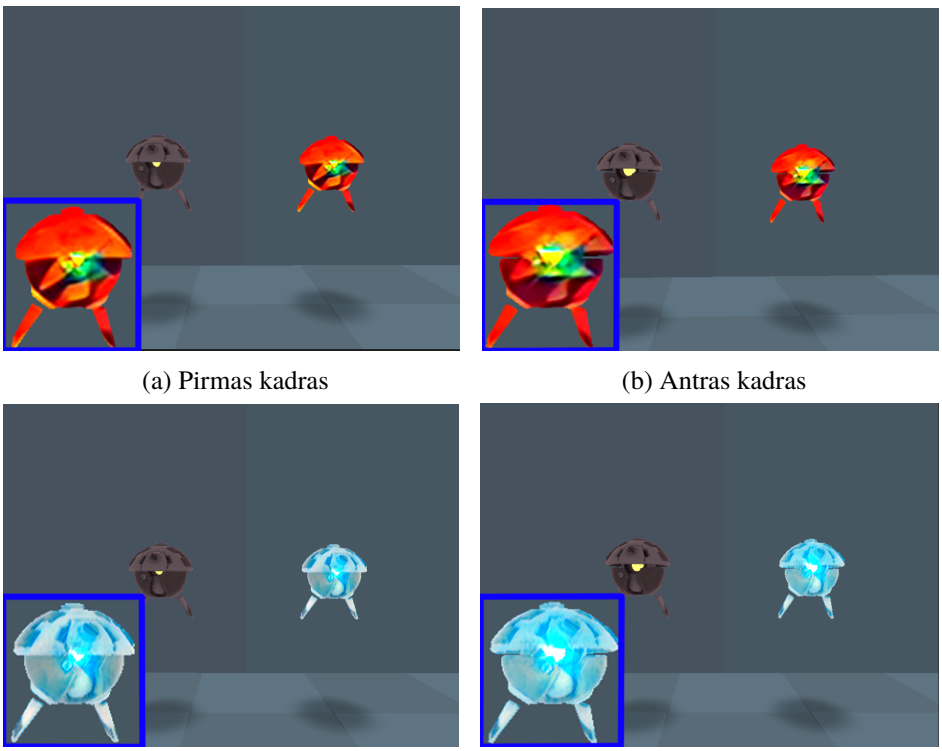
(b) Antras kadras, turinio pav.

(c) Pirmas kadras, stilizuotas pav.

(d) Antras kadras, stilizuotas pav.

10 pav. Matomas mirgėjimas - iššokančių stiliaus artefaktų atsiradimas stiliaus perkėlimo metu. Šie artefaktai sukelia stiliaus perkėlimo detalių nepastovumą tarp vienas po kito einančių žaidimo vaizdų. Priartintoje vaizdo srityje aiškiai matomos staigiai pasikeitusios stilizavimo detalės, šis staigus pasikeitimas taip pat matomos apskritimais pažymėtose srityse

Kadangi žaidimo metu aplinka, žaidėjo kameros pozicija ir rotacija pastoviai keičiasi, stiliaus perkėlimo DNT modeliui tampa sudėtinga vienodai stilizuoti vaizdą. Taikant stiliaus perkėlimą žaidimo metu net mažiausias pokytis įtakoją galutinio vaizdo rezultatą, todėl svarbu atsižvelgti į kameros filtrus, kitus vaizdo apdorojimo metodus ir pačios kameros poziciją. Kita vertus į šį nepastovumą galime pažvelgti ne tik, kaip į problemą, bet ir kaip į galimą tekstūrų animavimo problemos sprendimą, todėl trečiajam eksperimentui pasirinkau pritaikyti stiliaus perkėlimą įvairiems skirtingiems objektams žaidimuose, siekiant suteikti jiems animuotas tekstūras 11 pav.



(a) Pirmas kadras

(b) Antras kadras

(c) Pirmas kadras

(d) Antras kadras

11 pav. Stiliaus perkėlimo detalių nepastovumas, mirgėjimas tarp kadrų sudaro animuotų tekstūrų įspūdį. Pateiktoje nuotraukoje matoma, kaip skirtingose veikėjo animacijos stadijose skiriasi stilizavimo rezultatai

3. Eksperimentų rezultatai

3.1. Pirmojo eksperimento rezultatai

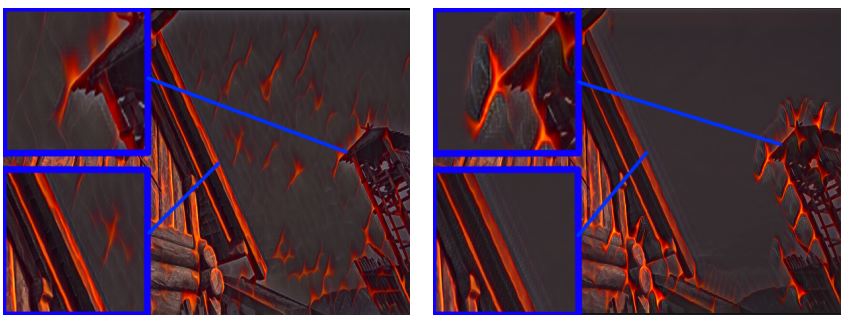
Iš 8 pav. yra matoma, kad naudojantis stiliaus perkėlimą galima atlikti aplinkos temos pakeitimą. Turint keletą išmokytų DNT modelių, temos pakeitimą testavimo metu galima keisti realiu laiku. Užtenka pakeisti vieną parametrą, nurodantį naudojamą DNT modelį, o tai galima atlikti Unity žaidimo redaktoriuje. Svarbu paminėti ir anksčiau aptartą bei svarbią problemą taikant stiliaus perkėlimą - mirgėjimą tarp kadrų, kuris nukreipia žaidėjo dėmesį, matomą 10 pav.

Laikinio vaizdų pastovumo išlaikymas yra viena pagrindinių vaizdo įrašų stilizacijos metodų užduotis. Kai atskiri kadrai stilizuojami nepriklausomai vienas nuo kito, stilizuotoje animacijoje paprastai atsiranda intensyvus mirgėjimas. Nors šis efektas yra būdingas ir tradiciniams rankomis spalvotiems vaizdo įrašams, jis gali tapti nepatogus stebėtojui, kai žiūrima ilgesnį laiko tarpą [FLJ+14].

Šiam nepastovumui sumažinti greitojo stiliaus perkėlimo modelyje įgyvendinama papildoma triukšmo nuostolių funkciją (angl. *noise loss*) pagal A.Swarnakar ir M.Bao straipsnį [Ank19]. Straipsnyje teigiama, kad vaizdų laikinis nestabilumas ir iššokančios stilizavimo detalės atsiranda dėl radikalaus stilizavimo pokyčio, esant labai nedideliam įvesties pokyčiui. Tokiai problemai spręsti yra sukuriama papildoma nuostolių funkcija, kuri apmokymo metu rankiniu būdu prideda triukšmo prie stilizuojamos nuotraukos ir apskaičiuoja nuostolius tarp stilizuoto įvesties paveikslėlio ir stilizuoto įvesties paveikslėlio su pridėtu triukšmu. Triukšmo nuostolių funkcija $\mathcal{L}_N$ apskaičiuoja vidutinę kvadratinę paklaidą tarp triukšmingo vaizdo įvesties ir originalaus vaizdo įvesties stilizavimo rezultatų. Matematinė formulė:

$$\mathcal{L}_N = \frac{1}{N}\sum_{j,k}(g(n(x))_{j,k} - g(x)_{j,k})^2$$

kur N yra bendras vaizdo pikselių skaičius j ir k - konkrečios pikselio x ir y koordinatės, x yra originalus įvesties vaizdas, n(x) - įvesties vaizdas su triukšmu, o g(z) yra metodas stilizuojantis vaizdą. Tokiu būdu yra atkartojami iššokančių stilizavimo detalių artefaktai, kuriuos modelis mokosi sumažinti.



(a) Įprastas DNT tinklo rezultatas (b) DNT tinklo su papildoma triukšmo nuostolių funkcija rezultatas

12 pav. Matomi skirtumai tarp įprastai apmokyto DNT tinklo ir DNT tinklo, apmokyto su pridėtine triukšmo nuostolių funkcija

12 pav. pateikiami skirtumai tarp įprasto apmokyto greitojo stiliaus perkėlimo DNT ir DNT, apmokyto su papildoma triukšmo nuostolių funkcija. Lyginant tinklų rezultatų nuotraukas galime matyti porą skirtumų. Pirma, tinklas apmokytas su triukšmo funkcija nesugeba stilizuoti dangaus. Dangus yra homogeninis vaizdo regionas, todėl stiliaus perkėlimo tinklas, nežinodamas, kaip jį stilizuoti, dažnai sukelia mirgėjimo artefaktus. Naudojant triukšmo nuostolių funkciją, modelis išmoksta vengti tokių objektų stilizavimo, tokiu būdu panaikindamas mirgėjimą. Pagrindinė šios nuostolių funkcijos problema - atsiradusios aureolės aplink objektų kontūrus. Šios aureolės atsiranda todėl, kad triukšmo nuostoliai lygina dvi stilizuotas nuotraukas, kuriose net ir netaikant triukšmo nuostolių funkcijos objektų kontūruose yra vaizduojami artefaktai, matomi įprasto DNT tinklo rezultato nuotraukoje. Modelis, lygindamas dvi stilizuotas nuotraukas ir bandydamas jas suvienodinti, išmoksta sustiprinti šį stilizavimo artefaktą. Tačiau šiai problemai spręsti buvo rastas lengvas sprendimo būdas - padidinant variacijos nuostolių funkcijos svorį, kad jis mokymosi metu būtų šiek tiek didesnis negu nuostolių funkcijos. Šios nuostolių funkcijos svorio padidinimas modelio mokymosi metu optimizuoja modelį vengti triukšmingų rezultatų, kas padeda iš rezultatų pašalinti aureolės artefaktus.



(a) DNT tinklo su papildoma triukšmo nuostolių funkcija rezultatas



(b) DNT tinklo, apmokyto su triukšmo nuostolių funkcija ir padidintu variacijos nuostolių funkcijos svoriu, rezultatas

13 pav. Matomi skirtumai - sumažinti aureolių artefaktai aplink objektus tarp DNT tinklo, apmokyto su papildoma triukšmo nuostolių funkcija, ir DNT tinklo, apmokyto su papildoma triukšmo nuostolių funkcija ir padidintu variacijos nuostolių funkcijos svoriu

Iš 13 pav. matome, kad variacijos nuostolių funkcijos svorių padidinimas padėjo sumažinti aureolių artefaktus, atsiradusius aplink stilizuojamoje nuotraukoje esančius objektus, tačiau šios nuostolių funkcijos taikymas padidino stilizuojamo vaizdo spalvų ryškumą ir sumažino detalumą.

14 pav. palyginus įprastu būdu apmokyto greitojo stiliaus perkėlimo DNT ir DNT, apmokytą su papildoma triukšmo funkcija, matomi geresni laikinio vaizdų pastovumo rezultatai.



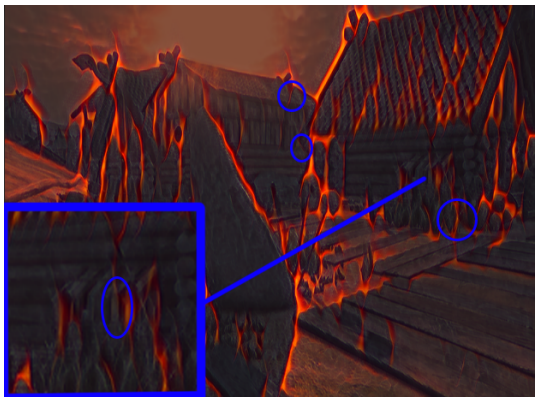
(a) Įprastai apmokytas DNT. Pirmas kadras



(b) Įprastai apmokytas DNT. Antras kadras



(c) DNT, apmokytas su papildoma triukšmo nuostolių funkcija. Pirmas kadras



(d) DNT, apmokytas su papildoma triukšmo nuostolių funkcija. Antras kadras

14 pav. Matomas laikinio vaizdų pastovumo pagerėjimas lyginant pirmoje eilėje pavaizduoto įprastai apmokyto DNT tinklo ir antroje eilėje DNT tinklo, naudojančio triukšmo nuostolių funkciją, rezultatai. DNT tinklas apmokytas su triukšmo nuostolių funkciją turi mažiau tarp kadrų iššokančių stilizavimo detalių (mėlynai pažymėtos iššokančios detalės)

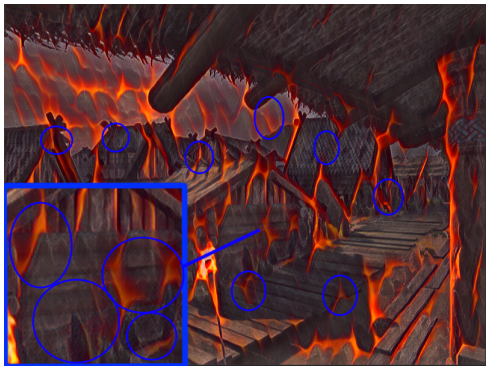
Nepastovumas arba mirgėjimas, taip pat gali būti sumažinamas naudojant antrąjį šiame darbe analizuotą DNT modelį - interaktyvų stiliaus perkėlimo DNT modelį, kuris apmokomas naudojantis žaidimo kadrus 15 pav. Kadangi šis stiliaus perkėlimo modelis yra apmokomas naudojantis keleto kadrų informacija, tai jis išmoksta atlikti taisyklingą stiliaus vertimą ir atlikdamas stiliaus perkėlimą atsižvelgia į tai, kaip tam tikros vaizdo detalės kinta per laiko tarpą. Nors šio modelio rezultatai turi geresnį laikinį vaizdų pastovumą, mirgėjimas vis tiek išlieka ir yra matomas.



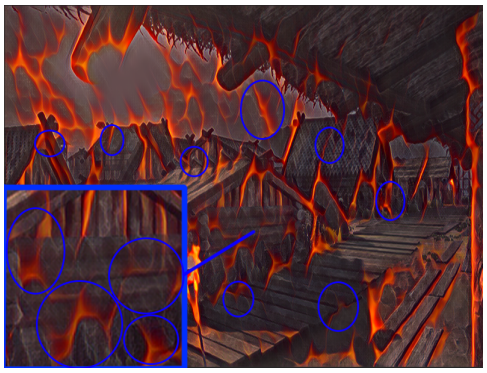
(a) Turinio pav.



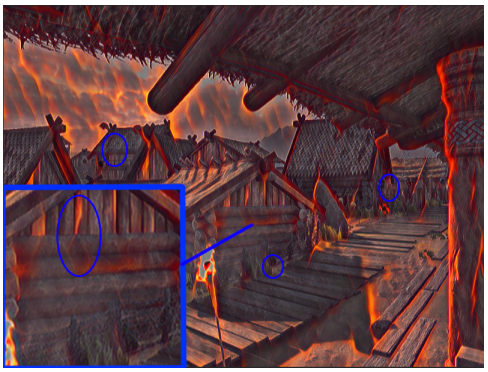
(b) Turinio pav.



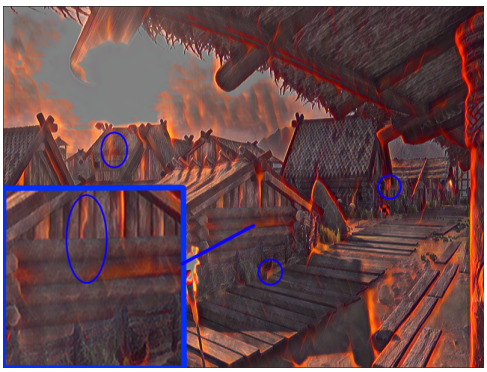
(c) Greito neuroninio stiliaus stilizuotas pav.



(d) Greito neuroninio stiliaus stilizuotas pav.



(e) Interaktyvaus stiliaus perkėlimo stilizuotas pav.



(f) Interaktyvaus stiliaus perkėlimo stilizuotas pav.

15 pav. Laikinio vaizdų pastovumo skirtumas tarp abiejų modelių. Paveikslėliuose lyginami rezultatai tarp DNT modelių, kurie buvo apmokyti nenaudojant triukšmo nuostolių funkcijos. Apskritimų srityse matomi stilizavimo pasikeitimai tarp kadrų

Laikinis vaizdų nepastovumas šiame modelyje atsiranda, todėl kad tekstūros ne visada turi pakankamai skiriamosios informacijos. Pvz.: homogeniški regionai, tokie kaip dangus, mirga todėl, kad vaizdas gali būti stilizuotas dviprasmiškai, kas koreliuoja su tinklo galimybe atkurti norimą vaizdą. Norint nuslopinti šį reiškinį galima naudoti aprašytą triukšmo nuostolių funkcija arba paruošti aplinką taip, kad joje būtų tik aiškiai ir gerai išsiskiriantys regionai, tačiau praktikoje tai sunkiai įgyvendinama. Laikinį vaizdų nepastovumą šiame modelyje taip pat gali sukelti ir triukšmas, esantis apmokymo nuotraukose, arba dublikatai, esantys apmokymo duomenyse, todėl papildomai rekomendacijų srityje yra pateikiamos rekomendacijas, kaip dar galima sumažinti laikinį vaizdų nepastovumą. Pirmojo eksperimento metu taip pat buvo nustatytas didelis neigiamas poveikis žaidimo greitaveikai, naudojant stiliaus perkėlimo DNT modelį. Šis neigiamas poveikis

žaidimo greitaveikai gali būti sumažinamas naudojant vaizdo erdvės dvikrypčio scenos atkūrimo (angl. temporal upsampling) algoritmą, aprašomą rekomendacijų dalyje. Naudotų DNT modelių greitaveikos rezultatai aprašyti 4 lentelėje.

3.2. Antrojo eksperimento rezultatai

Iš antrojo eksperimento nuotraukų, pateiktų 9 pav., matomi vaizdai su sėkmingai perkeltu stiliumi tik pasirinktiems objektams, o ne visam vaizdui. Palyginus gautus rezultatus matome, kad stilizuotame objekte tekstūros stilius perteikiamas visam objektui skiriasi tarp jo detalių - skirtingai stilizuojama namo horizontalūs rastai ir vertikalios lentos. O paprastoje nuotraukoje, naudojančioje tekstūrą, viso objekto detalės atrodo vienodai. Tačiau nors stiliaus perkėlimas įvykdomas sėkmingai, galima pastebėti ir keletą problemų. Viena iš jų - tai atspindį turintys objektai, tokie kaip skydas ant namo nebeatspindi saulės šviesos. Taip yra todėl, kad stiliaus perkėlimas įvyksta paskutiniame vaizdo apdorojimo etape, kai gauname atvaizdo išvestį ir todėl DNT modelis neatsižvelgia į objekto atspindį. Taip pat buvo pastebėta, kad objektai, esantys toliau nuo kameros, naudojant stiliaus perkėlimą turi mažiau detalių, palyginus su įprastu tekstūrų uždėjimu 9b pav., fone yra matoma kaip stilizuojama tik maža dalis pastato detalių. Taip nutinka todėl kad greitojo neuroninio stiliaus DNT modelis gauna gana mažai informacijos apie tolimą objektą ir todėl neturi užtektinai informacijos pastato detalėms stilizuoti. Abi šios problemos aktualios ir taikant stiliaus perkėlimą visam ekranui, tačiau jos čia yra mažiau išreikštos ir beveik nepastebimos, nes stilizuojant visą vaizdą atskiroms detalėms nėra skiriamas ypatingas dėmesys.

Naudojant interaktyvųjį stiliaus perkėlimo DNT modelį detalių problema tampa mažiau pastebima, nes interaktyvusis stiliaus perkėlimo DNT modelis apmokymui naudoja kadrų iškarpas (angl. *patch-based*). Naudojantis šiuo apmokymo metodu tinklas mokosi iš 32x32 dydžio kadrų dalių, o ne viso vaizdo, kaip greitojo stiliaus perkėlimo DNT modelis. Dėl šios priežasties rezultatai judant toliau nuo pasirinkto stilizuojamo objekto yra geresni ir tikslesni, kaip matome pateiktose nuotraukose 16 pav.



(a) Greito neuroninio stiliaus perkėlimo stilizuotas pav.



(b) Interaktyvaus stiliaus perkėlimo pav.



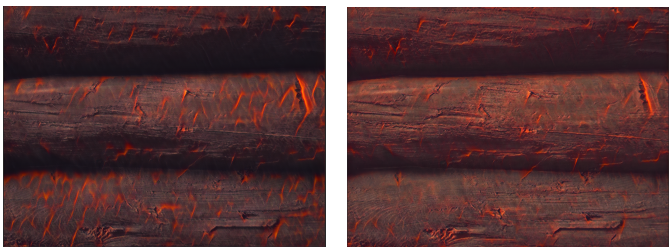
(c) Greito neuroninio stiliaus perkėlimo stilizuotas pav.



(d) Interaktyvaus stiliaus perkėlimo stilizuotas pav.

16 pav. Modelių stilizavimo skirtumai ir tikslumas

Vaizduose, pateiktuose 16 pav., matoma, kaip interaktyvaus stiliaus perkėlimo modelis išsaugo daugiau detalių ir todėl didesnė dalis kontūrų išlieka tokie patys, kaip ir žvelgiant iš arti. Greito neuroninio stiliaus DNT modelio rezultatai judant toliau nuo objekto sunkiau randa objekto kontūrus ir stilizavimo rezultate matosi detalių susiliejimas (pvz.: durų viršutinė stakta). Taip pat žvelgiant iš arti greito neuroninio stiliaus DNT modelis stengdamasis stilizuoti mažas detales pradeda iškraipyti objekto dalių kontūrus. Su šia problema interaktyvaus stiliaus perkėlimo modelis susiduria tik iš labai artimo atstumo tarp kameros ir objekto, tačiau tai galima sumažinti aukojant greitaveiką ir naudojant didesnius DNT modelio filtrus 17 pav. Interaktyvaus stiliaus perkėlimo DNT modelio filtrų padidinimo poveikis greitaveikai aprašomas 4 lentelėje.



(a) Interaktyvaus stiliaus perkėlimo modelio rezultatas

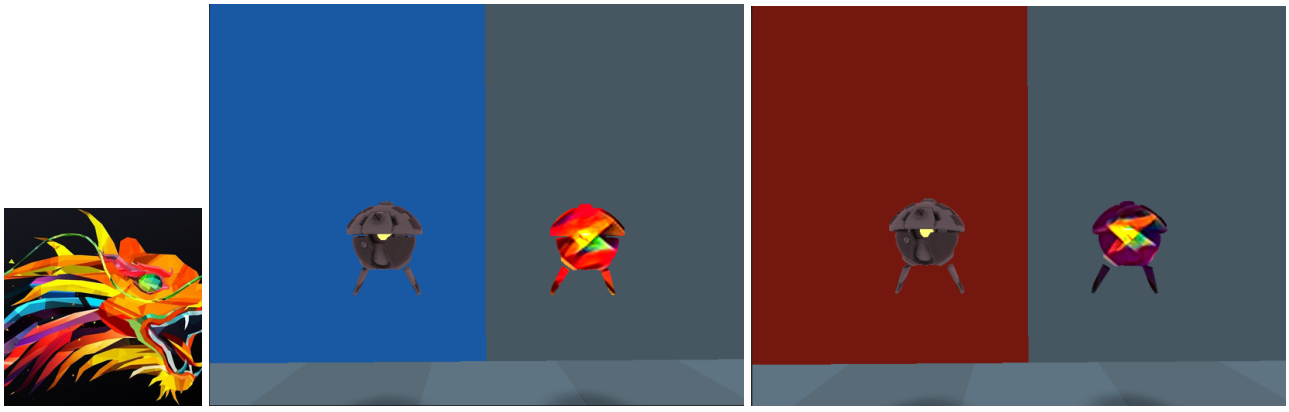
(b) Interaktyvaus stiliaus perkėlimo modelio su didesniais filtrais rezultatas

17 pav. Skirtumas tarp interaktyvaus stiliaus perkėlimo modelių su skirtingais filtrų dydžiais. DNT modelis su mažesniais filtrais stengiasi stilizuoti abstrakčiai, o DNT modelis su didesniais filtrais konkrečiau (pvz.: paveikslėlių kairė, apačia)

Rezultate interaktyvaus stiliaus perkėlimo modelis grąžina tikslesnius rezultatus dėl savo kadrų iškarpų apmokymo metodo. Remiantis šiais rezultatais galima teigti, kad interaktyvaus stiliaus perkėlimo modelis yra pranašesnis už greito neuroninio stiliaus DNT modelį objektų ir prototipų kūrime.

3.3. Trečiojo eksperimento rezultatai

Atlikę trečiąjį eksperimentą, kurio rezultatas matomas 11 pav., galima teigti, kad stiliaus perkėlimą galima naudoti kuriant unikalias animuotas tekstūras, kur animacijos atsiranda dėl mirgėjimo t.y. stiliaus perkėlimo nepastovumo tarp kadrų. Naudojant stiliaus perkėlimą svarbu atsižvelgti į tai, kaip objekto stilius keisis priklausomai nuo jo aplinkos, fono ir aplink esančių objektų spalvų. Tai ypač aktualu stilizuojant judančius objektus ir veikėjus kadangi jie nuolat juda ir todėl jų stilizavimas keičiasi po kiekvieno mažo pokyčio.



(a) Naudotas stiliaus pav.

(b) Veikėjo stilius kai fonas melynas

(c) Veikėjo stilius kai fonas raudonas

18 pav. Tas pats stiliaus perkėlimo DNT skirtingai stilizuoja veikėjus priklausomai nuo viso ekrano vaizdo

18 pav. yra matoma, kaip naudojant tą patį DNT modelį yra gaunami skirtingi stiliaus perkėlimo rezultatai priklausomai nuo to, koks fonas naudojamas stiliaus perkėlimo metu. Fono keitimas

lemia tekstūrų keitimąsi žaidimo metu. 11 pav. pavaizduota, kaip veikėjo neveiklumo animacija (angl. *idle animation*) keičia jo stilių. Stilius keičiasi, kadangi kiekvienoje veikėjo animacijos dalyje, keičiasi stilizuojamas vaizdas ir besikeičiančios smulkios veikėjo detalės įtakoja viso veikėjo stilizavimą. Kadangi šiame eksperimente pasinaudojom DNT modelio kadrų nepastovumu, o kadrų nepastovumas yra daug mažesnis interaktyvaus stiliaus perkėlimo DNT modelyje (16 pav.), tai šio modelio naudojimas gražintų prastesnius rezultatus ir todėl yra nerekomenduojamas.

3.4. Rezultatų palyginimas

3 lentelėje pateikiamos skirtingų modelių panaudojimo srities rekomendacijos. Naudojantis abiem modeliais galima taikyti stiliaus perkėlimą tiek visam vaizdui, tiek objektui. Šioje užduotyje geresnius rezultatus gražino interaktyvaus stiliaus perkėlimo modelis (16 pav.), nes interaktyvaus stiliaus perkėlimo DNT tiksliau stilizavo objekto detales, išlaikė objektų kontūrus ir turėjo geresnį laikiną pastovumą tarp kadrų, kai nebuvo naudojama triukšmo nuostolių funkcija. Kadangi šis DNT modelis turėjo geresnį laikiną pastovumą nerekomenduojama jo naudoti kuriant unikalias animuotas tekstūras. Iškirptinių scenų stilizavimui rekomenduojama naudoti interaktyvaus stiliaus perkėlimo DNT, kuris pagal gautus rezultatus grąžina kokybiškesnius vaizdus, tačiau veikia lėčiau. Tokiai pritaikymo sričiai greitaveika nėra svarbi, todėl rekomenduojamas būtent šis modelis. Svarbus šio modelio apribojimas - didelis neigiamas poveikis žaidimo greitaveikai, todėl interaktyvaus stiliaus perkėlimo DNT modelio taikymas nerekomenduojamas baigtame žaidime realiu laiku. Šiuo metu vartotojams siūlomos vaizdo plokštės dar nėra pakankamai pajėgios naudoti šį modelį realiu laiku, net ir su sumažintais filtrų dydžiais. Baigtame žaidime rekomenduojama naudoti greitojo stiliaus perkėlimo modelį su sumažintais filtrais ir apmokytą su papildoma triukšmo nuostolių funkcija.

3 lentelė. Konvoliucinių neuroninių tinklų panaudojimo rekomendacijos

	Greito neuroninio stiliaus DNT	Interaktyvus stiliaus perkėlimo DNT
Viso vaizdo stiliaus perkėlimas	Galima	Rekomenduojama
Pasirinkto objekto stiliaus perkėlimas	Galima	Rekomenduojama
Unikalių animuotų tekstūrų kūrimas	Rekomenduojama	Galima
Iškirptinių scenų (angl. *cutscene*) stilizavimas	Galima	Rekomenduojama
Stiliaus perkėlimo taikymas baigtame žaidime	Rekomenduojama	Galima

4 lentelėje pateikiami skirtingų modelių greitaveikos rezultatai. Greitaveika - kadrai per sekundę, apskaičiuojami paėmus kadrų per sekundę vidurkį per 1 minutės laiko tarpą. Per šį laiko tarpą žaidimo kamera juda tarp keleto nurodytų koordinačių, tai užtikrina vienodą lyginamąjį standartą. Greitaveikos skaičiavimui naudotas GeForce RTX 3070. Nenaudojant stiliaus perkėlimo žaidimas veikia 180,64 kadrų per sekunde greičiu.

4 lentelė. Konvoliucinių neuroninių tinklų greitaveika

DNT Modelis	Filtrų dydžiai	Kadrai per sekunde
1.Greito neuroninio stiliaus DNT (standartinis)	128, 64, 32 (1,679,235 parametrų)	20,22
2.Greito neuroninio stiliaus DNT (sumažintas)	32, 16, 8 (108,771 parametrų)	73,45
3.Interaktyvus stiliaus perkėlimo DNT (sumažintas)	32, 32, 32, 32, 32, 32 (285,635 parametrų)	18,9
4.Interaktyvus stiliaus perkėlimo DNT (standartinis)	32, 64, 128, 128, 128, 64 (2,748,547 parametrų)	7,07

19 pav. pateikiami skirtingų stiliaus perkėlimo DNT modelių rezultatai tarpusavio stilizavimo palyginimui.



(a) 1. (b) 2.

(c) 3. (d) 4.

19 pav. Skirtingų stiliaus perkėlimo DNT modelių rezultatai

4. Rekomendacijos

4.1. Mirgėjimo arba laikino nepastovumo mažinimas DNT

DNT mirgėjimas arba laikinis nepastovumas yra labai aktuali problema stiliaus perkėlimo taikyme vaizdo įrašuose arba realiu laiku. Mirgėjimas DNT modeliuose gali būti mažinamas naudojant mažesnius stiliaus įtakos parametrų svorius apmokant stiliaus perkėlimo modelius. O.Texler et al. straipsnyje pateikia keletą įrankių, kurie pagerina laikiną nepastovumą interaktyviame stiliaus perkėlimo DNT modelyje [TFK+20]. Pirmoji jo rekomendacija - tai triukšmo šalinimas iš mokymosi duomenų. Šiame darbe naudotas modelis buvo apmokytas naudojantis nuotraukomis, gautomis tiesiogiai iš pačio žaidimo, todėl šis triukšmas yra beveik neegzistuojantis palyginus su nuotraukomis gautomis iš kameros. Dėl šios priežasties apdorojus žaidimo nuotraukas O.Texler aprašytu įrankiu ir apmokius naują modelį skirtumai taikant stiliaus perkėlimą žaidime buvo nepastebimi. Paprasčiausias būdas sumažinti laikinį nepastovumą - tai naudoti DNT, skirtą laikiniam nepastovumui mažinti, kaip „Deep Video Prior“ [LXC20]. Svarbu paminėti, kad nors tokie modeliai sumažina laikinį nepastovumą, jie yra skirti vaizdo įrašams apdoroti ir galėtų būti naudojami tik iškirptinių scenų (angl. *cutscene*) stilizavimui.

4.2. Žaidimo optimizavimas

Kita rekomendacija būtų naudoti laiko imties didinimą (angl. *temporal upsampling*). Dabartiniuose žaidimuose dažnai naudojamos įvairios laikinumo schemos (angl. *temporalization schemes*), kurios skirtos pagerinti daug resursų naudojančių efektų kokybę arba veikimo greitį. Tokios laikinumo schemos pavyzdys yra laiko imties didinimas. Pagrindinė šios strategijos idėja yra naudoti informaciją iš prieš tai sugeneruotų kadrų, tokiu būdu pagerinant arba palengvinant sekančių kadrų generavimą. Vienas iš tokių algoritmų būtų vaizdo erdvės dvikryptis scenos atkūrimas (angl. *image-space bidirectional scene reprojection*), kurio pagalba galima generuoti aukštos kokybės tarpinius kadrus tarp kiekvieno neuroninio tinklo sugeneruoto kadro [YTS+11]. Naudojantis šiuo algoritmu galima būtu stipriai sumažinti stiliaus perkėlimo veikimo nuostolius.

4.3. DNT modelio optimizavimas

Optimizuoti galima ne tik žaidimą, bet ir patį DNT modelį. DNT Modelio filtrų dydžių keitimas yra tik vienas iš būdų optimizuoti DNT modelį. Kitas panašus būdas yra DNT modelio kvantizavimas (angl. *quantization*), kuris iškeičia DNT modelio tikslumą į greitaveiką. Kvantizavimas aproksimuoja DNT modelio svorių slankiojo kablelio skaičius į galutines reikšmes ir tokiu būdu dramatiškai sumažina atminties ir atliekamų skaičiavimų reikalavimus naudojant neuroninį tinklą [YSX+19]. Šiuo metu Unity „ML Agents” biblioteka nepalaiko reikiamų operacijų kvantizuotame ONNX DNT modelyje.

Rezultatai

1. Apmokyti du skirtingi stiliaus perkėlimo DNT modeliai ir atlikta jų analizė optimizuojant apmokytus DNT modelius stiliaus perkėlimui realiu laiku.
2. Nustatyta, jog sumažinus stiliaus perkėlimo DNT modelio filtrų skaičių atitinkamai padidėja žaidimo greitaveika, tačiau stiliaus perkėlimo DNT modelis sunkiau atpažįsta objekto detalių kontūrus.
3. Nustatyta, kad naudojant sumažintus filtrus greitojo stiliaus perkėlimo DNT turi pakankamą greitaveiką, kad šis DNT būtu taikomas žaidimuose.
4. Nustatyta, kad interaktyvaus stiliaus perkėlimo DNT grąžina tikslesnius stilizavimo rezultatus, todėl labiau tinka prototipų kūrimo problemai spręsti. Stiliaus perkėlimo DNT modelis leidžia greitai pakeisti viso vaizdo ar objekto stilių ir greitai jį išbandyti žaidime - 3D erdvėje.
5. Aptikta stiliaus perkėlimo problema žaidimuose - mirgėjimas, sukeliamas kadrų ir aplinkos nepastovumo judant 3D erdvėje.
6. Nustatyta, kad stiliaus perkėlimo metu sukeltas mirgėjimas gali būti išnaudotas kuriant unikalias animuotas tekstūras, kurių animacija priklauso nuo žaidimo aplinkos ir žaidėjo kameros pozicijos bei nuo objekto kuriam taikomas stiliaus perkėlimas.
7. Nustatyta, kad stiliaus perkėlimo metu sukeltas mirgėjimas yra stipriai sumažinamas pridėjus triukšmo nuostolių funkciją, tačiau aplink stilizuojamus objektus atsiranda aureolių artefaktai, kurie gali būti sumažinami koreguojant variacijos ir triukšmo nuostolių funkcijų svorius.
8. Nustatyta, kad stiliaus perkėlimo taikymas žaidimuose turi didelį neigiamą poveikį žaidimo greičiui ir jį sumažina nuo 2 iki 25 kartų priklausomai nuo naudojamo DNT ir filtrų dydžio.
9. Naudojantis skirtingų DNT modelių stilizavimo rezultatų nuotraukomis ir jų greitaveika bei analizuojant jų privalumus ir trūkumus pateiktos skirtingų DNT modelių panaudojimo sričių rekomendacijos.

Išvados

1. Stiliaus perkėlimo DNT modeliuose koreguojant filtrų dydžių reikšmes galime kurti mažesnius ir greitesnius DNT modelius, kurių dėka stiliaus perkėlimą įmanoma pritaikyti žaidimuose.
2. Lygių dizaineriai (angl. *level designer*), naudodamiesi stiliaus perkėlimu ir turėdami apmokytus DNT modelius, gali realiu laiku keisdami vieną parametrą išbandyti keletą skirtingų temų visam bendram vaizdui arba pasirinktiems objektams.
3. Interaktyvaus stiliaus perkėlimo DNT modelis turėjo geresnį laikinį nepastovumą, tačiau net ir šio tipo DNT jo pilnai neišvengia, o atsiradęs mirgėjimas tarp stilizuotų kadrų išlieka viena iš didžiausių problemų taikant stiliaus perkėlimą realiu laiku.
4. Naudodamiesi stiliaus perkėlimu žaidimų kūrėjai gali kurti unikalias animuotas tekstūras, nereikalaujančias papildomo kodo, kuris suteiktų tekstūrai animaciją.
5. Stiliaus perkėlimo taikymas sumažina žaidimo veikimo greitį nuo 2 iki 25 kartų, tačiau šiame darbe parodyta, kad stiliaus perkėlimas gali būti sėkmingai naudojamas žaidimų kūrime ir žaidimuose, pasitelkiant modelio ir žaidimo optimizavimą.

Literatūra

[AAA17] Saad Albawi, Tareq Abed Mohammed ir Saad ALZAWI. Understanding of a Convolutional Neural Network, 2017-08. DOI: 10.1109/ICEngTechnol.2017.8308186.

[Ank19] Michelle Bao Ankush Swarnakar. A Real-Time, Stable, Style-Invariant Architecture for Artistic Style Transfer of Videos, 2019. URL: https://github.com/ankushswar1/Fast-Stable-Neural-Style-Transfer-for-Videos/blob/master/Final%20Report.pdf.

[DLH+16] Chao Dong, Chen Change Loy, Kaiming He ir Xiaoou Tang. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016. DOI: 10.1109/TPAMI.2015.2439281.

[FYL20] Michael H. Fischer, Richard R. Yang ir Monica S. Lam. ImagineNet: Restyling Apps Using Neural Style Transfer. *CoRR*, abs/2001.04932, 2020. arXiv: 2001.04932. URL: https://arxiv.org/abs/2001.04932.

[FLJ+14] Jakub Fišer, Michal Lukáč, Ondřej Jamriška, Martin Čadík, Yotam Gingold, Paul Asente ir Daniel Sýkora. Color Me Noisy: Example-based Rendering of Hand-colored Animations with Temporal Noise Control. *Computers Graphics Forum (EGSR 2014)*, 33(4):1–10, 2014. ISSN: 1467-8659. DOI: 10.1111/cgf.12407.

[GEB15] Leon A. Gatys, Alexander S. Ecker ir Matthias Bethge. A Neural Algorithm of Artistic Style. *CoRR*, abs/1508.06576, 2015. arXiv: 1508.06576. URL: http://arxiv.org/abs/1508.06576.

[GW16] Sam Gross ir Michael Wilber. Training and investigating residual nets. 2016. URL: http://torch.ch/blog/2016/02/04/resnets.html.

[ISI16] Satoshi Iizuka, Edgar Simo-Serra ir Hiroshi Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2016)*, 35(4), 2016.

[YSX+19] Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang ir Xiansheng Hua. Quantization Networks. *CoRR*, abs/1911.09464, 2019. arXiv: 1911.09464. URL: http://arxiv.org/abs/1911.09464.

[YTS+11] Lei Yang, Yu-Chiu Tse, Pedro V. Sander, Jason Lawrence, Diego Nehab, Hugues Hoppe ir Clara L. Wilkins. Image-Based Bidirectional Scene Reprojection. *ACM Trans. Graph.*, 30(6):1–10, 2011-12. ISSN: 0730-0301. DOI: 10.1145/2070781.2024184. URL: https://doi.org/10.1145/2070781.2024184.

[JAF16] Justin Johnson, Alexandre Alahi ir Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *CoRR*, abs/1603.08155, 2016. arXiv: 1603.08155. URL: http://arxiv.org/abs/1603.08155.

[JYF+20] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu ir Mingli Song. Neural Style Transfer: A Review. *IEEE Transactions on Visualization and Computer Graphics*, 26(11):3365–3385, 2020. DOI: 10.1109/TVCG.2019.2921336.

[JST+19] Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman ir Daniel Sýkora. Stylizing Video by Example. *ACM Transactions on Graphics*, 38(4), 2019.

[LSD14] Jonathan Long, Evan Shelhamer ir Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *CoRR*, abs/1411.4038, 2014. arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

[LXC20] Chenyang Lei, Yazhou Xing ir Qifeng Chen. Blind Video Temporal Consistency via Deep Video Prior. H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan ir H. Lin, redaktoriai, *Advances in Neural Information Processing Systems*, tom. 33, p. 1083–1093. Curran Associates, Inc., 2020. URL: https://proceedings.neurips.cc/paper/2020/file/0c0a7566915f4f24853fc4192689aa7e-Paper.pdf.

[SC20] Anurag Sarkar ir Seth Cooper. Towards Game Design via Creative Machine Learning (GDCML). *CoRR*, abs/2008.13548, 2020. arXiv: 2008.13548. URL: https://arxiv.org/abs/2008.13548.

[ŞKS21] Deniz Şen, Hasan Tahsin Küçükkaykı ir Elif Sürer. Automated Game Mechanics and Aesthetics Generation Using Neural Style Transfer in 2D Video Games. *Bilişim Teknolojileri Dergisi*, 14:287–300, 2021. ISSN: 1307-9697. DOI: 10.17671/gazibtd.706884.

[SZ15] Karen Simonyan ir Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015. arXiv: 1409.1556 [cs.CV].

[TFK+20] Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menglei Chai, Sergey Tulyakov ir Daniel Sýkora. Interactive Video Stylization Using Few-Shot Patch-Based Training. *ACM Transactions on Graphics*, 39(4):73, 2020.

[Uly16] Dmitry Ulyanov. Fast Neural Doodle. https://github.com/DmitryUlyanov/fast-neural-doodle, 2016.

[ULV+16] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi ir Victor S. Lempitsky. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. *CoRR*, abs/1603.03417, 2016. arXiv: 1603.03417. URL: http://arxiv.org/abs/1603.03417.

[UVL16] Dmitry Ulyanov, Andrea Vedaldi ir Victor S. Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR*, abs/1607.08022, 2016. arXiv: 1607.08022. URL: http://arxiv.org/abs/1607.08022.

[ZIE16] Richard Zhang, Phillip Isola ir Alexei A. Efros. Colorful Image Colorization. *CoRR*, abs/1603.08511, 2016. arXiv: 1603.08511. URL: http://arxiv.org/abs/1603.08511.

Priedas nr. 1
Žaidimų kodai su stiliaus perkėlimu

Unity žaidimų kodai, su įgyvendintu stiliaus perkėlimu yra patalpinti GitHub tinklapyje: `https://github.com/Pilkas-git/bak2022`

Žaidimus su stiliaus perkėlimu sudaro 3 žaidimai - vikingų miestas, paprasta pirmojo asmens šaudyklė ir lenktynės kartingais. Visi jie buvo sukurti Unity, skirti naudoti kaip žaidimų šablonai. Kursinio darbo metu jiems buvo pritaikytas stiliaus perkėlimas. Stiliaus perkėlimo kodas ir naudoti modeliai randami atsidarius pasirinktą žaidimą -> Assets -> StyleTransfer.

Priedas nr. 2
Stiliaus perkėlimo modelio apmokymo kodas ir modeliai

Stiliaus perkėlimo modelio apmokymas ir versijavimas: `https://www.kaggle.com/augustasmasolas/kursstyletransfer`

DNT modeliai gali būti atsiunčiami keičiant versija ir parsisiunčiant apmokytą tos versijos ONNX modelį iš Data stulpelio.