Pregled i analiza metoda generiranja sintsetova za primjenu u domeni detekcije osoba i raspoznavanja akcija

Goran Paulin
Odjel za informatiku
Sveučilište u Rijeci
Rijeka, Hrvatska
gp@kreativni.hr

SAŽETAK

Sintsetovi nisu novost, u području računalnog vida koriste se od 1989. godine, ali značajan razvoj metoda i tehnika njihovog generiranja prisutan je tek posljednjih desetak godina. Ovaj rad donosi iscrpan pregled dosadašnje prakse generiranja sintsetova i analizu temeljem koje sistematizira proces generiranja. Uz to, fokusiran na primjenu sintsetova u domeni raspoznavanja rukometnih akcija, sadrži upute i preporuke za korištenje odabranih metoda generiranja sintsetova, te predlaže provedbu eksperimentalnog istraživanja kojim bi se potvrdila njihova učinkovitost.

Ključne riječi: računalni vid, sintetički podaci, sintset, metode generiranja, rukomet

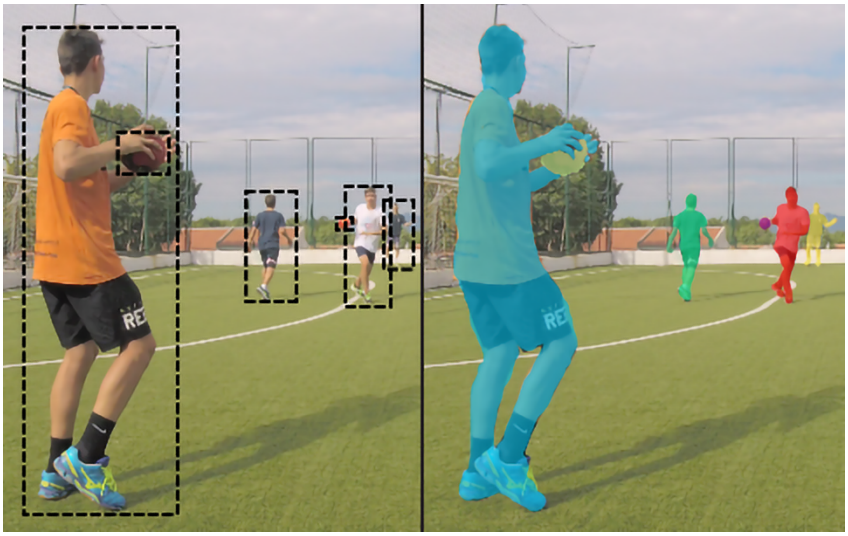
1 Uvod

Duboko učenje (eng. *deep learning*) zahtijeva velike količine podataka [1]. Iako su raspoloživi brojni javno dostupni skupovi podataka [2] (u daljnjem tekstu "datasetovi"), često je potrebno, zbog specifičnosti domene ili s namjerom unapređenja postojećeg, izgraditi novi dataset.

Jedna od takvih, deficitarnih domena, je rukomet, za koji, u vremenu nastanka ovog rada, postoje samo dva javno dostupna dataseta:

- CVBASE '06 Dataset – 10-minutna video snimka u niskoj rezoluciji (384x288 px), namjenski odigrane rukometne utakmice [3] i
- Sports-1M – kompilacija 1.133.158 YouTube video snimki, podijeljenih u 487 klasa od kojih se dvije odnose na rukomet (u dvorani i na plaži) [4].

Skup slika namijenjen nadziranom učenju u području računalnog vida, osim osnovnog podatka (slika) mora sadržavati i pripadno činjenično stanje (eng. *ground truth*) koje se dobiva anotiranjem. Anotiranje se može provesti na nivou cijele slike, označavanjem kojoj klasi slika pripada (npr. "rukometaš", "lopta"...) ili, na nivou pojedinih objekata na slici, njihovim označavanjem graničnim okvirom (eng. *bounding box*) odnosno označavanjem piksela koji pripadaju pojedinom objektu (Slika 1). Anotiranje realnih slika je spor i, u pojedinim područjima, poput medicine, stručan proces koji traži izrazitu pažnju anotatora te je kao takav podložan ljudskoj pogrešci. Sve navedeno kreiranje dataseta čini vremenski zahtjevnim i utoliko skupim.



Slika 1 - Označavanje objekata graničnim okvirom (lijevo) i označavanje piksela koji pripadaju pojedinom objektu (desno) (Izvor: [5]*)*

Potencijalno rješenje ovog problema su sintetički skupovi podataka (eng. *synthetic dataset,* u daljnjem tekstu ćemo ih nazivati "sintset"). Sintsetovi nisu novost, u području računalnog vida koriste se od 1989. godine [6], ali značajan razvoj metoda i tehnika njihovog generiranja prisutan je tek posljednjih desetak godina.

Motiviran idejom otkrivanja najbolje prakse za izgradnju optimalnog sintseta za potrebe raspoznavanja rukometnih akcija, ovaj rad istražuje postojeće sintsetove, načine njihovog nastanaka i primjenu, uz sljedeće doprinose:

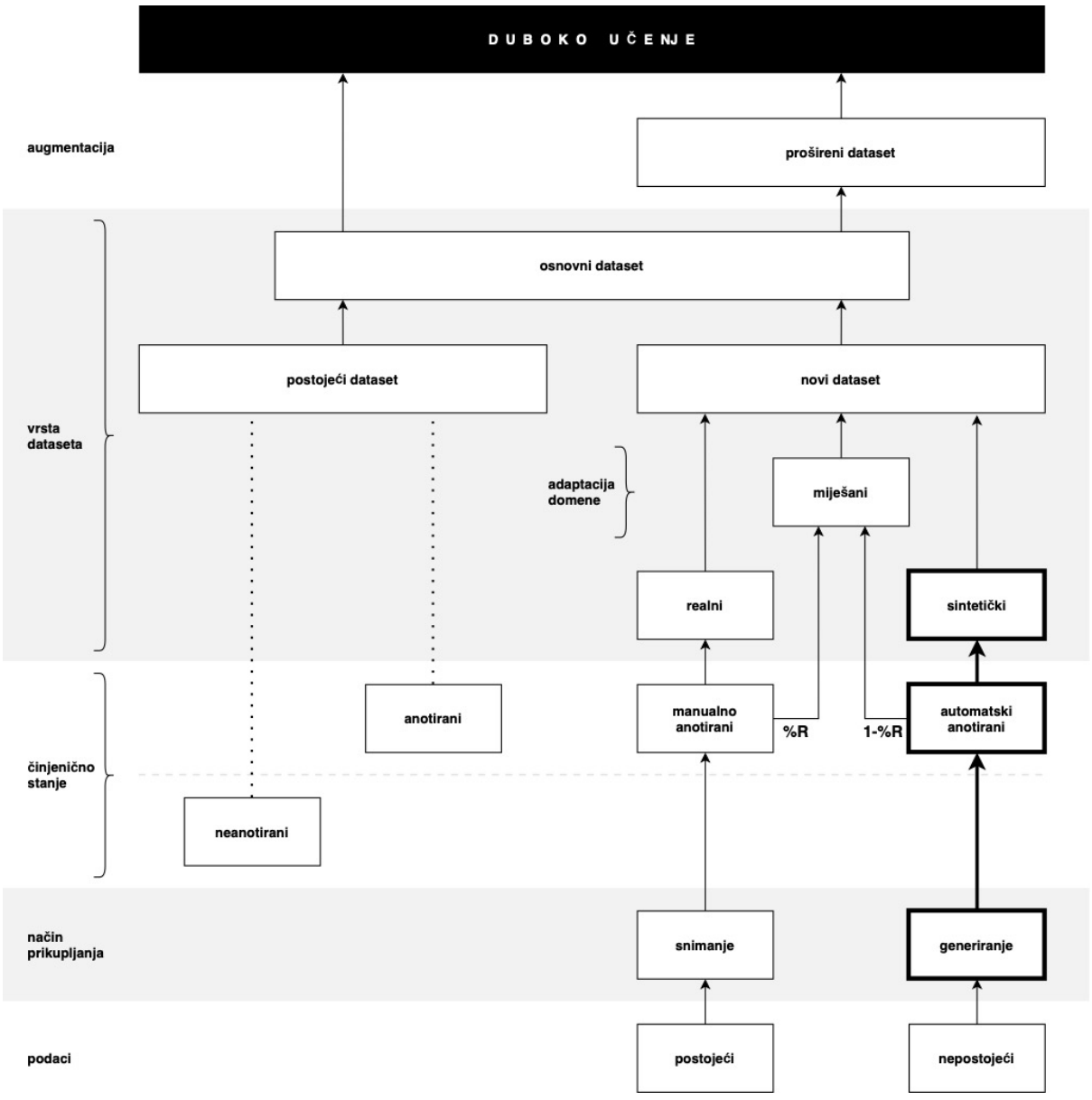
- kronološki pregled razvoja metoda i tehnika generiranja sintsetova u području računalnog vida
- sistematizacija procesa generiranja sintsetova u području računalnog vida
- prijedlog optimalnog načina generiranja sintseta za potrebe raspoznavanja rukometnih akcija.

Poglavlje 2 ukratko predstavlja sintetičke podatke, način njihove pripreme za korištenje u dubokom učenju te domene u kojima se koriste. Kronološki pregled radova (poglavlje 3) obuhvatio je sve radove iz [1] koji se odnose na područje računalnog vida te iterativno istražio povezane reference. Analiza (poglavlje 4) se osvrće na detektirane metode i tehnike te sistematizira generalni proces generiranja sintsetova i pojedine procese u njemu. Uz to predstavlja i zastupljenost pojedinih domena i u njima korištenih metoda generiranja. Diskusija (poglavlje 5) donosi upute i preporuke za primjenu predloženih metoda generiranja sintseta za potrebe raspoznavanja rukometnih akcija. U zaključku su dani prijedlozi za provedbu eksperimentalnog istraživanja kojim bi se potvrdila učinkovitost odabranih metoda.

2 Sintetički podaci

Skupovi podataka (eng. *datasets*) bili su jedni od pokretača napretka u računalnom vidu, obradi prirodnog jezika i drugim područjima umjetne inteligencije koja se oslanjaju na duboko učenje [7]. Svaki dataset je, zbog konkretnog sadržaja, na neki način pristran (eng. *biased*), što potiče istraživače na izgradnju novih i većih datasetova [8], ali kvaliteta dataseta ne mjeri se isključivo njegovom veličinom već različitim faktorima poput raznolikosti, cjelovitosti i distribucije podataka u njemu [9].

Kao što je vidljivo iz Slike 2, datasetovi nastaju prikupljanjem i anotacijom podataka. Pritom podaci mogu biti postojeći, dobiveni izravnim mjerenjem, ili nepostojeći, odnosno takvi koje je potrebno prethodno generirati.



Slika 2 - Priprema datasetova za duboko učenje

Sintetički podaci (eng. *synthetic data*) definirani su u [10] kao podaci koji se ne dobivaju izravnim mjerenjem. Povijest njihove primjene u strojnom učenju neposredno je vezana uz računalni vid i seže u 1989. godinu kada se po prvi put spominje njihovo korištenje u domenama autonomne vožnje [6] i analize optičkog toka [11]. Kao povod korištenju sintetičkih podataka navode se problem prikupljanja dostatne količine realnih podataka za učenje neuronske mreže te mogućnost automatskog anotiranja, što do danas ostaju dva primarna razloga njihove primjene.

Sintetički podaci mogu biti neograničeni izvor podataka i njima je moguće simulirati situacije s kojima se još nismo susreli. Uz to, omogućavaju prevladavanje ograničenja upotrebe realnih podataka uvjetovano poštivanjem privatnosti ili drugim propisima [12].

Studija o učinkovitosti sintetičkih podataka iz 2017. godine [13] pokazala je da se u 70% slučajeva, koristeći isključivo sintetičke podatke, moglo reproducirati rezultate postignute koristeći realne podatke.

U kontekstu troška, ali i vremena produkcije, podaci iz [14] govore u prilog isplativosti generiranja sintseta s obzirom da je trošak izrade pojedine sintetičke slike čak 444 puta manji od troška realnog ekvivalenta. Navedena ušteda u vremenu (27%) zanemaruje veličinu dataseta, a ukoliko ga uzmemo u obzir, produkcija jednako velikog realnog dataseta (1M slika) trajala bi 909 puta duže. Kada je jednom pripremljeno okruženje za generiranje sintetičkih slika i pripadnih anotacija, veličinu sintseta moguće je značajno povećati uz zanemarivo mali trošak po svakoj dodatnoj slici.

Tablica 1 - Trošak izrade i anotiranja pojedine slike u sintetičkom i realnom datasetu (Izvor: [14]*)*

dataset	sintetički	realni
veličina dataseta (broj slika)	1.000.000+	1.500
vrijeme potrebno za pripremu dataseta	88 sati	120 sati
trošak po slici (USD)	**0,0072**	**3,20**

Prema Tripathi i sur. [15], generirani sintetički podaci moraju biti učinkoviti, svjesni zadatka i realistični. Učinkovitost proizlazi iz istovremenog smanjivanja količine podataka, kako bi se štedilo na resursima potrebnim za učenje, uz povećanje raznovrsnosti uzoraka. Svjesni zadatka, sintetički podaci moraju stvoriti primjere koji pomažu u poboljšanju performansi ciljane neuronske mreže. Pritom moraju biti vizualno realistični kako bi minimizirali jaz domene prisutan između realnih i računalno generiranih slika (koje mogu varirati između nerealističnih i fotorealističnih), te na taj način poboljšali generalizaciju.

Ukoliko jaz domene nije minimiziran unutar samog sintseta, moguće je provesti adaptaciju domene jednostavnim miješanjem sintetičkog i realnog dataseta u određenim postocima [16]. Na taj način nastaju miješani odnosno hibridni datasetovi.

Unutar područja računalnog vida, sintetički podaci, kao parovi slika i pripadnih anotacija, primjenjuje se u sljedećim domenama:

- autonomna vožnja [6]
- detekcija objekata [17]
- određivanje poze [18]
- nadziranje [19]
- optički tok [11]
- razumijevanje scene [20]
- autonomno letenje [21]
- rekonstrukcija scene [22]
- medicinska segmentacija [23]
- navigacija robota [24]
- stereo disparitet [25]
- podržano učenje [26]
- raspoznavanje akcije [27]
- rekonstrukcija objekta [28]
- segmentacija [23]
- raspoznavanje objekta [29]
- robotski hvat [30]
- vizualno rasuđivanje [31]
- evaluacija značajki slike [32]
- praćenje objekata [33]
- utjelovljena umjetna inteligencija [7]
- analiza svjetlosnih polja [34]
- arhitektura neuronske mreže [35]
- generiranje tekstura [36]
- klasifikacija objekata [37]
- medicinska lokalizacija [38]
- određivanje gledišta [39]
- određivanje smjera gledanja [40]
- virtualna stvarnost [41]
- vizualna lokalizacija [42].

Sintetički podaci koriste se i za analizu anomalija u putanjama (sintetizirane putanje) [43], rekonstrukciju objekata iz mapa dubine (sintetizirani vokselizirani 3D objekti) [28], dodavanje sintetičkog šuma video zapisima [44], kreiranje sintsetova geometrije scene [45] i sintsetova 3D objekata razloženih na dijelove [46].

Izvan domene računalnog vida primjena sintetičkih podataka značajna je za neuronsko programiranje i bioinformatiku [1].

Metode i proces generiranja sintetičkih podataka istraženi su u ovom radu i obrađeni u poglavlju 4 (Analiza).

3 Pregled područja

Pregled područja prikazan je kronološki kako bi se moglo pratiti uvođenje i razvoj pojedinih metoda i tehnika generiranja sintsetova, te njihov međusobni utjecaj. Svi u pregledu navedeni radovi, osim tamo gdje je izričito navedeno suprotno, prijavljuju postizanje istih ili boljih rezultata od prethodno najboljih (eng. *state of the art*), korištenjem sintseta.

1989

Prva poznata primjena sintetičkih podataka za učenje neuronske mreže zabilježena je u domeni autonomne vožnje [6]. Potaknuta je problemom velike količine potrebnih podataka za treniranje mreže koje je teško prikupiti, pogotovo u različitim uvjetima vožnje i uz mogućnost promjene orijentacije kamere. Zato Pomerleau sa sur. kreira generator virtualnih prometnica. Za treniranje neuronske mreže koriste dva seta od po 1.200 slika prometnica simuliranih u različitim svjetlosnim uvjetima te s realističnim stupnjem šuma: jedan set u rezoluciji 30x32 px, koji predstavlja plavi kanal RGB kamere, i drugi u 8x32 px, koji predstavlja mapu dubine (eng. *depth map,* D), dobivenu laserskim mjeračem udaljenosti (eng. *range finder*). Nakon 40 epoha treninga, neuronska mreža postizala je točnost oko 90% na novim simuliranim slikama ceste i pokazala da može učinkovito upravljati vozilom "pod određenim terenskim uvjetima". Koristeći nisku rezoluciju sintetiziranih slika teško je razlikovati realne i sintetizirane ceste, što objašnjava visoku postignutu točnost modela.

Iste godine spominje se prvo korištenje sintetičkih podataka u domeni analize optičkog toka [11]. Iako Little i Verri ne prijavljuju detalje o sintetiziranim slikama, naglašavaju da su im omogućile usporedbu optičkih tokova generiranih različitim algoritmima sa pravim projiciranim poljima brzine (eng. *true projected velocity field*) koji su na sintetičkim slikama poznati i mogu se tretirati kao činjenično stanje (eng. *ground truth*, GT).

1994

Inspirirani s [11], Barron, Fleet i Beauchemin [47] generiraju četiri jednostavne ("Sinusoid1", "Square2", "Translating Tree", "Diverging Tree") i jednu kompleksnu ("Yosemite") sintetiziranu video sekvencu za evaluaciju performansi tehnika za analizu optičkog toka. Kao glavnu prednost sintetičkih inputa navode mogućnost upravljanja 2D poljem kretanja (eng. *motion field*) i svojstvima scene (eng. *scene properties*) te mogućnost metodičkog testiranja tijekom kojeg je moguće kvantificirati performanse. Uvode i naknadnu obradu (eng. *post-processing*) sintetiziranih podataka, prethodnim izglađivanjem (eng. presmoothing), koristeći Gaussov kernel sa standardnom devijacijom od 1,5 px u prostoru i vremenu. Time umanjuju temporalni *aliasing* i efekt kvantizacije. Napominju da sintetički podaci za testiranje, u odnosu na realne, daju bolje rezultate uz dvostruko manje izglađivanja.

1999

Hamarneh, koristeći MATLAB, za potrebe segmentacije medicinskih slika razvija prvi generator prostorno-vremenski deformabilnih oblika [23]. Njime sintetizira sekvence sa po 16 slika u sivim tonovima u rezoluciji 160x182 px koje koristi za detekciju i segmentiranje sličnih oblika u 2D sekvencama. Pritom u sintetizirane sekvence, kako bi ih uskladio s realnim primjerima, uvodi globalni i lokalni šum, preklapajuće i dodirujuće okluzije te nedostajuće frejmove. Koristeći Active Shape Models metodu dokazuje uspješnost primjene tako skrojenih sintsetova za učenje modela.

Tražeći način za svladavanje osnovnih problema računalnog vida poput analize pokreta (eng. *motion analysis*), raspoznavanja oblika i albeda na fotografija te ekstrapoliranja detalja slike, Freeman i Pasztor [22] razvijaju metodu VISTA (Vision by Image/Scene TrAining) kojoj je cilj za danu 2D sliku odrediti 3D scenu na kojoj se bazira. U tu svrhu kreiraju parove anotiranih 3D scena virtualnog svijeta i pripadnih renderiranih slika te modeliraju taj svijet koristeći Markovljevu mrežu, učeći parametre mreže kroz velik broj parova takvih primjera. Svoju metodu uspješno primjenjuju na problem "super-rezolucije" (određivanje visoko frekvencijskih detalja na slici niske rezolucije).

2001

Rosales i sur. predlažu sustav za određivanje 3D poze šake iz monokularne kolor sekvence metodom nadziranog učenja [18]. Za mapiranje značajki slika na vjerojatne 3D poze koriste Specialized Mappings Architecture (SMA), baziran na regresiji umjesto na klasifikaciji, što im omogućava kontinuum određenih poza umjesto konačnog broja klasa. Koristeći CyberGlove, rukavicu s 22 senzora koji prate kutne pokrete dlana i prstiju, prikupljaju podatke za učenje i apliciraju ih na 3D model šake koji potom renderiraju iz 86 različitih pogleda. Tako kreiraju 300.000 sintetičkih slika od kojih samo 8.000 (2,67%) koriste za učenje, dok je ostatak namijenjen testiranju. Ova disproporcija ukazuje na veliki potencijal generiranja masivnih sintsetova s raspoloživim hardverom, ali i na ograničenost (prvenstveno u kontekstu brzine) istog hardvera za potrebe treniranja modela.

2003

Grauman, Shakhnarovich i Darrell pripremaju sintset s 20.000 slika u rezoluciji 320x240 px koristeći POSER, komercijalni softver za animaciju koji omogućava manipulaciju s realističnim humanoidnim modelima, njihovo pozicioniranje na sceni i renderiranje iz željenog pogleda s teksturama ili bez [48]. Koristeći upravo renderirane slike bez tekstura (siluete) i unaprijed poznate, kao parametre 3D modela, lokacije 19 zglobova za Bayesian Multi-View Shape rekonstrukciju, određuju pozu tijela iz 2D slike, s prosječnom preciznošću unutar 3 cm. Za generiranje silueta koriste virtualne likove različitih proporcija tijela, kose i odjeće, a siluetama dodaju šum kako bi simulirali rupe ili nepostojeća proširenja prisutna u realnom setu nakon primjene mehanizama za uklanjanje pozadine.

2004

Nathan i Howard kreiraju Gazebo [24], jedan od prvih simulatora namijenjenih robotskoj navigaciji koji u virtualnom 3D svijetu omogućava učenje na sintetičkim slikama generiranim u realnom vremenu. Iako je kreiran prvenstveno kao simulator eksterijera njegova osnovna slabost je nemogućnost simulacije različitih fizičkih modela tla. Uz to, ne podržava deformabilne objekte te dinamiku fluida i termodinamiku.

2006

Desurmont i sur. grade prvi sportski sintetički video set namijenjen praćenju igrača nogometa [33]. Sastoji se od 13 sekvenci u trajanju od po 400 sekundi, u rezoluciji 1.400 x 1.050 px. GT je raspoloživ za pozicije igrača te pozicije i parametre 7 statičnih i 3 PTZ (eng. *pan-tilt-zoom*) kamere kojima se simuliraju standardni uvjeti televizijskog prijenosa nogometne utakmice. Unatoč vjernim pozicijama kamera, osnovna slabost ovog seta je manjak bilo kakvog realističnog šuma i distorzija slike.

2007

Taylor, Chosak i Brewer uočavaju problem resursa potrebnih za kreiranje virtualnog svijeta te odlučuju iskoristiti postojeći svijet komercijalne igre Half-Life 2 koja dopušta modifikacije od strane igrača (eng. *modding*) [19]. Skriptanim kontrolama omogućavaju izvođenje ponovljivih scenarija s podesivim parametrima poput pozicija višestrukih međusobno sinkroniziranih PTZ kamera, rasvjete (doba dana, umjetni izvori svjetla) te sekvenci različitih događanja (akcija) na sceni. Tako grade ObjectVideo Virtual Video (OVVV), sustav za generiranje sintsetova u domeni nadziranja, koji im omogućava renderiranje 4 simultane sekvence u rezoluciji 320x240 px u 10 fps, bez GT, odnosno 2 sekvence ukoliko paralelno generiraju i GT. GT za sve vidljive objekte uključuje 3D poziciju centroida, projekciju centroida na renderiranu sliku, granični okvir oko cijelog objekta i granični okvir oko vidljivog dijela objekta. GT kamere sastoji se od njene 3D pozicije, orijentacije, horizontalnog vidnog polja (eng. *field of view*, FOV) i dimenzija slike. Uz to, generira se i segmentacijska mapa, na nivou piksela. Realizam sintetičkih slika povećavaju dodajući šum na nivou piksela, video *ghostingom* (multipliciranje slike zakašnjelom kopijom prethodnog signala) i radijalnom distorzijom koja simulira nesavršenost leće kamere. Za razliku od realne snimke u kojoj je, zbog nesavršene optike, automatski prisutan *antialiasing*, efekt nazubljenih rubova (eng. *aliasing*) na sintetičkim slikama reduciraju SSAA (Super-Sampling Antialiasing Algorithm) algoritmom, renderirajući slike u dvostruko većoj rezoluciji i izglađujući prilikom smanjenja blokove 2x2 piksela. U usporedbi korištenja realnih i sintetičkih podataka za ocjenu algoritama praćenja (eng. *tracking*) postižu slične rezultate (mjereći ukupnu pogrešku).

2008

Ragheb i sur. generiraju ViHASi (Virtual Human action Silhouette) sintset za evaluaciju metoda za raspoznavanje akcija na bazi silueta (eng. S*ilhouette-Based Human Action Recognition*, SBHAR) [27]. Koristeći MotionBuilder, namijenjen filmskoj industriji, kombiniraju 9 3D modela virtualnih karaktera i 20 klasa radnji te ih renderiraju u 640x480 px

koristeći do 40 fiksnih kamera. U naknadnoj obradi uz šum dodaju i parcijalne okluzije (parovi vodoravne i okomite linije širine 0-20 px). Testirajući vlastitu SBHAR metodu, uočavaju da dodavanje do 15% šuma reducira sposobnost raspoznavanja na 81,73%, a 50% šuma na svega 25%. Kod primjene okluzije značajno veći utjecaj na degradaciju raspoznavanja ima manjak vodoravnih (raspoznavanje pada na 20%) nego okomitih piksela (raspoznavanje pada na 69,70%). Osnovni nedostatak ViHASi sintseta je manjak prirodnih tranzicija između pojedinih radnji.

Saxena, Driemeyer i Ng generiraju sintetičke slike za potrebe nadziranog učenja, manualno anotirajući lokaciju ispravnog robotskog hvata na 5 klasa 3D modela [49]. Za sintetiziranje slika koriste POV-Ray renderer i zaključuju da se s povećanjem grafičkog realizma poboljšava točnost algoritma za određivanje hvata. Tijekom generiranja 2.500 slika rendomiziraju različita svojstva objekata: boju, veličinu i tekst (na koricama knjiga) čime zapravo uvode metodu rendomizacije domene koju će [50] popularizirati 2017. godine. Ukazuju na vremenski zahtjevnu pripremu 3D objekata napominjući da se problem može riješiti koristeći 3D objekte raspoložive na Internetu, uz minorne modifikacije.

2010

Baker i sur. kreiraju hibridni (kombinacija realnih i sintetičkih slika) Middlebury dataset za evaluaciju metoda analize optičkog toka [51]. 3Delight Renderman-kompatibilan renderer omogućava im korištenje linearnog prostora boje i ambijentalne okluzije (eng. *ambient occlusion*) kojom aproksimiraju globalnu iluminaciju. GT kreiraju projiciranjem 3D kretanja na 2D sliku zbog čega vektor optičkog toka pohranjen u pojedinom pikselu može predstavljati kretanje više od jednog objekta, što je slabost ove metode. Autori predlažu da se prilikom kreiranja budućih datasetova za ovu namjenu uzmu u obzir različiti materijali, promjene osvjetljenja, atmosferski efekti i transparencija.

Tarel i sur. kreiraju dva sintseta (FRIDA i FRIDA2) namijenjena evaluaciji algoritama za obnavljanje vidljivosti i kontrasta [52]. FRIDA sadrži 90 slika 18 isključivo urbanih scena, a FRIDA2 330 slika 66 različitih prometnica. Koristeći SiVIC za izgradnju virtualnog okruženja, fizikalni model kretanja vozila i renderiranje (RGB+D), svakoj od generiranih slika dodaju 4 varijante koje, po prvi put u domeni autonomne vožnje, sadrže različite oblike magle: uniformnu, heterogenu, oblačastu (eng. *cloudy*) i heterogeno oblačastu.

Metodu Taylora i sur. [19] za generiranja sintseta, prilagodivši kameru kako bi se njenim pomicanjem i rotacijom mogli snimati isključivo virtualni prolaznici (a ne cesta ispred vozila), koriste 2010. godine Marín i sur., [53], za detektiranje pješaka i dokazuju da je moguće učenje modela na virtualnom setu za uspješnu detekciju na snimkama realnih scena.

Za potrebe medicinskih segmentacija, Hamarneh u suradnji s Jassiem gradi novi sintset [54] u kojem iterativnim rastom simulira vaskularna stabla i kreira njihove volumetrijske prikaze koristeći rasterizaciju voksela (eng. *voxel*). S obzirom da vaskularna stabla, kao i prethodno deformabilne oblike [23] modelira parametarski, područje medicine pokazuje se kao jedino u kojem se koriste proceduralno generirani 3D modeli.

Queiroz i sur. razvijaju prototip generatora video sekvenci animiranih sintetičkih lica s pripadnim GT u svakom frejmu [17]. Pomoću njega grade Virtual Human Faces Database (VHuF). Generator koristi kombinaciju fotografija lica apliciranih na morfabilni model i realistične animacije ponašanja ljudskih lica (govor, facijalne ekspresije i kretanje očiju). Baza je namijenjena treniranju i evaluaciji algoritama za praćenje i detekciju osoba, a u testovima je pokazala pouzdanost iznad 95%.

2011

Kaneva, Torralba i Freeman kreiraju dva fotorealistična sintseta, Virtual City i Statue of Liberty, kako bi na njima istražili robusnost deskriptora značajki (eng. *feature descriptor*) u uvjetima promjenjivog osvjetljenja i pogleda na scenu [32]. Za rendering koriste 3ds Maxov Mental Ray renderer koji im omogućava korištenje Daylight sustava za iluminaciju scene. Njime variraju 5 doba (od 11 ujutro do 5 popodne) istog ljetnog dana, izbjegavajući noć, a kamerama variraju širinu leće od 50 do 200 mm. Uspoređujući svoj Statue of Liberty set sa realnim (fotografije Kipa slobode) utvrđuju izrazito slične performanse i rangiranje opisnika. Značajnu degradaciju performansi uočavaju kod promjene osvjetljenja što se može objasniti kretanjem sjena i refleksija uslijed kretanja Sunca tijekom dana.

Pishchulin i sur. za generiranje sintseta koriste snimanje pokreta (eng. *motion capture*, *mo-cap*) s 8 HD kamera ispred uniformno obojane pozadine. Iz snimke ekstrahiraju siluete i prilagođavaju im morfabilni 3D model (eng. *3D Morphable Model*, 3DMM) tijela [16]. Nakon toga nasumično uzorkuju oblik tijela (visina/debljina) i smještaju ga na nasumično odabranu foto pozadinu. Koristeći rendomizirane poglede kamere renderiraju tisuće sintetičkih primjera koristeći samo 11 snimljenih ljudi. Dodajući realne podatke (čime zapravo provode adaptaciju domene) nadmašuju najbolje rezultate prethodnika. Problem ove metode generiranja sintseta je pozicioniranje sintetiziranih karaktera na nerealnim odnosno nemogućim mjestima na pozadinama, neprirodan spoj ljudi i pozadina te manjak odgovarajućih sjena.

Shotton i sur. predlažu metodu za brzo i točno određivanje 3D poze iz samo jedne 640x480 px dubinske slike bez temporalnih informacija, koristeći Kinect [55]. Generiraju velik (500K slika) i jako raznolik sintset za treniranje modela koji omogućava klasifikatoru postizanje invarijantnost u odnosu na pozu, oblik tijela i odjeću. Cilj je bio izgraditi realističan set (u kontekstu dubinske mape) kako bi renderi različitih pogleda bili što bliži slikama kamere i imali veliki raspon varijacija poza u skladu s onima koje će se testirati. Kolor kodiranjem dijelova tijela (njih 31), koristeći teksturu, svode problem određivanja poze na klasifikaciju, a to im omogućava i razlikovanje lijeve i desne strane tijela.

2012

Vacavant i sur., koristeći SiVIC simulator, kreiraju BMC (Background Model Challenge) sintset namijenjen usporedbi algoritama za oduzimanje pozadine [56]. Fokusiraju se na variranje atmosferskih uvjeta i uvode vjetar u simulaciju.

Butler i sur., kao i prethodnici [51], upozoravaju na problem transparencije kod kreiranja sintsetova u domeni optičkog toka [57]. Za generiranje svog MPI-Sintel sintseta koriste "Sintel", Blenderov animirani film otvorenog koda (eng. *open-source*). Modificiraju Blenderovo interno zamućenje pokreta (eng. *motion blur*) kako bi dalo točan vektor kretanja za svaki piksel (u funkciji GT). Za razliku od [51], MPI-Sintel sadrži duže sekvence, veće pokrete, zrcalne refleksije (eng. *specular reflections*), zamućenje pokreta, zamućenje fokusa (eng. *defocus blur*) i atmosferske efekte. Savjetuju oprez kod korištenja ovog sintseta prilikom treniranja i evaluacije algoritama koji ovise o zakonima fizike jer ih animacija nužno ne slijedi.

Satkin, Lin i Hebert predlažu korištenje postojećih baza 3D modela za rješavanje problema razumijevanja scene iz monokularnih slika [20]. Kreću od pretpostavke da čovjekova okruženja nisu slučajna već se sastoje od različitih veličina, oblika, orijentacija i položaja objekata koji se nalaze u određenim međusobnim odnosima, a koje je moguće naučiti ako postoji dovoljna količina podataka. Zato kreiraju sintset od 500 slika u kategorijama "spavaća soba" i "dnevni boravak". Koriste Google SketchUp koji im omogućava automatsku kalibraciju kamere označavanjem točke nestajanja (eng. *vanishing point*) i popunjavanje scene gotovim Google 3D Warehouse objektima. S obzirom da im fotorealističnost nije potrebna, za renderiranje objekata koriste OpenGL i osim RGB slike generiraju pripadnu masku objekta i normale površina (eng. *surface normals*). Za automatsko grupiranje srodnih objekata, kao i za procjenu slobodnog prostora na sceni, koriste vokselizaciju.

Pepik i sur. smatraju da 2D granični okvir nije optimalna reprezentacija detekcije objekata koji sadrže pokretne dijelove [58]. Zato kreiraju sintset generirajući nerealistične, gradijente renderinge 3D CAD modela (automobili i bicikli) iz kojih izravno uče HOG značajke. Njihov model automatski uči volumetrijske dijelove što im omogućava određivanje gledišta (eng. *viewpoints*) i lokacije individualnih dijelova objekta. Uspoređuju realni, sintetički i miješani dataset za treniranje modela i zaključuju da se samostalno korišten sintset, zbog statistike značajki, ponaša najlošije, ali da u kombinaciji s realnim, u svim testiranim slučajevima daje najbolje rezultate.

Peris i sur. kreiraju Tsukuba Stereo dataset za evaluacija stereo dispariteta [25]. Za modeliranje objekata koriste Pixologic ZBrush, a za fotorealistično renderiranje Autodesk Mayu. Na scenama primarno variraju rasvjetu, simulirajući utjecaje različitih vrsta svjetala u rasponu od fluorescentnog do sunčevog. Osim RGB slika, sintset sadrži, kao GT, mapu dispariteta (eng. *disparity map*), mapu neprekrivanja (eng. *non-occlusion map*) i mapu diskontinuiteta dubine (eng. *near depth discontinuity mask*), koje su izvedene iz renderirane mape dubine.

2013

Zambanin i Kampel kreiraju SIDIRE sintset za istraživanje utjecaja promjene iluminacije na izgled objekta [29]. Za svaki od 14 3D modela novčića, u Blenderu renderiraju po 12 setova slika u kojima 4 materijala s različitim BRDF (dvosmjerna funkcija distribucije refleksije, eng. *Bidirectional Reflectance Distribution Function*) kombiniraju s teksturom u 3 stupnja intenziteta (pri čemu prvi stupanj predstavlja nepostojanje teksture) i osvjetljavaju iz 65

različitih smjerova. Evaluirajući objekte bez teksture, utvrdili su da se značajke za koje se prethodno tvrdilo da su neosjetljive na svjetlosne uvjete, poput smjera gradijenta (eng. *Gradient Direction*, GD) ili Self Quotient Image (SQI), ponašaju značajno lošije na objektima bez teksture nego na teksturiranim objektima. Umjesto njih, u uvjetima jakih svjetlosnih promjena, preporučuju korištenje Jets of Even Gabor Filter Responses (JEG), kako za neteksturirane tako i za teksturirane objekte.

Haltakov, Unger i Ilic koriste simulator vožnje otvorenog koda VDrift, koji im omogućava kreiranje različitih prometnih scenarija, sličnih stvarnima, za generiranje sintseta za višeklasnu (eng. *multi-class*) segmentaciju slike u domeni autonomne vožnje [59]. Za renderiranje segmentacijske anotacije koriste uniformno bojanje tekstura objekata koji pripadaju različitim klasama, a prilikom renderiranja takvih slika isključuju rasvjetu, sjene, refleksije, *antialiasing* i MIP mapiranje (eng. *mip-mapping*) kako bi dobili ispravnu vrijednost boje za svaki piksel. Njihov CRF model postizao je dobre rezultate (89,0% točnosti) koristeći samo rendere s teksturama. Korištenje dubinskih značajki ili značajki optičkog toka davalo je relativno loše rezultate (80,6% i 75%). Kombinacija tekstura i dubinskih značajki ili značajki optičkog toka dala je najbolje rezultate (91,2% odnosno 90,1%), a kombinacija svih značajki (tekstura, dubina i optički tok) dala je nešto lošiji rezultat (91,0%) što sugerira da su informacije enkodirane u dubinskim značajkama i značajkama optičkog toka relativno slične.

Henry i sur. kreiraju Synthetic Migrating Cells sintset za medicinsku segmentaciju modelirajući 6 umjetnih neutrofila sa anizotropnim Gaussovim oblicima različitih orijentacija i ručno im ucrtavaju putanje po kojima se kreću (eng. *motion path*) [60]. Tako kreirani sintset, poput [44], naknadno obrađuju dodajući bijeli šum s 5 različitih intenziteta, kako bi povećali sličnost između neutrofila i pozadine, te time otežali segmentaciju.

Mnih i sur. koriste osmobitne Atari 2600 igre za generiranje slika u rezoluciji 210x160 px i 128 boja koje, nakon konverzije u sive tonove (eng. *greyscale*), smanjivanja na 110x84 px i rezanja na 84x84 px (zona igre) u realnom vremenu prosljeđuju konvolucijskog neuronskoj mreži i treniraju je igrati koristeći varijantu Q-learninga [26]. Testirajući tako naučenu mrežu na 7 igara, na 6 postižu bolje rezultate u odnosu na prethodne pristupe, a na 3 bolje i od ljudi, ekspertnih igrača.

Za razliku od [25] koji teže fotorealizmu, Haeusler i Kondermann zastupaju mišljenje da se problemi pojedinih algoritama u domeni stereo dispariteta bolje uočavaju koristeći specifične ne-fotorealistične primjere [61]. Kao tipični primjer problema navode neteksturirane pozadine koje, u slučaju određivanja dubine piksela na slici u domeni autonomne vožnje, rezultiraju smještanjem cijelog neba na udaljenost maksimalno udaljenog vidljivog objekta na slici. Njihov sintset je naizgled apstraktan, ali kreiran tako da omogući kombinatoričko iteriranje kroz prostor svog dizajna kako bi se istražile najrelevantnije situacije za pojedinu primjenu.

Moiseev i sur. evaluiraju metode raspoznavanja prometnih znakova koristeći sintset i ukazuju na značaj njegove veličine te postupka prethodne segmentacije [62]. Sintset grade koristeći kombinaciju crteža prometnih znakova i slučajno odabranih foto pozadina te variraju 17 parametara kako bi se postigla velika varijabilnost unutar klasa. Zbog praktičnijeg variranja nijansi boja, RGB prostor tekstura transformiraju u HSV. Unatoč relativno velikom

broju (100.000) sintetičkih slika u niskoj rezoluciji (30x30 px), generiranih za trening, postižu značajno lošije rezultate kod korištenja LDA i SVM klasifikatora (43,6% i 79,01%) u odnosu na trening korištenjem isključivo realnih slika (93,28% i 95,7%) što objašnjavaju neodgovarajućom linearnom prirodom ta dva modela koja nije dostatna za obuhvatiti visoku varijabilnost sintseta. No, kada povećaju sintset na 650.000 slika, te prije klasifikacije obave segmentaciju (koristeći HOG), tako izuzmu pozadinu i smanje kompleksnost podataka, rezultat sintseta se se znatno popravlja (83,22% i 91%). k-NN postiže bolje metode na sintsetu (93,15%) nego na realnom (72,81%), bez obzira na veličinu skupa za treniranje, a segmentacija pritom korištena na sintsetu dodatno poboljšava rezultat (96,91%). Ukupno najbolje rezultate postiže korištenje CNN-a na sintsetu već s 100.000 slika (97,87%), dok su na realnom nešto lošiji (96,3%) i u rangu onih kad se koristi k-NN na segmentiranom sintsetu sa 650.000 slika.

Za potrebe fluoroskopije Heimann i sur. generiraju sintset volumetrijskih slika ultrazvučnih pretvornika (eng. *transducer*) [38]. Koristeći manji broj slika fluoroskopskih sekvenci (bez pretvornika) kao pozadinu, ucrtavaju na njih pretvornik (u obliku cijevi zadebljane na kraju) koristeći rendomiziranu 3D krivulju (eng. *spline*) odgovarajuće debljine. S obzirom da je riječ o sekvenci volumetrijskih slika, ključno za postizanje realističnosti je korištenje različitog stupnja transparencije po slojevima kroz koje krivulja prolazi. Adaptacijom domene, koristeći 8 puta veći i savršeno anotirani sintset u odnosu na realni, postigli su trostruko smanjenje broja pogrešnih detekcija.

2014

Handa i sur. kreiraju temelj za vrednovanje (eng. *benchmark*) za RGB-D vizualnu odometriju, 3D rekonstrukciju scene i SLAM (eng. *Simultaneous Localization And Mapping*) u interijeru [63]. Kod kreiranja sintseta, koji se sastoji od snimki putanje kroz dvije različite scene (dnevni boravak i ured), ciljajući fotorealističnost, posebnu pažnju posvećuju svjetlosnim fenomenima prisutnim na realnim slikama: zrcalne refleksije, sjene i curenje boje (eng. *color bleeding*). Uredska scena je u potpunosti kreirana proceduralno, a nedostatak toga, zbog korištenog alata (POV-Ray), je nemogućnost korištenja poligonalnog modela za evaluaciju rekonstrukcije površina. Šum, osim u RGB dodaju i u mapu dubine, ali izostavljaju zamućenje pokreta.

Vázquez i sur. [64] nastavljaju Marínov rad [53] i uočavaju da, unatoč potencijalnoj visokoj vizualnoj sličnosti sintetičkog i realnog seta, već promjena osobina kamere dovodi do problema pomaka dataseta (eng. *dataset shift*). Kako bi ga anulirali, razvijaju V-AYLA integrirani okvir (eng. *framework*), koji za adaptaciju domene koristi 10 puta manje podataka iz realnog seta u odnosu na sintset, i s njim postižu jednako dobre rezultate detekcije kao kad se koristi ručno anotirani realni set za učenje i testiranje.

Xu i sur. [65] također nastavljaju rad [53]. Za generiranje svog sintseta koriste prednost tekstura u višim rezolucijama i veću varijabilnost objekata (automobili, zgrade, drveće i pješaci). Koriste i adaptaciju domene, a za detekciju primijenjuju pristup mješavine dijelova (eng. *mixture-of-parts*) kombiniran s modelom deformabilnih dijelova (eng. *deformable*

part-based model, DPM) s kojim nadmašuju dotadašnji najbolji rezultat postignut latentnom SVM metodom.

Rozantsev, Lepetit i Fua uvode algoritamsku estimaciju parametara za sjenčanje prema manjem uzorku realnih slika [66]. Kombinacijom tih parametara, koristeći grubi 3D model objekta kojeg detektiraju, položen na 2D pozadinu, sintetiziraju željeni broj slika za učenje. Njihova ključna pretpostavka je da sintetizirane slike moraju biti što sličnije realnima, ali ne nužno u smislu realističnosti već u smislu značajki koje sadrže, a na koje se oslanja korištena metoda detekcije. Potvrđuju da sintset značajno popravlja performanse modela u odnosu na treniranje isključivo s realnim slikama, ali ukazuju na postojanje točke u kojoj preveliki broj sintetičkih slika počinje djelovati negativno. Prema njihovom iskustvu, ta točka ovisi o metodi detekcije pa tako za AdaBoost preporučuju koristiti 100, za DPM 50, a za CNN 15-20 sintetičkih slika za svaku realnu.

Aubry i sur. renderiraju 1.393 detaljnih 3D modela stolica na bijeloj pozadini iz 62 različita kuta i tako grade sintset od 86.366 slika [67]. Kombiniranjem modela diskriminacijskih dijelova (eng. *part-based discriminative* model) i podudaranja na temelju primjera (eng. *exemplar-based matching*) problem detekcije kategorije svode na problem 2D-3D poravnavanja (eng. *alignment*), te koriste sintset za naučiti raspoznavati tip, položaj i orijentaciju stolice na danoj slici.

Courty i sur. grade Agoraset, sintset realistično renderiranih virtualnih ljudi promatranih s 64 kamere, koji se, tretirani kao čestice (eng. *particles*), kreću po realističnim dinamičnim putanjama kroz 8 različitih scena [68]. Set je namijenjen praćenju i segmentaciji u video analizi mnoštva. S obzirom da koriste između 200 i 2.000 avatara pješaka, novost koju uvode je pohrana identifikacijske oznake svakog pješaka u 16-bitnu segmentacijsku masku sivih tonova. Naglašavaju potrebu za realizmom sintseta kako bi se naučeno moglo transferirati u realne situacije. Manjak Agoraseta je ograničenost varijacija u geometriji kroz koju se kreću pješaci, mali broj korištenih tekstura te vremensko rastezanje (eng. *time-stretch*) *mo-cap* animacije hodanja, kako bi se dotična prilagodila brzini svakog pojedinog pješaka, što dovodi do nerealistične dinamike kretanja.

Sun i Saenko osporavaju prethodne zaključke o nužnosti fotorealizma u domeni detekcije objekata [69]. Gradeći paralelno dva sintseta, prvi s realističnim renderima 3D objekata na slučajno odabranoj 2D podlozi iz ImageNet dataseta, a drugi sa renderima sivih tonova na bijeloj pozadini, dolaze do jednako uspješnih rezultata detekcije koristeći pristup brze adaptacije baziran na dekoreliranim značajkama (whitened HOG, WHO). Metoda je uspješna jer odbacuje "pozadinsku statistiku" i zadržava samo oblik i teksturu karakterističnu za cijelu kategoriju, ali ne i za pojedini objekt.

2015

Nakon što je estimacija optičkog toka formulirana kao potencijalni zadatak za nadzirano učenje koji je moguće uspješno riješiti koristeći konvolucijske mreže, uočena je slabost MPI-Sintel [57] sintseta: premala količina podataka (1.628 stereo slika). Kako bi omogućili treniranje CNN, Mayer i sur., koristeći prilagođenu verziju Blendera, paralelno kreiraju 3

sintseta (FlyingThings3D, Monkaa i Drivings) s ukupno preko 35.000 stereo slika [70]. Za razliku od MPI-Sintel, njihov sintset, kao dio GT, sadrži mape promjene dispariteta (eng. *disparity change*) i granica pokreta (eng. *motion boundaries*), a u naknadnu obradu uvode odsjaj sunca (eng. *sunlight glare*) i manipulaciju gama krivulje (eng. *gamma curve*). Za razliku od prethodnih sintsetova koji su pohranjivani na disk u standardnim formatima zapisa slike (PNG, JPG), zbog velike količine generiranih podataka (2,5 TB), autori zapisuju sve RGB slike u WebP4 formatu (s gubitkom informacija, eng. *lossy*), a ne-RGB u LZO5 (bez gubitka, eng. *lossless*). Učinkovitost kreiranog sintseta dokazuju treniranjem FlowNet CNN kojim postižu trenutno najbolje rezultate i to čak 1.000 puta brže od prethodnika.

Istim problemom (estimacija optičkog toka korištenjem CNN) bave se Fischer i sur. [71], ali za razliku od [70] koriste nerealistične rendere i potvrđuju da s njima učeni CNN modeli mogu dobro generalizirati. Time ujedno potvrđuju i da zaključak o uspješnom korištenju nerealističnih sintsetova na problemima detekcije [69] vrijedi i u domeni optičkog toka. Njihov sintset imenovan je Flying Chairs i sastoji se od 22.872 parova slika i polja toka. Građen je kombiniranjem 964 slika s Flickera, razrezanih u 4 kvadranta, kao pozadina, i različitog broja stolica iz seta renderiranih 3D modela stolica iz [67], u prednjem planu. S obzirom na uspješno testiranje modela učenog na Sintel sintsetu koji uopće ne sadrži stolice, autori naglašavaju da dobar sintset u domeni optičkog toka ne mora biti treniran na podacima koji semantički odgovaraju onima za test.

Rivera-Rubio, Alexiou i Bharath dopunjuju realni RSM dataset za lokalizaciju u interijeru sa 7 sekvenci prolaza kroz sintetičke hodnike izrađene u Unityu. Pritom, kao automatski kreirani GT, uvode mjeru udaljenosti od početne točke [72].

Chen i sur. koriste 12 sati vožnje snimljene unutar The Open Racing Car Simulatora (TORCS) od čega za treniranje CNN modela od nule koriste 484.815 slika u rezoluciji 280x210 px s pripadnim anotacijama [73]. Model testiraju koristeći video snimljen kamerom pametnog telefona i, bez obzira na dvije različite domene, postižu dobre performanse, pogotovo u modulu za percepciju prometne trake.

Hattori i sur. u domeni videonadzora grade masivni (2.5M slika) sintset kojemu je cilj podesiti sustav nadzora za novu lokaciju koristeći poznati razmještaj geometrije, virtualno rekonstruiran, i 36 različitih modela virtualnih pješaka sa po 3 moguće konfiguracije hoda [74]. Posebnu pozornost obraćaju na perspektivnu distorziju kamere s obzirom da je prilikom korištenja kamera sa širokim vidnim poljem i malom žarišnom duljinom potrebno naučiti raspoznavati distorzirane slike ljudi (npr. nagnuti pješaci s velikim glavama). Koristeći isključivo sintetičke podatke za učenje i vlastitu metodu detekcije (SLSV) postižu najbolji rezultat (u odnosu na korištenje HOG+SVM i DPM metoda, u kombinaciji sintetičkih i realnih podataka za učenje).

Veeravasarapu i sur. analiziraju utjecaj različitih faktora prilikom izgradnje virtualnih scena (geometrija, izgled, osvjetljenje, fizika, okruženje, kamera, parametri rendera i dr.) na pojedine značajke [75]. Zaključuju da utjecaj fotorealističnosti sintseta na performanse modela (za testiranje na realnim podacima) ovisi primarno o bliskosti distribucija između sintetičkih i realnih podataka. Za provedbu analize grade vlastiti sintset u Blenderu, a s obzirom da uvode kišu, koja može značajno utjecati na izgled dva susjedna frejma zbog

efekta zamućenja pojedinih dijelova slike, napominju da njena implementacija nije fizikalno ispravna te da ne utječe na promjenu vizualnih svojstava površina koje su u dodiru s njom.

Su i sur. [39] grade veliki sintset (preko 2.4M slika u 12 klasa) kojeg kombiniraju s realnim (12K slika), za potrebe određivanja gledišta na 2D slikama koristeći CNN. Sintset nastaje polaganjem renderiranih ShapeNet 3D modela na 2D pozadine iz SUN397 baze, koristeći pritom *alpha blending* kojemu je svrha spriječiti klasifikatore da nauče nerealistične uzorke na spojevima renderiranih 3D objekata s pozadinom.

Papon i Schoeler koriste CNN za razumijevanje scene. Zbog manjka odgovarajućeg RGB-D trening seta s anotiranim pozama objekata, kreiraju protočnu strukturu (eng. *pipeline*) renderinga "u letu" (eng. *on-the-fly*) koji generira interijere realistično popunjene objektima, paralelno s treningom [76]. Po završetku treninga na sintsetu provode transfer učenja (eng. *transfer learning*) iz relativno malog anotiranog realnog seta zahvaljujući čemu njihov model uspješno anotira realne scene. Generiranje sintetičkih scena vrši se u Blenderu, koristeći CPU i sekundarni GPU, sekvencijalnim smještanjem objekata u virtualnu prostoriju, na slučajna mjesta, pazeći pritom da se ne preklapaju. Primarni GPU koristi se za paralelni trening. Zbog manjka tekstura i jednostavnog modela rasvjete rendering nije realističan, ali s obzirom da se koristi samo informacija o intenzitetu pojedinih piksela u kombinaciji s mapom dubine, dostatan je za učenje modela.

Peng i sur. uočavaju da većini slobodno dostupnih 3D objekata često nedostaju realistične teksture, odgovarajuće poze i pozadine [77]. Kako bi testirali učinak spomenutih faktora na kvalitetu CNN detektora grade vlastiti sintset i dokazuju da ukoliko se CNN uči na sintsetu ugođenom za ciljani zadatak, pokazat će visok stupanj invarijantnosti u odnosu na spomenute faktore, ali ako je prethodno treniran za klasifikaciju na, primjerice, ImageNet setu, bolje uči kada se spomenuti faktori eksplicitno stimuliraju.

2016

Handa i sur. uočavaju da, za potrebe razumijevanja scene, postojeći anotirani setovi ograničavaju svoj fokus na 3D objekte umjesto na scene odnosno na pojedine slike umjesto na video sekvence. Zato grade vlastiti generator, SceneNet [78], koji analizom statistike povezanih pojavljivanja i prostornih odnosa objekata u prostoru na postojećim anotiranim setovima, uči, a potom i generira neograničeni broj potencijalno povezanih i smisleno popunjenih virtualnih interijera. Pritom koriste veliki broj pojedinačnih, ali i grupiranih 3D modela, te automatsko uzorkovanje fotografskih tekstura iz postojećih baza. Za renderiranje koriste Blender, ali ne forsiraju fotorealizam u korist veće količine i varijacija sintetiziranih slika (zbog bržeg renderinga). Šum koji dodaju mapi dubine ovisan je, osim o dubini, i o kutu gledanja.

Richardson, Sela i Kimmel koriste morfabilni 3D model lica za generiranje relativno fotorealističnog sintseta s kojim uče CNN baziran na ResNetu rekonstruirati teksturirani 3D model lica iz samo jedne fotografije [79]. Relativna fotorealističnost postignuta je korištenjem Phong modela sjenčanja koji zanemaruje fizička svojstva kože poput podpovršinskog raspršivanja svjetla (eng. *subsurface scattering*, SSS), ali uspješnost

postignute rekonstrukcije dokazuje da ni u ovom slučaju fotorealističnost nije presudna za kvalitetu CNN modela.

Mueller, Smith i Ghanem [80] grade hibridni UAV123 set kao *benchmark* za praćenje bespilotnih letjelica (eng. *unmanned aerial vehicles*, UAV) u niskom letu. Sastoji se od 123 sekvence i 112.578 slika. Od toga je 8 sekvenci (u različitim virtualnim okruženjima) sintetizirano koristeći Unreal Engine 4 (UE4), bez ikakvih efekata u naknadnoj obradi. Simulacija unutar UE4, koju koriste za generiranje sintetičkih slika, ujedno se može koristiti i za *on-line* evaluaciju različitih algoritama za praćenje.

Honauer i sur. uvode sintsetove kao *benchmark* u domeni analize svjetlosnih polja [34]. Njihov 4D Light Fields set sadrži 24 sintetički dizajnirana i gusto uzorkovana svjetlosna polja primarno difuzno osvjetljenih scena (koristeći Lambert model sjenčanja) i pripadni GT (disparitet) visoke točnosti.

Lin i sur. kreiraju simulator za istraživanje interakcije čovjeka i scene u virtualnoj stvarnosti (eng. *virtual reality*, VR) [41]. Za razliku od prethodnika ([20], [76], [78]) koji za potrebe razumijevanja scene grade virtualne interijere koristeći objekte kao nepomične cjeline, Lin i sur. prvo razlažu gotove 3D objekte (npr. kuhinjski elementi) na njihove anotirane pomične komponente (npr. vrata kuhinjskih ormarića, ladice...) te koriste UE4 fiziku za simuliranje dinamike krutih objekata (eng. *rigid objects*) u interakciji s čovjekom. Pritom koriste Oculus Rift sa Kinect senzorom, podlogu za ples (eng. *dancing pad*) za kretanje i Leap Motion za prikupljanje finih pokreta prstiju i manipulaciju objektima u virtualnom okruženju. Ovaj pristup, dinamičkog kreiranja dataseta, može se koristiti za planiranje zadataka robota i semantičku segmentaciju slika.

Johnson i sur. koriste slučajno uzorkovanje predefiniranog grafa scene (eng. *scene graph*) za postavljanje jednostavnih geometrijskih oblika različitih veličina, materijala i boja na različite pozicije, bez preklapanja, tako da budu vidljivi kameri i pazeći da obavezno postoje male horizontalne i vertikalne margine između centara svih parova objekata na slici kako bi se izbjegla dvosmislenost prostornih relacija [31]. Njihov sintset, CLEVR, sadrži 100K slika i namijenjen je vizualnom rasuđivanju (eng. *visual question answering*, VQA) koristeći CNN i LSTM.

Massa, Russell i Aubry [81] nastavljaju razvoj metode za detekciju prema primjeru (eng. *exemplar-based detection*) [67]. Svoj sintset kreiraju preklapajući renderirane teksturirane 3D modele preko foto pozadina, što im omogućava naučiti CNN kako zanemariti pozadinu. Pritom, za razliku od [77], gdje se preferirao realizam komponiranja (eng. *compositing*), ne koriste anotacije objekata za odabrati odgovarajuće pozadinske slike već ih biraju nasumično. Na taj način adaptaciju vrše u smjeru od realnih prema renderiranim slikama smatrajući da je teže halucinirati značajke koje odgovaraju nedostajućim dijelovima slike (poput okruženja objekta ili njegove teksture) nego ih ukloniti. Kako bi izbjegli artefakte boje u komponiranim slikama, koriste isključivo sive tonove. Vezano uz 3D modele, zaključuju da korištenje baze objekata iz različitih kategorija, u odnosu na isključivo one koje ciljaju detektirati, daje bolje rezultate i pokazuju da se unaprijed izračunate značajke renderiranih pogleda na objekte mogu dodati kao potpuno spojeni sloj u CNN, čime se povećavaju točnost i brzina detekcije.

Johnson-Roberson i sur. za izgraditi sintset GTAVision koriste komercijalnu računalnu igru Grand Theft Auto V (GTA V) čiji izdavači dopuštaju korištenje snimki iz igre u nekomercijalne svrhe [82]. Autori nemaju mogućnost prilagodbe igre vlastitim potrebama pa iz nje ekstrahiraju samo dostupne podatke među kojima je i segmentacijska mapa na kojoj su svi objekti istog tipa (npr. automobili) jednako označeni. Kako bi iz takve segmentacijske mape izvukli precizne granične okvire kao anotacije pojedinih automobila, koriste diskontinuitet prisutan u dubinskoj mapi. Prilikom testiranja na realnim podacima uočavaju da između učenja Faster-RCNN s 10K i 50K sintetičkih slika nastaje značajan kvalitativni skok odnosno da postoji prag iznad kojega mreža uči puno bolje razlikovati modele pojedinih automobila. Točnost modela je nastavila, iako puno sporije, rasti do 200K sintetičkih slika, a nakon toga se počela smanjivati, što pripisuju pretreniranosti mreže sa redundantnim podacima. To ukazuje na zaključak da isključivo povećanje količine podataka, ali ne i njihove varijabilnosti, dovodi do degradacije modela. Autori predlažu istražiti mogućnost generiranja manjeg, ali sadržajno kompleksnijeg sintseta kako bi se istovremeno smanjilo vrijeme potrebno za treniranje mreže i povećao utjecaj pojedinih slika na učenje.

Movshovitz-Attias, Kanade i Sheikh analiziraju utjecaj fotorealizma na kvalitetu sintseta [8]. Za tu potrebu grade dva sintseta renderiranih automobila: RenderCar (819.000 slika renderiranih 3D modela naknadno položenih na slučajno odabrane pozadine) za učenje, i RenderScene (1.800 slika rendera 3D modela na pripadnoj 3D sceni) za validaciju CNN modela detekcije. Varirajući kompleksnost materijala i rasvjete zaključuju da kompleksni materijali i rasvjeta, jedinstvenost 3D scene (u odnosu na kombinaciju 3D modela i 2D pozadine) kao i sama postavke kvalitete renderera, pridonose kvaliteti sintseta jer uzimaju u obzir elemente interakcije 3D modela sa scenom kao što sjene i refleksije. Praktični nedostatak tog pristupa je zahtjevnost resursa za renderiranje što još uvijek ograničava veličinu tako kreiranih sintsetova i čini ih neupotrebljivima za trening. No, dodavanje čak i male količine fotorealističnog sintseta realnom daje bolje rezultate od korištenja kombinacije različitih realnih setova za učenje jer može uvesti značajke koje nisu prisutne u realnom setu i tako potaknuti model da bolje generalizira. Problem količinskog praga na koji ukazuju [82] predlažu riješiti većim brojem 3D modela, ali ne uzimaju u obzir mogućnost njihovog proceduralnog kreiranja. Eksperimentirajući s okluzijama zaključuju da ne nalaze značajnu razliku bez obzira kakav oblik okluzije se koristi (od jednobojnih kvadrata do zakrpi fotografskim teksturama različitih oblika) već da učinak okluzije ovisi isključivo o klasi objekta. Naglašavaju i važnost usklađivanja sintseta za treniranje sa statistikom kutova pod kojim su snimljeni objekti u realnom setu. U postupak pripreme sintseta uvode varijabilnu brzinu zatvarača kamere (eng. *shutter speed*) i vinjetu. Slike inicijalno snimaju u PNG formatu, ali ih naknadno komprimiraju u JPG jer, iako kompresija nije vizualno uočljiva, uočeno je da utječe na performanse klasifikatora. Njihov CNN koristi weighted SoftMax (wSM) funkciju gubitka koja ne dopušta modelu nasumično "pogađanje" klase već zahtjeva da slični pogledi na 3D objekt imaju slične vjerojatnosti, čime postižu stabilniju predikciju.

Koristeći Menge integrirani okvir kao simulatora mnoštva i UE4 za renderiranje, Cheung i sur. grade generator masivnog sintseta (1M video sekvenci s ukupno 20M slika) uvodeći u automatsko anotiranje putanje, broj pješaka i informaciju o ponašanju mnoštva u pojedinom frejmu [83]. Za razliku od [68], omogućavaju korištenje kompleksne geometrije u

ulozi statičnih i dinamičkih prepreka. Realizam povećavaju kombinirajući sintetički generirano mnoštvo s realnim video snimkama.

Wood i sur. za potrebe određivanja smjera gledanja (eng. *gaze estimation*) generiraju 1M sintetičkih slika očiju koristeći Unity i proceduralno animiran morfabilan model oka [40]. Model oka kreiran je 3D skeniranjem područja oka u visokoj rezoluciji (5M točaka) i naknadnim retopologiziranjem sveden na 229 verteksa. S obzirom da u vrijeme nastanka njihovog rada praćenje zrake (eng. *ray tracing*) nije bilo dostupno u Unityu, a realističnost prikaza očne jabučice ovisi o korištenju refrakcije koju nije moguće postići standardnom rasterizacijom, autori simuliraju fizikalno ispravan efekt refrakcije koristeći program za sjenčanje fragmenata (eng. *fragment shader*, GPU program koji procesira svaki piksel tijekom rasterizacije). Scenu osvjetljavaju rasvjetom baziranom na slici s visokim rasponom boja (eng. *high dynamic range*, HDR).

Lerer, Grossi Fergus dodaju Torch integrirani okvir za strojno učenje u UE4, u glavnu petlju igre (eng. *game loop*), kako bi omogućili njihovu *on-line* interakciju [84]. Kreiraju simulaciju padajućih blokova za potrebe vizualnog rasuđivanja i na njima istražuju utjecaj okluzije na predikciju koristeći DeepMask model. Zaključuju da model uspješno radi predikcije na bazi relevantnih lokalnih značajki slike koje ukazuju na fizikalnu nestabilnost geometrije, a ne pamteći određene scene.

Veeravasarapu, Rothkopf i Ramesh nastavljaju istraživati utjecaj fotorealističnosti [85]. Koristeći stohastičko generiranje 3D scene grade svoj sintset. Tijekom renderiranja u Blenderu, koristeći Monte Carlo metodu, variraju broj uzoraka po pikselu i zaključuju da je treniranje različitih CNN arhitektura invarijantno na broj uzoraka nakon 40 uzoraka po pikselu čak i kada rezultirajuća slika sadrži vidljive ostatke šuma. Analizirajući performanse mreže na različitim dijelovima slike uočavaju da su najproblematičnije granice 3D objekata na kojima, u odnosu na realne slike, izostaju očekivani efekti (curenje boje, penumbra). Stoga predlažu modeliranje odgovarajućih efekata senzora i leće, a s tim ciljem u naknadnu obradu uvode i efekt kromatske aberacije (eng. *chromatic aberration*).

Shafaei, Little i Schmidt grade sintset u domeni autonomne vožnje na, zbog pravnih pitanja, neimenovanoj računalnoj igri [86] i, potaknuti ograničenjima koje postavlja korištenje komercijalnih igara, predlažu suradnju s proizvođačima igara kako bi se putem pristupačnijeg sučelja (eng. *interface*) ubuduće omogućilo bolje korištenje takvih resursa u domeni računalnog vida.

Richter i sur. [87] su problemu na koji ukazuju Shafaei i sur. [86] doskočili koristeći tehniku poznatu kao "zaobilaženje" (eng. *detouring*). Ubrizgavaju omotač (eng. *wrapper*) između operativnog sustava i igre, koji im omogućava snimanje, mijenjanje i reprodukciju naredbi za renderiranje. Na taj način označavaju različite resurse u procesu renderiranja (geometriju, teksture, programe za sjenčanje) i prate ih iz frejma u frejm. Koristeći rudarenje pravila (eng. *rule mining*), koje predlaže povezivanje tako pribavljenih oznaka sa sadržajem renderiranih slika, i ljudskog anotatora koji ih validira i potvrđuje, anotiraju prosječnom brzinom od 7 sekundi po slici. Autori kao neke od ključnih prednosti korištenja računalnih igara za ekstrahiranje sintsetova navode prirodnost rasporeda objekata na scenama, realistične

teksture, realistično kretanje vozila i likova te prisutnost sitnih objekata koji obogaćuju kadar detaljima i tako pridonose dojmu realističnosti.

Mahendran i sur., modificiraju stariju računalnu igru (Doom) s kojom generiraju CocoDoom, sintset od 500K sintetičkih slika i pripadni GT za raspoznavanje, detekciju i praćenje objekata, segmentaciju, monokularnu procjenu dubine i procjenu ego-kretanja (eng. *ego-motion*). GT uključuje i dnevnik događaja (eng. *log of events*) s kojim je moguće rekonstruirati tijek igre [88]. Sličan pristup koriste Kempka i sur. [89], ali ne grade sintset fiksne veličine već omogućavaju generiranje "u letu" slika za podržano učenje, poput [26], pri čemu generiraju do 7.000 slika u sekundi umjesto 60 slika u sekundi, u realnom vremenu izvođenja igre. Pritom nalaze da efektivni bot ne mora vidjeti svaku sliku već da se, zbog brzine učenja, isplati preskakati između 4-10 slika.

S ciljem istovremene predikcije volumetrijskog zauzeća scene i kategorije objekta iz samo jedne dubinske slike, Song i sur., koristeći Planner5D, grade SUNCG, sintset s 45.622 manualno dizajniranih kompleksnih interijera, s realističnim rasporedom namještaja, renderiranih u 130.269 slika [90]. Kao i [20], koriste vokselizaciju, ali, kako bi ubrzali proces, zasebno i jednokratno vokseliziraju svaki pojedini objekt koji se koristi za slaganje scene. S obzirom da njihov SSCNet model koristi isključivo informacije o dubini, bez boje, objekti poput prozora predstavljaju mu problem, kao i objekti sa sličnom geometrijom, ali različitom funkcijom.

Chen i sur. nalaze da je za učinkovito određivanje poze, uz dostatnu količinu podataka za treniranje neuronske mreže, ključni sastojak različitost korištenih tekstura odjeće [91]. Koristeći morfabilni model (SCAPE), generiraju 10.556 različitih ljudskih 3D modela. Potom prikupljaju veliki broj slika sportske odjeće na jednostavnoj pozadini koja im olakšava segmentaciju. Iz takvih slika segmentiraju po 1.000 različitih odjevnih predmeta za gornji i donji dio tijela koje koriste kao teksture u kombinaciji s prethodno pripremljenim setom tekstura glava, ruku, cipela i kože čije boje dodatno rendomiziraju. Na 3D modele apliciraju poze iz CMU MoCap dataseta i renderiraju na nekoj od 795 realnih 2D pozadina. Tako generiraju ukupno 5.099.405 slika za treniranje, a selekciju od 1.574 uvrštavaju u Human3D+ sintset.

Ros i sur. grade SYNTHIA sintset namijenjen segmentaciji u domeni autonomne vožnje [92] i s njim uvode simulaciju godišnjih doba koja drastično mijenjaju izgled slike. Na taj način povećavaju realizam s obzirom da je riječ o scenama u eksterijeru. Za učenje CNN modela kombiniraju realne i sintetičke slike pri čemu koriste Balanced Gradient Contribution (BGC) koji gradi serije podataka (eng. *batch*) iz obje domene u predefiniranom omjeru i na taj način koristi sintetičke slike za sofisticiranu regularizaciju. Iako su slike renderirane u 960x720 px, za trening se koriste u rezoluciji 180x120 px kako bi se uštedjela memorija i ubrzao trening. Negativna posljedica toga je gubitak prepoznatljivosti manjih objekata poput prometnih znakova i stupova uz prometnice. S obzirom na način nastanka i automatsku anotaciju, osnovni nedostatak ovog sintseta, na koji ukazuju i autori [93], je nemogućnost korištenja za druge namjene u domeni računalnog vida (poput praćenja i detekcije objekata).

Tabernik i sur. smatraju da CNN-ovima nedostaje eksplicitna struktura u značajkama što često dovodi do pretreniranosti, nemogućnosti rekonstrukcije iz djelomičnih opservacija (npr. prilikom okluzije) i ograničenih generativnih sposobnosti (npr. za potrebe vizualizacije dijelova mreže) [35]. Zato predlažu korištenje hijerarhijskih kompozicijskih modela u kojima je eksplicitna struktura inherentna. Svoju hipotezu testiraju na PacMan sintsetu i postižu ubrzanje zaključivanja modela (eng. *inference*) te jednostavniju vizualizaciju.

Zhang i sur. se fokusiraju na granične slučajeve u domeni stereo dispariteta poput refleksivnosti (eng. *specularity*), manjka teksture, skokova u disparitetu i transparencije [94]. Smatraju da je često teško ili čak nemoguće kreirati odgovarajući realni set za takve potrebe, posebno kada snimanje ovisi o vremenskim uvjetima (primjerice, intenzivne refleksije sunca ili kišni artefakti) na koje nije moguće utjecati. Postojećim sintetičkim setovima zamjeraju zatvorenost u smislu da iako postoje slike i GT, promjenom parametara nije moguće podesiti set vlastitim potrebama jer virtualne scene iz kojih su izvedeni nisu dostupne za modifikaciju. Zato preporučaju gradnju generatora i, koristeći UE4, kreiraju vlastiti s kojim pripremaju UnrealStereo sintset za evaluaciju nabrojenih graničnih slučajeva. Prednost korištenja alata za razvoj igara (eng. *game engine*) vide u postojanju tržnice (eng. *marketplace*) na kojoj se, često i besplatno, mogu pronaći odgovarajuće virtualne scene za generirati potreban dataset, te u mogućnosti generiranja neograničenog broja slika na zahtjev korisnika (eng. *on demand*). U GT uvode mape refleksivnih i transparentnih područja s kojim se problematične situacije mogu automatski detektirati. Za slučajeve manjka teksture preporučaju korištenje DispNet CNN arhitekture koja zbog svojih velikih receptivnih polja uzima u obzir informaciju o kontekstu.

Gaidon i sur. gradeći Virtual KITTI [95] sintset, ispravljaju nedostatak uočen u [92], dodajući GT za detekciju i praćenje objekata onome za segmentaciju. Mišljenja su da prethodni sintsetovi u domeni autonomne vožnje nisu dovoljno detaljni (od rasporeda objekata do toga da ne sadrže tablice automobila) te prilikom konstrukcije 3D scena, sadržaj prilagođavaju referentnom video setu (KITTI) kako bi minimizirali jaz (eng. *gap*) između sintetičkog i realnog seta. Prednost sintsetova vide u mogućnosti provedbe analize utjecaja pojedinog faktora (*lat. ceteris paribus*) i što-ako (eng. *what-if*) analize, pogotovo za rijetka događanja. U GT uvode vidljivost pojedinog objekta u magli.

Bochinski, Eiselein i Sikora treniraju detektor objekata za potrebe video nadzora koristeći vlastiti MOCAT sintset [96]. Generiraju ga unutar *sandbox* igre Garry's Mod koja im omogućava kreiranje različitih scenarija kojima potiču na kretanje vozila, ljude, ali i životinje. Koriste promjenu doba dana i uočavaju da scene u sumrak rezultiraju značajno većim brojem lažnih pozitivnih detekcija što sugerira da postoji prag potrebnog osvjetljenja ispod kojeg algoritmi za uklanjanje pozadine, a posljedično i CNN, postaju neučinkoviti.

Zhang i sur. koriste modificirane SUNCG [90] 3D scene za izgraditi svoj MLT sintset [97]. Renderiraju 779.342 slika od kojih koriste 568.793 koje po svojoj statistici distribucije boje i dubine odgovaraju onima u realnom NYUv2 datasetu. Istražuju utjecaj realizma na scene interijera pa renderiraju koristeći kombinaciju Open GL renderera i fizikalno baziranog Mitsuba renderera sa Path Space Metropolis Light Transport (MLT) integratorom te različite tipove unutarnje i vanjske rasvjete. Zaključuju da povećani realizam značajno utječe na kvalitetu predikcije.

Kako bi olakšali korištenje podataka iz UE4 u domeni računalnog vida, Qiu i Yuille pokreću projekt UnrealCV [98] u sklopu kojeg izrađuju UE4 dodatak (eng. *plug-in*) putem kojega je moguće eksterno pristupiti aplikacijama izrađenim u UE4 odnosno proslijediti njihov izlaz u neki integrirani okvir za duboko učenje poput Caffea. Na taj način istraživači mogu imati pristup GT, kontrolirati agente i sl., bez poznavanja UE4. Autori navode i neke još neriješene probleme virtualnih svjetova značajne za generiranja sintsetova: ograničenost varijabilnosti 3D sadržaja u njima, manjak interne strukture 3D geometrije i postizanje vjerne fizikalne simulacije.

2017

McCormac i sur. generiraju masivni (5M slika) fotorealistični SceneNet RGB-D sintset za razumijevanje scene, kombinirajući nasumično uzorkovane rasporede scena iz SceneNeta i objekte iz ShapeNeta [99]. Umjesto realistične konfiguracije objekata na sceni kakvu koriste u [90], autori upotrebljavaju fizikalni engine (Chrono Engine) kojim simuliranju ispuštanje objekata na scenu kako bi se, u padu, samostalno rasporedili u fizikalno mogućim kombinacijama. Za renderiranje, u rezoluciji 320x240 px, koristeći GPU podržani softver za praćenje zrake Opposite Renderer, na 4-12 GPU-a, trebalo im je mjesec dana (prosječno 3 sekunde po slici). Uz praćenje zrake koristili su i mapiranje fotona (eng. *photon mapping*) za globalnu iluminaciju čime se aproksimira indirektno osvjetljenje, curenje boja i kaustičnost (eng. *caustics*). Iako su testirali brzinu renderinga u odnosu na broj uzoraka po pikselu i broj mapa fotona (i u konačnici se opredijelili za 16 uzoraka i 4 mape fotona), nisu testirali kako se rezultirajuće sekvence ponašaju na učenje CNN-a, što bi bila vrijedna informacija za optimalno konfiguriranje renderera. SceneNet RGB-D je moguće koristiti za učenje CNN-a od nule i tako postići bolje rezultate u odnosu na učenje sa ImageNet slikama, što ukazuje na značaj kombinacije veličine i fotorealističnosti seta.

Za razliku od prethodnika koji prakticiraju vizualno podržano učenje (eng. *reinforced learning*, RL) na retro igrama ([26] i [89]), Zhu i sur. koriste AI2-THOR [100] simulator baziran na fotorealističnim 3D scenama interijera [101]. Simulator je kreiran u Unityu, integriran s TensorFlowom i omogućava izravnu komunikaciju podsustava za fiziku (eng. *physics engine*) i integriranog okvira za duboko učenje. Sadrži 120 scena u 4 kategorije (kuhinja, dnevni boravak, spavaća soba i kupaonica) koje su, kao i u [90], manualno dizajnirane prema referentnim fotografijama. Autori simulatora nedostatkom smatraju manjak detalja na 3D modelima u odnosu na objekte u realnom svijetu. Zhu i sur. navode da je njihovom ciljem vođenom (eng. *target-driven*) navigacijskom modelu za značajno nadmašiti standardne RL modele bilo potrebno oko 100M sintetičkih slika za trening.

Carlucci, Russo i Caputo smatraju da je perceptualni potpis dviju različitih vrsta slika (RGB i D) vrlo različit, pri čemu prve karakteriziraju teksture, a druge siluete objekata [37]. Kako bi istražili primjenu isključivo potonjih za duboko učenje, renderiraju veliki sintset (4.1M) dubinskih slika koristeći pritom slučajno skaliranje 3D objekata po različitim osima, s ciljem postizanja veće varijabilnosti geometrije. Vizualizacijom filtera u prvom sloju odabrane arhitekture (CaffeNet) potvrđuju da se značajno razlikuju od onih koje ista neuronska mreža nauči koristeći ImageNet RGB slike.

Mitash, Bekris i Boularias koriste vlastiti sintset kako bi robotu omogućili trajno učenje iz budućih, automatski anotiranih realnih slika većeg broja različitih pogleda na objekt [102]. To rade tako što prvo uče detektor (Faster-RCNN) označavati granične okvire koristeći sintetizirane slike. Potom na skupovima realnih slika, različitih pogleda na objekt, rade detekciju i uzimaju oblak točaka (eng. *point cloud*) u zoni detekcije. Spajanjem svih oblaka točaka rade 3D segmentaciju (ekstrahiraju objekt iz pozadine) i, koristeći Super4PCS algoritam, rotiraju odgovarajući 3D objekt u skladu s onim na slikama, prilagođavaju mu veličinu, smještaju ga na 3D scenu i transferiraju 3D granični okvir iz takve 3D scene na realne slike.

Veeravasarapu, Rothkopf i Visvanathan uvode generativni suparnički trening (eng. *generative adversarial training*) u izgradnju 3D scena namijenjenih fotorealističnom renderiranju sintsetova [103]. 3D scenu tretiraju kao parametarski generativni model koji variraju u smjeru realnih slika, koristeći GAN. Na taj način mijenjaju raspored objekata na sceni i osvjetljenje. Dinamiku (kretanje vozila i pješaka) zanemaruju, a ne variraju ni intrinzične atribute 3D objekata (oblik i teksturu).

Shah i sur. [21] razvijaju simulator (AirSim) u domeni autonomnog letenja, koristeći gotove scene na koje su ukazali u [94]. U standardni UE4 fizikalni *engine* dodaju magnetizam, a modeliraju i promjene u tlaku i gustoći zraka ovisne o visini letenja.

Dosovitskiy i sur. također koriste UE4 za razvoj simulatora, u domeni autonomne vožnje. Tako nastaje CARLA [104]. Autori uočavaju da se generalizaciju u odnosu na nove vremenske uvjete postiže jednostavnije nego generalizaciju u odnosu na nove gradove. To se može objasniti činjenicom da su CNN model trenirali koristeći slike samo jednog grada (drugi je bio rezerviran za testiranje), ali pri različitim vremenskim uvjetima (što je utjecalo na promjenu osvjetljenja i šum na slikama te doprinijelo većoj varijabilnosti). Ovaj problem je prisutan i u drugim sintsetovima koji preferiraju 3D modele građevina specifičnih arhitekturnih stilova. CARLA u automatsko anotiranje uvodi GPS lokaciju, kompas, brzinu kretanja, vektor akceleracije i akumulirani utjecaj sudara.

Za razliku od [99] koji nasumičnim uzorkovanjem kombiniraju rasporede scena iz SceneNeta i objekte iz ShapeNeta, Jiang i sur. uče gramatiku scene iz SUNCG [90] i ShapeNeta i opisuju je u prostornom i-ili grafu (eng. *Spatial And-Or Graph*, S-AOG) [105]. Uzorkovanjem SAOG-a nastaju različite konfiguracije scena. Autori koriste Mantra PBR renderer i nailaze na problem odabira parametara za sjenčanje s obzirom da im niže postavke kvalitete (manje uzoraka po pikselu) ne omogućavaju sintetizirati slike s kojima bi nadmašili modele prethodnika koji postižu najbolje rezultate. Pri zadovoljavajućim postavkama, vrijeme (CPU) renderiranja za sliku veličine 640x480 px im iznosi 3-5 minuta, ovisno o sadržaju i veličine scene te korištenoj iluminaciji. Kada se uzme u obzir i rezolucija slike (koja je ovdje 4x veća), to je čak 25 puta sporije od GPU metode korištene u [99], što govori u prilog GPU renderinga. Autori u GT uvode anotiranje iluminacije (pozicija, orijentacija, vrsta, intenzitet) te mapu korištenih materijala.

Dok se [19] i [63] bave simuliranjem šuma fizičkih kamera, Planche i sur. fokusiraju se na unapređivanje virtualnih senzora dubine te, u svoju DepthSynth protočnu strukturu [106],

uvode nove vrste šuma: aksijalan i lateralan šum, šum refleksivne površine, šum nerefleksivne površine, strukturalan šum, šum distorzije leće i efekata, šum kvantizacijskog koraka, šum pokreta i brzine zatvarača, te šum sjene.

Tobin i sur. prvi sustavno istražuju rendomizaciju domene [50] i zaključuju da uz dovoljno varijabilnosti u simulatoru, stvaran svijet može modelu izgledati kao samo još jedna varijacija. Koristeći ne-fotorealistični renderer ugrađen u MuJoCo Physics Engine generiraju stotine tisuća slika rendomiziranih jednostavnih geometrijskih objekata kojima variraju oblik, poziciju na sceni, nerealistične teksture, rasvjetu i poziciju kamere. Tako kreiran sintset koriste za treniranje VGG-16 mreže za detekciju objekata. Ukazuju na značajan utjecaj teksture kod korištenja manjih datasetova: performanse seta od 10.000 slika koji koristi samo 1.000 različitih tekstura odgovara performansama seta od samo 1.000 slika koji koristi sve raspoložive teksture. U tom smislu prilikom rendomizacije važnija je rendomizacija tekstura nego pozicija objekata. Autori provode ablacijsku studiju metodologije treninga pri čemu procjenjuju utjecaj sljedećih faktora: broj slika za treniranje, broj unikatnih tekstura, korištenje rendom šuma, prisutnost distraktora (okluzije), rendomizacija pozicije kamere i korištenje različitih prethodno treniranih težina u modelu detekcije. Zaključuju da svi faktori, izuzev rendom šuma, utječu na trening.

Larumbe i sur. kreiraju parametarski simulator poze glave i u njemu generiraju UPNA Synthetic Head Pose Database sintset [107]. U automatsko anotiranje uvode 2D projekciju 3D točaka lica.

Varol i sur. grade SURREAL, masivni (6.5M slika) sintset namijenjen estimaciji poze [108]. Kako bi osigurali realizam, sintetička tijela kreiraju koristeći SMPL model čovjeka čiji parametri su podešeni MoSh metodom prema 3D podacima *mo-cap* markera. Takvim 3D tijelima dodaju teksturu odjeće, stavljaju u poze i renderiraju preko pozadinskih 2D slika. Unatoč velikom broju sintetiziranih slika, autori su zaključili da, zbog velike varijabilnosti sintseta, dobre rezultate mogu postići već s 55K slika. Utvrdili su i da povećanje varijacija odjeće pozitivno utječe na performanse modela, što govori u prilog rendomizaciji domene [50]. Za *mo-cap* preporučuju da bude podudaran s distribucijom ciljanog skupa.

Zimmermann i Brox apliciraju 39 akcija iz biblioteke *mo-cap* animacija u Mixamo animacijskom alatu na 20 različitih virtualnih karaktera i tako grade Rendered Handpose Dataset [109], fokusiran na sintetičke modele ruku, za treniranje CNN-ova, koristeći isključivo RGB podatke. Prilikom odabira 2D slika gradova i pejzaža, koje rendomski koriste za pozadinu, paze da ne sadrže ljude jer njihova prisutnost negativno utječe na učenje modela. Prilikom renderiranja koriste 0 do 2 usmjerena svjetla i globalnu iluminaciju za uskladiti izgled renderiranih modela s pozadinom. Na materijalu kože rendomiziraju zrcalne refleksije. U anotacije uvode ključne točke (eng. *keypoints*) s pripadnim 3D i UV koordinatama na rezultirajućoj slici te indikator vidljivosti svake ključne točke.

Savva i sur. dizajniraju MINOS simulator [110] baziran na SUNCG [90] sintsetu. MINOS podržava multisenzorske (eng. *multisensory*) modele za navigaciju usmjerenu na ciljeve (eng. *goal-directed navigation*) u kompleksnim interijerima. Eksperimenti koji provode pokazuju da aktualni pristupi dubokog podržanog učenja podbacuju u velikim realističnim okruženjima, ali i da je multimodalnost (pri čemu se istovremeno koriste slika, dubina,

normale površina, sile dodira / kontakta i semantička segmentacija) korisna za svladati navigaciju na objektima pretrpanim scenama.

Rajpura, Bojinov i Hegde, na primjeru detekcije pakiranih prehrambenih proizvoda grupiranih u hladnjaku, pokazuju da se, koristeći prijenos učenja, učinkoviti detektor objekata može trenirati gotovo isključivo koristeći nerealistično renderirani sintset [111]. Sa samo 4.000 sintetičkih slika postižu mAP 24 za 55 različitih proizvoda, uz 17 objekata za odvraćanje pažnje (eng. *distractors*). Dodatnih 12% postižu dodavanjem 400 realnih slika. Uočavaju da podešavanje Inception modula mreže pomaže prijenosu učenja, kao i da s povećavanjem broja varijacija 3D objekata od 10 do 200, mAP raste, a nakon toga počinje lagano padati. Prilikom detekcije problem im rade vertikalno naslagani objekti i izrazito kosi kadrovi – u oba slučaja dolazi do lažnih predviđanja.

Richter, Hayder i Koltun kritiziraju tehnike prethodnika za prikupljanje podataka iz komercijalnih igara (bez mogućnosti pristupa kodu): prvenstveno nemogućnost generiranja anotacija brzinom kojom se izvodi igra (u realnom vremenu) i nemogućnost segmentiranja na nivou instanci [112]. Razvijaju pristup koji integrira dinamičko ažuriranje softvera te analizu i prepisivanje međukoda (eng. *bytecode rewriting*), s kojim u anotacije uvode vizualnu odometriju (ego-kretanje). Atmosferskim uvjetima dodaju snijeg i bilježe njegov negativan utjecaj na optički tok (zbog nametnutog kretanja u prednjem planu) i detekciju (jer smanjuje kontrast). Sposobnost detekcije umanjuje i kretanje svjetala automobila (noću), refleksije sunca u leći (u sumrak) te refleksije u lokvama (tijekom kiše). Autori provode statističku analizu svojeg sintseta kako bi utvrdili da se blisko poklapa s fizičkim okruženjem.

De Souza i sur. istražuju proceduralno kreiranje sintetičkih videa s različitim akcijama ljudi u virtualnom svijetu, za potrebe treniranja dubokih mreža [113]. U Unityu definiraju parametarski generativni model akcija koji se oslanja na fiziku, pravila za slaganje scena i tehnike proceduralne animacije poput fizike krpene lutke (eng. *ragdoll physics*). S njim generiraju 6M sintetičkih slika (Procedural Human Action Videos, PHAV dataset) odnosno 39.982 različitih videa s više od 1.000 primjera za svaku od 35 kategorija akcije, od čega 21 baziranu na *mo-cap* podacima i 14 potpuno sintetički proceduralno generiranih. Uz protagonista na scenu postavljaju pozadinske likove i likove s kojima je protagonist u interakciji. Koristeći tehnike proceduralne animacije bazirane na fizici, tijekom nenadziranog generiranja velike količine rendom varijacija, autori nailaze na problem rubnih slučajeva s kojim se fizikalni model unutar alata za razvoj igara ne može nositi, što vizualno rezultira tipičnim greškama vidljivim u računalnim igrama.

S ciljem primjene neuronskih mreža u domeni rekonstrukcije objekata nepoznate kategorije, Schöning i sur. razvijaju Osnabrück Synthetic Scalable Cube Dataset, baziran na vokseliziranim objektima [114] različitih dimenzija (od 3x3x3 do 16x16x16). Za rekonstrukciju koriste realativno plitku neuronsku mrežu sa samo 5 slojeva u koju ulazi 2D slika objekta kojeg je potrebno rekonstruirati, a rezultat je slijed voksela koji čine njegov volumen. Renderiraju ukupno 830.000 rendom generiranih vokseliziranih objekata, svakog iz 12 različitih pogleda, naglašavajući da je to minimalan broj pogleda iz kojih čovjek može riješiti isti zadatak. Najveću točnost uočavaju na rubovima volumena, a najveći, i još

neriješen, problem im predstavljaju zaklonjeni i unutarnji dijelovi objekta. Zaključuju da 2D konvolucija nije nužno odgovarajuća ahitektura za riješiti ovaj zadatak.

Georgakis i sur., poput [111], bave se problemom detekcije objekata u pretrpanim interijerima i grade svoj sintset, ali, umjesto 3D objekata, na fotografije interijera umeću izrezane 2D slike prethodno renderiranih objekata, i to isključivo na plohe na kojima mogu fizički biti smješteni u odgovarajućoj veličini, koristeći u tu svrhu sementičku segmentaciju scene [115].

Dwibedi, Misra i Hebert kritiziraju [115] navodeći da korak semantičke segmentacije ne generalizira dobro na novim scenama. Zato predlažu novi, jednostavniji pristup, kojemu je cilj, za potrebe treniranja detektora, osigurati samo realizam na nivou zakrpe (eng. *patch-level realism*) [116]. Izrezuju objekte iz 2D slika (Big Berkeley Instance Recognition Dataset), koristeći pritom CNN za segmentaciju, te ih lijepe na rendom pozadine (UW Scenes), ne uzimajući pritom u obzir veličinu, rasvjetu ili kompoziciju scene. Ključ njihove metode je u načinu spajanja zalijepljenih objekata s pozadinom (eng. *blending*). Treniraju model koristeći 3 različita moda *blendinga*: bez *blendinga*, Gaussian Blurring i Poisson Blending. Na taj način model postaje invarijantan prema *blendingu* što povećava performanse (AP) za 8%. Uvođenjem augmentacije u obliku okluzije i sakaćenja (eng. *truncation*) rezultat se poboljšava za dodatnih 10%, a dodavanjem distraktor objekata za još 3%.

Müller i sur. smatraju da su gotove igre nepraktične za korištenje kao generator sintsetova zbog izuzetno ograničenih mogućnosti prilagodbe [117]. Prednost daju modernim, potpuno prilagodljivim, alatima za razvoj igara koje ne karakterizira samo fotorealizam već i realistična simulacija fizike čime se jaz između simuliranih i realnih svjetova značajno smanjuje. Koristeći UE4 kreiraju vlastiti simulator (Sim4CV) namijenjen autonomnoj vožnji i letenju. Simulator sadrži velik izbor PBR (eng. *Physically-Based Rendering*) teksturiranih 3D objekata s velikim brojem poligona (eng. *high-poly*), kao građevnih blokova, i generira frejmove u rezoluciji 320x180 px kako bi latenciju, koja je bitna za treniranje s kraja na kraj (eng. *end-to-end training*), sveo na minimum.

Tsirikoglou i sur. koriste detaljnu 3D geometriju, fizikalno bazirane materijale, Monte Carlo bazirani rendering i vjernu simulaciju optike i senzora kamere te proizvode uvjerljivo najrealističniji sintset za primjenu u domeni autonomne vožnje [118]. Njihova metoda kombinira proceduralno generiranje unikatnog svijeta za svaki renderirani frejm sa distribuiranim renderingom "u oblaku" (eng. *cloudbased*), koristeći infrastrukturu namijenjenu filmskoj industriji. Računski zahtjevno generiranje unikatnih svjetova optimiziraju generirajući samo geometriju vidljivu kameri izravno ili u refleksijama i sjenama. Naglašavaju da je teško postići fotorealizam koristeći nedovoljno detaljnu geometriju ili fizikalno netočan transport svjetla te identificiraju 5 ključnih ciljeva za postizanje realizma: cjelokupna kompozicija scene, geometrija, osvjetljenje, svojstva materijala i optički efekti. U eksperimentima treniraju DFCN sa samo 25.000 sintetičkih slika (u rangu prethodnika) za semantičku segmentaciju i postižu 36.93%, u usporedbi s GTA V [87] (31.12%) i SYNTHIA [92] (20.7%), te zaključuju da se isplati fokusirati na maksimiziranje varijacija i realizma.

Za razliku od [118], Lopez i sur. [119] smatraju da ekstremni fotorealizam nije nužan te da Virtual KITTI [95] i SYNTHIA [92] mogu biti dovoljno realistični uz primjenu adaptacije

domene. Napominju da jaz domena nije problem na relaciji virtualno-realno već općenitiji problem različitih senzora i okruženja.

Ključna sposobnost za sigurnu navigaciju tijekom autonomnog letanja je detekcija žica koje su na slikama široke svega nekoliko piksela. Madaan, Maturana i Scherer grade sintset renderirajući 3D žice različitih oblika na 67.702 pozadinskih slika u rezoluciji 640x480 px preuzetih iz 154 različitih video zapisa letenja [120]. Koristeći pretragu po rešetki (eng. *grid search*), nalaze CNN ahitekturu na bazi VGG kojom postižu AP 0,73.

2018

U domeni navigacije robota, istaknuti problem podržanog učenja je generalizacija. Da bi mogao generalizirati u još neviđenim okruženjima, agent mora biti otporan na varijacije na niskoj razini (eng. *low-level*) poput boje, teksture ili promjene objekta, ali i one na visokoj razini (eng. *high-level*) poput promjene rasporeda elemenata koji čine okruženje. Agenti učeni u specifičnim okruženjima, loše se snalaze u novima. Wu i sur. [121] pokazuju da se taj problem može riješiti koristeći različite stupnjeve augmentacije: na nivou piksela, za rendomizaciju domene; na nivou objekta, za prisliti agenta svladati različite koncepte objekta simultano; i na nivou scene, forsirajući generalizaciju okruženja. Autori koriste SUNCG [90] za izgraditi različita okruženja i tako nastaje House3D sintset. Primjenom navedenih augmentacija na RGB slikama unapređuju uspješnost navigacije za 8%, a dodatno poboljšanje postižu koristeći dubinske i segmentacijske mape. Autori koriste prilagođeni OpenGL renderer koji na jednom NVIDIA Tesla M40 GPU omogućava renderiranje 600 slika veličine 120x90 px u sekundi, a podržava i paralelno renderiranje na većem broju GPU-ova što ga čini pogodnim za primjenu u RL.

Procjena spontanih pokreta dojenčadi omogućava predvidjeti neurorazvojne poremećaje u vrlo ranoj dobi. S obzirom da korištenje sintsetova namijenjenih određivanju poze odraslih osoba, poput [108], dovodi do degradacije točnosti kad se primijeni na dojenčad, Hesse i sur. kreiraju Moving INfants In RGB-D (MINI-RGBD) sintset koristeći SMIL model tijela te realistične oblike tijela i teksture generirane kombiniranjem različitih oblika tijela i tekstura prave djece, kako bi se sačuvala njihova privatnost [122]. Za renderiranje RGB-D slika koriste OpenDR renderer. Limitacije korištenog SMIL modela su nemogućnost pokreta prstiju, manjak facijalnih ekspresija i kose.

Ward, Moghadam i Hudson [123] bave se segmentacijom pojedinih listova biljaka što je izuzetno zahtjevan problem zbog varijabilnosti oblika listova, ali i njihove deformacije tijekom života biljke. Dodatni problem su međusobna preklapanja i zaklanjanja listova. Autori grade sintset (Synthetic Arabidopsis Dataset) na bazi jednog manualno pripremljenog 3D lista nastalog ocrtavanjem (eng. *trace*) skenirane slike. Tako pripremljen list deformiraju rendom skaliranjem po svim osima, teksturiraju s različitom teksturom (što im izravno popravlja rezultat segmentacije) i raspoređuju na sceni. Prilikom kreiranja segmentacijske mape isključuju *antialiasing* kako bi osigurali da pojedini piksel pripada samo jednom listu. Autori predlažu za postizanje geometrijskih varijacija biljaka ubuduće koristiti neki integrirani okvir za proceduralno generiranje poput L-Systema što je pristup koji će uspješno koristiti [124].

Istraživači u domeni nadziranja zanemarivali su utjecaj osvjetljenja na sposobnost identifikacije osoba. Drastične razlike u osvjetljenju imaju izrazito negativan utjecaj, što su uočili Bąk, Carr i Lalonde [125]. Zato uvode HDR rasvjetu, po uzoru na [40], i koristeći 140 HDR mapa generiraju veliki (1.68M) Synthetic Person Re-Identification (SyRI) sintset. Zahvaljujući raznolikosti osvjetljenja, modeli učeni na ovim podacima značajno bolje detektiraju osobe u novim svjetlosnim uvjetima, čime autori problem identifikacije osoba tretiraju kao problem adaptacije domene. Autori napominju da za učenje diskriminativnih značajki koje dobro generaliziraju, broj klasa tijekom treninga treba biti značajno veći od dimenzija zadnjeg skrivenog sloja neuronske mreže. Zato su dimenziju zadnjeg skrivenog sloja fiksirali na 256, a mrežu trenirali sa preko 3.000 klasa (različitih identiteta) koje su generirali koristeći Adobe Fuse. Unatoč velikom broju slika pripremljenih za trening koristili su i realne slike za dodatnu adaptaciju domene jer ni sa 140 HDR mapa nisu uspjeli pokriti sve moguće slučajeve, a uočili su i jaz u distribuciji značajki između sintetičkih i realnih slika koji sugerira da, fokusirani na varijabilnost rasvjete, nisu uzeli u obzir i druge faktore značajne za premošćivanje jaza, predložene u [118].

Saleh i sur. uočavaju da pomak domene (eng. *domain shift*) nema jednak utjecaj na klase u prednjem i stražnjem planu slike te da ih zato treba različito tretirati. Stražnji plan često djeluje realističnije dok objekti u prednjem planu imaju realističan oblik, ali ne i teksturu, što ih čini pogodnijima za detekciju nego za segmentaciju [126]. Autori u Unityu grade VIES, minimalno realistično virtualno okruženje namijenjeno segmentaciji u domeni autonomne vožnje, i prijavljuju da je za izgraditi ga bio potreban samo 1 dan jednoj osobi. Koristeći isključivo sintset generiran VIES-om, uspješno kombiniraju maske objekata u prednjem planu, proizvedene s Mask R-CNN, sa segmentacijom na nivou piksela, koristeći DeepLab, i potvrđuju da je fuzionirani pristup, u kojem je segmentacija bazirana na prethodnoj detekciji, uspješniji. Time ujedno pokazuju da između realne i sintetičke domene postoji veći jaz kada se trening oslanja primarno na podatke o teksturi jer su oblici robusniji u odnosu na pomak domene.

Tremblay i sur. nastavljaju Tobinova [50] istraživanja o potencijalu rendomizacije domene kao jednostavnije, a time i jeftinije, alternative generiranju ekstremno fotorealističnih sintsetova [127]. Grade svoj DR sintset rendom smještajući proizvoljan broj 3D objekata od interesa (u njihovom slučaju to su automobili) na rendom 2D pozadinu te uvode novu komponentu, leteće distraktore (eng. *flying distractors*), u obliku različitih jednostavnih geometrijskih tijela. Sve objekte na sceni (i one od interesa i distraktore) teksturiraju koristeći rendom teksture, što je ključna novost. Provode iscrpnu ablacijsku studiju koja potvrđuje važnost rendomizacije osvjetljenja, različitosti tekstura i augmentacije, a za leteće distraktore, kojima je svrha naučiti mrežu ignorirati susjedne uzorke i nositi se s parcijalnim okluzijama objekata od interesa, navode da se bez njih performanse smanjuju za 1,1%. Vezano za veličinu trening seta, do saturiranja performansi dolazi već nakon 10K korištenih slika kada se koriste prethodno trenirane težine odnosno nakon 50K bez njih. No, prethodni trening im pomaže do čak 1M slika što se može objasniti činjenicom da slike iz DR sintseta nisu fotorealistične.

Nakon nerealističnog DR sintseta [50], Tremblay, To i Birchfield, kreiraju realistični Falling Things (FAT) [128] sintset, služeći se metodom fizikalno simuliranog ispuštanja 3D objekata

na scenu predstavljenom u [99]. Set je namijenjen detekciji objekata u kućanstvu, a autori izmjerenom statistikom kvantitativno potvrđuju optimalnu distribuciju varijabilnosti postignutu zahvaljujući primijenjenoj metodi. Autori predvođeni Tremblayem koriste FAT u domeni određivanja poze za robotski hvat u [129], i pokazuju da se kombinacijom fotorealističnih slika i rendomizacije domene može postići dostatna varijabilnost dataseta za korištenje tako treniranog modela u realnom okruženju, bez dodatnog podešavanja. Mjereći učinak veličine datasetova uočili su da set baziran isključivo na rendomizaciji domene postiže najbolji rezultat pri korištenju 300K slika (66,64 AUC), isključivo fotorealistični set tek sa 600K (62,94 AUC), a kada se koristi kombinacija na način da niti jedan set nije zastupljen s manje od 40%, postiže se najbolji rezultat (77,00 AUC) već sa 120K slika.

Sorokin i sur. kreiraju generator sintetičkih sekvenci filopodija, za potrebe medicinske segmentacije [130]. Generator se sastoji od 3 modula zadužena za generiranje geometrije (izrađen u MATLAB-u), te deformacija i tekstura (izrađeni u C++). Evolucija filopodija je modelirana koristeći stohastičku simulaciju rasta na molekularnom nivou, a kretanje i deformacija tijela stanice i pripadnih filopodija bazirana je na teoriji elastičnosti i implementirana korištenjem FEM-a (eng. *Finite Element Method*). Time je u generatore sintsetova uvedena interna struktura geometrije, čime je eliminiran nedostatak na kojeg su ukazivali Qiu i Yuille [98].

Rahmani, Mian i Shah razvijaju Robust Non-linear Knowledge Transfer Model (R-NKTM) za nenadzirano učenje u domeni raspoznavanja radnje [131]. R-NKTM je potpuno povezana neuronska mreža koja transferira znanje o radnjama promatranim iz bilo kojeg nepoznatog kuta u dijeljeni (eng. *shared*) virtualni pogled, tražeći nelinearni put koji ih povezuje. Mreža je skalabilna i potrebno ju je trenirati samo jednom, koristeći sintetičke podatke, nakon čega dobro generalizira s realnim podacima. Autori generiraju potrebne sintetičke podatke aplicirajući *mo-cap* animacije na 3D model čovjeka iz MakeHuman aplikacije. Potom renderiraju 108 pogleda na takvu video sekvencu i iz njih računaju realistične putanje dijelova tijela služeći se prethodno poznatim pozama. S tim putanjama, koristeći algoritam k-srednjih vrijednosti (eng. *k-means*), formiraju generalnu kodnu knjigu (eng. *codebook*) s kojom potom generiraju sintetičke putanje koje koriste za treniranje R-NKTM.

Monokularna procjena dubine koristi se u domeni autonomne vožnje kada nisu raspoloživi parovi stereo slika. No, treniranje modela za procjenu dubine koristeći sintetičke podatke podložno je utjecaju domene (eng. *bias*) zbog čega takvi modeli ne funkcioniraju u realnom okruženju. Opisani problem adaptacije domene, Atapour-Abarghouei i Breckon rješavaju transferom stila (eng. *style transfer*), kojim se minimizira razlika između distribucija u dvije različite domene (realna i sintetička), te suparničkim (eng. *adversarial*) treningom, odnosno koristeći GAN [132]. Problem na koji nailaze u ovom pristupu je nemogućnost adaptacije u slučajevima iznenadne promjene osvjetljenja i saturacije. Uz to, kada se dvije domene drastično razlikuju u intenzitetu između osvjetljenih područja i područja u sjeni, sjene, nakon transfera stila, mogu biti pogrešno raspoznate kao izdignuta područja ili nepostojeći objekti u prednjem planu.

Tekstura, odsjaji svjetla (eng. *highlights*) i sjenčanje neki su od vizualnih znakova koji omogućavaju ljudima pojmiti materijal od kojeg je građen objekt na nekoj slici. Deschaintre i sur. traže način za, koristeći neuronsku mrežu, iz slike ekstrahirati 4 slikovne mape (difuzni

albedo, reflektivni albedo, hrapavost refleksije i normale) pomoću kojih je, koristeći Cook-Torrance BRDF model sjenčanja, moguće rekreirati materijal sa slike prilikom renderinga [36]. U tu svrhu grade Synthetic SVBRDFs And Renderings sintset, varirajući 800 SVBRDF (eng. *spatially-varying bi-directional reflectance distribution functions*) u Allegorithmic Substance proceduralnim materijalima kreiranim od strane grafičara iz filmske i video industrije. Iako postiže dobre rezultate, nedostatak ove metode je korištenje ulaznih slika isključivo iz frontalnog pogleda, na kojima nije vidljivo ponašanje refleksija pod blagim kutovima u odnosu na kameru, zbog čega se ne može naučiti rekonstrukcija Fresnellovog efekta.

Wrenninge i Unger slijede [118] i generiraju jednako fotorealističan sintset, Synscapes [133]. U anotacije uvode metapodatke scene, kojima opisuju sva svojstva scene za svaku generiranu sliku, a za pohranu slika koriste OpenEXR format. Nalaze da su zamućenje pokreta (kao posljedica brzine kretanja promatrača) i doba dana (visina Sunca) parametri koji najviše utječu na prediktivne performanse mreže. Zamućenje pokreta je posebno zanimljivo jer se povećava s povećanjem brzine vozila na kojem se nalazi kamera i razmazuje značajke na slici koje ostaju prepoznatljive čovjeku, ali se CNN-u čine značajno drugačije od prethodno naučenih. Približavanje Sunca horizontu smanjuje kontrast na slici, ali, zbog korištene automatske ekspozicije (eng. *auto-exposure*), slika nije nužno tamnija već nestaju jake sjene bez kojih postaje teže razlikovati naučene značajke. Autori naglašavaju da nema naznake da neuronske mreže mogu samostalno poništiti pomak domene te da zbog toga realizam treba ugraditi u sintsetove prilikom njihovog generiranja.

Tian i sur. proceduralno generiraju gradove koristeći konfiguracije preuzete iz OpenStreetMap (OSM) podataka i Computer Generated Architecture (CGA) pravila unutar specijaliziranog alata za generiranje 3D urbanih okruženja, CityEngine [93]. Tako nastaje njihov sintset, ParallelEye, namijenjen detekciji objekata u domeni autonomne vožnje. Autori napominju da, iako je moguće koristiti GAN-ove za kreiranje fotorealističnih slika, problem s njihovim korištenjem u funkciji generiranja sintsetova je što takvim slikama manjkaju odgovarajuće anotacije, a to poništava ključnu prednost sintsetova u odnosu na realne. Ovaj zaključak potvrdit će i [134].

Lai i sur. kreiraju VIVID, simulator namijenjen učenju vizualnog raspoznavanja, koji nudi velike i raznolike scene u eksterijerima i interijerima [135]. Za razliku od prethodnih simulatora, fokusiranih na vozila i letjelice, novost koju uvode su akcije ljudi.

Mayer i sur. analiziraju različite načine izgradnje sintsetova u domenama optičkog toka i određivanja dispariteta: proceduralnu rendomizaciju i manualno modeliranje [136]. Manualno modeliranje geometrije i manualno slaganje cijele scene isključuju kao opciju jer rezultira s manje generiranih podataka uz isto uloženog truda, ali pritom ne uzimaju u obzir mogućnost proceduralnog modeliranja i generiranja scene koja može pomiriti oba pristupa. Nalaze da je trening koji koristi različite datasetove najučinkovitiji kada se provodi po fazama, koristeći prvo jedan dataset za učenje, a potom, zasebno, svaki sljedeći. Također nalaze da u pojedinim domenama, poput optičkog toka, realizam nije nužan – potenciranje realizma kompleksnim osvjetljenjem scene neće nužno pomoći čak ni kada su testni podaci realistično osvjetljeni. Ističu i važnost trenutka u kojemu se podaci prezentiraju neuronskoj mreži: rana faza učenja preferira jednostavnije podatke, a kasnija složenije.

Na tragu ideje o različitoj važnosti prednjeg i stražnjeg plana [126], Alhaija i sur. predlažu jednostavniji način za augmentaciju realnih slika, korištenih kao okruženje, virtualnim objektima [137]. Takav pristup omogućava brzu proizvodnju realističnih pozadinskih slika (fotografiranjem) uz istovremenu mogućnost generiranja proizvoljnog broja kombinacija 3D objekata u prednjem planu. Tom metodom grade sintsetove KITTI-360 i KITTI-15 te eksperimentima dokazuju da modeli učeni na takvim slikama generaliziraju bolje od onih učenih isključivo na sintetičkim podacima ili na ograničenim količinama realnih podataka. Uočavaju da dodavanje samo jedne augmentacije ne doprinosi poboljšanju modela (u odnosu na treniranje na samo realnim slikama), ali performanse se povećavaju sa svakom dodanom augmentacijom i do saturacije dolazi nakon dodanih 20. Vezano uz broj 3D objekata (automobili) korištenih za augmentaciju, zaključuju da više od 5 dodanih na scenu utječe negativno na performanse jer dolazi do značajnog zaklanjanja manjih vozila (u stražnjom planu) koja, svojom raznolikošću, više doprinose raznolikosti značajki. Ispituju i utjecaj refleksija, naknadne obrade i pozicioniranja objekata, kao ključnih aspekata realizma, te zaključuju da refleksije realne okoline (korištenjem HDR mapa za rasvjetu) imaju minimalan utjecaj, naknada obrada (osobito korištenje zamućenja, da renderi ne budu oštriji od pozadina) značajan, a prilikom pozicioniranja 3D modela pomaže ako su, fizikalno ispravno, smješteni na tlo. Problem njihovog pristupa je što funkcionira samo kada su svi umetnuti 3D objekti isključivo u prednjem planu, a, s obzirom da u kontekstu rasvjete 3D objekti nisu u interakciji s okruženjem, ugrožen je realizam jer ne poštuju sjene koje bi trebale padati na njih (zgrade, drveće...).

Ludl i sur. kreiraju SIM i BIG-SIM sintsetove, u domeni određivanja poze, namijenjene rijetkim aktivnostima u urbanim područjima [138]. Prilikom generiranja setova koriste 2D pozadine, ali, kao i kod [109], uz uvjet da se na njima ne pojavljuju ljudi jer njihova neanotirana prisutnost radi problem prilikom učenja mreže. U anotacije uvode metapodatke za ljude (spol, dob, rasa, visina i težina), koji omogućavaju selektivne upite prema modelu, bazirane na tim parametrima, te anotiraju radnje i namjere.

Koristeći veliku količinu neanotiranih video podataka i manji set anotiranih, Khodabandeh i sur. sintetiziraju realističan video set podataka za treniranje neuronskih mreža namijenjenih raspoznavanju radnje [139]. To čine tako što prvo treniraju generativni model na manjem, anotiranom setu putanja 18 pojedinih zglobova (koristeći 2D kostur). Ovaj model generira sekvence pokreta kostura za danu oznaku radnje. Drugi generativni model, treniran na neanotiranom setu, potom generira sekvencu fotorealističnih frejmova koristeći zadanu oznaku radnje, set referentnih slika i proizvoljnu pozadinu. Tako nastaje sintetizirani video na kojem osoba iz referentnih slika izvodi zadanu radnju. Metodu koriste za dopuniti postojeće realne video setove za raspoznavanje radnji. Čine to u omjeru 1:20 i time postižu bolje rezultate od prethodnika koristeći Inflated 3D ConvNet i Convolutional 3D mreže.

Rojas-Perez, Munguia-Silva i Martinez-Carranza koriste vlastiti sintset za određivanje sigurne zone za slijetanje pri autonomnom letenju i zaključuju da korištenje dubinskih mapa u tu svrhu daje bolje rezultate od korištenja RGB podataka [140]. Pritom koriste Inception module za ekstrahirati značajke različitih veličina iz mapa dubine, omogućavajući na taj način mreži da nauči više detalja iz njih.

Nakon što su se prethodno fokusirali na rendomizaciju tekstura [50], Tobin i sur. nastavljaju istraživati mogućnosti rendomizacije domene tako što generiraju masivni (1M) set nerealističnih proceduralno generiranih objekta [30]. U tu svrhu koriste 40.000 objekata iz ShapeNet dataseta koje rastavljaju u više od 400.000 konveksnih dijelova koristeći V-HACD biblioteku za dekompoziciju.

Prakash i sur. predstavljaju strukturiranu rendomizaciju domene (Structured Domain Randomization, SDR) koja uzima u obzir strukturu i kontekst scene [141]. Za razliku od DR [127] koji objekte i distraktore smješta nasumično prema jednolikoj raspodjeli vjerojatnosti, SDR to čini prema raspodjeli vjerojatnosti koje proizlaze iz specifičnog problema. Na taj način SDR omogućava neuronskoj mreži da prilikom detekcije uzme u obzir i područje oko objekta, kao kontekst. U formulaciji autora, SDR uključuje 3 komponente: globalne parametre, jednu ili više krivulja koje predstavljaju kontekst i objekte koji su raspoređeni duž tih krivulja. SDR pristup balansira između ekstremnog fotorealizma i nerealističnosti karakteristične za DR, proizvodeći slike koje su prilično realistične, ali primarno sadrže veliku raznolikost. Za razliku od DR, kod kojeg se performanse saturiraju tek sa 50K slika [50], SDR saturaciju postiže već se 10K slika, zbog čega je SDR moguće koristiti i za inicijalizaciju mreže kada ne postoji dovoljno anotiranih realnih podataka. [127] je pokazao da je za DR rasvjeta najznačajniji parametar, ali kod korištenja SDR to su kontekst, saturacija i kontrast, pri čemu saturacija ukazuje na važnost usklađivanja tekstura između dvije domene.

U prilog korištenju SDR u odnosu na DR govore i rezultati Dvornika, Mairala i Schmida koji, u domeni detekcije objekata, provode eksperimente smještajući objekte izrezane iz 2D slika na različite pozicije na 2D pozadinama, te zaključuju da točnost detekcije značajno opada kada su objekti smješteni na nerealističnim pozicijama [142]. To ukazuje na činjenicu da vizualni kontekst postaje krucijalan izvor informacije kad god su vizualne informacije oštećene, dvosmislene ili nepotpune.

Tylecek i sur. kreiraju 3DRMS Challenge Dataset 2018 namijenjen 3D rekonstrukciji vrtnih scena za potrebe učenja kretanja robota kroz njih [143]. Većina prethodno analiziranih sintsetova u domeni autonomnog kretanja sadrži objekte i okruženja kreirana od strane ljudi. U takvim okruženjima elementi flore pojavljuju se primarno dekorativno, dok 3DRMS naglasak stavlja upravo na mogućnost navigacije kroz, pod utjecajem vjetra i drugih atmosferskih uvjeta, pomične fine strukture poput lišća, stabljika i grana.

2019

Pumarola i sur. grade masivni 3DPeople Dataset (2.5M) sa 80 virtualnih karaktera koji izvode 70 različitih radnji [144]. Za razliku od SURREAL [108] seta u kojem autori tijelima dodaju teksturu odjeće, Pumarola i sur. za odjenuti likove, ali i opremiti ih dodacima poput naočala i šešira, koriste zasebnu geometriju. Na takvom setu primjenjuju vlastiti Spherical Area-Preserving Parameterization (SAPP) algoritam, baziran na metodi optimalnog transporta mase, koji 3D geometriju svodi na slikovnu reprezentaciju (tzv. geometrijska slika). Posljedica ovog postupka je da će sve geometrijske slike biti semantički usklađene odnosno svaka UV koordinata odgovarat će približno istom semantičkom dijelu modela, što će

značajno olakšati učenje neuronske mreže. Mreža učena s ovim podacima, za svaku 2D sliku čovjeka u pozi generira odgovarajuću 3D geometriju, ali bez ikakvih tekstura.

Gotovo desetljeće nakon Bakerovog hibridnog Middlebury seta [51], Bayraktar, Yigit i Boyraz odlučuju se za izgradnju vlastitog (ADORESet), u omjeru 10:3 (realni:sintetički), u domeni detekcije objekata, namijenjenog robotici [145]. Odluku o izgradnji hibridnog seta objašnjavaju činjenicom da većina dostupnih datasetova sadrži ili realne ili sintetičke slike, što smatraju nedostatnima za primjenu na njima učenih modela u stvarnom svijetu, pogotovo uzevši u obzir da su prethodna istraživanja ([95], [115] i [127]) u ovoj domeni zanemarila testiranje modela učenih na sintetičkim podacima na realnim i obrnuto. Koristeći različite CNN arhitekture dokazuju da učenjem na hibridnom setu postižu uvjerljivo najbolje rezultate.

Li i sur. polaze od pretpostavke da složenost i raznolikost realnog svijeta nije moguće realistično replicirati u virtualnom okruženju te unapređuju ideju o polaganju 3D modela vozila na pozadinske fotografije iz [137] tako što umjesto fotografija koriste video snimke stvarnih prometnica koje kombiniraju s LiDAR podacima kako bi proizveli precizne dubinske mape [146]. Ovaj pristup omogućava im ne samo uklopiti 3D objekte u bilo koji plan već i generiranje realističnog toka prometa vozila te dinamike kretanja pješaka. U sklopu prethodne obrade uklanjaju pomične objekte, docrtavaju uklonjeno, preuzimaju iluminaciju i poboljšavaju teksture. Rezultirajuće slike, proizvedene PBRT rendererom, su fotorealistične i čine njihov AADS sintset. Problemi ovog pristupa, u odnosu na korištenje potpuno virtualnih svjetova, su ograničeno vidno polje LiDAR-a zbog kojeg, iako postoji 3D scena okruženja, gledište na sintetiziranim slikama ne može značajno odstupiti od izvornog, te nemogućnost variranja rasvjete i vremenskih uvjeta.

Krishnan i sur. kreiraju Air Learning, simulator namijenjen autonomnom letenju [147]. Od prethodnika [21] se razlikuje po tome što omogućava generiranje različitih rendom okruženja te evaluaciju performansi RL modela s naglaskom na njihovu energetsku učinkovitost pri korištenju na pojedinim hardverskim platformama.

U domeni vizualnog rasuđivanja primijenjenog na video, Girdhar i Ramanan grade sintset CATER [148] namijenjen učenju raspoznavanja kompozicija pokreta objekata koje zahtijevaju dugoročno rasuđivanje. CATER je baziran na CLEVR-u [31], ali uvodi 2 dodatna objekta (izvrnuti konus i tri isprepletena torusa) te 4 različita pokreta (rotaciju, promjenu mjesta, klizanje i poklapanje jednog objekta drugim). Autori zaključuju da su za zadatke višeg nivoa, poput raspoznavanja radnje, aktualne arhitekture neuronskih mreža upotrebljive uz dovoljno veliki dataset za učenje, ali da zadaci srednjeg nivoa, poput praćenja objekata, predstavljaju veliki izazov u slučaju dugotrajnih okluzija.

Savva i sur. kreiraju Habitat, simulator za istraživanje utjelovljene umjetne inteligencije [7]. Njegova ključna prednost je brzina kojom generira realistične slike, koristeći samo 1 GPU: preko 10.000 frejmova u sekundi, što u praksi znači da je njime brže generirati sintetičke slike nego ih učitavati s diska. Mason, Vejdan i Grijalva [149] analiziraju učinke takvih generiranja sintsetova "u letu" i zaključuju da su pozitivni, pogotovo u smislu uštede vremena korištenog za zapisivanje i čitanje pohranjenih podataka, za sve metode strojnog učenja koje ovise o velikim skupovima podataka.

Qiang Wang i sur. grade IRS sintset namijenjen evaluaciji stereo dispariteta u scenama interijera [150] i u njemu posebnu pozornost posvećuju oblikovanju svjetlosnih efekata (promjena svjetloće, refleksija i transmisija svjetla, odbljesak objektiva), smatrajući ih značajnim za primjenu naučenog modela u realnom okruženju. Pritom koriste odgođen (eng. *deferred*) put sjenčanja, optimiziran za scene s velikim brojem svjetala u kojima je potrebna visoka razina vjernosti osvjetljenja.

Qi Wang i sur. koriste realističnost igre GTA V za kreiranje GCC sintseta namijenjenog brojanju ljudi na sceni [151]. Njihovi prethodnici ([87], [82] i [112]) to čine bez zadiranja u igru, ali s obzirom da GTA V ne sadrži potrebne scene mnoštva ljudi, autori su prisiljeni kreirati vlastite scene koristeći dodatak baziran na Script Hook V C++ biblioteci. Dodatni problem je što GTA V ne podržava više od 256 ljudi na sceni, zbog čega razdvajaju ciljana područja na više scena, zasebno ih renderiraju i potom spajaju slike. Na taj način dovode na scene ukupno 7.625.843 različito animiranih i međusobno variranih virtualnih karaktera. Broj karaktera na pojedinim slikama varira između 0 i 3.995, a prosječno ih je vidljivo 501. Za potrebu adaptacije domene uvode Structural Similarity Index (SSIM) u tradicionalni Cycle GAN, što im omogućava zadržati lokalnu teksturu i strukturnu sličnost koje bi bez SSIM bile izgubljene ili distorzirane. Prema primjerima koje autori prikazuju, čini se da je prvenstvena uloga adaptacije domene korekcija boje (smanjivanje saturacije i promjena tona slike) pa vrijedi postaviti pitanje može li se do istog rezultata doći jednostavnom kolor korekcijom, koristeći odgovarajući 3D LUT (eng. *lookup table*, tablica za pretraživanje) kojom se mapira jedan prostor boje u drugi.

Solovev i sur. koriste X-Plane, komercijalni simulator letenja, za izgraditi istoimeni sintset namijenjen učenju reprezentacije stanja za potrebe slijetanja zrakoplova [152]. Uz svaku sliku generiraju podatke o stanju 1.090 različitih senzora, a prilikom svake od 8.011 simulacija, koriste do 12 različitih perturbacija poput naleta vjetra i neispravnosti raznih sustava u zrakoplovu. Autori treniraju različite modele koristeći samo slike, samo podatke senzora i obje vrste podataka zajedno te nalaze da modeli koji koriste slike postižu bolje rezultate od onih koji ih ne koriste, ali samo pod uvjetom da su ključni sadržaji na slici (horizont i staza za slijetanje) jasno vidljivi (što noću nije slučaj).

Tripathi i sur. smatraju da generiranje sintetičkih podataka metodom komponiranja ima dvije bitne prednosti: jaz između domena je minimalan i ovisi primarno o artefaktima *blendinga*, a sama metoda je široko aplikativna jer može koristiti već postojeće datasetove anotiranih 2D slika [15]. Problem artefakata *blendinga* i moguća rješenja razmatrana su u [116], a Tripathi i sur. im predlažu alternativu namjernim kreiranjem artefakata polaganjem (eng. *pasting*) dijelova drugih pozadina, izrezanih u obliku prethodno ekstrahiranih maski objekata s trećih slika. Na taj način uklanjaju diskriminatorne značajke nastale komponiranjem i postižu jednako dobre rezultate koristeći 50% manje podataka za učenje.

Fonder i Droogenbroeck, koristeći UE4, grade Mid-Air, multi-modalni sintset namijenjen dronovima u niskom letu kroz nestrukturirana okruženja (nenaseljena područja) [153]. U dosadašnju praksu variranja doba dana i godine, uvode volumetrijske i dinamičke oblake koji bacaju sjenu i time doprinose realizmu. Autori modeliraju dostupnost GPS signala dronu i čine to optički tako što uvode dodatnu kameru koja gleda prema satelitima i utvrđuje

vidljivost. Bolji postupak bio bi napraviti projekciju zrake (eng. *raycast*) od satelita prema dronu, što bi im omogućilo, testiranjem fizikalnih slojeva (bazirano na materijalima), provjeriti da li signal može proći kroz pojedinu prepreku pa čak i koliko ga se pritom gubi. Autori ukazuju i na problem korištenja LOD-ova (eng. *level of details*, nivo detalja) koje je nužno koristiti na prostorno velikim scenama (njihova pokriva 100 km^2) kako bi se reducirala količina prikazane 3D geometrije. LOD-ovi, prilikom približavanja 3D objektu, uzrokuju njegovu naglu vizualnu promjenu jer se promjenom LOD-a mijenja prikazani objekt (objekt u nižoj geometrijskoj rezoluciji zamjenjuje se objektom u višoj), a to može negativno utjecati na učenje modela. Preporučuju da se zamjena objekata vrši kada su objekti još dovoljno daleko.

Jung i sur. grade Apollo Synthetic Dataset [154] i prvi spominju tehniku fotogrametrije s kojom, koristeći 3D skenove objekata u visokoj rezoluciji, retopologiziranjem dolaze do realističnih 3D modela s optimalnim brojem poligona za izvođenje u realnom vremenu. Vremenski najzahtjevniji korak ove tehnike je faza retopologizacije koja će se moći izbjeći koristeći mikro-poligonske renderere poput Nanite renderera sadržanog u nadolazećem UE5 alatu za razvoj igara. Mikro-poligonski renderi ujedno eliminiraju i problem LOD-ova spomenut u [153].

Chociej, Welinder i Weng kreiraju ORRB (OpenAI Remote Rendering Backend), simulator namijenjen vizualnom treniranju robota [155]. ORRB je fokusiran na rendomizaciju domene i optimiziran za upotrebu u oblaku: 88 poslužitelja za renderiranje, koristeći svaki po 8 V100 GPU-a, proizvodi 3.438 frejmova u sekundi, što omogućava hiper-produckiju masivnih sintsetova. Autori prvi spominju korištenje ambijentalne okluzije kao efekta u naknadnoj obradi, za razliku od [51] gdje je inicijalno korišten u renderingu. Ukazuju i na važnost sjemena (eng. *seed*) kojeg koriste kako bi rendomizaciju i rendering učinili deterministčkim u onoj mjeri u kojoj alat za razvoj igre (Unity) to dopušta. Determinizam je izuzetno važan u kreiranju sintsetova jer omogućava reproduktibilnost, a time i kontrolirane promjene pojedinih parametara koji utječu na ishod rendomizacije.

Sharma i D'Amico grade Spacecraft Pose Estimation Dataset (SPEED), sintset namijenjen određivanju poze svemirske letjelice u orbiti Zemlje [156]. Koristeći OpenGL renderer polažu renderirani 3D model Tango letjelice preko fotografija Zemlje načinjenih meteorološkim satelitom (Himawari-8) i definiraju tako stvorene slike AR (eng. *Augmented Reality*, dopunjena stvarnost) izvorom sintseta.

Dosadašnji pokušaji simuliranja LiDAR-a unutar GTA V ili UE4/Unity baziranih simulatora udarili su u 2 problema: koristeći projeciranje zrake u ulozi LiDAR-a, registrirali bi pojednostavljenu geometriju sudarača (eng. *collider*), npr. cilindar za pješaka ili kuglu za krošnju stabla, umjesto vidljive geometrije – i to samo na udaljenosti do 30 m. Zato Hurl, Czarnecki i Waslander umjesto projeciranja zrake koriste mapu dubine i grade puno precizniji LiDAR-ov oblak točaka iz nje [157]. Problem ovog pristupa je nedostatak informacija o vrijednosti refleksije (boja objekta), koju LiDAR inače sadrži, zbog čega korištenje podataka u ovom obliku može značajno umanjiti performanse modela prilikom detekcije [158].

Khan i sur. [159] metodu za proceduralno generiranje 3D scene koristeći OSM kartografiju i CityEngine iz [93] kombiniraju prosljeđivanjem tako nastale scene u CARLA simulator [104]. Eksperimentiraju s kišnim uvjetima i nalaze da dodavanje 3% kišnih slika poboljšava segmentaciju (kišnih slika) oko 10%, ali da treniranje mreže s više od 15% kišnih slika ima negativan utjecaj na performanse, što se može objasniti manjom zastupljenošću kišnih slika u testnim setovima. Također nalaze da oblaci i lokve smanjuju performanse mreže više no što to čini kiša svojim okluzijama. Oblaci, zbog svojih oblika, često budu raspoznati kao neka druga klasa (npr. automobil), a lokve kreiraju refleksije po Y osi i utoliko imaju izrazito negativan utjecaj na segmentaciju svih klasa čije slike reflektiraju.

Jalal i sur. grade SIDOD sintset koristeći NVIDIA Deep Learning Data Synthesizer (NDDS), namijenjen efikasnijem snimanju sintetiziranih slika i pripadnih anotacija, unutar UE4 [160]. SIDOD u anotacije uvodi rotaciju objekta. Metoda generiranja je slična [128], ali, za razliku od njih, ne generiraju više od jedne slike tijekom ispuštanja i pada objekata već rendomiziraju parametre za svaku sliku. Koriste leteće distraktore uvedene u [127], ali pritom ne koriste kolizije već dopuštaju penetraciju distraktora u objekte od interesa, svodeći ih tako na 2D okludere različitih oblika (koji nastaju potencijalnom penetracijom).

Kako bi obuhvatili cijelu scenu jednom slikom, za razliku od prethodnika koji koriste *pinhole* model kamere (jednosmjeran i bez leće), Zioulis i sur. uvode višesmjernu (eng. *omnidirectional*) kameru u domenu rekonstrukcije scene i s njom renderiraju 3D60 sintset [161]. Tako generirane slike moguće je koristiti u CNN-ovima nakon što ih se savijanjem (eng. *warping*) razvije u pravokutni oblik koristeći kubične mape (eng. *cubemap*) ili pravokutne (eng. *equirectangular*, ERP) projekcije. Sličnim problemom bave se i Yogamani i sur. [162], pri čemu koriste 4 kamere s ribljim okom (eng. *fisheye*) za postići višesmjernost pogleda, te ukazuju da se prilikom procesiranja takvih slika standardnim CNN-ovima gubi svojstvo nepromjenjivosti značajki prilikom translacije.

Temel, Chen i AlRegib [163] koriste alat za vizualne efekte, Adobe After Effects, i u njemu različite tehnike naknadne obrade za generirati CURE-TSD sintset prometnih znakova s kojim mjere utjecaj različitih faktora na degradaciju raspoznavanja. Dok su autori [62] koristili isključivo geometrijske transformacije, promjenu boja i zamućivanje, Temel i sur. eksperimentiraju sa kišom, snijegom, sjenama, sumaglicom, promjenom osvjetljenja, šumom, greškama kodeka (eng. *codec*), prljavim lećama, okluzijama i oblačnošću. Nalaze da najznačaniji utjecaj na degradaciju imaju greške u kodeku i ekspozicija (preko 80%), a najmanji sjene (16%). Utjecaj svih ostalih uvjeta je u značajnom rasponu od 30% do 63% što ukazuje na zaključak da kod generiranja sintsetova dodavanje efekata u naknadnoj obradi treba tretirati kao dodavanje šuma, a ne samo kao alat za postizanje realizma.

Weinzaepfel i sur. istražuju vizualnu lokalizaciju u prostoru, koristeći slike na zidovima virtualne galerije kao objekte od interesa [42]. Za tu potrebu grade Virtual Gallery sintset u kojemu posebnu pozornost posvećuju variranju rasvjete i različitim stupnjevima okluzije. Koriste homografiju u funkciji augmentacije i njome značajno poboljšavaju lokalizaciju (sa 25% na 69%).

Xie i sur. kreiraju simulator za istraživanje utjelovljene umjetne inteligencije [164]. Za razliku od Habitata [7], fokusiranog primarno na brzinu generiranja sintetičkih slika, VRGym osim

fizikalne simulacije krutih tijela (eng. *rigid body*) omogućava i simulaciju mekih tijela (eng. *soft body*), kolizija, fluida, gibanja odjeće, te rezanja i lomljenja objekata. Kako bi omogućio bolju integraciju čovjeka u virtualnoj stvarnosti (eng. *virtual reality*), koristi Oculus Touch, LeapMotion i podatkovnu rukavicu (eng. *data glove*) za prikupljanje podataka u ulozi anotacija koje se mogu koristiti za *on-line* i *off-line* trening.

Tome i sur. ukazuju na porast interesa za xR (eng. *eXtended Reality*) tehnologijama (AR, VR i MR) u različitim domenama te činjenicu da je čovjek u njima zastupljen pogledom iz vlastite perspektive [165]. Takav egocentričan pogled karakteriziraju izrazita samo-okluzija i jake perspektivne distorzije koje rezultiraju drastičnom razlikom u percepciji gornjeg i donjeg dijela tijela. Tim problemima moguće je doskočiti noseći inercijalne mjerne uređaje (eng. *Inertial Measurement Unit*, IMU), ali oni su intruzivni i kompleksni za kalibraciju. Zato autori grade fotorealističan xR-EgoPose sintset, namijenjen učenju modela raspoznavanju poze iz prepoznatih pozicija zglobova. U anotacije uvode 3D poziciju zglobova.

Nowruzi i sur. istražuju performanse detektora objekata u uvjetima ograničene količine stvarnih podataka [9]. Koriste postojeće sintsetove ([133], [112], [104]) iz domene autonomne vožnje i traže odgovor na pitanje koliko je realnih podataka, u odnosu na sintetičke, zapravo potrebno. Koristeći samo realne setove zaključuju da uklanjanje čak 90% seta ima manji negativni učinak na pouzdanost detektora nego uklanjanje sljedećih 5% čime se može objasniti dostatnost korištenja minimalnog realnog seta (u odnosu na sintset) za poboljšavanje performansi modela, a time i preporuka optimalnog omjera (korištenje 15-20 sintetičkih slika za svaku realnu) iz [66].

Hinterstoisser i sur. nude rješenje u istoj domeni, za slučaj kada realni podaci uopće nisu dostupni za specifičan trening. Pokazuju da se detektore objekata bazirane na neuronskim mrežama (Faster-RCNN, R-FCN, MaskRCNN) može učinkovito trenirati koristeći isključivo sintetičke podatke tako da se zamrznu slojevi odgovorni za ekstrakciju značajki generičkih modela inicijalno treniranih na realnim podacima [166]. Ova metoda je uspješna jer su inicijalni ekstraktori značajki (InceptionResnet i Resnet) već dovoljno istrenirani da se, kada se primijene na sintetičke podatke, ponašaju kao "projektori" i rezultiraju značajkama koje su bliže realnim.

Wang i sur. bave se detekcijom proizvoda u automatima za prodaju i u svrhu adaptacije domene, kao i [132], koriste transfer stila kako bi renderirane objekte svog sintseta učinili realističnijima [167]. Pritom transfer stila primjenjuju isključivo na pojedine objekte od interesa, izostavljajući okolinu jer je Cycle GAN previše promijeni. Oblik pojedinih proizvoda u automatima može se značajno razlikovati zbog deformacija ambalaže, zbog čega je simuliraju deformacijom geometrije, rendomizirajući pozicije točaka površine. Za detekciju koriste PVANET, SSD i YOLOv3. Najbolje rezultate postižu s PVANET-om (95.54%), dok SSD (90,02%) i pogotovo YOLOv3 (62,33%) imaju problema sa većim okluzijama i distorzijama oblika objekata koje su posljedica korištenja širokokutne kamere.

Kar i sur. kreiraju Meta-Sim integrirani okvir namijenjen generiranju sintetičkih urbanih okruženja [134] u domeni autonomne vožnje. Pritom koriste SDR [141], ali distribuciju uče iz realnih podataka, relevantnih za rješavanje konkretnog zadatka. Njihov generativni model koristi Maximum Mean Discrepancy (MMD) metriku za usporedbu distribucija, koja im

omogućava optimizirati parametre scene na nivou svakog pojedinog objekta. Na taj način zadržavaju strukturu koju generira probabilistička gramatika, ali transformiraju distribuciju atributa. Čine to zato što smatraju da se problem adaptacije domene sastoji od dvije komponente: izgleda (vizualni stil objekata), kojim su se bavili i prethodnici, ali i sadržaja (raspored i tipovi različitih objekata na sceni), zbog čega uvode pojam "jaz distribucije" (eng. *distribution gap*).

Bailo, Ham i Shin koriste pix2pixHD GAN model iz [168] za generirati sintset mikroskopskih slika krvnih zrnaca [169] i u naknadnu obradu uvode utiskivanje (eng. *embossing*). Iako postižu zadovoljavajuće rezultate koristeći isključivo sintset, kritiziraju upotrebljivost GAN-a u ulozi generatora sintetičkih podataka zbog vremena potrebnog za njegovo dizajniranje, dugog i nestabilnog treninga, te velikih računskih zahtjeva, čime ne mogu opravdati marginalno poboljšanje performansi u ovom specifičnom zadatku.

Burić, Paulin i Ivašić-Kos kreiraju prvi sintset za detekciju objekata u domeni rukometa [5]. Kombiniraju tradicionalno 3D modeliranje s proceduralnim i optimiziraju generator i anotator za produkciju 3 slike HD formata u sekundi. Tako grade jedan od najmanjih sintsetova, koji sadrži samo 984 slike, s kojim utrostručuje prethodno postojeći minimalni realni set i postižu 3 puta bolji rezultat detekcije (25,73% u odnosu na inicijalnih 8,49% mAP) koristeći YOLOv2.

2020

Kong i sur. grade Synthinel-1 [170], sintset namijenjen segmentaciji zgrada na slikama iz zraka. Za postizanje rendomizacije domene koriste 9 različitih arhitekturalnih stilova čime rješavaju problem uočen u [104]. Novi problem s kojim se susreću je veličina potrebnog virtualnog svijeta u svakom kadru jer već 10-20 km^2 urbane sredine gusto populirane 3D objektima postavlja značajne memorijske zahtjeve koje korištenje LOD-ova, kritizirano i u [153], ne može riješiti.

4 Analiza

Analizom 165 radova iz poglavlja 3 (Pregled područja) utvrđena je primjena sintsetova u 31 domeni (Tablica 2). Redak u tablici u kojemu je domena označena zvjezdicom (*) odnosi se na 4 rada koji se bave sintsetovima nevezano uz domenu njihove primjene ([98], [135], [149] i [14]).

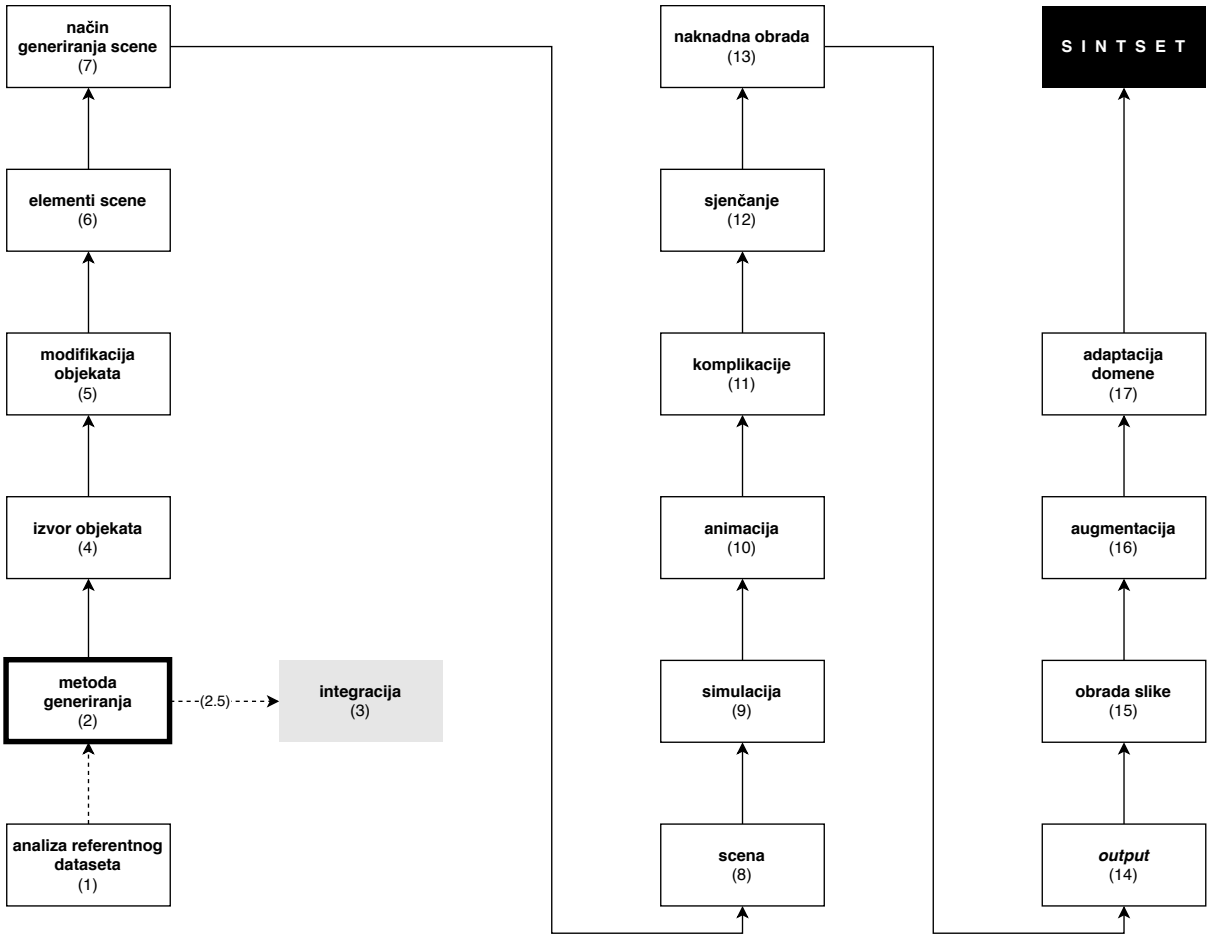
Tablica 2 - zastupljenost domena u analiziranim radovima (N=165)

domena	broj radova
autonomna vožnja	33
detekcija objekata	21
određivanje poze	12
nadziranje	10
optički tok	8
razumijevanje scene	8
autonomno letenje	7
rekonstrukcija scene	6
medicinska segmentacija	5
navigacija robota	5
stereo disparitet	5
*	4
podržano učenje	4
raspoznavanje akcije	4
rekonstrukcija objekta	4
segmentacija	4
raspoznavanje objekta	3
robotski hvat	3
vizualno rasuđivanje	3
evaluacija značajki slike	2
praćenje objekata	2
utjelovljena umjetna inteligencija	2
analiza svjetlosnih polja	1
arhitektura neuronske mreže	1
generiranje tekstura	1
klasifikacija objekata	1
medicinska lokalizacija	1
određivanje gledišta	1
određivanje smjera gledanja	1
virtualna stvarnost	1
vizualna lokalizacija	1
zaštita privatnosti	1

Najveći broj sintsetova (20%) nastao je u domeni autonomne vožnje i utoliko treba imati na umu da je značajan dio metoda i tehnika njihovog generiranja optimiziran upravo za tu primjenu. Sintsetovi posvećeni raspoznavanju akcije slabo su zastupljeni (2,42%), a sintsetovi namijenjeni primjeni u sportu koriste se samo u dva slučaja: po jednom u domeni praćenja objekata (nogomet, [33]) i u domeni detekcije objekata (rukomet, [5]).

4.1 Generalni proces generiranja sintsetova

Iz analiziranih radova može se izlučiti generalni proces generiranja sintsetova, predstavljen na Slici 3.



Slika 3 - Pregled procesa generiranja sintsetova u analiziranim radovima (N=165)

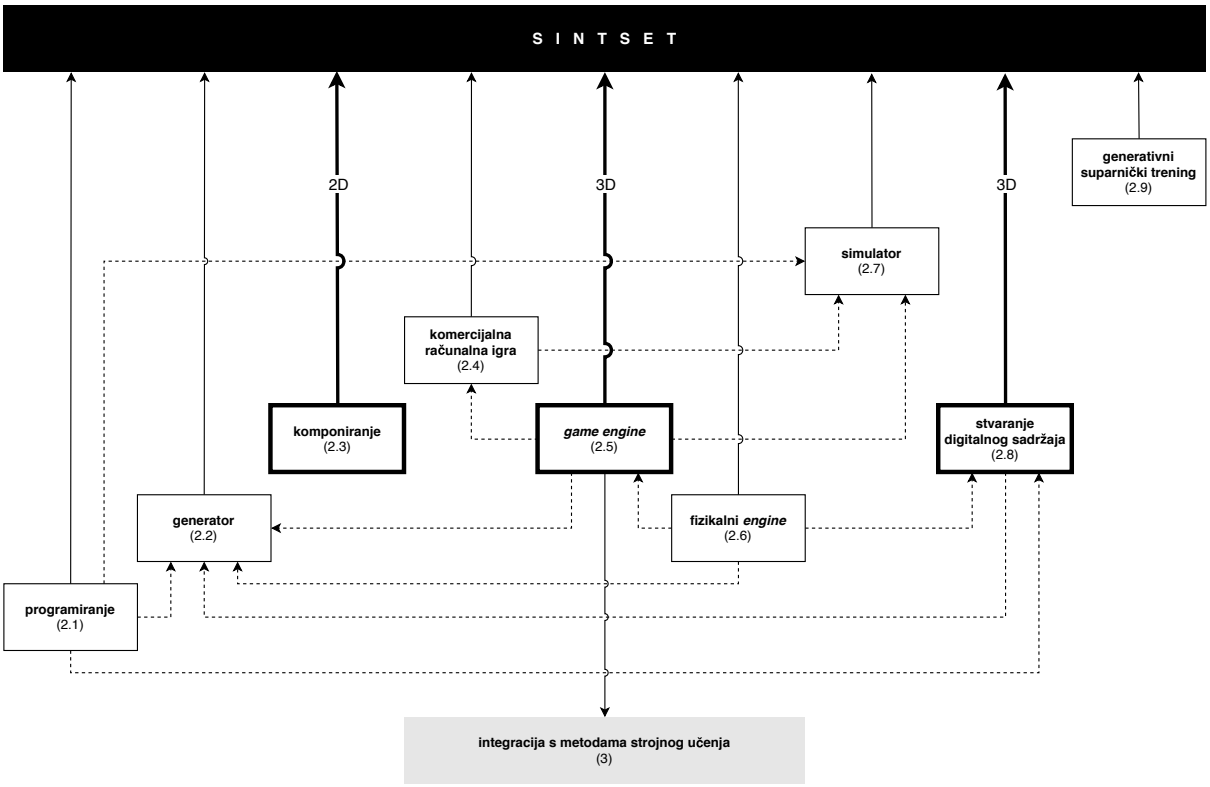
Generalni proces sastoji se od 17 pojedinih procesa pri čemu ih 16 (1 – 2 i dalje 4 – 17) čini linearnu sekvencu. Izuzetak je proces pod brojem 3 (integracija), koji predstavlja alternativni put korištenja sintetiziranih podataka izravnom integracijom metoda generiranja s nekom od platformi za strojno učenje, i kao takav nije predmet interesa ovog rada.

Numeracija korištena za označavanje pojedinih procesa proizlazi iz sistematizacije procesa generiranja sintseta (Prilog 8.4), izgrađene za potrebe ovog rada, prema kronološki prvom pojavljivanju pojedine stavke u radovima obrađenim u poglavlju 3 (Pregled područja) i onim redoslijedom (od 1 do 17) kojim pojedine stavke sudjeluju u procesu generiranja sintseta. Numeracija se koristi i dalje u tekstu, unutar oblih zagrada, za referenciranje pojedinih stavki hijerarhije sistematizacije.

4.2 Procesi

Ukoliko je prilikom izgradnje sintseta raspoloživ referentni dataset, njegovom opcionom analizom (1) moguće je izvesti algoritamsku estimaciju parametara za sjenčanje (12) i utvrditi distribuciju značajki koju je poželjno postići sintsetom.

Prvi obavezan proces je odabir metode generiranja sintseta (2). U analiziranim radovima utvrđeno je 9 različitih metoda (2.1 – 2.9), prikazanih na Slici 4.



Slika 4 - Metode generiranja sintseta

Pojedine metode opciono su sadržane u kompleksnijim metodama, što je označeno isprekidanom crtom na Slici 3. *Game engine* metoda (2.5) omogućava izravnu integraciju s platformama za strojno učenje (3), a istaknute metode (2.3, 2.5 i 2.8) su predloženi kandidati za izgradnju optimalnog sintseta za potrebe raspoznavanja rukometnih akcija – oznake "2D" i "3D" odnose se na tip scene (2D ili 3D) koji pritom nastaje.

U metodi programiranja (2.1) sintset se kreira algoritamski, izravnim programiranjem *outputa*. Metoda korištenja generatora (2.2) oslanja se na postojeći generator koji može prethodno biti isprogramiran ili izgrađen korištenjem alata za stvaranje digitalnog sadržaja, odnosno fizikalnog ili *game enginea*. Generator omogućava promjenu parametara sinteze od strane korisnika, a *output* ne mora nužno generirati u realnom vremenu. Metoda komponiranja (2.3) tretira scenu kao skup 2D slojeva položnih jedan na drugog pri čemu se koriste minimalno 2 sloja (po jedan za stražnji i prednji plan slike). Mogućnost korištenja metode komercijalne računalne igre (2.4) ovisna je o licenci kojom se regulira smije li igra uopće biti korištena u svrhu generiranja sintseta. Ukoliko smije, najveći problem predstavlja način kako pristupiti sadržajima unutar igre, eventualno ih prilagoditi vlastitim potrebama i izvesti u odgovarajućem obliku (finalna slika uz pripadnu anotaciju). Tehničku osnovu svake računalne igre čini *game engine* (2.5), koji je ujedno, kada se koristi kao produkcijski alat, i zasebna metoda kojom se može proizvesti sintset bez da se prethodno proizvede igra. Sastavni dio većine *game enginea* je fizikalni *engine* (2.6), koji može postojati i kao zasebni softver, specijaliziran za pojedinu vrstu fizikalne simulacije, kojim se također može izravno proizvesti sintset, što njegovu primjenu čini zasebnom metodom. Metoda korištenja simulatora (2.7) oslanja se na programiranje dotičnog ili korištenje *game enginea* odnosno modificirane komercijalne računalne igre za provođenje konkretne simulacije. Za razliku od generatora, simulatori *output* najčešće generiraju u realnom vremenu. Metoda stvaranja digitalnih sadržaja (2.8) oslanja se na alate koji često u sebi sadrže fizikalni *engine* i omogućavaju automatizaciju programiranjem. Kronološki najnovija metoda je generativni suparnički trening (2.9) koji koristi neuronske mreže (GAN) tijekom procesa generiranja sadržaja sintseta, ali ograničen je nemogućnošću automatskog kreiranja odgovarajućih anotacija pa je zapravo podobniji za korištenje u ulozi pomoćnog alata za potrebe adaptacije domene (17.4).

Procese 4 – 17, njihovo grananje i konvergenciju pojedinih stavki možemo promatrati na Slici 4.

U slučaju kreiranja 2D scena (8.1) najčešći izvori 2D objekata namijenjenih *blendingu* (7.1.1) su prethodno renderirane slike (4.1.1), koje mogu sadržavati maske za automatsku ekstrakciju objekta od interesa, te fotografije (4.1.2), koje zahtijevaju manualnu ekstrakciju. Kad je riječ o 3D scenama (8.2) 3D objekti mogu biti pronađeni na Internetu (4.2.1), proceduralno (4.2.2) ili manualno (4.2.3) generirani, izgrađeni konverzijom OSM mapa (4.2.4) u 3D geometriju, generirani kao L-System (4.2.5) ili fotogrametrijom (4.2.7) te korišteni kao oblak točaka, dobiven LiDAR-om (4.2.6), kojeg je također moguće konvertirati u 3D geometriju.

3D objekti se nerijetko predprocesiraju prilikom uvođenja na 3D scenu, a njihova modifikacija može biti parametarska (5.1.1), manualna (5.1.2), te svedena na retopologiziranje (5.1.3) ili na skaliranje po različitim osima (5.1.4). Ukoliko su 3D objekti dobiveni LiDAR-om, moguće je odraditi predprocesiranje cjelovitih LiDAR scena (5.1.5).

Dok su tipični elementi (6) 2D scene 2D objekti (6.1) i pozadina (6.5), pozadina (kao 2D objekt) može biti i dio 3D scene, uz 3D objekte (6.2), svjetla (6.3) i barem jedne kamere (6.4).

3D scenu moguće je generirati manualno (7.2.1), proceduralno (7.2.2), fizikalnim simuliranjem (7.2.3), generativnim suparničkim treningom (7.2.4) i koristeći dopunjenu stvarnost (7.2.5).

Iako ih je tehnički moguće provesti i na 2D scenama, simulacija (9) i animacija (10) za potrebe generiranja sintsetova provode se isključivo na 3D scenama. Pritom se simuliraju fizika (9.1), atmosferski uvjeti (9.2), kretanje mnoštva (9.3), promjena godišnjih doba (9.4), perturbacije (9.5) i deformacije modela (9.6). Animacija uključuje *mo-cap* odnosno snimanje pokreta (10.1), manualnu animaciju (10.2), automatske tranzicije između animacija (10.3), predefinirane putanje kretanja (10.4), proceduralnu animaciju (10.5) i animaciju krpene lutke (10.6).

Prva konvergencija 2D i 3D puteva tijekom generiranja sintseta prisutna je u procesu dodavanja komplikacija (11). Komplikacije se najčešće uvode kao neki od oblika okluzije (11.1), a za specifične potrebe, kada su u pitanju video sekvence, i kao nedostajući frejmovi (11.2).

Proces sjenčanja (12) odnosi se isključivo na 3D scene. U njemu se definira vrsta sjenčanja (12.1) te se odabire renderer (12.2), hardver za renderiranje (12.3), postavke *outputa* (12.4) i, ograničeno prethodnim odabirima, način isporuke *outputa* modelu za učenje (12.5).

Do druge konvergencije 2D i 3D puteva dolazi u procesu naknadne obrade (13) kada se generiranoj slici dodaje šum (13.1); obavlja izglađivanje (13.2); uvode distorzija slike (13.3) i, za video sekvence, video *ghosting* (13.4); obavlja *antialiasing* (13.5); dodaju magla (13.6), zamućenje pokreta (13.7) i fokusa (13.8), te odsjaj sunca (13.9); obavlja manipulacija krivulje game (13.10); dodaju vinjeta (13.11), kromatska aberacija (13.12), automatska ekspozicija (13.13) i ambijentalna okluzija (13.14); te simuliraju greške kodeka (13.15) i zaprljanost leće (13.16).

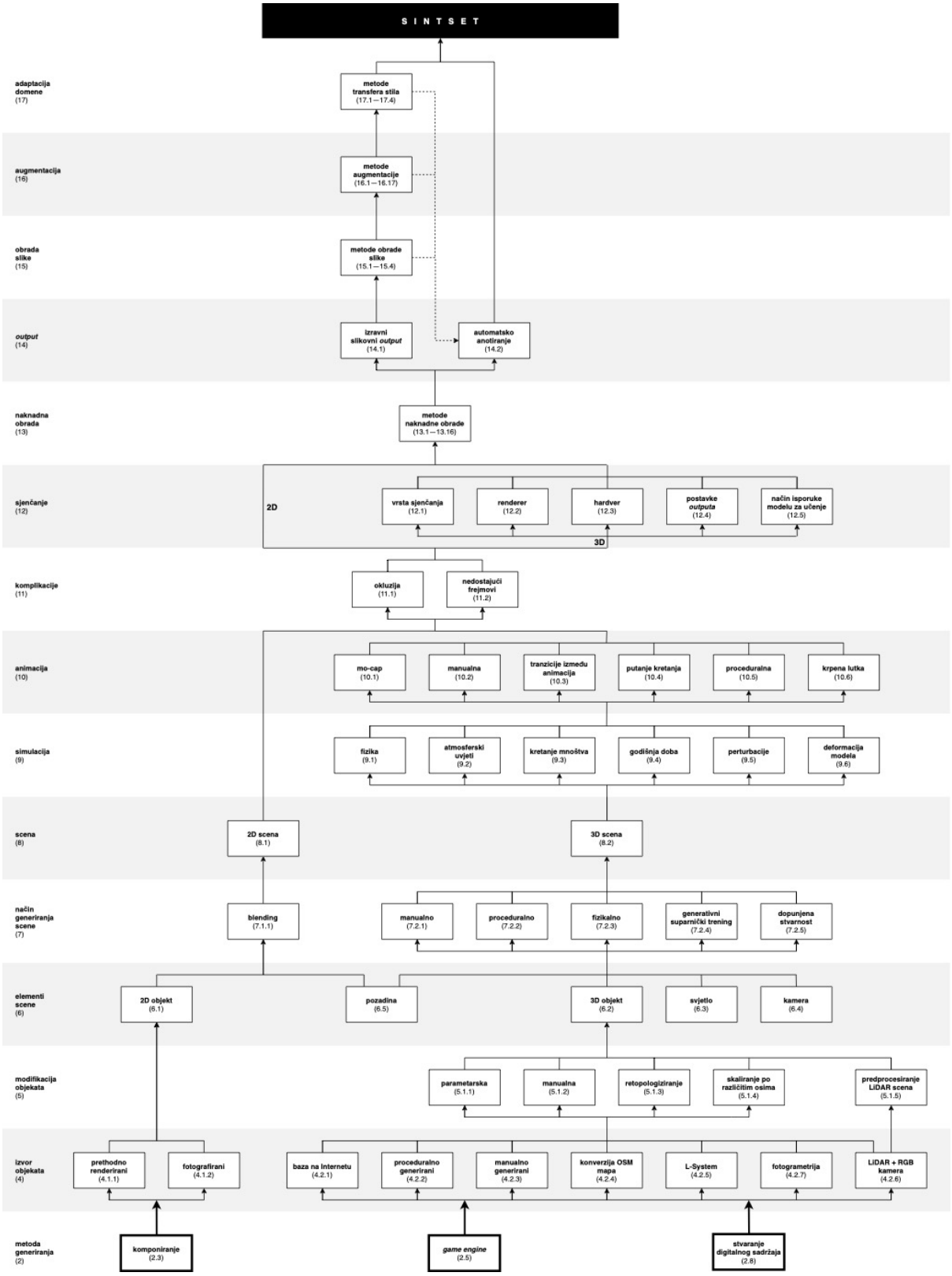
U procesu *outputa* (14) nastaju izravni slikovni *output* (14.1) i automatska anotacija (14.2) na koju može utjecati i rezultat svakog od predstojeća 3 procesa (15 – 17).

Proces obrade slike (15) koristi metode smanjivanja (15.1), rezanja (15.2), uklanjanja pozadine segmentacijom prije treniranja (15.3) i savijanje (15.4).

Ukoliko je osnovni sintset potrebno proširiti augmentacijom (16), koristi se neka od metoda: okluzija (16.1), rezanje (16.2), promjena kontrasta mape dubine (16.3), promjena svjetloće mape dubine (16.4), zamjena bijele boje pozadine u mapi dubine rendom bojom udaljenijom od centra mase objekta (16.5), kosa transformacija (16.6), 2D rotacija (16.7), 3D rotacija (16.8), sakaćenje (16.9), distraktor objekti (16.10), promjena svjetloće (16.11), promjena kontrasta (16.12), zrcaljenje (16.13), homografija (16.14), oštrenje (16.15), utiskivanje (16.16) i inverzija kanala boje (16.17).

Ako se adaptacija domene (17) neće provesti miješanjem sintseta s realnim datasetom (Slika 1), moguće ju je odraditi u sklopu procesa generiranja sintseta, korištenjem nekog u tu svrhu kreiranog integriranog okvira (17.1), transferom učenja (17.2) ili stila (17.3) te suparničkim treningom (17.4).

Tijek navedenih procesa (za metode: komponiranje, *game engine* i stvaranja digitalnog sadržaja), prikazan je na Slici 5.



Slika 5 - Proces generiranja sintseta (za metode: komponiranje, game engine, stvaranje digitalnog sadržaja)

4.3 Zastupljenost domena i metoda generiranja

Analizirani radovi obuhvaćaju 85 proizvedenih, javno dostupnih i imenovanih sintsetova (Tablica 7 - kronološki popis sintsetova), te 12 proizvedenih simulatora (Tablica 8 - kronološki popis simulatora) i 6 generatora (Tablica 9 - kronološki popis generatora).

Promatrajući zastupljenost domena u nabrojenim grupama, možemo uočiti već spomenutu pristranost u korist autonomne vožnje i unutar sintsetova (24,7%). Izrada simulatora minimalno se preferira za navigaciju robota (25%) iako se u značajnom postotku izrađuju i za potrebe autonomne vožnje (12,5%). Izrada generatora minimalno se preferira za medicinsku segmentaciju (33,3%), a za autonomnu vožnju se uopće ne koriste. Jedina domena za koju se istovremeno proizvode sintsetovi, simulatori i generatori je nadziranje.

Tablica 3 - zastupljenost domena u analiziranim sintsetovima (N=85)

domena	broj baza
autonomna vožnja	21
određivanje poze	10
detekcija objekata	6
optički tok	6
nadziranje	5
rekonstrukcija scene	4
autonomno letenje	3
stereo disparitet	3
evaluacija značajki slike	2
medicinska segmentacija	2
navigacija robota	2
praćenje objekata	2
raspoznavanje akcije	2
raspoznavanje objekta	2
razumijevanje scene	2
rekonstrukcija objekta	2
segmentacija	2
vizualno rasuđivanje	2
analiza svjetlosnih polja	1
arhitektura neuronske mreže	1
generiranje tekstura	1
klasifikacija objekata	1
određivanje smjera gledanja	1
robotski hvat	1
vizualna lokalizacija	1

Tablica 4 - zastupljenost domena u analiziranim simulatorima (N=12)

domena	broj simulatora
navigacija robota	3
autonomna vožnja	2
autonomno letenje	2
utjelovljena umjetna inteligencija	2
*	1
nadziranje	1
podržano učenje	1

Tablica 5 - zastupljenost domena u analiziranim generatorima (N=6)

domena	broj generatora
medicinska segmentacija	2
detekcija objekata	1
nadziranje	1
razumijevanje scene	1
stereo disparitet	1

Promatrajući zastupljenost metoda generiranja sintsetova u pojedinim domenama (Tablica 6) uočava se najčešće korištenje metode stvaranja digitalnog sadržaja (35,29%) i to u domenama optičkog toka i rekonstrukcije scene, ali i u domeni raspoznavanja akcije gdje je ujedno i jedina korištena metoda. Popularnost ove metode može se objasniti fleksibilnošću koju alati za stvaranje digitalnog sadržaja pružaju u generiranju scena, neopterećeni imperativom izvođenja u realnom vremenu svojstvenom *game engine* metodi. Uz to, ova metoda omogućava i postizanje najvećeg stupnja fotorealizma.

Druga metoda po učestalosti korištenja je *game engine* (11,76%) koja je, uz metodu korištenja komercijalne računalne igre, ujedno i jedna od preferiranih metoda u domeni autonomne vožnje, a i preferirana metoda u domeni određivanja poze. Njezina ključna prednost, dobrodošla u spomenutim domenama, je mogućnost izvođenja u realnom vremenu, ali nauštrb fotorealizma.

Slabu zastupljenost (1,18%) metoda komponiranja, korištenja fizikalnog *enginea* i kombinacije stvaranja digitalnog sadržaja, korištenja *game enginea* i simulatora može se objasniti isključivom primjenom metode komponiranja u domeni optičkog toka u kojoj je, s vremenom, prevladalo korištenje metode stvaranja digitalnog sadržaja, potom ugradnjom fizikalnog *enginea* u *game engine* i, u slučaju korištenja kombinacije metoda, preferiranjem jednostavnijih procesa produkcije sintsetova od strane većine autora.

Stupac označen upitnikom odnosi se na čak 11,76% radova u kojima autori ne navode metodu generiranja vlastitog sintseta.

Tablica 6 - zastupljenost metoda generiranja sintsetova u pojedinim domenama (N=85)

metode generiranja sintseta / domene	programiranje	komponiranje	komercijalna računalna igra	*game engine*	fizikalni *engine*	simulator	stvaranje digitalnog sadržaja (DCC)	DCC + *game engine* + simulator	?	**Σ**
analiza svjetlosnih polja							1			**1**
arhitektura neuronske mreže									1	**1**
autonomna vožnja	1		6	6		2	3	1	2	**21**
autonomno letenje				1		1			1	**3**
detekcija objekata	1			1		1	3			**6**
evaluacija značajki slike							2			**2**
generiranje tekstura									1	**1**
klasifikacija objekata							1			**1**
medicinska segmentacija	2									**2**
nadziranje			2	1		1	1			**5**
navigacija robota				1					1	**2**
određivanje poze	1			4			3		2	**10**
određivanje smjera gledanja				1						**1**
optički tok		1					4		1	**6**
praćenje objekata	1			1						**2**
raspoznavanje akcije							2			**2**
raspoznavanje objekta				1			1			**2**
razumijevanje scene	1				1					**2**
rekonstrukcija objekta	1						1			**2**
rekonstrukcija scene							4			**4**
robotski hvat	1									**1**
segmentacija							2			**2**
stereo disparitet				2			1			**3**
vizualna lokalizacija				1						**1**
vizualno rasuđivanje							1		1	**2**
Σ	**9**	**1**	**8**	**20**	**1**	**5**	**30**	**1**	10	85

5 Diskusija

Izgradnju optimalnog sintseta za potrebe raspoznavanja rukometnih akcija poželjno je započeti analizom referentnog dataseta s ciljem ugađanja sintseta za konkretan zadatak [77]. S obzirom da će se trenirani model primjenjivati na realnim podacima (video zapis rukometne utakmice), oni su ujedno i dobar izbor referentnog dataseta.

Utjecaj fotorealističnosti sintseta na performanse modela (za testiranje na realnim podacima) ovisi primarno o bliskosti distribucija između sintetičkih i realnih podataka [75], pa je prilikom analize potrebno utvrditi:

- podudarnost virtualnog sa fizičkim okruženjem (dvorana/igralište, igrači, lopta) [112]
- podudarnost tekstura [77]
- podudarnost poza [77], odnosno *mo-capa* [108]
- statistiku kutova pod kojima su snimljeni objekti od interesa [8]
- svojstva kamere [74]
- parametre za sjenčanje [66]
- 3D LUT, ako je prvenstvena uloga adaptacije domene korekcija boja (kao u [151]).

Rezultati analize koriste se u različitim fazama procesa generiranja sintseta i svaka (buduća) promjena unutar referentnog dataseta može značajno utjecati na njih. Kako bi se osigurala trajna primjenjivost sintseta za učenje modela, moguće je izgraditi proceduralnu protočnu strukturu u kojoj rezultati analize predstavljaju ulazne parametre pojedinih procesa, što omogućava automatski generirati optimalan sintset prilikom svake promjene referentnog dataseta.

Od 9 utvrđenih metoda generiranja, 3 konkuriraju za izgradnju optimalnog sintseta za potrebe raspoznavanja rukometnih akcija: stvaranje digitalnog sadržaja, *game engine* i komponiranje. Dok odabir prve dvije metode slijedi prethodno utvrđen trend zastupljenosti metoda generiranja u postojećim sintsetovima i zadovoljava zahtjeve fotorealizma ([97], [133]), komponiranje je odabrano kao treća opcija jer ga je moguće realizirati paralelno uz jednu od prethodne dvije metode, koristeći resurse koji pritom nužno nastaju.

Programiranje nije preporučeno koristiti kao metodu generiranja već isključivo za automatizaciju pojedinih procesa u okviru odabranih metoda. Slično vrijedi i za metodu korištenja generatora koje nije optimalno graditi od nule s obzirom da je potrebna infrastruktura za generiranje već razvijena unutar različitih alata za stvaranje digitalnog sadržaja i *game enginea*. Korištenje komercijalnih računalnih igara nije opcija jer smo, za razliku od primjene u domeni autonomnih vozila, u slučaju rukometa ograničeni na svega 5 igara nastalih u periodu između 2011. i 2018. godine ([171], [172], [173], [174] i [175]), a pritom se susrećemo i sa poznatim tehničkim [117] i pravnim [86] problemima. Metoda korištenja samostalnog fizikalnog *enginea* je isključena jer fizikalna simulacija nije nužna (što isključuje i metodu korištenja simulatora), a ukoliko je želimo koristiti (primjerice, za ispucavanje lopte prilikom dodavanja ili šuta), fizikalni *engine* raspoloživ je unutar alata za stvaranje digitalnog sadržaja i *game enginea*. Metodu generativnog suparničkog treninga izbjegavamo zbog kompleksnosti i problematičnih anotacija [93].

Osnovna prednost korištenja metode stvaranja digitalnog sadržaja, uz postizanje fotorealizma, je neograničena mogućnost upravljanja svim aspektima 3D scene, što je preduvjet učinkovite rendomizacije domene [50]. Generativni karakter alata za stvaranje digitalnog sadržaja omogućava proceduralno generiranje unikatne scene za svaki sintetizirani frejm [118], ali i svakog pojedinog elementa scene, uključno s geometrijom, što se, iz tehničkih razloga, teško može postići koristeći isključivo *game engine*. Dodatne prednosti ove metode su mogućnost distribucije sjenčanja na više računala i mogućnost nelinearnog renderiranja proizvoljnih frejmova. No, u slučaju ograničenosti resursa za sjenčanje, moguće je, nauštrb maksimalnog fotorealizma, primijeniti kompromis te provesti odgovarajuće prilagodbe geometrije i drugih elemenata scene za primjenu unutar *game enginea* i tako realizirati sjenčanje u realnom vremenu, uz dostatni fotorealizam.

Metoda komponiranja ovisi o raspoloživosti elemenata za izgradnju prednjeg (igrači, lopta) i stražnjeg plana (dvorana/igralište). Elemente prednjeg plana moguće je automatski generirati tijekom primjene metode stvaranja digitalnog sadržaja i/ili korištenja *game enginea*, a elementi stražnjeg plana mogu nastati fotografiranjem dvorane/igrališta bez prisustva igrača [109] i lopti, tijekom kreiranja referentnog (realnog) dataseta.

Problem ograničene varijabilnosti sintseta koji nastaje zbog korištenja ograničenog broja gotovih modela [176], umjesto većim brojem takvih modela [8], preporuča se riješiti njihovim proceduralnim generiranjem. Proceduralno generirane objekte moguće je modificirati uzorkovanjem SAOG-a, što se prethodno koristilo isključivo za konfiguriranje scena [105].

Sve objekte na sceni (i one od interesa i distraktore), s naglaskom na odjeću igrača [91], potrebno je teksturirati koristeći što veći broj unikatnih rendom tekstura [127], usklađenih s referentnim datasetom. Pojedinu teksturu nužno je zadržati tijekom izvođenja jednog ciklusa anotirane akcije (npr. skok-šut), koristeći isto sjeme za rendomizaciju [155] u svakom frejmu akcije.

Iako je kosu (igrača) tehnički moguće tretirati kao deformabilni dio 3D modela i, po potrebi, fizikalno simulirati, zahtjevnost takvog postupka, kao i manjak informacija o učinkovitosti istog za potrebe treniranja modela, navode na zaključak da može biti tretirana kao kruti objekt. Isto vrijedi i za loptu, imajući na umu potencijalne deformacije prilikom kolizija. Nije nužno koristiti ni fotorealistični model sjenčanja kože [79].

Scenu, bez obzira da li je smještena u interijer ili eksterijer, poželjno je osvijetliti primarno rasvjetom baziranom na slici s visokim rasponom boja [40], a sekundarno koristeći dodatne umjetne izvore svjetla (pogotovo za interijere i oblikovanje sjena) koje je jednostavnije varirati [32].

Kamere je potrebno uskladiti s kutovima i svojstvima utvrđenim prilikom analize referentnog dataseta, uz dodavanje realističnog šuma i distorzija (npr. kromatske aberacije) u naknadnoj obradi, čime se ispravlja nedostatak uočen u [33].

3D scene, koje nastaju prilikom korištenja metoda stvaranja digitalnog sadržaja i korištenja *game enginea*, poželjno je generirati proceduralno, koristeći strukturalnu rendomizaciju

domene [141], pazeći na realističnost pozicija objekata na sceni. Za automatsko grupiranje objekata, kao i za procjenu slobodnog prostora na sceni, može se koristiti vokselizacija [20].

Prilikom generiranja 2D scena, komponiranjem, nije nužno paziti na realističnost pozicija objekata na sceni već je potrebno osigurati samo realizam na nivou zakrpe i trenirati model koristeći različite modove *blendinga* [116].

Poželjno je izbjeći ručnu animaciju jer dotična ne slijedi nužno zakone fizike [57] odnosno ne replicira prirodni pokret. Umjesto nje preporuča se koristiti *mo-cap* i odgovarajuće inercijalne mjerne uređaje. Kod apliciranja snimljenog pokreta potrebno je paziti da se ne dogodi vremensko rastezanje koje dovodi do nerealistične dinamike kretanja, uočene u [68]. Ukoliko je potrebno povezivati sekvence akcija, potrebno je omogućiti prirodne tranzicije između njih.

S obzirom da je za raspoznavanje rukometnih akcija nužno praćenje objekata, što predstavlja veliki izazov u slučaju dugotrajnih okluzija [148], potrebno je, u ulozi komplikacija, uvesti okluzije, pazeći na značajnu degradaciju koju može izazvati manjak vodoravnih piksela [27].

Prilikom korištenja metode stvaranja digitalnog sadržaja, za sjenčanje se preporuča koristiti PBR renderer [105] i ne više od 40 uzoraka po pikselu [85], a prilikom korištenja *game engine* metode odgođen put sjenčanja [150]. U oba slučaja poželjno je koristiti linearni prostor boje [51] i *antialiasing* [19]. Ciljajući fotorealističnost, potrebno je posvetiti pažnju svjetlosnim fenomenima prisutnim na realnim slikama (zrcalne refleksije, sjene i curenje boje) [63], pri čemu je globalnu iluminaciju moguće aproksimirati koristeći ambijentalne okluzije [51]. Preporuča se snimanje slika u formatima bez gubitka (npr. PNG), uz mogućnost naknadne konverzije u format s gubitkom (JPG). Dobrodošlo je ciljati varijabilne dimenzije *outputa*, u skladu s rezolucijama koju očekuju pojedine mreže, ali pritom treba paziti i na rezoluciju korištenih tekstura, pri čemu može pomoći korištenje proceduralnih tekstura [36].

Primarni *output* moraju biti video sekvence [47], koje sadrže barem jedan ciklus anotirane akcije. Kao osnovna anotacija mogu se koristiti granični okvir oko cijelog objekta i granični okvir oko vidljivog dijela objekta [19], a za renderiranje segmentacijske anotacije može se koristiti uniformno bojanje tekstura objekata koji pripadaju različitim klasama [59] pri čemu je potrebno isključiti rasvjetu, sjene, refleksije, *antialiasing* i MIP mapiranje. Alternativno, s obzirom da se u rukometu nalazi samo po 7 igrača u svakom timu, moguće je za anotiranje igrača (i lopte) koristiti 16-bitnu segmentacijsku masku sivih tonova [68].

Ukoliko je raspoloživ i realni dataset, u svrhu adaptacije domene preporuča se oformiti hibridni dataset [58], a ukoliko se generira više od jednog sintseta koristeći različite metode, preporuča se da niti jedan sintset u hibridnom setu ne bude zastupljen s manje od 40% [129]. Prilikom korištenja hibridnog seta poželjno je učenje modela provesti po fazama, koristeći svaki dio dataseta zasebno, počevši od vizualno jednostavnijih prema kompleksnijima [136], pri čemu stupanj fotorealističnosti može biti mjerilo kompleksnosti.

6 Zaključak

Tijekom protekla tri desetljeća sintsetovi su postali rješenje mnogobrojnih problema vezanih uz pripremu velike količine kvalitetnih podataka za duboko učenje. Uz potencijalno neograničenu količinu podataka koje je moguće brzo i ekonomično generirati, odlikuje ih mogućnost automatiziranog anotiranja. Automatizirano anotiranje ujedno eliminira i potencijalnu ljudsku pogrešku svojstvenu manualnoj anotaciji, koja može ugroziti proces učenja.

U ovom radu istražene su dosadašnje prakse generiranja sintsetova, sistematiziran je proces generiranja i pritom je identificirano 9 različitih metoda generiranja.

Tražeći najbolje prakse za izgradnju optimalnog sintseta za potrebe raspoznavanja rukometnih akcija, odabrane su dvije metode generiranja kao primarne (stvaranje digitalnog sadržaja i *game engine*) i jedna sekundarna (komponiranje), te su dane upute i preporuke za njihovu primjenu.

Za odabrane primarne metode generiranja ne postoji prethodna komparacija učinkovitosti pa se, za utvrditi koja je, i po kojim kriterijima, najbolja u praksi, preporuča provesti eksperiment i, po prvi put, paralelno izgraditi i evaluirati dva primarna ili sva tri predložena sintseta, koristeći pritom predloženo proceduralno generiranje ne samo pojedinih scena već svih elemenata scene.

Tako stvoren komplet rukometnih sintsetova može se koristiti i kao temelj za vrednovanje budućih sintsetova u istoj domeni.

7 Literatura

[1] S. I. Nikolenko, „Synthetic Data for Deep Learning“, str. 1–156, ruj. 2019.

[2] „CVonline: Image Databases“. [Na internetu]. Dostupno na: http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm. [Pristupljeno: 16-tra-2020].

[3] J. Pers, M. Bon, i G. Vuckovic, „CVBASE 06 Dataset“. [Na internetu]. Dostupno na: http://vision.fe.uni-lj.si/cvbase06/dataset.html. [Pristupljeno: 05-velj-2020].

[4] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, i L. Fei-Fei, „Large-scale Video Classification with Convolutional Neural Networks“, u *CVPR*, 2014.

[5] M. Burić, M. Ivašić-Kos, i G. Paulin, „Object Detection Using Synthesized Data“, 2020.

[6] D. a Pomerleau, „Alvinn: An autonomous land vehicle in a neural network“, *Adv. Neural Inf. Process. Syst. 1*, str. 305–313, 1989.

[7] M. Savva *i ostali*, „Habitat: A platform for embodied AI research“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2019-Octob, str. 9338–9346, 2019.

[8] Y. Movshovitz-Attias, T. Kanade, i Y. Sheikh, „How useful is photo-realistic rendering for visual learning?“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 9915 LNCS, izd. September, str. 202–217, 2016.

[9] F. E. Nowruzi, P. Kapoor, D. Kolhatkar, F. Al Hassanat, R. Laganiere, i J. Rebut, „How much real data do we actually need: Analyzing object detection performance using synthetic and real data“, 2019.

[10] S. P. Parker, *McGraw-Hill Dictionary of Scientific and Technical Terms*, 6th ed. New York: McGraw-Hill Education, 2003.

[11] J. J. Little i A. Verri, „Analysis of differential and matching methods for optical flow“, u *[1989] Proceedings. Workshop on Visual Motion*, 1989, str. 173–180.

[12] D. B. Rubin, „Discussion of statistical disclosure limitation“, 1993.

[13] N. Patki, R. Wedge, i K. Veeramachaneni, „The Synthetic Data Vault“, u *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016, str. 399–410.

[14] D. Lange, „Synthetic Data: A Scalable Way to Train Perception Systems“. [Na internetu]. Dostupno na: https://www.nvidia.com/en-us/gtc/on-demand/. [Pristupljeno: 31-svi-2020].

[15] S. Tripathi, S. Chandra, A. Agrawal, A. Tyagi, J. M. Rehg, i V. Chari, „Learning to generate synthetic data via compositing“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, sv. 2019-June, str. 461–470, 2019.

[16] L. Pishchulin, A. Jain, C. Wojek, M. Andriluka, T. Thormählen, i B. Schiele, „Learning people detection models from few training samples“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, str. 1473–1480, 2011.

[17] R. Queiroz, M. Cohen, J. L. Moreira, A. Braun, J. C. Jacques, i S. R. Musse, „Generating facial ground truth with synthetic faces“, *Proc. - 23rd SIBGRAPI Conf. Graph. Patterns Images, SIBGRAPI 2010*, izd. August, str. 25–31, 2010.

[18] R. Rosales, V. Athitsos, L. Sigal, i S. Sclaroff, „3D hand pose reconstruction using specialized mappings“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 1, izd. 2000, str. 378–387, 2001.

[19] G. R. Taylor, A. J. Chosak, i P. C. Brewer, „OVVV: Using virtual worlds to design and evaluate surveillance systems“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2007.

[20] S. Satkin, J. Lin, i M. Hebert, „Data-driven scene understanding from 3D models“, *BMVC 2012 - Electron. Proc. Br. Mach. Vis. Conf. 2012*, str. 1–11, 2012.
[21] S. Shah, D. Dey, C. Lovett, i A. Kapoor, „AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles“, str. 621–635, 2018.
[22] W. T. Freeman, E. C. Pasztor, i O. T. Carmichael, „Learning Low-Level Vision“, *Int. J. Comput. Vis.*, sv. 40, izd. 1, str. 25–47, 2000.
[23] G. Hamarneh, „Deformable Spatio-Temporal Shape Modeling“, 1999.
[24] N. Koenig i A. Howard, „Design and use paradigms for Gazebo, an open-source multi-robot simulator“, *2004 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, sv. 3, str. 2149–2154, 2004.
[25] M. Peris, S. Martull, A. Maki, Y. Ohkawa, i K. Fukui, „Towards a simulation driven stereo vision system“, *Proc. - Int. Conf. Pattern Recognit.*, izd. December, str. 1038–1042, 2012.
[26] V. Mnih *i ostali*, „Playing Atari with Deep Reinforcement Learning“, str. 1–9, 2013.
[27] H. Ragheb, S. Velastin, P. Remagnino, i T. Ellis, „ViHASi: Virtual human action silhouette data for the performance evaluation of silhouette-based action recognition methods“, *2008 2nd ACM/IEEE Int. Conf. Distrib. Smart Cameras, ICDSC 2008*, izd. September, 2008.
[28] Z. Wu, S. Song, A. Khosla, X. Tang, i J. Xiao, „3D ShapeNets for 2.5D Object Recognition and Next-Best-View Prediction“, lip. 2014.
[29] S. Zambanini i M. Kampel, „Evaluation of low-level image representations for illumination-insensitive recognition of textureless objects“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 8156 LNCS, izd. PART 1, str. 71–80, 2013.
[30] J. Tobin *i ostali*, „Domain Randomization and Generative Models for Robotic Grasping“, *IEEE Int. Conf. Intell. Robot. Syst.*, str. 3482–3489, 2018.
[31] J. Johnson, L. Fei-Fei, B. Hariharan, C. L. Zitnick, L. Van Der Maaten, i R. Girshick, „CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning“, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, sv. 2017-Janua, str. 1988–1997, 2017.
[32] B. Kaneva, A. Torralba, i W. T. Freeman, „Evaluation of image features using a photorealistic virtual world“, *Proc. IEEE Int. Conf. Comput. Vis.*, str. 2282–2289, 2011.
[33] X. Desurmont, J. B. Hayet, J. F. Delaigle, J. Piater, i B. Macq, „Trictrac video dataset: Public hdtv synthetic soccer video sequences with ground truth“, *Work. Comput. Vis. Based Anal. Sport Environ.*, str. 92–100, 2006.
[34] K. Honauer, O. Johannsen, D. Kondermann, i B. Goldluecke, „A dataset and evaluation methodology for depth estimation on 4D light fields“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 10113 LNCS, str. 19–34, 2017.
[35] D. Tabernik, M. Kristan, J. L. Wyatt, i A. Leonardis, „Towards deep compositional networks“, *Proc. - Int. Conf. Pattern Recognit.*, sv. 0, str. 3470–3475, 2016.
[36] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, i A. Bousseau, „Single-image SVBRDF capture with a rendering-aware deep network“, *ACM Trans. Graph.*, sv. 37, izd. 4, 2018.
[37] F. M. Carlucci, P. Russo, i B. Caputo, „A deep representation for depth images from synthetic data“, *Proc. - IEEE Int. Conf. Robot. Autom.*, str. 1362–1369, 2017.
[38] T. Heimann, P. Mountney, M. John, i R. Ionasec, „Learning without labeling: Domain

adaptation for ultrasound transducer localization“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 8151 LNCS, izd. PART 3, str. 49–56, 2013.

[39] H. Su, C. R. Qi, Y. Li, i L. J. Guibas, „Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2015 Inter, str. 2686–2694, 2015.

[40] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, i A. Bulling, „Learning an Appearance-Based Gaze Estimator from One Million Synthesised Images“, u *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, 2016, str. 131–138.

[41] J. Lin, X. Guo, J. Shao, C. Jiang, Y. Zhu, i S.-C. Zhu, „A Virtual Reality Platform for Dynamic Human-Scene Interaction“, u *SIGGRAPH ASIA 2016 Virtual Reality Meets Physical Reality: Modelling and Simulating Virtual Humans and Environments*, 2016.

[42] P. Weinzaepfel, G. Csurka, Y. Cabon, i M. Humenberger, „Visual Localization by Learning Objects-Of-Interest Dense Match Regression“, u *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, str. 5627–5636.

[43] C. Piciarelli, C. Micheloni, i G. L. Foresti, „Trajectory-based anomalous event detection“, *IEEE Trans. Circuits Syst. Video Technol.*, sv. 18, izd. 11, str. 1544–1554, 2008.

[44] N. Habeeb i S. Hasson Aljebori, „Improving Video Streams Summarization Using Synthetic Noisy Video Data“, *Int. J. Adv. Comput. Sci. Appl.*, sv. 6, pros. 2015.

[45] S. Elanattil, P. Moghadam, S. Sridharan, C. Fookes, i M. Cox, „Non-rigid Reconstruction with a Single Moving RGB-D Camera“, *Proc. - Int. Conf. Pattern Recognit.*, sv. 2018-Augus, str. 1049–1055, 2018.

[46] K. Mo *i ostali*, „Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, sv. 2019-June, str. 909–918, 2019.

[47] J. L. Barron, D. J. Fleet, i S. S. Beauchemin, „Systems and Experiment Performance of optical flow techniques“, *Int. J. Comput. Vis.*, sv. 12, izd. 1, str. 43–77, 1994.

[48] K. Grauman, G. Shakhnarovich, i T. Darrell, „Inferring 3D structure with a statistical image-based shape model“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 1, izd. Iccv, str. 641–648, 2003.

[49] A. Saxena, J. Driemeyer, J. Kearns, i A. Y. Ng, „Robotic grasping of novel objects“, *Adv. Neural Inf. Process. Syst.*, str. 1209–1216, 2007.

[50] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, i P. Abbeel, „Domain randomization for transferring deep neural networks from simulation to the real world“, *IEEE Int. Conf. Intell. Robot. Syst.*, sv. 2017-Septe, str. 23–30, 2017.

[51] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, i R. Szeliski, „A database and evaluation methodology for optical flow“, *Int. J. Comput. Vis.*, sv. 92, izd. 1, str. 1–31, 2011.

[52] J. P. Tarel, N. Hautière, A. Cord, D. Gruyer, i H. Halmaoui, „Improved visibility of road scene images under heterogeneous fog“, *IEEE Intell. Veh. Symp. Proc.*, str. 478–485, 2010.

[53] J. Marín, D. Vázquez, D. Gerónimo, i A. M. López, „Learning appearance in virtual scenarios for pedestrian detection“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, str. 137–144, 2010.

[54] G. Hamarneh i P. Jassi, „VascuSynth: Simulating vascular trees for generating

volumetric image data with ground-truth segmentation and tree analysis“, *Comput. Med. Imaging Graph.*, sv. 34, izd. 8, str. 605–616, 2010.
[55] J. Shotton *i ostali*, „Real-Time human pose recognition in parts from single depth images“, *Commun. ACM*, sv. 56, izd. 1, str. 116–124, 2013.
[56] A. Vacavant, T. Chateau, A. Wilhelm, i L. Lequièvre, „A Benchmark Dataset for Outdoor Foreground/Background Extraction“, sv. 7728, izd. November 2012, 2013, str. 291–300.
[57] D. J. Butler, J. Wulff, G. B. Stanley, i M. J. Black, „A Naturalistic Open Source Movie for Optical Flow Evaluation“, 2012, str. 611–625.
[58] B. Pepik, M. Stark, P. Gehler, i B. Schiele, „Teaching 3D geometry to deformable part models“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, izd. June, str. 3362–3369, 2012.
[59] V. Haltakov, C. Unger, i S. Ilic, „Framework for Generation of Synthetic Ground Truth Data for Driver Assistance Applications BT - Pattern Recognition“, 2013, str. 323–332.
[60] K. M. Henry, L. Pase, C. F. Ramos-Lopez, G. J. Lieschke, S. A. Renshaw, i C. C. Reyes-Aldasoro, „PhagoSight: An Open-Source MATLAB® Package for the Analysis of Fluorescent Neutrophil and Macrophage Migration in a Zebrafish Model“, *PLoS One*, sv. 8, izd. 8, 2013.
[61] R. Haeusler i D. Kondermann, „Synthesizing Real World Stereo Challenges BT - Pattern Recognition“, 2013, str. 164–173.
[62] B. Moiseev, A. Konev, A. Chigorin, i A. Konushin, „Evaluation of traffic sign recognition methods trained on synthetically generated data“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 8192 LNCS, str. 576–583, 2013.
[63] A. Handa, T. Whelan, J. McDonald, i A. J. Davison, „A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM“, *Proc. - IEEE Int. Conf. Robot. Autom.*, str. 1524–1531, 2014.
[64] D. Vazquez, A. M. Lopez, J. Marin, D. Ponsa, i D. Geronimo, „Virtual and real world adaptationfor pedestrian detection“, *IEEE Trans. Pattern Anal. Mach. Intell.*, sv. 36, izd. 4, str. 797–809, 2014.
[65] J. Xu, D. Vázquez, A. López, J. Marín, i D. Ponsa, „Learning a Part-Based Pedestrian Detector in a Virtual World“, *IEEE Trans. Intell. Transp. Syst.*, sv. 15, lis. 2014.
[66] A. Rozantsev, V. Lepetit, i P. Fua, „On rendering synthetic images for training an object detector“, *Comput. Vis. Image Underst.*, sv. 137, str. 24–37, 2015.
[67] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, i J. Sivic, „Seeing 3D Chairs: Exemplar Part-Based 2D-3D Alignment Using a Large Dataset of CAD Models“, u *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, str. 3762–3769.
[68] N. Courty, P. Allain, C. Creusot, i T. Corpetti, „Using the Agoraset dataset: Assessing for the quality of crowd video analysis methods“, *Pattern Recognit. Lett.*, sv. 44, str. 161–170, 2014.
[69] B. Sun i K. Saenko, *From Virtual to Reality: Fast Adaptation of Virtual Object Detectors to Real Domains*. 2014.
[70] N. Mayer *i ostali*, „A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, sv. 2016-Decem, str. 4040–4048, 2016.
[71] A. Dosovitskiy *i ostali*, „FlowNet: Learning optical flow with convolutional networks“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2015 Inter, str. 2758–2766, 2015.

[72] J. Rivera-Rubio, I. Alexiou, i A. A. Bharath, „Appearance-based indoor localization: A comparison of patch descriptor performance“, *Pattern Recognit. Lett.*, sv. 66, str. 109–117, 2015.
[73] C. Chen, A. Seff, A. Kornhauser, i J. Xiao, „DeepDriving: Learning affordance for direct perception in autonomous driving“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2015 Inter, izd. Figure 1, str. 2722–2730, 2015.
[74] H. Hattori, V. N. Boddeti, K. Kitani, i T. Kanade, „Learning scene-specific pedestrian detectors without real data“, u *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, str. 3819–3827.
[75] V. S. R. Veeravasarapu, R. N. Hota, C. Rothkopf, i R. Visvanathan, „Model Validation for Vision Systems via Graphics Simulation“, 2015.
[76] J. Papon i M. Schoeler, „Semantic pose using deep networks trained on synthetic RGB-D“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2015 Inter, str. 774–782, 2015.
[77] X. Peng, B. Sun, K. Ali, i K. Saenko, „Learning deep object detectors from 3D models“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2015 Inter, str. 1278–1286, 2015.
[78] A. Handa, V. Patraucean, S. Stent, i R. Cipolla, „Scenenet: An annotated model generator for indoor scene understanding“, *Proc. - IEEE Int. Conf. Robot. Autom.*, sv. 2016-June, str. 5737–5743, 2016.
[79] E. Richardson, M. Sela, i R. Kimmel, „3D face reconstruction by learning from synthetic data“, *Proc. - 2016 4th Int. Conf. 3D Vision, 3DV 2016*, str. 460–467, 2016.
[80] M. Mueller, N. Smith, i B. Ghanem, „A Benchmark and Simulator for UAV Tracking BT - Computer Vision – ECCV 2016“, 2016, str. 445–461.
[81] F. Massa, B. C. Russell, i M. Aubry, „Deep exemplar 2D-3D detection by adapting from real to rendered views“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, sv. 2016-Decem, str. 6024–6033, 2016.
[82] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, i R. Vasudevan, „Driving in the Matrix: Can virtual worlds replace human-generated annotations for real world tasks?“, *Proc. - IEEE Int. Conf. Robot. Autom.*, str. 746–753, 2017.
[83] E. Cheung, T. K. Wong, A. Bera, X. Wang, i D. Manocha, „LCrowdV: Generating labeled videos for simulation-based crowd behavior learning“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 9914 LNCS, str. 709–727, 2016.
[84] A. Lerer, S. Gross, i R. Fergus, „Learning physical intuition of block towers by example“, *33rd Int. Conf. Mach. Learn. ICML 2016*, sv. 1, str. 648–656, 2016.
[85] V. S. R. Veeravasarapu, C. Rothkopf, i V. Ramesh, „Model-driven Simulations for Deep Convolutional Neural Networks“, str. 1–10, 2016.
[86] A. Shafaei, J. J. Little, i M. Schmidt, „Play and learn: Using video games to train computer vision models“, *Br. Mach. Vis. Conf. 2016, BMVC 2016*, sv. 2016-Septe, str. 26.1-26.13, 2016.
[87] S. R. Richter, V. Vineet, S. Roth, i V. Koltun, „Playing for data: Ground truth from computer games“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 9906 LNCS, str. 102–118, 2016.
[88] A. Mahendran, H. Bilen, J. F. Henriques, i A. Vedaldi, „ResearchDoom and CocoDoom: Learning Computer Vision with Games“, 2016.
[89] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, i W. Jaskowski, „ViZDoom: A Doom-based AI research platform for visual reinforcement learning“, *IEEE Conf. Comput. Intell. Games, CIG*, sv. 0, 2016.

[90] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, i T. Funkhouser, „Semantic scene completion from a single depth image“, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, sv. 2017-Janua, str. 190–198, 2017.
[91] W. Chen *i ostali*, „Synthesizing training images for boosting human 3D pose estimation“, *Proc. - 2016 4th Int. Conf. 3D Vision, 3DV 2016*, str. 479–488, 2016.
[92] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, i A. M. Lopez, „The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, sv. 2016-Decem, izd. 600388, str. 3234–3243, 2016.
[93] Y. Tian, X. Li, K. Wang, i F. Y. Wang, „Training and testing object detectors with virtual images“, *IEEE/CAA J. Autom. Sin.*, sv. 5, izd. 2, str. 539–546, 2018.
[94] Y. Zhang, W. Qiu, Q. Chen, X. Hu, i A. Yuille, „UnrealStereo: A Synthetic Dataset for Analyzing Stereo Vision“, pros. 2016.
[95] A. Gaidon, Q. Wang, Y. Cabon, i E. Vig, „VirtualWorlds as Proxy for Multi-object Tracking Analysis“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, sv. 2016-Decem, str. 4340–4349, 2016.
[96] E. Bochinski, V. Eiselein, i T. Sikora, „Training a convolutional neural network for multi-class object detection using solely virtual world data“, u *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2016, str. 278–285.
[97] Y. Zhang *i ostali*, „Physically-based rendering for indoor scene understanding using convolutional neural networks“, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, sv. 2017-Janua, str. 5057–5065, 2017.
[98] W. Qiu i A. Yuille, „UnrealCV: Connecting computer vision to unreal engine“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 9915 LNCS, str. 909–916, 2016.
[99] J. McCormac, A. Handa, S. Leutenegger, i A. J. Davison, „SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2017-Octob, str. 2697–2706, 2017.
[100] E. Kolve *i ostali*, „AI2-THOR: An Interactive 3D Environment for Visual AI“, str. 2–5, 2017.
[101] Y. Zhu *i ostali*, „Target-driven visual navigation in indoor scenes using deep reinforcement learning“, *Proc. - IEEE Int. Conf. Robot. Autom.*, str. 3357–3364, 2017.
[102] C. Mitash, K. E. Bekris, i A. Boularias, „A self-supervised learning system for object detection using physics simulation and multi-view pose estimation“, *IEEE Int. Conf. Intell. Robot. Syst.*, sv. 2017-Septe, str. 545–551, 2017.
[103] V. S. R. Veeravasarapu, C. Rothkopf, i R. Visvanathan, „Adversarially tuned scene generation“, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, sv. 2017-Janua, str. 6441–6449, 2017.
[104] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, i V. Koltun, „CARLA: An Open Urban Driving Simulator“, izd. CoRL, str. 1–16, 2017.
[105] C. Jiang *i ostali*, „Configurable, Photorealistic Image Rendering and Ground Truth Synthesis by Sampling Stochastic Grammars Representing Indoor Scenes“, ožu. 2017.
[106] B. Planche *i ostali*, „DepthSynth: Real-Time Realistic Synthetic Data Generation from CAD Models for 2.5D Recognition“, *Proc. - 2017 Int. Conf. 3D Vision, 3DV 2017*, izd. May, str. 1–10, 2018.
[107] A. Larumbe, M. Ariz, J. J. Bengoechea, R. Segura, R. Cabeza, i A. Villanueva, „Improved

Strategies for HPE Employing Learning-by-Synthesis Approaches“, *Proc. - 2017 IEEE Int. Conf. Comput. Vis. Work. ICCVW 2017*, sv. 2018-Janua, str. 1545–1554, 2017.
[108] G. Varol *i ostali*, „Learning from synthetic humans“, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, sv. 2017-Janua, str. 4627–4635, 2017.
[109] C. Zimmermann i T. Brox, „Learning to Estimate 3D Hand Pose from Single RGB Images“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2017-Octob, str. 4913–4921, 2017.
[110] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, i V. Koltun, „MINOS: Multimodal Indoor Simulator for Navigation in Complex Environments“, str. 1–14, 2017.
[111] P. S. Rajpura, H. Bojinov, i R. S. Hegde, „Object Detection Using Deep CNNs Trained on Synthetic Images“, 2017.
[112] S. R. Richter, Z. Hayder, i V. Koltun, „Playing for Benchmarks“, ruj. 2017.
[113] C. R. De Souza, A. Gaidon, Y. Cabon, i A. M. López, „Procedural generation of videos to train deep action recognition networks“, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, sv. 2017-Janua, str. 2594–2604, 2017.
[114] J. Schöning, T. Behrens, P. Faion, P. Kheiri, G. Heidemann, i U. Krumnack, „Structure from Motion by Artificial Neural Networks BT - Image Analysis“, 2017, str. 146–158.
[115] G. Georgakis, A. Mousavian, A. C. Berg, i J. Košecká, „Synthesizing training data for object detection in indoor scenes“, *Robot. Sci. Syst.*, sv. 13, 2017.
[116] D. Dwibedi, I. Misra, i M. Hebert, „Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2017-Octob, str. 1310–1319, 2017.
[117] M. Müller, V. Casser, J. Lahoud, N. Smith, i B. Ghanem, „Sim4CV: A Photo-Realistic Simulator for Computer Vision Applications“, *Int. J. Comput. Vis.*, sv. 126, izd. 9, str. 902–919, 2018.
[118] A. Tsirikoglou, J. Kronander, M. Wrenninge, i J. Unger, „Procedural Modeling and Physically Based Rendering for Synthetic Data Generation in Automotive Applications“, 2017.
[119] A. M. López, J. Xu, J. L. Gómez, D. Vázquez, i G. Ros, „From virtual to real world visual perception using domain adaptation—The DPM as example“, *Adv. Comput. Vis. Pattern Recognit.*, izd. 9783319583464, str. 243–258, 2017.
[120] R. Madaan, D. Maturana, i S. Scherer, „Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles“, u *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, str. 3487–3494.
[121] Y. Wu, Y. Wu, G. Gkioxari, i Y. Tian, „Building generalizable agents with a realistic and rich 3D environment“, *6th Int. Conf. Learn. Represent. ICLR 2018 - Work. Track Proc.*, str. 1–15, 2018.
[122] N. Hesse, C. Bodensteiner, M. Arens, U. G. Hofmann, R. Weinberger, i A. Sebastian Schroeder, „Computer vision for medical infant motion analysis: State of the art and RGB-D data set“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 11134 LNCS, str. 32–49, 2019.
[123] D. Ward, P. Moghadam, i N. Hudson, „Deep leaf segmentation using synthetic data“, *Br. Mach. Vis. Conf. 2018, BMVC 2018*, 2019.
[124] J. Ubbens, M. Cieslak, P. Prusinkiewicz, i I. Stavness, „The use of plant models in deep learning: An application to leaf counting in rosette plants“, *Plant Methods*, sv. 14, izd. 1, str. 1–10, 2018.
[125] S. Bąk, P. Carr, i J. F. Lalonde, „Domain Adaptation Through Synthesis for

Unsupervised Person Re-identification“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 11217 LNCS, str. 193–209, 2018.
[126] F. S. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, i J. M. Alvarez, „Effective Use of Synthetic Data for Urban Scene Semantic Segmentation“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 11206 LNCS, str. 86–103, 2018.
[127] J. Tremblay *i ostali*, „Training deep networks with synthetic data: Bridging the reality gap by domain randomization“, *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, sv. 2018-June, str. 1082–1090, 2018.
[128] J. Tremblay, T. To, i S. Birchfield, „Falling things: A synthetic dataset for 3D object detection and pose estimation“, *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, sv. 2018-June, str. 2119–2122, 2018.
[129] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, i S. Birchfield, „Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects“, izd. CoRL, str. 1–11, 2018.
[130] D. V. Sorokin *i ostali*, „FiloGen: A Model-Based Generator of Synthetic 3-D Time-Lapse Sequences of Single Motile Cells with Growing and Branching Filopodia“, *IEEE Trans. Med. Imaging*, sv. 37, izd. 12, str. 2630–2641, 2018.
[131] H. Rahmani, A. Mian, i M. Shah, „Learning a Deep Model for Human Action Recognition from Novel Viewpoints“, *IEEE Trans. Pattern Anal. Mach. Intell.*, sv. 40, izd. 3, str. 667–681, 2018.
[132] A. Atapour-Abarghouei i T. P. Breckon, „Real-Time Monocular Depth Estimation Using Synthetic Data with Domain Adaptation via Image Style Transfer“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, str. 2800–2810, 2018.
[133] M. Wrenninge i J. Unger, „Synscapes: A Photorealistic Synthetic Dataset for Street Scene Parsing“, 2018.
[134] A. Kar *i ostali*, „Meta-sim: Learning to generate synthetic datasets“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2019-Octob, str. 4550–4559, 2019.
[135] K.-T. Lai, C.-C. Lin, C.-Y. Kang, M.-E. Liao, i M.-S. Chen, *VIVID: Virtual Environment for Visual Deep Learning*. 2018.
[136] N. Mayer *i ostali*, „What Makes Good Synthetic Training Data for Learning Disparity and Optical Flow Estimation?“, *Int. J. Comput. Vis.*, sv. 126, izd. 9, str. 942–960, 2018.
[137] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, i C. Rother, „Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes“, *Int. J. Comput. Vis.*, sv. 126, izd. 9, str. 961–972, 2018.
[138] D. Ludl, T. Gulde, S. Thalji, i C. Curio, „Using Simulation to Improve Human Pose Estimation for Corner Cases“, *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, sv. 2018-Novem, str. 3575–3582, 2018.
[139] M. Khodabandeh, H. R. V. Joze, I. Zharkov, i V. Pradeep, „DIY human action dataset generation“, *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, sv. 2018-June, str. 1529–1539, 2018.
[140] L. O. Rojas-perez *i ostali*, „Real-Time Landing Zone Detection for UAVs using Single Aerial Images“.
[141] A. Prakash *i ostali*, „Structured domain randomization: Bridging the reality gap by context-aware synthetic data“, *Proc. - IEEE Int. Conf. Robot. Autom.*, sv. 2019-May, str. 7249–7255, 2019.

[142] N. Dvornik, J. Mairal, i C. Schmid, „Modeling visual context is key to augmenting object detection datasets“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 11216 LNCS, str. 375–391, 2018.
[143] R. Tylecek *i ostali*, „The Second Workshop on 3D Reconstruction Meets Semantics: Challenge Results Discussion BT - Computer Vision – ECCV 2018 Workshops“, 2019, str. 631–644.
[144] A. Pumarola, J. Sanchez, G. P. T. Choi, A. Sanfeliu, i F. Moreno, „3Dpeople: Modeling the geometry of dressed humans“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2019-Octob, str. 2242–2251, 2019.
[145] E. Bayraktar, C. B. Yigit, i P. Boyraz, „A hybrid image dataset toward bridging the gap between real and simulation environments for robotics: Annotated desktop objects real and synthetic images dataset: ADORESet“, *Mach. Vis. Appl.*, sv. 30, izd. 1, str. 23–40, 2019.
[146] W. Li *i ostali*, *AADS: Augmented Autonomous Driving Simulation using Data-driven Algorithms*. 2019.
[147] S. Krishnan, B. Borojerdian, W. Fu, A. Faust, i V. J. Reddi, „Air Learning: An AI Research Platform for Algorithm-Hardware Benchmarking of Autonomous Aerial Robots“, 2019.
[148] R. Girdhar i D. Ramanan, „CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning“, str. 1–16, 2019.
[149] K. Mason, S. Vejdan, i S. Grijalva, „An ‚On the Fly‘ Framework for Efficiently Generating Synthetic Big Data Sets“, *Proc. - 2019 IEEE Int. Conf. Big Data, Big Data 2019*, str. 3379–3387, 2019.
[150] Q. Wang, S. Zheng, Q. Yan, F. Deng, K. Zhao, i X. Chu, *IRS: A Large Synthetic Indoor Robotics Stereo Dataset for Disparity and Surface Normal Estimation*. 2019.
[151] Q. Wang, J. Gao, W. Lin, i Y. Yuan, „Learning from synthetic data for crowd counting in the wild“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, sv. 2019-June, str. 8190–8199, 2019.
[152] P. Solovev *i ostali*, „Learning State Representations in Complex Systems with Multimodal Data“, str. 1–10, 2018.
[153] M. Fonder i M. Droogenbroeck, *Mid-Air: A Multi-Modal Dataset for Extremely Low Altitude Drone Flights*. 2019.
[154] J. Jung, „Synthetic data generation for camera-based perception“.
[155] M. Chociej, P. Welinder, i L. Weng, „ORRB -- OpenAI Remote Rendering Backend“, 2019.
[156] S. Sharma, C. Beierle, i S. D'Amico, „Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks“, u *2018 IEEE Aerospace Conference*, 2018, str. 1–12.
[157] B. Hurl, K. Czarnecki, i S. Waslander, „Precise synthetic image and LiDAR (PreSIL) dataset for autonomous vehicle perception“, *IEEE Intell. Veh. Symp. Proc.*, sv. 2019-June, str. 2522–2529, 2019.
[158] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, i A. L. Sangiovanni-Vincentelli, „A LiDAR point cloud generator: From a virtual world to autonomous driving“, *ICMR 2018 - Proc. 2018 ACM Int. Conf. Multimed. Retr.*, str. 458–464, 2018.
[159] S. Khan, B. Phan, R. Salay, i K. Czarnecki, „CVPR Workshops - ProcSy: Procedural Synthetic Dataset Generation Towards Influence Factor Studies Of Semantic Segmentation Networks“, str. 88–96, 2019.

[160] M. Jalal, J. Spjut, B. Boudaoud, i M. Betke, *SIDOD: A Synthetic Image Dataset for 3D Object Pose Recognition with Distractors*. 2019.
[161] N. Zioulis, A. Karakottas, D. Zarpalas, F. Alvarez, i P. Daras, „Spherical View Synthesis for Self-Supervised 360° Depth Estimation“, *Proc. - 2019 Int. Conf. 3D Vision, 3DV 2019*, str. 690–699, 2019.
[162] S. Yogamani *i ostali*, „WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2019-Octob, str. 9307–9317, 2019.
[163] D. Temel, M.-H. Chen, i G. AlRegib, „Traffic Sign Detection Under Challenging Conditions: A Deeper Look into Performance Variations and Spectral Characteristics“, *IEEE Trans. Intell. Transp. Syst.*, str. 1–11, 2019.
[164] X. Xie *i ostali*, „Vrgym: A virtual testbed for physical and interactive AI“, *ACM Int. Conf. Proceeding Ser.*, 2019.
[165] D. Tome, P. Peluse, L. Agapito, i H. Badino, „XR-EgoPose: Egocentric 3D human pose from an HMD camera“, *Proc. IEEE Int. Conf. Comput. Vis.*, sv. 2019-Octob, str. 7727–7737, 2019.
[166] S. Hinterstoisser, V. Lepetit, P. Wohlhart, i K. Konolige, „On pre-trained image features and synthetic images for deep learning“, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, sv. 11129 LNCS, str. 682–697, 2019.
[167] K. Wang, F. Shi, W. Wang, Y. Nan, i S. Lian, „Synthetic Data Generation and Adaption for Object Detection in Smart Vending Machines“, 2019.
[168] T. C. Wang, M. Y. Liu, J. Y. Zhu, A. Tao, J. Kautz, i B. Catanzaro, „High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs“, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, str. 8798–8807, 2018.
[169] O. Bailo, D. Ham, i Y. M. Shin, „Red blood cell image generation for data augmentation using conditional generative adversarial networks“, *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, sv. 2019-June, str. 1039–1048, 2019.
[170] F. Kong, B. Huang, K. Bradbury, i J. M. Malof, „The synthinel-1 dataset: A collection of high resolution synthetic overhead imagery for building segmentation“, *Proc. - 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV 2020*, str. 1803–1812, 2020.
[171] „IHF Handball Challenge 12“. [Na internetu]. Dostupno na: https://store.steampowered.com/app/283490/IHF_Handball_Challenge_12/. [Pristupljeno: 16-tra-2020].
[172] „IHF Handball Challenge 14“. [Na internetu]. Dostupno na: https://store.steampowered.com/app/279460/IHF_Handball_Challenge_14/. [Pristupljeno: 16-tra-2020].
[173] „Handball 16“. [Na internetu]. Dostupno na: https://store.steampowered.com/app/384320/Handball_16/. [Pristupljeno: 16-tra-2020].
[174] „Handball 17“. [Na internetu]. Dostupno na: https://store.steampowered.com/app/526980/Handball_17/. [Pristupljeno: 16-tra-2020].
[175] „Handball Action Total“. [Na internetu]. Dostupno na: https://store.steampowered.com/app/486940/Handball_Action_Total/. [Pristupljeno: 16-tra-2020].
[176] A. Ng, „Machine Learning Yearning“, 2018. [Na internetu]. Dostupno na:

https://www.deeplearning.ai/machine-learning-yearning/. [Pristupljeno: 20-tra-2019].

8 Prilozi

8.1 Kronološki popis sintsetova

Tablica 7 - kronološki popis sintsetova, sadrži pregled javno dostupnih sintsetova nastalih u periodu od 2006. do 2020. godine, istraženih za potrebe ovog rada.

Stupac 1 navodi godinu nastanka sintseta, a stupac 2 njegovo puno i (u zagradi) skraćeno ime. U stupcu 3 naveden je referentni rad u kojem je sintset predstavljen.

Stupac 4 navodi domenu za primjenu u kojoj je sintset nastao.

Ukoliko su u referentnom radu navedene jedna ili više metoda korištenih za evaluaciju sintseta, nabrojene su u stupcu 5 (i međusobno odvojene zarezom).

Metoda generiranja sintseta (prema uvedenoj sistematizaciji, u prilogu 8.4) ukoliko je poznata, navedena je u stupcu 6. Oznaka "DCC" (eng. *digital content creation*) odnosi se na metodu "stvaranje digitalnog sadržaja" (2.8).

Stupac 7 navodi primarni alat za generiranje sintseta (ukoliko je poznat), a stupac 8 pripadni korišteni renderer (ukoliko je poznat).

Broj klasa, ukoliko je primjenjiv i poznat, naveden je u stupcu 9.

Ukoliko sintset sadrži video sekvence, njihov broj naveden je u stupcu 10, a ukupno trajanje svih sekvenci (u sekundama) u stupcu 11. Broj anotiranih (samostalnih) slika naveden je u stupcu 12. Pojedini sintsetovi sastoje se od kombinacije video sekvenci i samostalnih slika.

Stupci 13 – 15 navode širinu i visinu slike (u pikselima) te broj slika u sekundi koji se odnosi na video sekvence.

U stupcu 16 navedeni su radovi koji koriste dotični sintset.

Tablica 7 - kronološki popis sintsetova

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
godina	**sintset**	**rad**	**domena**	**metoda evaluacije**	**metoda generiranja**	**alat za generiranje**	**renderer**	**broj klasa**	**broj video sekvenci**	**ukupno trajanje videa (sec)**	**broj anotiranih slika**	**širina slike**	**visina slike**	**fps**	**radovi koji koriste bazu**
2006	**TRICTRAC**	[33]	praćenje objekata		programiranje	Ogre 3D	Open GL		13	400		1.400	1.050	25	
2008	**ViHASi**	[27]	raspoznavanje akcije	SBHAR	DCC	MotionBuilder		20			980	640	480	30	
2010	**Middlebury Flow**	[51]	optički tok				3Delight		8		32	640	480		[57]
2010	**Foggy Road Image DAtabase (FRIDA)**	[52]	autonomna vožnja	MSR, FSS, NBPC, NBPC+PA	simulator	SiVIC					450	640	480		
2010	**FRIDA2**	[52]	autonomna vožnja		simulator	SiVIC					1.650	640	480		
2010	**CVC**	[53]	autonomna vožnja	HOG, linear SVM	komercijalna računalna igra	Half-Life 2			5		28.095	640	480		[16]
2010	**VascuSynth**	[54]	medicinska segmentacija	HOG, linear SVM	programiranje	Graph eXchange Language									
2010	**VHuF**	[17]	detekcija objekata		programiranje	Facial Description Language									
2011	**Virtual City**	[32]	evaluacija značajki slike	SIFT, GLOH, DAISY, HOG, SSIM	DCC	3ds Max	Mental Ray				3.000	640	480		
2011	**Statue of Liberty**	[32]	evaluacija značajki slike	SIFT, GLOH, DAISY, HOG, SSIM	DCC	3ds Max	Mental Ray				625	640	480		
2012	**Background Model Challenge (BMC)**	[56]	nadziranje	GMM, BC, CB, VM	simulator	SiVIC			20	1.200				25	
2012	**MPI-Sintel**	[57]	optički tok		DCC	Blender					1.628	1.024	436		[70], [71], [150]
2012	**Tsukuba Stereo**	[25]	stereo disparitet		DCC	ZBrush + Maya					1.800				
2013	**Synthetic Image Dataset for Illumination Robustness Evaluation (SIDIRE)**	[29]	raspoznavanje objekta		DCC	Blender		14			10.920	500	500		

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
godina	sintset	rad	domena	metoda evaluacije	metoda generiranja	alat za generiranje	renderer	broj klasa	broj video sekvenci	ukupno trajanje videa (sec)	broj anotiranih slika	širina slike	visina slike	fps	radovi koji koriste bazu
2013	**Synthetic Migrating Cells**	[60]	medicinska segmentacija		programiranje	MATLAB		1			1.078	275	275		
2014	**ICL-NUIM**	[63]	rekonstrukcija scene		DCC	POVRay			8	312	9.202	640	480	30	
2014	**Agoraset**	[68]	nadziranje		DCC	Maya	Mental Ray		87		21.750	640	480	30	
2015	**Scene Flow / FlyingThings3D**	[70]	optički tok		DCC	Blender			2.247		26.066	950	540		[150]
2015	**Scene Flow / Monkaa**	[70]	optički tok		DCC	Blender			8		8.591	950	540		
2015	**Scene Flow / Driving**	[70]	optički tok		DCC	Blender			1		4.392	950	540		
2015	**Synthetic RSM Dataset**	[72]	navigacija robota		game engine	Unity			7	52	3.129	200	150		
2015	**Flying Chairs**	[71]	optički tok		komponiranje						22.872	512	384		
2016	**UAV123**	[80]	praćenje objekata		game engine	UE4			123		112.578				
2016	**4D Light Fields**	[34]	analiza svjetlosnih polja		DCC	Blender	interni + Cycles				1.944	512	512		
2016	**CLEVR**	[31]	vizualno rasuđivanje	CNN + LSTM	DCC	Blender					100.000				
2016	**GTAVision**	[82]	autonomna vožnja	Faster-RCNN	komercijalna računalna igra	GTA V					200.000	1.914	1.052		
2016	**RenderCar**	[8]	detekcija objekata	AlexNet	DCC	3ds Max	V-Ray				819.000				
2016	**RenderScene**	[8]	detekcija objekata	AlexNet	DCC	3ds Max	V-Ray				1.800				
2016	**UnityEyes**	[40]	određivanje smjera gledanja		game engine	Unity					1.000.000	400	300		
2016	**VG**	[86]	autonomna vožnja	FCN	komercijalna računalna igra	neimenovana igra		12			60.000	1.024	768	1	
2016	**GTAV**	[87]	autonomna vožnja		komercijalna računalna igra	GTA V		19			24.966	1.914	1.052		[118], [119],

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
godina	sintset	rad	domena	metoda evaluacije	metoda generiranja	alat za generiranje	renderer	broj klasa	broj video sekvenci	ukupno trajanje videa (sec)	broj anotiranih slika	širina slike	visina slike	fps	radovi koji koriste bazu
															[126], [133]
2016	**CocoDoom**	[88]	raspoznavanje objekta		game engine	ResearchDoom		94			500.000				
2016	**SUNCG**	[90]	rekonstrukcija scene	SSCNet	DCC	Planner5D		84			130.269				[99], [97], [105], [110], [121], [161]
2016	**Human3D+**	[91]	određivanje poze	AlexNet, VGG							1.574				
2016	**SYNTHIA**	[92]	autonomna vožnja	Target-Net	game engine	Unity		13	4		213.400	960	720		[112], [118], [119], [126], [133]
2016	**PaCMan**	[35]	arhitektura neuronske mreže	DNC				20			102.400	640	480		
2016	**UnrealStereo**	[94]	stereo disparitet	DispNet	game engine	UE4			4	33	na zahtjev				
2016	**Virtual KITTI**	[95]	autonomna vožnja	Fast-R-CNN	game engine	Unity			35		17.000	1.242	375		[112], [119], [127], [137], [141], [146]
2016	**TU Berlin Multi-Object and Multi-Camera Tracking Dataset (MOCAT)**	[96]	nadziranje	CNN	komercijalna računalna igra	Garry's Mod					101.638	1.366	768	30	
2016	**MLT**	[97]	razumijevanje scene	VGG-16	programiranje	OpenGL	Mitsuba				568.793				
2017	**SceneNet RGB-D**	[99]	razumijevanje scene	CNN	fizikalni engine	Chrono Engine	Opposite	255			5.000.000	320	240		
2017	**VANDAL**	[37]	klasifikacija objekata	CaffeNet	DCC	Blender		319			4.106.340				

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
godina	sintset	rad	domena	metoda evaluacije	metoda generiranja	alat za generiranje	renderer	broj klasa	broj video sekvenci	ukupno trajanje videa (sec)	broj anotiranih slika	širina slike	visina slike	fps	radovi koji koriste bazu
2017	**UPNA Synthetic Head Pose Database**	[107]	određivanje poze		programiranje	vlastiti			120	1.200	36.000	1.280	720	30	
2017	**SURREAL**	[108]	određivanje poze	CNN	DCC	Blender		23	67.582		6.536.752	320	240		
2017	**Rendered Handpose Dataset**	[109]	određivanje poze	HandSegNet + PoseNet + GestureNet	DCC	Blender		33			43.986	320	320		
2017	**Syn2Real**	[111]	detekcija objekata	GoogleNet FCN	DCC	Blender	Cycles				4.000	512	512		
2017	**VIsual PERception benchmark (VIPER)**	[112]	autonomna vožnja	CNN	komercijalna računalna igra	GTA V		11			254.064	1.920	1.080		[126], [9]
2017	**Procedural Human Action Videos (PHAV)**	[113]	raspoznavanje akcije	Cool-TSN	DCC	Unity			39.982		5.996.286				
2017	**Osnabrück Synthetic Scalable Cube Dataset**	[114]	rekonstrukcija objekta	CNN	programiranje	MATLAB					9.960.000	100	100		
2018	**House3D**	[121]	navigacija robota	CNN + LSTM			vlastiti OpenGL renderer				na zahtjev	120	90	600	
2018	**Moving INfants In RGB-D (MINI-RGBD)**	[122]	određivanje poze				OpenDR		12		12.000	640	480		
2018	**Synthetic Arabidopsis Dataset**	[123]	segmentacija	Mask-RCNN	DCC	Blender					10.000	550	550		
2018	**Synthetic Person Re-Identification (SyRI)**	[125]	nadziranje	SpindleNeT	game engine	UE4		100	28.000	560.000	1.680.000			30	
2018	**Virtual Environment for Instance Segmentation (VIES)**	[126]	autonomna vožnja	DeepLab + Mask R-CNN	game engine	Unity					61.305				
2018	**DR**	[127]	detekcija objekata	Faster RCNN	game engine	UE4		36			100.000	1.200	400	30	[141]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
godina	sintset	rad	domena	metoda evaluacije	metoda generiranja	alat za generiranje	renderer	broj klasa	broj video sekvenci	ukupno trajanje videa (sec)	broj anotiranih slika	širina slike	visina slike	fps	radovi koji koriste bazu
2018	**Falling Things (FAT)**	[128]	određivanje poze	PoseCNN, DOPE	game engine	UE4		21			61.500	960	540		
2018	**Synthetic SVBRDFs and renderings**	[36]	generiranje tekstura	U-Net			Mitsuba				200.000				
2018	**Synscapes**	[133]	autonomna vožnja	FRRN, DeepLab			nepoznati renderer filmske kvalitete				25.000	2.048	1.024		[9]
2018	**ParallelEye**	[93]	autonomna vožnja	DPM, Faster R-CNN	game engine	Unity		3			15.931				
2018	**KITTI-360**	[137]	autonomna vožnja	MNC, Faster-RCNN	DCC	Blender	Cycles				4.000				
2018	**KITTI-15**	[137]	autonomna vožnja	MNC, Faster-RCNN	DCC	Blender	Cycles				4.000				
2018	**SIM**	[138]	određivanje poze	OpenPose	game engine	Unity					36.225				
2018	**BIG-SIM**	[138]	određivanje poze	OpenPose	game engine	Unity					453.998				
2018	**Random-1M**	[30]	robotski hvat		programiranje	V-HACD biblioteka					1.000.000				
2018	**SDR**	[141]	autonomna vožnja	Faster-RCNN	game engine	UE4					25.000				
2018	**3DRMS Challenge Dataset 2018**	[143]	rekonstrukcija scene	DeepLabV3	DCC	Blender		9			25.000	640	480		
2019	**3DPeople Dataset**	[144]	rekonstrukcija objekta	GimNet	DCC	Blender		22	22.400		2.500.000	640	480		
2019	**ADORESet**	[145]	detekcija objekata	VGGNet, InceptionV3, ResNet, Xception	simulator	Gazebo		30			97.500	300	300		
2019	**AADS**	[146]	autonomna vožnja	Mask-RCNN			PBRT	60			143.906	1.920	1.080		
2019	**A diagnostic dataset for Compositional Actions and TEmporal**	[148]	vizualno rasuđivanje	I3D, TSN, LSTM				301	23	10	5.500	320	240	24	

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
godina	sintset	rad	domena	metoda evaluacije	metoda generiranja	alat za generiranje	renderer	broj klasa	broj video sekvenci	ukupno trajanje videa (sec)	broj anotiranih slika	širina slike	visina slike	fps	radovi koji koriste bazu
	Reasoning (CATER)														
2019	**Indoor Robotics Stereo (IRS)**	[150]	stereo disparitet	DispNet-CSS + DispNorm Net	game engine	UE4					100.025	960	540		
2019	**GTA5 Crowd Counting (GCC)**	[151]	nadziranje	SFCN + SE Cycle GAN	komercijalna računalna igra	GTA V					15.212	1.080	1.920		
2019	**X-Plane**	[152]	autonomno letenje	PCA Autoencoder, ResNet34 Autoencoder, TS Regr. ConvLSTM+LSTM, TS Regr. ConvLSTM, Dynamics	simulator	X-Plane			8.011	921.265	7.000.000	256	256		
2019	**Mid-Air**	[153]	autonomno letenje		game engine	UE4			54	4.740	119.000	1.024	1.024	25	
2019	**Apollo Synthetic Dataset**	[154]	autonomna vožnja		game engine	Unity					273.000	1.920	1.080		
2019	**Spacecraft Pose Estimation Dataset (SPEED)**	[156]	autonomno letenje	Spacecraft Pose Network			vlastiti OpenGL renderer				12.000	1.920	1.200		
2019	**PreSIL**	[157]	autonomna vožnja	AVOD-FPN	komercijalna računalna igra	GTA V					50.000	1.920	1.080		
2019	**ProcSy**	[159]	autonomna vožnja	Deeplab v3+	DCC + game engine + simulator	CityEngine + UE4 + CARLA					11.000	2.048	1.024		
2019	**A Synthetic Image Dataset for 3D Object Pose Recognition with Distractors (SIDOD)**	[160]	određivanje poze		game engine	UE4		21			144.000				
2019	**3D60**	[161]	rekonstrukcija scene	CoordNet	DCC	Blender					224.406				
2019	**CURE-TSD**	[163]	autonomna vožnja	U-Net, ResNet, VGG, GoogLeNet	DCC	After Effects		14	2.989		896.700	1.628	1.236		
2019	**Virtual Gallery**	[42]	vizualna lokalizacija	Mask R-CNN	game engine	Unity		42			55.572	1.920	1.080		

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
godina	**sintset**	**rad**	**domena**	**metoda evaluacije**	**metoda generiranja**	**alat za generiranje**	**renderer**	**broj klasa**	**broj video sekvenci**	**ukupno trajanje videa (sec)**	**broj anotiranih slika**	**širina slike**	**visina slike**	**fps**	**radovi koji koriste bazu**
2019	**WoodScape**	[162]	autonomna vožnja	ENet	programiranje	vlastiti		40			10.000				
2019	**xR-EgoPose**	[165]	određivanje poze	ResNet	DCC	Maya					383.000	1.024	1.024	30	
2020	**SynthineI-1**	[170]	segmentacija	U-net, DeepLabV3	DCC	CityEngine					2.108	572	572		

8.2 Kronološki popis simulatora

Tablica 8 - kronološki popis simulatora, sadrži pregled javno dostupnih simulatora nastalih u periodu od 2004. do 2019. godine, istraženih za potrebe ovog rada.

Stupac 1 navodi godinu nastanka simulatora, a stupac 2 njegovo puno i (u zagradi) skraćeno ime. U stupcu 3 naveden je referentni rad u kojem je simulator predstavljen.

Stupac 4 navodi domenu za primjenu u kojoj je simulator nastao.

Primarni alat za izradu simulatora, ukoliko je poznat, naveden je u stupcu 5, a njegova vrsta u stupcu 6. Oznaka "IDE" (eng. *integrated development environment*) odnosi se na metodu "programiranje" (2.1), prema uvedenoj sistematizaciji (u prilogu 8.4).

Broj klasa, ukoliko je primjenjiv i poznat, naveden je u stupcu 7.

Stupci 8 – 10 navode širinu i visinu slike (u pikselima) te broj slika u sekundi koji simulator generira.

U stupcu 11 navedeni su radovi koji koriste dotični simulator.

Tablica 8 - kronološki popis simulatora

1	2	3	4	5	6	7	8	9	10	11
godina	**simulator**	**rad**	**domena**	**alat za izradu simulatora**	**vrsta alata**	**broj klasa**	**širina slike**	**visina slike**	**fps**	**radovi koji koriste simulator**
2004	**Gazebo**	[24]	navigacija robota							
2007	**OVVV**	[19]	nadziranje	Half-Life 2	komercijalna računalna igra		320	240		
2017	**AI2-THOR**	[100]	podržano učenje	Unity	game engine					
2017	**AirSim**	[21]	autonomno letenje	UE4	game engine					
2017	**Car Learning to Act (CARLA)**	[104]	autonomna vožnja	UE4	game engine	12				[146], [159], [9]
2017	**MINOS**	[110]	navigacija robota							
2017	**Sim4CV**	[117]	autonomna vožnja	UE4	game engine		320	180		
2018	**VIVID**	[135]	*	UE4	game engine					
2019	**Air Learning**	[147]	autonomno letenje	UE4	game engine					
2019	**Habitat**	[7]	utjelovljena umjetna inteligencija	C++	IDE				10.000	
2019	**OpenAI Remote Rendering Backend (ORRB)**	[155]	navigacija robota	Unity	game engine		200	200		
2019	**VRGym**	[164]	utjelovljena umjetna inteligencija	UE4	game engine					

8.3 Kronološki popis generatora

Tablica 9 - kronološki popis generatora, sadrži pregled javno dostupnih generatora nastalih u periodu od 1999. do 2018. godine, istraženih za potrebe ovog rada.

Stupac 1 navodi godinu nastanka generatora, a stupac 2 njegovo puno ime. U stupcu 3 naveden je referentni rad u kojem je generator predstavljen.

Stupac 4 navodi domenu za primjenu u kojoj je generator nastao.

Primarni alat za izradu generatora, ukoliko je poznat, naveden je u stupcu 5, a njegova vrsta u stupcu 6.

Tablica 9 - kronološki popis generatora

1	2	3	4	5	6
godina	**generator**	**rad**	**domena**	**alat za izradu generatora**	**vrsta alata**
1999	**Synthetic Sequence Generator**	[23]	medicinska segmentacija	MATLAB	IDE
2010	**VHuF**	[17]	detekcija objekata		
2016	**SceneNet**	[78]	razumijevanje scene	Blender	DCC
2016	**LCrowdV**	[83]	nadziranje	Menge + UE4	crowd simulation engine + game engine
2016	**UnrealStereo**	[94]	stereo disparitet	UE4	game engine
2018	**FiloGen**	[130]	medicinska segmentacija	MATLAB + C++	IDE

8.4 Sistematizacija procesa generiranja sintseta

Sistematizacija procesa generiranja sintseta izgrađena je prema kronološki prvom pojavljivanju pojedine stavke u radovima obrađenim u poglavlju 3 (Pregled područja) i onim redoslijedom (od 1 do 17) kojim pojedine stavke sudjeluju u procesu generiranja sintseta.

1. analiza referentnog dataseta
- 1.1. algoritamska estimacija parametara za sjenčanje [66]
- 1.2. distribucija značajki [75]

2. metoda generiranja
- 2.1. programiranje
  - 2.1.1. alati
    - 2.1.1.1. Ogre 3D [33]
    - 2.1.1.2. Graph eXchange Language [54]
    - 2.1.1.3. Facial Description Language [17]
    - 2.1.1.4. MATLAB [60]
    - 2.1.1.5. OpenGL [97]
    - 2.1.1.6. V-HACD biblioteka [30]
- 2.2. generator [6]
  - 2.2.1. alati za izradu generatora
    - 2.2.1.1. MATLAB [23]
    - 2.2.1.2. Blender [78]
    - 2.2.1.3. Menge [83]
    - 2.2.1.4. Unreal Engine 4 [94]
    - 2.2.1.5. C++ [130]
- 2.3. komponiranje [81]
- 2.4. komercijalna računalna igra
  - 2.4.1. Half-Life 2 [19]
  - 2.4.2. GTA V [82]
  - 2.4.3. Garry's Mod [96]
- 2.5. game engine
  - 2.5.1. Unity [72]
  - 2.5.2. Unreal Engine 4 [80]
- 2.6. fizikalni engine
  - 2.6.1. Chrono Engine [99]
  - 2.6.2. MuJoCo Physics Engine [50]
- 2.7. simulator
  - 2.7.1. Gazebo [24]
  - 2.7.2. OVVV [19]
  - 2.7.3. SiVIC [52]
  - 2.7.4. VDrift [59]
  - 2.7.5. TORCS [73]
  - 2.7.6. AI2-THOR [100]
  - 2.7.7. AirSim [21]

2.7.8. CARLA [104]
2.7.9. MINOS [110]
2.7.10. Sim4CV [117]
2.7.11. VIVID [135]
2.7.12. Air Learning [147]
2.7.13. Habitat [7]
2.7.14. X-Plane [152]
2.7.15. ORRB [155]
2.7.16. VRGym [164]
2.8. stvaranje digitalnog sadržaja
2.8.1. alati
2.8.1.1. 3ds Max [32]
2.8.1.2. Blender [57]
2.8.1.2.1. prilagođeni [70]
2.8.1.3. SketchUp [20]
2.8.1.4. Planner5D [90]
2.8.1.5. CityEngine [93]
2.8.1.6. After Effects [163]
2.9. generativni suparnički trening [139]

3. integracija
3.1. Unreal Engine 4
3.1.1. strojno učenje
3.1.1.1. Torch [84]
3.1.1.2. Caffe [98]
3.1.1.3. Tensorflow [117]
3.2. Unity
3.2.1. podržano učenje
3.2.1.1. Tensorflow [101]

4. izvor objekata
4.1. 2D objekt
4.1.1. prethodno renderirani [70]
4.1.2. fotografirani [115]
4.1.2.1. baza na Internetu
4.1.2.1.1. Big Berkeley Instance Recognition Dataset (BigBIRD) [115]
4.2. 3D objekt
4.2.1. baza na Internetu
4.2.1.1. besplatni
4.2.1.1.1. ModelNet [28]
4.2.1.1.2. 3D Warehouse [20]
4.2.1.1.3. ShapeNet [39]
4.2.1.1.4. Stanford Database [78]
4.2.1.2. tržnice [94]
4.2.2. proceduralno generirani
4.2.2.1. alati
4.2.2.1.1. MATLAB [23]

4.2.2.1.2. POVRay [63]
4.2.3. manualno generirani
4.2.3.1. alati
4.2.3.1.1. ZBrush [25]
4.2.3.1.2. Fuse [125]
4.2.3.1.3. MakeHuman [131]
4.2.4. konverzija OSM mapa [93]
4.2.5. L-System [124]
4.2.6. LiDAR + RGB kamera [146]
4.2.7. fotogrametrija [154]

5. modifikacija objekata
5.1. 3D objekt
5.1.1. parametarska
5.1.1.1. morfabilni model [17]
5.1.1.1.1. tijelo [16]
5.1.1.1.1.1. SMPL (Skinned Multi-Person Linear) [108]
5.1.1.1.1.2. SMIL (Skinned Multi-Infant Linear) [122]
5.1.1.1.2. lice [17]
5.1.1.1.3. oko [40]
5.1.2. manualna
5.1.3. retopologiziranje [40]
5.1.4. skaliranje po različitim osima [37]
5.1.5. predprocesiranje LiDAR scena [146]
5.1.5.1. uklanjanje pomičnih objekata [146]
5.1.5.2. docrtavanje uklonjenog [146]
5.1.5.3. preuzimanje iluminacije [146]
5.1.5.4. poboljšavanje tekstura [146]

6. elementi scene
6.1. 2D objekt
6.2. 3D objekt
6.2.1. struktura
6.2.1.1. poligonalni
6.2.1.1.1. FEM (Finite Element Method) [130]
6.2.1.2. volumetrijski [54]
6.2.2. mogućnost deformacije
6.2.2.1. krut
6.2.2.2. deformabilan
6.2.2.2.1. vozilo [96]
6.2.2.2.1.1. pokretni dijelovi [58]
6.2.2.2.2. humanoid [96]
6.2.2.2.3. životinja [96]
6.2.3. svojstva
6.2.3.1. transformacije
6.2.3.1.1. pozicija
6.2.3.1.2. rotacija

6.2.3.1.3. veličina [49]
6.2.3.2. boja [49]
6.2.3.3. tekstura
6.2.3.3.1. način nastanka
6.2.3.3.1.1. crtana [62]
6.2.3.3.1.2. fotografska
6.2.3.3.1.2.1. baza na Internetu
6.2.3.3.1.2.1.1. OpenSurfaces [28]
6.2.3.3.1.2.1.2. ArchiveTextures [28]
6.2.3.3.1.2.1.3. Image*After [70]
6.2.3.3.1.2.1.4. Flicker [71]
6.2.3.3.1.3. proceduralna
6.2.3.3.1.3.1. alati
6.2.3.3.1.3.1.1. ImageMagick [70]
6.2.3.3.1.3.1.2. Substance [36]
6.2.3.3.2. prostor boje
6.2.3.3.2.1. RGB
6.2.3.3.2.2. HSV [62]
6.2.3.3.3. rezolucija [65]
6.2.3.4. materijal
6.2.3.4.1. svojstva
6.2.3.4.1.1. BRDF [29]
6.2.3.4.1.2. transparencija [38]
6.2.3.4.1.3. podpovršinskog raspršivanja svjetla [79]
6.2.3.4.1.4. refleksivnost [94]
6.2.3.4.1.5. mokroća [112]
6.3. svjetlo [6]
6.3.1. izvor
6.3.1.1. doba dana [19]
6.3.1.2. umjetni izvori [19]
6.3.1.3. slika s visokim rasponom boja [40]
6.3.2. svojstva
6.3.2.1. sjena [32]
6.3.2.2. penumbra [85]
6.4. kamera
6.4.1. vrsta
6.4.1.1. statična
6.4.1.2. PTZ (pan-tilt-zoom) [33]
6.4.2. svojstva
6.4.2.1. transformacije
6.4.2.1.1. pozicija [53]
6.4.2.1.2. rotacija [6]
6.4.2.2. širina vidnog polja [74]
6.4.2.2.1. perspektivna distorzija[74]
6.4.2.3. žarišna duljina [74]
6.4.2.4. brzina zatvarača [8]
6.4.3. prema smjeru

6.4.3.1. jednosmjerna
6.4.3.2. višesmjerna [161]
6.4.4. u odnosu na objekt promatranja
6.4.4.1. iz trećeg lica
6.4.4.2. egocentrična [165]
6.5. pozadina
6.5.1. fotografija [16]
6.5.1.1. baza na Internetu
6.5.1.1.1. ImageNet [69]
6.5.1.1.2. SUN397 [39]
6.5.1.1.3. UW Scenes [116]
6.5.1.1.4. LSUN [144]
6.5.2. uniformna boja [69]

7. način generiranja scene

7.1. 2D
7.1.1. blending
7.1.1.1. vrste
7.1.1.1.1. alpha [39]
7.1.1.1.2. Gaussian Blurring [115]
7.1.1.1.3. Poisson [115]
7.1.1.2. prema planovima
7.1.1.2.1. prednji [126]
7.1.1.2.2. pozadina [126]
7.2. 3D
7.2.1. manualno [136]
7.2.2. proceduralno
7.2.2.1. strukturirano
7.2.2.1.1. graf scene [31]
7.2.2.2. stohastičko (rendom) [136]
7.2.2.2.1. pazeći na preklapanje [76]
7.2.3. fizikalno
7.2.3.1. ispuštanje objekata na scenu [99]
7.2.4. generativni suparnički trening [103]
7.2.5. dopunjena stvarnost [156]

8. scena

8.1. 2D scena
8.2. 3D scena

9. simulacija

9.1. fizika
9.1.1. kruta tijela [41]
9.1.2. vrsta tla [24]
9.1.3. deformabilni objekti [24]
9.1.4. dinamika fluida [164]
9.1.4.1. tlak zraka [21]

11. komplikacije

- 11.1. okluzija [23]
  - 11.1.1. pozicija
    - 11.1.1.1. preklapajuća [23]
    - 11.1.1.2. dodirujuća [23]
    - 11.1.1.3. samo-okluzija [165]
  - 11.1.2. tekstura
    - 11.1.2.1. uniformna boja [8]
    - 11.1.2.2. fotografske teksture različitih oblika [8]
- 11.2. nedostajući frejmovi [23]

12. sjenčanje

- 12.1. vrsta sjenčanja
  - 12.1.1. realistično
    - 12.1.1.1. fotorealistično [118]
  - 12.1.2. nerealistično [58]
    - 12.1.2.1. gradijentno [58]
- 12.2. renderer
  - 12.2.1. alati
    - 12.2.1.1. OpenGL [33]
    - 12.2.1.2. POV-Ray [49]
    - 12.2.1.3. 3Delight [51]
    - 12.2.1.4. SiVIC [52]
    - 12.2.1.5. Mental Ray [32]
    - 12.2.1.6. Blender [57]
    - 12.2.1.7. Maya [25]
    - 12.2.1.8. Cycles [34]
    - 12.2.1.9. V-Ray [8]
    - 12.2.1.10. Mitsuba [97]
    - 12.2.1.11. Opposite Renderer [99]
    - 12.2.1.12. Mantra PBR [105]
    - 12.2.1.13. MuJoCo Physics Engine [50]
    - 12.2.1.14. OpenDR [122]
    - 12.2.1.15. PBRT [146]
  - 12.2.2. postavke
    - 12.2.2.1. prostor boje
      - 12.2.2.1.1. gama
      - 12.2.2.1.2. linearni [51]
    - 12.2.2.2. ambijentalna okluzija [51]
    - 12.2.2.3. globalna iluminacija [51]
      - 12.2.2.3.1. metoda
        - 12.2.2.3.1.1. mapiranje fotona [99]
      - 12.2.2.3.2. svojstva
        - 12.2.2.3.2.1. curenje boje [63]
        - 12.2.2.3.2.2. indirektno osvjetljenje [99]
        - 12.2.2.3.2.3. kaustičnost [99]
    - 12.2.2.4. refleksije [59]

12.2.2.5. antialiasing [59]
12.2.2.6. MIP mapiranje [59]
12.2.2.7. refrakcija [40]
12.2.2.8. broj uzoraka po pikselu [85]
12.2.2.9. nivo detalja [153]
12.2.3. program za sjenčanje fragmenata [40]
12.2.3.1. refrakcija [40]
12.2.4. put sjenčanja
12.2.4.1. odgođen [150]
12.2.4.2. unaprijedni
12.3. hardver
12.3.1. CPU
12.3.2. GPU [76]
12.3.3. distribuirano
12.3.3.1. u oblaku [118]
12.4. postavke outputa
12.4.1. stereoskopija
12.4.1.1. monokularni render
12.4.1.2. stereo par [25]
12.4.2. temporalnost
12.4.2.1. slika
12.4.2.2. sekvenca [47]
12.4.3. format
12.4.3.1. PNG
12.4.3.2. JPG [8]
12.4.3.3. OpenEXR [133]
12.4.4. kompresija podataka
12.4.4.1. RGB
12.4.4.1.1. WebP4 [70]
12.4.4.2. ne-RGB
12.4.4.2.1. LZO5 [70]
12.5. način isporuke modelu za učenje
12.5.1. sekvencijalno (serija podataka)
12.5.2. u realnom vremenu [26]
12.5.3. u letu [76]
12.5.4. na zahtjev [94]

13. naknadna obrada
13.1. šum [6]
13.1.1. obuhvat
13.1.1.1. globalni [23]
13.1.1.2. lokalni [23]
13.1.1.3. na nivou piksela [19]
13.1.2. prema kanalu
13.1.2.1. RGB
13.1.2.2. mapa dubine [63]
13.1.2.2.1. prema kutu gledanja [78]

13.1.3. prema vrsti
13.1.3.1. aksijalan i lateralan šum [106]
13.1.3.2. šum refleksivne površine [106]
13.1.3.3. šum nerefleksivne površine [106]
13.1.3.4. strukturalan šum [106]
13.1.3.5. šum distorzije leće i efekata [106]
13.1.3.6. šum kvantizacijskog koraka [106]
13.1.3.7. šum pokreta i brzine zatvarača [106]
13.1.3.8. šum sjene [106]
13.2. izglađivanje [47]
13.3. distorzija slike
13.3.1. radijalna [19]
13.4. video ghosting [19]
13.5. antialiasing
13.5.1. SSAA [19]
13.6. magla [52]
13.6.1. unifromna [52]
13.6.2. heterogena [52]
13.6.3. oblačasta [52]
13.6.4. heteogeno oblačasta [52]
13.7. zamućenje pokreta [57]
13.8. zamućenje fokusa [57]
13.9. odsjaj sunca [70]
13.10. manipulacija krivulje game [70]
13.11. vinjeta [8]
13.12. kromatska aberacija [85]
13.13. automatska ekspozicija [133]
13.14. ambijentalna okluzija [155]
13.15. greške kodeka [163]
13.16. zaprljanost leće [163]

14. output
14.1. izravni output
14.1.1. RGB [6]
14.1.1.1. izvedeni output
14.1.1.1.1. sivi tonovi [23]
14.1.2. mapa dubine [6]
14.1.2.1. izvedeni output
14.1.2.1.1. mapa dispariteta [25]
14.1.2.1.2. mapa neprekrivanja [25]
14.1.2.1.3. mapa diskontinuiteta dubine [25]
14.1.2.1.4. mapa promjene dispariteta [70]
14.1.3. mapa optičkog toka [11]
14.1.4. siluete [48]
14.1.5. segmentacijska mapa [19]
14.1.5.1. izvedeni output
14.1.5.1.1. 16-bitni sivi tonovi [68]

15.1. smanjivanje [26]
15.2. rezanje [26]
15.3. uklanjanje pozadine segmentacijom prije treniranja [62]
15.4. savijanje [161]

16. augmentacija
16.1. okluzija [23]
16.2. rezanje [37]
16.3. kontrast mape dubine [37]
16.4. svjetloća mape dubine [37]
16.5. zamjena bijele boje pozadine u mapi dubine rendom bojom udaljenijom od centra mase objekta [37]
16.6. kosa transformacija (smicanje) [37]
16.7. 2D rotacija [116]
16.8. 3D rotacija [116]
16.9. sakaćenje [116]
16.10. distraktor objekti [116]
16.11. svjetloća [127]
16.12. kontrast [127]
16.13. zrcaljenje [127]
16.14. homografija [42]
16.15. oštrenje [169]
16.16. utiskivanje [169]
16.17. inverzija kanala boje [169]

17. adaptacija domene
17.1. integrirani okviri
17.1.1. V-AYLA [64]
17.2. transfer učenja [76]
17.3. transfer stila [132]
17.3.1. cijele slike [132]
17.3.2. objekata od interesa [167]
17.4. suparnički trening [132]