SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1547

Učenje korespondencijske metrike za gustu stereoskopsku rekonstrukciju

Marin Oršić

Zagreb, srpanj 2017.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

Zahvaljujem mentoru, izv. prof. dr. sc. Siniši Šegviću, na savjetima i vođenju tijekom rada na diplomskom studiju. Također, zahvaljujem svojoj obitelji, prijateljima i kolegama na potpori kroz dosadašnje obrazovanje.

SADRŽAJ

1. Uvod

Stereoskopska rekonstrukcija područje je računalnog vida čiji je cilj ostvariti položaj točaka u stvarnom 3D prostoru na temelju dvije ili više slika. Algoritam stereoskopske rekonstrukcije ostvaruje se na rektificiranom stereu: paru slika kalibriranom tako da obje kamere budu usmjerene u istom pravcu te da retci obje kamere odgovaraju istoj ravnini u prostoru. Postupak stereoskopske rekonstrukcije pronalazi odgovarajuće piksele na slikama i na temelju njihovog relativnog položaja određuje udaljenost 3D točke i referentne kamere. Relativan odnos odgovarajućih piksela naziva se disparitet. Bitna stavka postupka upravo je korespondencijska metrika koja se koristi za određivanje sličnosti slikovnih okana. Klasične metode koriste ručno definirane postupke za računanje korespondencijskih metrika, no pojavom dubokog učenja i njegovim uspjehom u području računalnog vida pokazalo se kako je korespondencijsku metriku moguće naučiti na podacima. Najnovije metode koriste duboko učenje kako bi ostvarile stereo postupak učen s kraja na kraj: za rektificirani stereo par i poznatu dubinsku informaciju ostvaruju stereo algoritam kojem je svaki korak postupka naučen. Takvi postupci postižu najbolje rezultate na javno dostupnim skupovima slika za stereoskopsku rekonstrukciju.

U ovom radu opisana je geometrija stereoskopskog para, kalibracija te koraci stereoskopske rekonstrukcije. Definirano je duboko učenje kao grana strojnog učenja i njegova primjena u rješavanju problema računalnog vida. Opisan je duboki model koji ostvaruje korespondencijsku metriku za korištenje kako u stereoskopskoj rekonstrukciji, tako u drugim postupcima računalnog vida koji koriste metriku sličnosti slikovnih okana. Također je opisan duboki model koji inspiriran klasičnim postupcima ostvaruje sve korake algoritma stereoskopske rekonstrukcije te daje najbolje rezultate. Opisani su postupci pripreme skupova za učenje, arhitekture modela, treniranje i validacija hiperparametara. Točnost postupaka je mjerena na skupovima slika s poznatom informacijom o dubini.

2. Geometrija kalibriranog stereoskopskog para

Postupak koji prethodi stereoskopskoj rekonstrukciji je kalibracija i rektifikacija para (ili više) kamera. Ostvarivanjem ovog postupka, normale na ravnine kamere su paralelne a svaki par korespondentnih piksela se na obje kamere nalazi u istom redu. Na ovaj način se omogućuje postupak računanja dispariteta: za svaki piksel lijeve kamere odgovarajući piksel desne kamere nalazi se na udaljenosti d piksela po vodoravnoj osi. Disparitet se definira kao vertikalan pomak d između položaja korespondentih piksela u lijevoj i desnoj slici. Mapa dispariteta je matrica koja sadrži disparitete za svaki piksel referentne kamere. Odnos para piksela lijeve i desne kamere može se definirati kao:

$$I_L(x, y) = I_D(x, y - d) \tag{2.1}$$

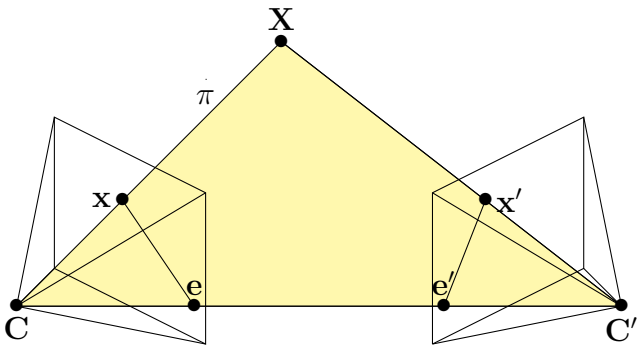
gdje su I_L i I_D matrice vrijednosti slikovnih elemenata a d je disparitet. Na ovaj način pretraživanje korespondentih piksela potrebno je vršiti isključivo na vodoravnim linijama slika te se složenost postupka smanjuje.

2.1. Epipolarna geometrija

Slika 2.1 prikazuje točku $\mathbf{X}$ u prostoru gledanu iz dviju kamera. Točke $\mathbf{C}$ i $\mathbf{C}'$ označavaju centre kamera. Projekcija točke $\mathbf{X}$ gledane iz lijeve i desne kamere su $\mathbf{x}$ i $\mathbf{x}'$. Projicirane točke $\mathbf{x}$ i $\mathbf{x}'$ te centri kamera leže na istoj ravnini kao i točka $\mathbf{X}$.

Ključni pojmovi potrebni za opis geometrije dvaju pogleda su [6]:

- epipol: točka u kojoj pravac određen centrima kamera C i C' siječe slikovnu ravninu.
- epipolarna ravnina: ravnina koja sadrži pravac određen centrima kamera.
- epipolarni pravac: presjek epipolarne ravnine sa slikovnom ravninom. Svi epipolarni pravci sijeku se u epipolu.



Slika 2.1: Točka $\mathbf{X}$ u sceni snimljena s dvije kamere. Žutom bojom označena je ravnina π određena točkama $\mathbf{C}$, $\mathbf{C}'$ i $\mathbf{X}$. Projekcije snimljene točke $\mathbf{x}$ i $\mathbf{x}'$ na lijevu odnosno desnu sliku, te točke na epipolarnom pravcu $\mathbf{e}$ i $\mathbf{e}'$

Točka $\mathbf{x}$ perspektivna je projekcija točke $\mathbf{X}$: $\mathbf{x} = \mathbf{PX}$. Inverzno preslikavanje porodica je rješenja definirana s:

$$\mathbf{X}(\lambda) = \mathbf{P}^{+}\mathbf{x} + \lambda\mathbf{C} \tag{2.2}$$

gdje je $\mathbf{P}^{+}$ pseudo-inverz matrice $\mathbf{P}$ [2]:

$$\mathbf{P}^{+} = (\mathbf{P}^{*}\mathbf{P})^{-1}\mathbf{P}^{*} \tag{2.3}$$

Za $\lambda = 0$ rješenje je točka $\mathbf{P}^{+}\mathbf{x}$ dok je za $\lambda = \infty$ rješenje centar kamere $\mathbf{C}$. Gledano iz desne kamere, ove dvije točke su $\mathbf{P}'\mathbf{P}^{+}\mathbf{x}$ i $\mathbf{P}'\mathbf{C}$. Epipolarni pravac određuju dvije projicirane točke $\mathbf{l}' = (\mathbf{P}'\mathbf{C}) \times (\mathbf{P}'\mathbf{P}^{+}\mathbf{x})$. Točka $\mathbf{e}' = \mathbf{P}'\mathbf{C}$ je epipol desne slike, odnosno projekcija centra lijeve kamere u desnu sliku. Slijedi da je $\mathbf{l}' = [\mathbf{e}']_{\times}\mathbf{P}'\mathbf{P}^{+}\mathbf{x} = \mathbf{F}\mathbf{x}$ gdje je $\mathbf{F}$ fundamentalna matrica:

$$\mathbf{F} = [\mathbf{e}']_{\times}\mathbf{P}'\mathbf{P}^{+} \tag{2.4}$$

gdje je $[\mathbf{e}']_{\times}$ matrični zapis vektorskog produkta:

$$[\mathbf{e}']_{\times} = \begin{bmatrix} 0 & -e'_3 & e'_2 \\ e'_3 & 0 & -e'_1 \\ -e'_2 & e'_1 & 0 \end{bmatrix} \tag{2.5}$$

Fundamentalna matrica $\mathbf{F}$ određuje relativan odnos projiciranih točaka na lijevu i desnu slikovnu ravninu. Projiciranu točku $\mathbf{x}$ lijeve slike moguće je preslikati u njenu korespondentnu točku u desnoj slici, $\mathbf{x}'$. Za svaki par korespondentnih točaka projiciranih na lijevu i desnu slikovnu ravninu vrijedi epipolarno ograničenje:

$$\mathbf{x}'^{\mathbf{T}}\mathbf{F}\mathbf{x} = 0 \tag{2.6}$$

Esencijalna matrica specijalan je oblik fundamentalne matrice za slučaj kada su slikovne koordinate normalizirane [13]. Esencijalna matrica češće je korištena za rješavanje problema u geometriji dva pogleda zato što u odnosu na fundamentalnu matricu ima manje stupnjeva slobode zbog čega je njena estimacija jednostavnija.

Neka je projekcijska matrica kamere $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ gdje je $\mathbf{K}$ kalibracijska matrica, $\mathbf{R}$ matrica rotacijske transformacije a $\mathbf{t}$ vektor translacije. Projekcija točke $\mathbf{X}$ tada je $\mathbf{x} = \mathbf{PX}$. Inverzna transformacija točke $\mathbf{x}$ korištenjem kalibracijske matrice je $\hat{\mathbf{x}} = \mathbf{K}^{-1}\mathbf{x}$. Tada je $\hat{\mathbf{x}} = [\mathbf{R}|\mathbf{t}]\mathbf{X}$, a točka $\hat{\mathbf{x}}$ je slikovna točka izražena u normaliziranim koordinatama. Matrica kamere $\mathbf{K}^{-1}\mathbf{P} = [\mathbf{R}|\mathbf{t}]$ zove se normalizirana matrica kamere kojoj je efekt kalibracijske matrice uklonjen. Kada je lijeva kamera postavljena u ishodište globalnog koordinatnog sustava te je njena rotacijska matrica jedinična, tada su normalizirane matrice lijeve i desne kamere $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ i $\mathbf{P}' = [\mathbf{R}|\mathbf{t}]$. Fundamentalna matrica koja određuje odnos ovako definiranog para kamera je esencijalna matrica:

$$\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R} = \mathbf{R}[\mathbf{R^T t}]_{\times} \tag{2.7}$$

Za esencijalnu matricu vrijedi epipolarno ograničenje $\hat{\mathbf{x}}^{'\mathbf{T}}\mathbf{E}\hat{\mathbf{x}} = 0$. Transformacijom normaliziranih slikovnih koordinata korištenjem odgovarajućih kalibracijskih matrica dobiva se $\mathbf{x}^{'\mathbf{T}}(\mathbf{K}^{'\mathbf{T}})^{-\mathbf{1}}\mathbf{E}\mathbf{K}^{-\mathbf{1}}\mathbf{x} = 0$. Iz epipolarnog orgraničenja za fundamentalnu matricu iz jednadžbe 2.6 dobiva se veza između esencijalne i fundamentalne matrice:

$$\mathbf{E} = \mathbf{K}^{'\mathbf{T}}\mathbf{F}\mathbf{K} \tag{2.8}$$

Kalibracijska matrica definira se kao:

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

gdje su f_x i f_y fokalne udaljenosti kamere za x i y os, s modelira smicanje između osi, a c_x i c_y su središta optičkih osi u pikselima. Parametri matrice $\mathbf{K}$ mogu se pronaći koristeći uzorak poput primjerice uzorka šahovnice. Geometrija uzorka je poznata a postupak kalibracije pronalazi specifične točke i procjenjuje parametre krivulja koje prolaze točkama. Konstrukcijom kalibracijske matrice uklanjaju se linearna izobličenja koja leća kamere unosi u sliku.

Za estimaciju esencijalne matrice koristi se algoritam 5 točaka [15]. S obzirom na epipolarno ograničenje, potrebno je poznavati najmanje 5 korespondentnih parova točaka lijeve i desne slike koje ubacivanjem u epipolarno ograničenje daju linearni sustav.

2.2. Rektifikacija

U prethodnom odjeljku pokazani su postupci kojima se pronalaze esencijalna, odnosno fundamentalna matrica u određivanju epipolarne geometrije scene. Nakon što je određena geometrija scene, epipolarni pravci mogu se koristiti kako bi se uparili pikseli obje slike koji odgovaraju istoj točki scene. Postupak je moguć i bez pronalaženja epipolarnih pravaca, no u tom slučaju pretraživanje je ekstenzivnije i time računalno zahtjevnije. Rektifikacijom se postiže efekt da su epipolarni pravci slika paralelni s x osi slike te se u obje slike nalaze na istoj y vrijednosti. Posljedično, disparitet među korespondentnim pikselima lijeve i desne slike postoji samo po y-osi.

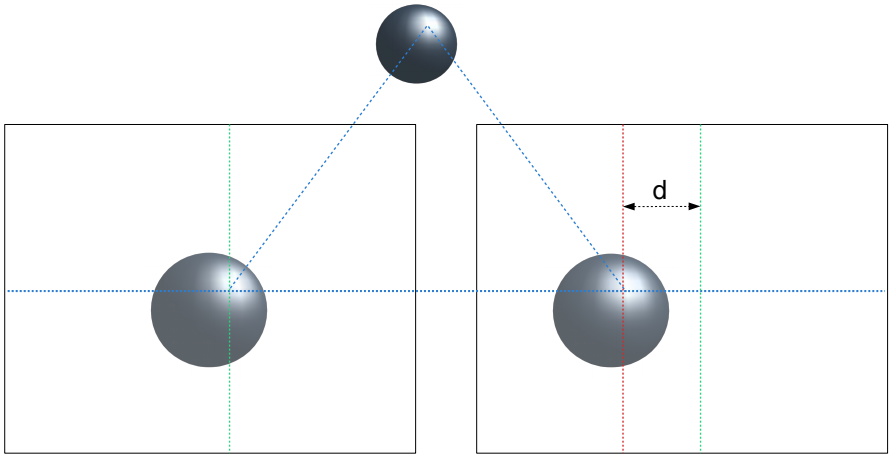
S ciljem uporabe efikasnijeg algoritma ostvarivanja korespondencije, ulazne slike se prethodno *rektificiraju* (primjenjuje se niz transformacija nad matricama slikovnih vrijednosti) na način da su odgovarajući horizontalni pravci upravo epipolarni pravci. Postupak koji ostvaruje rektifikaciju ulaznih slika može se opisati u nekoliko koraka:

- pronaći minimalno 5 korespondentih točaka $\mathbf{x}_i$ i $\mathbf{x}'_i$
- na temelju pronađenih parova točaka, estimirati esencijalnu matricu te epipolove $\mathbf{e}$ i $\mathbf{e}'$.
- pronaći transformaciju $\mathbf{H}'$ koja epipol desne slike $\mathbf{e}'$ preslikava u točku u beskonačnosti $(1, 0, 0)^T$.
- pronaći transformaciju $\mathbf{H}$ koja minimizira udaljenost $\sum_i d(\mathbf{H}\mathbf{x}_i, \mathbf{H}'\mathbf{x}'_i)$
- lijevu i desnu sliku transformirati koristeći pronađene matrice $\mathbf{H}$ i $\mathbf{H}'$

Ovako rektificiran par kamera dovodi piksele kamera i dubinu scene za popratne piksele u odnos izražen s

$$d = f\frac{B}{Z} \tag{2.10}$$

gdje je f fokalna udaljenost kamere mjerena u pikselima, B je udaljenost između središta kamera, Z dubina, a d disparitet. Iz jednadžbe 2.10 vidi se obrnuto proporcionalna veza između dubine Z u 3D prostoru i dispariteta d. Kada je položaj 3D točke blizu kamere, disparitet će biti velik. Vrijedi i obrnut slučaj: disparitet točke na beskonačnoj udaljenosti jednak je 0. Pikseli rektificiranih slika mogu se jednostavno uspoređivati po sličnosti i spremati u *mapu dispariteta* za daljnju obradu. Na slici 2.2 vidljivo je kako su nakon rektifikacije odgovarajući epipolarni pravci paralelni i po y osi se nalaze na istom mjestu.



Slika 2.2: Geometrija rektificiranog stereo para. Plavom bojom označena je epipolarna ravnina te su zelenom i crvenom označene okomite osi projicirane točke u lijevoj, odnosno desnoj slici. Slovom d označen je disparitet projicirane točke.

2.3. Ostvarivanje korespondencije u rektificiranom stereu

Jednom kad je ostvarena rektifikacija stereosustava, moguće je uz smanjenu računalnu složenost ostvariti korespondenciju piksela kamera. Općenito, ostvarivanje korespondencije je pronalaženje parova piksela koji odgovaraju istoj točki u prostoru.

Prve metode ostvarivanja stereo korespondencije ostvarivale su takozvane *rijetke korespondencije* u kojima se ne izračunava disparitet za sve piksele. Danas se rijetke korespondencije ne koriste u velikoj mjeri. Metode za ostvarivanje *guste korespondencije* se danas koriste zahvaljujući snažnoj računalnoj snazi koja je dostupna. Ovi postupci su teži za ostvariti pošto postoje izazovi s kojima se svaki stereo algoritam suočava:

- područja bez teksture,
- refleksivne podloge od kojih se svjetlosna zraka lomi na različit način pri ulasku u svaku od kamera,
- stereoskopska sjena: točke koje se vide iz jedne kamere no ne i iz druge.

U ovom poglavlju opisuje se općeniti postupak stereoskopske rekonstrukcije koji je primjenjiv na većinu algoritama koji rješavaju ovaj problem.

Stereo algoritam može se opisati u četiri koraka [17], [12]

1. izračun podatkovnih cijena - traži se razlika između dvaju slikovnih okana prilikom uparivanja značajki. Unaprijed je definirana mjera razlike korištena u postupku.

2. agregacija cijene - prikupljanje podatkovnih cijena za disparitete u razmatranju.

3. izračun dispariteta i optimizacija - dispariteti se računaju tako da se odaberu najmanje agregirane cijene za svaki piksel.

4. zaglađivanje dispariteta.

Također, algoritmi mogu biti *lokalni* i *globalni* s obzirom na optimizacijski postupak koji koriste. Lokalne metode se ograničavaju na područje u konačnom prozoru oko značajke koja se uparuje dok globalne metode minimiziraju globalni kriterij nad cijelom slikom. Lokalne metode su brze dok globalne generiraju glatke mape dispariteta.

Mjera sličnosti koristi se kod izračuna podatkovne cijene. Preciznost algoritama stereoskopske rekonstrukcije uvelike ovisi o podatkovnim cijenama koje pak ovise o definiranoj mjeri sličnosti slikovnih okana. Pokazuje se da odabir mjere sličnosti može utjecati na konačan rezultat, stoga je ključno koristiti metriku koja je robusna. Neke od mjera sličnosti su:

- srednja kvadratna razlika
- srednja apsolutna razlika
- kvadrat razlike intenziteta piksela
- apsolutna razlika intenziteta piksela

Lokalne metode agregiraju podatkovne cijene nad ograničenim područjem unutar mape dispariteta. Područje pretraživanja može biti dvodimenzionalno (x-y prostor) ili trodimenzionalno (x-y-d prostor). Agregacija podatkovne cijene nad fiksnim područjem pretraživanja može se računati 2D ili 3D konvolucijom kao na je 2.11. Također su moguće izvedbe koje koriste varijabilni prozor pretraživanja.

$$C(x, y, d) = w(x, y, d) * C_0(x, y, d) \tag{2.11}$$

gdje je $C_0(x, y, d)$ početna mapa dispariteta koja konvolucijom s $w(x, y, d)$(postoji više izvedbi) izvodi agregaciju. Ova operacija može biti dvodimenzionalna, ako je fiksiran disparitet d, ili trodimenzionalna u $x - y - d$ prostoru.

Globalne metode ostvarivanja korespondencije optimizacijski postupak vrše odmah nakon izračuna podatkovne cijene, najčešće preskačući korak agregacije cijene.

Definira se funkcija cilja koju je potrebno minimizirati

$$E(d) = E_d(d) + \lambda E_s(d) \tag{2.12}$$

gdje je $E_d(d)$ mjera u kojoj disparitet d odgovara ulaznom paru slika

$$E_d(d) = \sum_{(x,y)} C(x, y, d(x, y)) \tag{2.13}$$

a $E_s(d)$ je mjera glatkosti algoritma te je najčešće ograničena samo na susjedne piksele

$$E_s(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1)) \tag{2.14}$$

gdje je ρ neka monotona rastuća funkcija mjere razlike dispariteta.

3. Duboko učenje kao grana strojnog učenja

Duboko učenje grana je strojnog učenja koja pretpostavlja da kompozitna struktura modela može kvalitetnije opisati probleme koji se rješavaju strojnim učenjem. Duboko učenje postoji već dulje vrijeme, no donedavno se smatralo kako modele nije moguće naučiti bolje od plitkih modela. Razlog tome bio je nedostatak računalne snage. Glavni razlog popularizaciji dubokog učenja je razvoj grafičkog sklopovlja, koje je, uz primjenu u 3D igrama, moćno oružje za učenje dubokih modela zbog sposobnosti paralelne obrade podataka velike memorijske propusnosti. Također su otkrivene metode koje omogućuju brže učenje i korištenje modela većeg kapaciteta te je zadovoljena potreba za velikim podatkovnim skupovima bez kojih bi bilo nemoguće naučiti kvalitetne modele.

U ovom poglavlju definirani su glavni pojmovi u strojnom učenju te su opisani građevni elementi dubokih modela za računalni vid, tehnike učenja i regularizacije.

3.1. Glavni pojmovi u strojnom učenju

Svaki algoritam strojnog učenja čine [19]:

- Model. Funkcija preslikavanja $h(\mathbf{x}|\theta)$ gdje je $\mathbf{x}$ primjer a θ parametri modela.
- Funkcija gubitka. Funkcija koja vrednuje kvalitetu modela za parametre θ na podacima $\mathbf{x}$
- Optimizacijski postupak. Algoritam koji mijenja parametre modela θ s ciljem da smanji vrijednost funkcije gubitka.

Učenje modela uključuje izvođenje optimizacijskog postupka s ciljem minimizacije gubitka na skupu za učenje. Dva razreda problema nadziranog učenja su klasifikacija i regresija.

3.1.1. Klasifikacija i regresija

Klasifikacija je problem u strojnom učenju koji primjeru $\mathbf{x}$ dodjeljuje jednu od C oznaka. Takve oznake nazivaju se razredi ili klase. Izlaz modela definira se kao:

$$h(\mathbf{x}|\theta) = k, k \in [1..C] \tag{3.1}$$

S druge strane, regresija definira izlaz modela kao kontinuiranu vrijednost. Izlaz modela jednak je:

$$h(\mathbf{x}|\theta) = y, y \in \mathbb{R} \tag{3.2}$$

Zadatak regresije je na temelju podataka iz skupa za učenje naučiti kontinuiranu funkciju. Klasifikacija na temelju podataka uči odrediti razred neviđenim primjerima.

3.1.2. Kapacitet modela

Kapacitet modela određuje sposobnost modela za prilagođavanje primjerima za učenje. Objasnit ćemo kapacitet na primjeru regresije polinoma. Na slici 3.1 prikazan je primjer učenja modela koji je polinom n-tog stupnja:

$$h(x|\mathbf{a}) = \sum_{i=0}^{n} a_i x^i \tag{3.3}$$

Gdje su $\mathbf{a}$ parametri modela, a x je ulaz. Funkcija gubitka je srednja kvadratna pogreška:

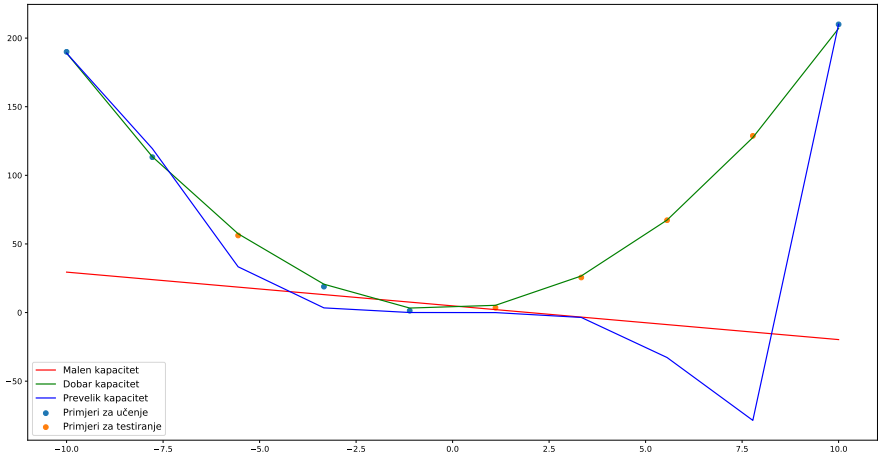
$$L(h|\mathbf{x}) = \frac{1}{N} \sum_{i=0}^{N} (h(x_i) - y_i)^2 \tag{3.4}$$

Model se uči gradijentnim spustom na način da se optimiraju parametri $\mathbf{a}$ uz minimizaciju funkcije gubitka. Na primjeru sa slike, distribucija podataka odgovara polinomu drugog stupnja te su plavom bojom označeni primjeri za učenje, odnosno narančastom primjeri za testiranje. Parametri modela optimiraju se s obzirom na primjere za učenje. Iscrtani su izlazi modela i to:

- Crvenom bojom model polinoma prvog stupnja
- Zelenom bojom model polinoma drugog stupnja
- Plavom bojom model polinoma šestog stupnja

Model polinoma prvog stupnja nema dovoljan kapacitet za distribuciju podataka, dok druga dva modela mogu opisati distribuciju podataka. Razlika između druga dva modela je u tome što model polinoma šestog stupnja ima prevelik kapacitet za dane podatke. Posljedica tome je činjenica da je pogreška na skupu za testiranje znatno veća

kod modela polinoma šestog stupnja zato što se previše prilagodio podacima iz skupa za treniranje. Preveliki kapacitet za distribuciju podataka dovodi do prenaučenosti modela.



Slika 3.1: Usporedba modela s različitim kapacitetima. Crvenom bojom označen je model koji za podatkovni skup ima premalen kapacitet. Zelenom bojom označen je model s dobrim kapacitetom, dok je plavom bojom označen model koji za distribuciju podataka ima prevelik kapacitet i pokazuje preveliko prilagođavanje primjerima za učenje.

3.2. Glavni pojmovi u dubokom učenju

Duboko učenje kao grana strojnog učenja pretpostavlja kompozitnu strukturu modela, odnosno da je izlaz modela kompozicija više funkcija. Duboki modeli imaju velik kapacitet stoga su potrebni veliki skupovi podataka kako bi dobro generalizirali. U ovom odjeljku opisuju se glavni pojmovi u dubokom učenju.

3.2.1. Optimizacijski postupak

Duboki modeli mogu se prikazati kao kompozicija funkcija $\mathbf{h}_l(\mathbf{h}_{l-1}(...\mathbf{h}_1(\mathbf{x}|\theta_1)|\theta_{l-1})|\theta_l)$ pri čemu je $\mathbf{h}_i$ aktivacija i-tog sloja uz parametre θ_i a $\mathbf{x}$ je ulazni podatak. Postupak učenja traži takve parametre modela za koje je funkcija gubitka minimalna. Kod dubokih modela, optimizacijski algoritam je algoritam širenja unazad:

Algoritam 1 Širenje unazad stohastičkim gradijentnim spustom

Ulaz: Primjer $\mathbf{x}$, očekivani izlaz $\mathbf{y}$, stopa učenja η, početni parametri modela $\theta_{1..l}$

$\mathbf{o}_0 \leftarrow \mathbf{x}$

Za $i \in 1..l$ **Ponavljaj**

$\mathbf{o}_i \leftarrow \mathbf{h}_i(\mathbf{o}_{i-1}|\theta_i)$

Kraj

Za $i \in l..1$ **Ponavljaj**

$\theta_i \leftarrow \theta_i - \eta \frac{\delta L}{\delta \mathbf{o}_i} \frac{\mathbf{o}_i}{\delta \theta_i}$

Kraj

Učenje širenjem unazad je osnovni optimizacijski postupak učenja dubokih modela. U međuvremenu su se pojavili algoritmi koji ostvaruju ubrzanje konvergencije. Neki od tih algoritama su:

- Stohastički gradijentni spust s momentom
- Učenje s Nesterovljevim momentom [3]
- AdaGrad [4]
- ADAM [11]

Za velik broj modela, algoritam učenja ADAM (engl. *Adaptive Moments*) ostvaruje najbržu konvergenciju. Algoritam održava eksponencijalni pomični prosjek gradijenata i njihovih kvadrata. Ažuriranje vrijednosti parametara modela vrši se korištenjem procjena prvih i drugih momenata gradijenata. Pseudokod algoritma dan je s:

Algoritam 2 Učenje algoritmom ADAM

Ulaz: Primjeri $\mathbf{x}_{1..N}$, očekivani izlaz $\mathbf{y}$, stopa učenja η, početni parametri modela θ

Ulaz: ϵ, ρ_1, ρ_2

$\mathbf{s} = \mathbf{0}$, $\mathbf{r} = \mathbf{0}$

t = 0

Sve dok uvjet konvergencije nije zadovoljen **Ponavljaj**

$\mathbf{x} \leftarrow mini\_grupa(\mathbf{x}_{1..N})$

$\mathbf{g} \leftarrow \frac{1}{m}\nabla_\theta \sum_{i=0}^{m} L(\mathbf{h}(\mathbf{x_i}|\theta), \mathbf{y_i})$

$\mathbf{s} \leftarrow \rho_1\mathbf{s} + (1-\rho_1)\mathbf{g}$

$\mathbf{r} \leftarrow \rho_2\mathbf{r} + (1-\rho_2)\mathbf{g} \odot \mathbf{g}$

$\hat{\mathbf{s}} \leftarrow \frac{\mathbf{s}}{1-\rho_1^t}$

$\hat{\mathbf{r}} \leftarrow \frac{\mathbf{r}}{1-\rho_2^t}$

$\theta \leftarrow \theta - \eta\frac{\hat{\mathbf{s}}}{\sqrt{\hat{\mathbf{r}}} + \delta}$

Kraj

3.2.2. Konvolucijski slojevi

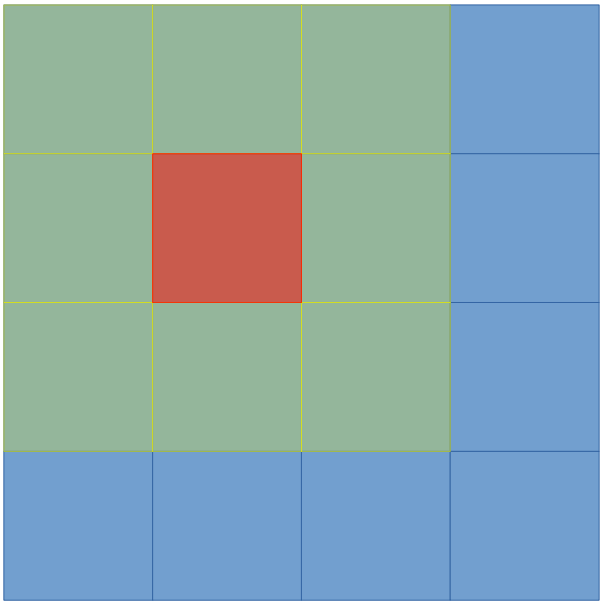
Konvolucija je operacija filtriranja signala koja se definira kao:

$$\mathbf{I}(i) * \mathbf{K}(i) = \sum_m \mathbf{I}(m)\mathbf{K}(i-m) \tag{3.5}$$

gdje je I ulazni signal a K filtar. U obradi slika i dubokom učenju za računalni vid, koristi se operacija 2D konvolucije izražena s:

$$\mathbf{I}(i,j) * \mathbf{K}(i,j) = \sum_{c\in C} \sum_{m=i-\left\lfloor \frac{k_x}{2} \right\rfloor}^{m=i+\left\lceil \frac{k_x}{2} \right\rceil} \sum_{n=j-\left\lfloor \frac{k_y}{2} \right\rfloor}^{n=j+\left\lceil \frac{k_y}{2} \right\rceil} \mathbf{I}(m,n,c)\mathbf{K}(i-m, j-n, c) \tag{3.6}$$

pri čemu ulazna slika $\mathbf{I}$ ima C kanala. Veličina konvolucijske jezgre je pritom $k_x \times k_y$. 2D konvolucija je osnovni građevni element dubokih konvolucijskih modela. Konvolucijski sloj čini konvolucija ulaza s više jezgara gdje model može imati više slojeva. Izlaz 2D konvolucije naziva se mapa značajki te izlaz iz konvolucijskog sloja čini više mapa značajki. Izlazne mape značajki su 3D struktura. Za ulaznu sliku $W \times H \times C$ i F konvolucijskih jezgara dimenzija $K \times K \times C$, izlaz iz konvolucijskog sloja bit će tenzor dimenzija $(H-(K-1)) \times (W-(K-1)) \times F$. Razlog smanjenju rezolucije



Slika 3.2: Konvolucija nad prvim mogućim pikselom ulazne slike. Plavom bojom označena je ulazna slika, žutom bojom označeni elementi konvolucijske jezgre a crvenom bojom piksel ulazne slike nad kojim se vrši konvolucija.

opisan je slikom 3.2. Za konvolucijsku jezgru dimenzija 3×3, konvolucija se može izvesti tek nad drugim retkom i drugim stupcem slike. U općenitom slučaju, za konvolucijsku jezgru dimenzija $K \times K$, prvi element nad kojim se može izvesti konvolucija je $I(\lfloor \frac{K}{2} \rfloor, \lfloor \frac{K}{2} \rfloor)$. U slučaju kada je potrebno zadržati rezoluciju nakon konvolucijskog sloja, potrebno je popuniti rubove oko izlaza. Najčešća metoda koja se pritom koristi je popunjavanje nulama.

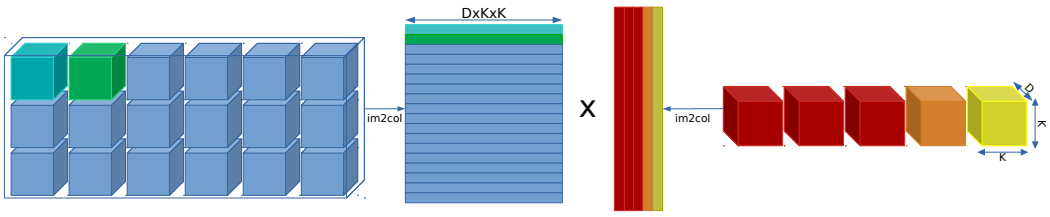
Operacija konvolucije može se izvoditi i na način da se preskaču neki elementi ulaza. Za takvu konvoluciju definira se korak P koji označuje da se konvolucija izvodi za svaki P-ti element ulaza. Konvolucija s korakom 1 upravo je operacija definirana jednadžbom 3.6.

3.2.3. Paralelizacija konvolucije na GPU

U konvolucijskim slojevima, broj jezgara određuje broj izlaznih mapa značajki sloja. Dimenzija konvolucijskog filtra je $K \times K \times D$, gdje je K najčešće 3, 5 ili 7, a D odgovara dubini ulaza. Za konvolucijski sloj sa 64 jezgre dimenzije $3 \times 3 \times 4$ i ulaznu sliku $100 \times 200 \times 4$, potrebno je napraviti 77616 skalarnih produkata od kojih se svaki sastoji od 36 množenja i 35 zbrajanja i tako za svaku od 64 jezgre. Pritom su

proračuni pojedinih elemenata izlaza međusobno nezavisni, stoga je operaciju moguće paralelizirati.

Implementacija matričnog množenja također je efikasna na GPU te se konvolucija izvodi uz pomoć matričnog množenja. Na slici 3.3 prikazane su ulazna slika te konvolucijske jezgre. Oko svakog piksela nad kojim se radi konvolucija uzima se okno dimenzija jezgara te se svaki 3D tenzor pretvara u vektor. Na slici je to označeno operacijom *im2col* [8]. Zbog preglednosti ilustracije okna se ne preklapaju, što u stvarnoj implementaciji nije slučaj već odmak među oknima određuje korak konvolucije. Slaganjem okana dobiva se matrica dimenzija $(K \cdot K \cdot D) \times broj\_okana$. Slično se radi za konvolucijske jezgre: svaka jezgra pretvara se u vektor te se za više jezgara dobiva matrica dimenzija $(K \cdot K \cdot D) \times F$, gdje je F broj izlaznih mapa značajki te odgovara broju konvolucijskih filtara. Za prethodni primjer, bilo bi potrebno pomnožiti matrice dimenzija 77616×36 i 36×64. Dobivenu matricu potrebno je presložiti u tenzor dimenzija 98x198x64. Vrijedi primijetiti kako okna oko susjednih piksela dijele elemente, tako da se većina elemenata kopira više puta. Ovo je prihvatljivo s obzirom da je brzina izvođenja faktor koji je većini ključan, ali objašnjava kako duboki modeli zahtijevaju veliku količinu radne memorije.



Slika 3.3: Izvedba konvolucije pomoću matričnog množenja.

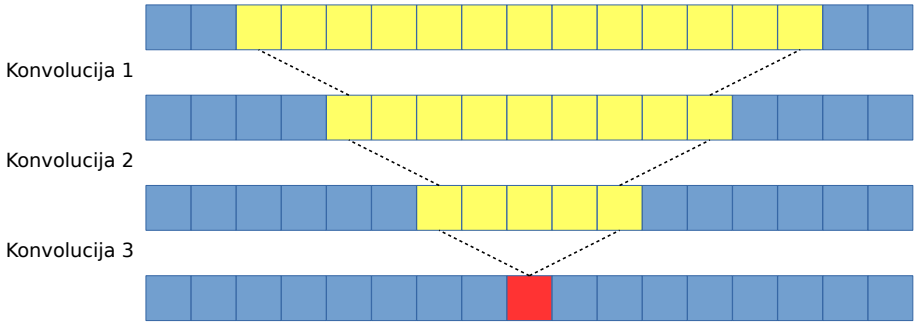
3.2.4. Receptivno polje

Duboki konvolucijski modeli za računalni vid na ulazu primaju slike. Za mnoge zadatke potrebni su modeli s velikim receptivnim poljem [5]. Receptivno polje definira se kao broj piksela ulazne slike koji utječu na svaki element izlazne mape značajki nekog sloja. Receptivno polje omogućuje modelu zaključivanje na temelju većeg odsječka slike. Postizanje velikog receptivnog polja moguće je ostvariti na više načina te često korištene arhitekture modela koriste više njih. Najčešće metode postizanja većeg receptivnog polja su:

– povećanje broja konvolucijskih slojeva

– slojevi sažimanja

Ranije je navedeno kako operacija konvolucije djeluje nad središnjim pikselom istovremeno djelujući na okolinu u veličini konvolucijske jezgre. Za konvolucijsku jezgru veličine $K \times K$, receptivno polje veličine je $K \times K$. Nizanjem konvolucijskih slojeva receptivno polje modela se proširuje. Slika 3.4 pokazuje kako se receptivno polje proširuje nizanjem konvolucijskih slojeva s jezgrom veličine $K = 5$. Crvenom bojom je označen element izlaza na koji djeluju žuto označeni elementi svakog sloja. Gornji sloj na slici predstavlja ulaznu sliku. Usprkos činjenici da modeli s više konvolucijskih slojeva postižu veće receptivno polje, dodavanjem slojeva model se produbljuje i ima više parametara, stoga ga je teže naučiti.



Slika 3.4: Porast receptivnog polja modela nizanjem konvolucijskih slojeva. Zbog jednostavnosti prikaza, no bez smanjenja općenitosti, izostavljena je druga dimenzija slike. Svi konvolucijski slojevi sa slike imaju jezgru $K = 5$.

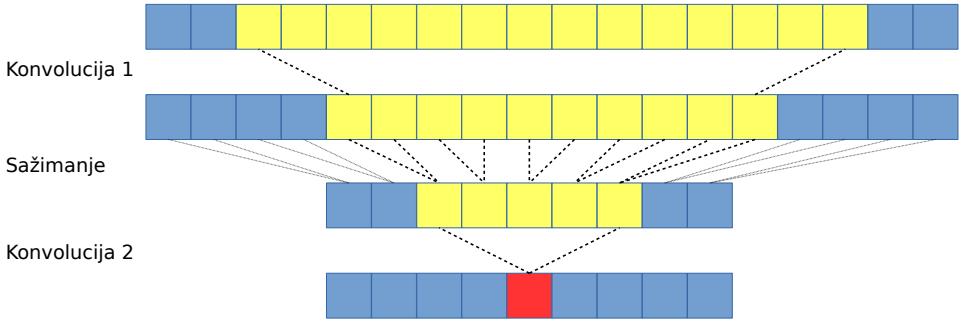
Slojevi sažimanja ostvaruju proširenje receptivnog polja modela stapajući elemente ulaza. Dvije najčešće izvedbe slojeva sažimanja su:

– sažimanje najvećom vrijednosti: u klizećem prozoru dimenzija $P \times P$ uzima se najveći element prozora.

– sažimanje prosječnom vrijednosti: u klizećem prozoru dimenzija $P \times P$ uzima se prosječna vrijednost elemenata.

Korak sloja sažimanja P najčešće se postavlja na 2. Pritom se postiže smanjenje rezolucije P puta u obje slikovne dimenzije. Bitno je primijetiti kako sažimanje smanjuje ulaz te ima značajan utjecaj na porast performansi. U slučaju 2D konvolucijskog sloja, sloj sažimanja koji mu prethodi smanjuje broj elemenata P^2 puta, što je značajna ušteda memorije. Klizeći prozor sloja sažimanja, za razliku od konvolucije, najčešće

ne vrši preklapanje elemenata. Efekt smanjenja rezolucije konvolucijom moguće je ostvariti korakom konvolucije.



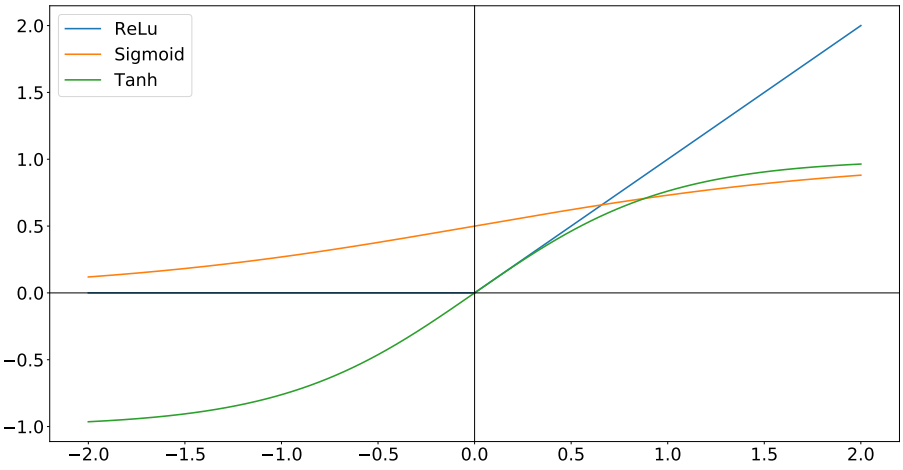
Slika 3.5: Porast receptivnog polja modela konvolucijskim slojevima uz sloj sažimanja. Zbog jednostavnosti prikaza, no bez smanjenja općenitosti, izostavljena je druga dimenzija slike. Svi konvolucijski slojevi sa slike imaju jezgru $K = 5$ dok sažimanje smanjuje ulaz za 2 puta.

3.2.5. Aktivacijska funkcija

Kako bi bilo moguće modelirati nelinearne funkcije, potrebna je aktivacijska funkcija koja je također nelinearna. U dubokim modelima najčešće se koriste:

- Zglobnica: $ReLu(x) = max(0, x)$
- Sigmoidalna aktivacija: $\sigma(x) = \dfrac{1}{1 + e^{-x}}$
- Tangens hiperbolni: $tanh(x) = \dfrac{1 - e^{-2x}}{1 + e^{-2x}}$

i prikazane su na slici 3.6. U konvolucijskim modelima, nelinearna aktivacija nalazi se nakon izlaza konvolucijskog sloja.



Slika 3.6: Najčešće korištene aktivacijske funkcije.

3.2.6. Aktivacijska funkcija softmax

Kod klasifikacijskih problema korisno je da izlaz modela ima vjerojatnosnu reprezentaciju. Operacija $softmax$ pruža upravo to:

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{C} e^{x_j}} \tag{3.7}$$

Vrijedi $\sum_{i=0}^{C} \sigma(x_i) = 1$ stoga se može odrediti vjerojatnost svakog od razreda za primjer x_i. Funkcija $softmax$ je generalizacija sigmoidalne aktivacije logističke regresije. Također, funkcija gubitka za model sa $softmax$ izlazom je unakrsna entropija:

$$-\frac{1}{N}\sum_{n=0}^{N}\sum_{c=0}^{C} P(y_n = c)\log h(\mathbf{x}_n|\theta)_c \tag{3.8}$$

gdje je y_n oznaka razreda za primjer $\mathbf{x}_n$, $P(y_n = c)$ je vjerojatnost da je oznaka n-tog primjera c dok je $h(\mathbf{x}_n|\theta)$ izlaz modela.

3.2.7. Regularizacija dubokih modela

Regularizacija je u strojnom učenju metoda ili skup metoda kojima se sprečava prenaučenost modela. Prenaučenost se može definirati kao velika razlika između točnosti na skupu za učenje i skupu za testiranje, gdje je na skupu za testiranje točnost manja. Metode regularizacije ugrađuju se u funkciju gubitka, definiraju se kao zasebni slojevi ili se koriste prilikom treniranja modela. Neke od tih metoda su:

– rano zaustavljanje

– kažnjavanje L norme parametara: u funkciju gubitka ugrađuje se dodatni gubitak koji ograničava parametre modela

– normalizacija po grupi

Rano zaustavljanje izvodi se tijekom učenja modela: iz skupa za učenje izdvajaju se primjeri nad kojima se model ne uči, no mjeri se gubitak nad njima. Kao konačni parametri postavljaju se oni za koje je gubitak na izdvojenom skupu minimalan. Dodatno se može definirati broj epoha učenja tijekom kojih nisu pronađeni bolji parametri te se učenje prekida.

L-norme parametara kažnjavaju model na način da u izvornu funkciju $L(\mathbf{x})$ gubitka ugrađuju normu na parametre:

$$L^{'}(\mathbf{x}) = L(\mathbf{x}) + \lambda \left\| \theta \right\|_N \tag{3.9}$$

gdje je λ regularizacijski faktor. Primjerice, L2 norma parametara $\mathbf{w}$ može se definirati kao:

$$\sqrt{\sum_{i=0}^{N} w_i^2} \tag{3.10}$$

Normalizacija po grupama metoda je koja ubrzava učenje dubokih modela istovremeno imajući regularizacijski učinak [9]. Uzevši u obzir činjenicu da se parametri svakog sloja mijenjaju tijekom učenja, mijenjaju se aktivacije svakog sloja. Ovisnost nekog sloja o svim prethodnim slojevima te promjena njihovih izlaza može znatno otežati učenje. Normalizacija po grupama može poništiti takvo ponašanje. Ideja postupka je normalizirati izlaze svakog sloja na razdiobu $\mathcal{N}(0, 1)$. Posljedica je da ulazi u svaki sloj kojem prethodi normalizacija grupe imaju sličnu razdiobu tijekom procesa učenja te se postiže ubrzanje učenja te bolja propagacija gradijenata. Algoritam sloja normalizacije po grupama je slijedeći:

Algoritam 3 Normalizacija po grupama

Ulaz: Vrijednosti ulaza $\mathbf{x}$ u mini grupi: $B = \mathbf{x}_{1..m}$

Izlaz: $\mathbf{y}_i = BN_{\gamma,\beta}(\mathbf{x}_i)$

$\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i$

$\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (\mathbf{x}_i - \mu_B)^2$

$\hat{\mathbf{x}}_i \leftarrow \frac{\mathbf{x}_i - \mu_B}{\sqrt{\sigma_B + \epsilon}}$

$\mathbf{y}_i \leftarrow \gamma \hat{\mathbf{x}}_i + \beta \equiv BN_{\gamma,\beta}(\mathbf{x}_i)$

Varijable regularizacije po grupama γ i β su parametri modela te se optimiraju prilikom učenja. U konkretnoj implementaciji normalizacije po grupi za konvolucijske slojeve, srednja vrijednost i varijanca izlaza računaju se posebno za svaku izlaznu mapu značajki. Za izlazni tenzor dimenzija $N \times H \times W \times F$, gdje je N veličina mini-grupe a F broj mapa značajki, μ_B i σ_B^2 su vektori s F elemenata.

Regularizacijom modela postiže se manja pristranost modela podacima za učenje s ciljem da model ostvari bolju mogućnosti generalizacije.

3.2.8. Memorijski izazovi prilikom učenja

Neka sloj čine 3 operacije: konvolucija, normalizacija po grupama te neka nelinearnost. Neka je ulazna slika dimenzija 256×512, konvolucija koristi popunjavanje nulama zbog zadržavanja rezolucije i ima 64 izlazne značajke. Za ažuriranje parametara konvolucije potrebno je poznavati aktivacije sve 3 operacije gdje je tenzor aktivacija svake od operacija $256 \times 512 \times 64 = 8MB$, dakle ukupno $24MB$ po aktivacijama sloja. Za 20 slojeva aktivacije su veličine $480MB$ i ako se koristi mini grupa veličine 10 ukupne aktivacije iznose $4800MB$. Sve aktivacije je potrebno čuvati u memoriji prilikom izračuna gradijenata.

Vidljivo je kako je memorija ključan resurs za učenje dubokih modela. Moderne grafičke kartice imaju kapacitet i do $12GB$ te je moguće koristiti više kartica u jednom računalu, no usprkos tome postoje ograničenja na arhitekturu modela.

4. Primjena dubokog učenja za stereoskopsku rekonstrukciju

Do sad su opisani postupci koji prethode algoritmu stereo rekonstrukcije: kalibracija i rektifikacija stereo para. Opisane su i glavne komponente dubokog učenja kao grane strojnog učenja te postupci treniranja modela. U ovom poglavlju opisane su dvije arhitekture dubokih modela koje se koriste za stereoskopsku rekonstrukciju. Prva metoda ostvaruje ugrađivanje slikovnih okana u visokodimenzionalan metrički prostor s mogućnošću uspoređivanja po sličnosti. Vrijedna značajka ove metode je primjenjivost u svim postupcima koji koriste metriku sličnosti slikovnih okana poput računanja optičkog toka ili određivanje vlastitog gibanja kamere. Druga arhitektura za ulazni stereo par na izlazu daje glatku mapu dispariteta te je takav model učen s kraja na kraj. Ovakva arhitektura je kompletan stereo postupak za sebe te se pokazalo kako naučeni stereo postupci postižu najbolje rezultate na javno dostupnim podatkovnim skupovima.

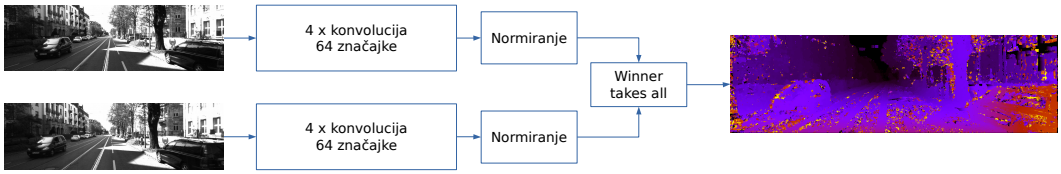
4.1. Ugrađivanje okana u metrički prostor za ostvarivanje korespondencije

Metrika korištena pri izračunu podatkovnih cijena bitan je dio stereo algoritma. Metrika mora biti robusna s obzirom na šum i velike razlike u svjetlosnim razinama susjednih piksela. S druge strane, poželjno je da metrika nosi korisnu informaciju u područjima slike bez teksture što predstavlja najveći problem stereo algoritmima. Ovi uvjeti zadovoljeni su dubokim modelom naučenim na slikama s poznatom dubinskom informacijom.

Metričko korespondencijsko ugrađivanje može se ostvariti dubokim modelom [18]. Model se sastoji od 4 konvolucijska sloja u kojem svaki sloj ima 64 značajke te je veličina svake konvolucijske jezgre 3×3. Ulaz u model je slikovno okno dimenzija $P \times P$ te je izlaz ugrađivanje okna u 64-dimenzionalni vektor. Prednost konvolucijske

arhitekture je činjenica da se ugrađivanje može izvesti nad čitavom slikom u jednom prolazu, tako da je izlaz za sliku dimenzija $W \times H$ ugrađivanje dimenzija $W \times H \times 64$. Nakon zadnjeg konvolucijskog sloja izlazi se normiraju na jedinični vektor uzduž osi značajki.

Korištenjem ugrađivanja dobivenih dubokim modelom, moguće je izračunati mapu dispariteta. Na slici 4.1 prikazana je arhitektura modela za računanje dispariteta. Za ulazni par slika, I_L i I_D, konvolucijski slojevi modela daju ugrađivanja E_L i E_D koja su dimenzija $W \times H \times 64$. Uzevši da je slika I_L referentna te da se razmatraju dispariteti do D, ugrađivanja desne slike posmiču se za disparitet $d \in [0..D]$ uzduž vodoravne osi te se za svaki piksel vrši skalarni umnožak vektora reprezentacija na odgovarajućim mjestima. Rezultat je 3D tenzor dimenzija $W \times H \times D$ gdje se uzduž treće dimenzije nalaze mjere sličnosti piksela na odgovarajućim disparitetima. Ako se za svaki piksel uzduž treće osi napravi operacija $argmin$, dobiva se mapa dispariteta. Opisani postupak na slici je označen kao blok *Winner takes all*, zato što se kao konačni disparitet razmatraju samo najsličniji pikseli ne uzimajući šire susjedstvo u obzir. U usporedbi s globalnim algoritmima stereoskopske rekonstrukcije, ovakav pristup ne uzima u obzir kriterij glatkoće.



Slika 4.1: Model za korespondencijsku metriku. Ulazni stereo par slika prolazi kroz 4 konvolucijska sloja gdje se parametri dijele za lijevu i desnu sliku. Slijedi normiranje po osi značajki te pronalazak najsličnijeg piksela kao izlaznog dispariteta.

Zadaća modela je ostvariti ugrađivanje koje ima svojstvo preslikavanja u prostor u kojem su slična slikovna okna bliska po kosinusnoj udaljenosti. S tim ciljem stvaraju se primjeri za učenje nad skupom slika za koju je poznata mapa dispariteta i to na slijedeći način:

- Za okno oko piksela lijeve slike $R = I_L^{PxP}(x, y)$ pronalazi se piksel desne slike na odgovarajućem disparitetu: $P = I_D^{PxP}(x, y - d)$
- Za isti piksel lijeve slike R pronalazi se piksel desne slike na nasumičnom odmaku od odgovarajućeg dispariteta: $Q = I_D^{PxP}(x, y - d - o)$
- Trojka $< R, P, Q >$ čine jedan primjer za učenje.

Optimizacijski postupak minimizira slijedeću funkciju gubitka:

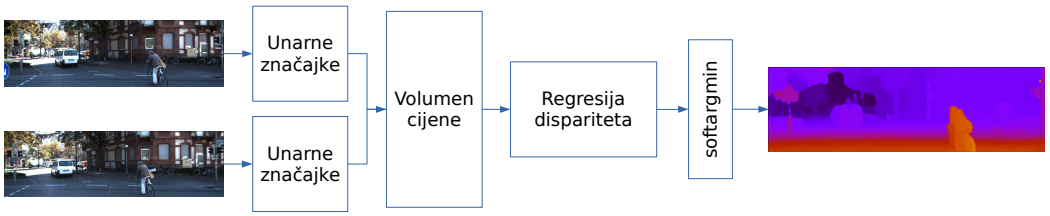
$$L(R, P, Q) = max(0, m + \mathbf{h}(\mathbf{R}|\theta) \cdot \mathbf{h}(\mathbf{Q}|\theta) - \mathbf{h}(\mathbf{R}|\theta) \cdot \mathbf{h}(\mathbf{P}|\theta)) \tag{4.1}$$

gdje je h izlaz modela uz parametre θ. Ovakva funkcija gubitka zahtjeva da je ugrađivanje referentnog okna bliže ugrađivanju pozitivnog primjera za najmanje m. Drugim riječima, funkcija gubitka implicira kako model uči samo za one primjere u kojem je sličnost R i P manja od sličnosti između R i Q za više od m.

4.2. Integrirani rekonstrukcijski model koji se uči s kraja na kraj

Duboki model opisan u prethodnom poglavlju koristi se u postupku stereoskopske rekonstrukcije s ciljem da klasičnoj metodi, poput SGM-a, ostvari precizne korespondencijske metrike te se pokazalo kako takav pristup ostvaruje precizne rekonstrukcije. U ovom poglavlju opisan je duboki model koji za ulazni par slika na izlazu daje mapu dispariteta [10]. Drugim riječima, ovakav model sposoban je naučiti čitav stereo postupak i pritom postiže bolje rezultate od prethodno opisanih metoda. Usprkos činjenici da model u potpunosti zamjenjuje tipične stereo postupke, arhitektura modela inspirirana je klasičnim postupcima.

Na slici 4.2 prikazani su dijelovi modela. Slojevi unarnih značajki sastoje se od 18 konvolucijskih slojeva s rezidualnim vezama. Ulaz u sloj unarnih značajki su lijeva i desna slika stereo para te se prilikom prolaza dijele težine za obje slike. Dobivene reprezentacije lijeve i desne slike koriste se za izgradnju volumena cijene: za svaki disparitet d reprezentacije desne slike posmiču se za d mjesta udesno po vodoravnoj osi te se konkateniraju s reprezentacijama lijeve slike. Dobiveni tenzor za ulazni par slika širine W i visine H, najveći disparitet u razmatranju D i F unarnih značajki dimenzija je WxHxDx2F.



Slika 4.2: Duboki model za stereoskopsku rekonstrukciju. Ulaz u model čini stereo par gdje slojevi unarnih značajki dijele parametre za obje slike. Značajke lijeve i desne slike sažimaju se u volumen cijene te regresija dispariteta na svom izlazu daje aktivacije dispariteta za svaki piksel. Konačno, $softargmin$ računa mapu dispariteta čitave slike.

Volumen cijene, koji je 4D tenzor, u nastavku modela ulazi u niz 3D konvolucijskih slojeva. Slično kao i kod algoritma SGM, disparitet na izlazu je onaj koji minimizira aktivacije (kod SGM-a cijene) zadnjeg konvolucijskog sloja. Izračun minimalne cijene vrši operacija $softargmin$ definirana formulom 4.2, gdje je $\sigma(-c_d)$ aktivacija $softmax$ definirana formulom 3.7 nad elementima izlaza zadnjeg konvolucijskog sloja, uzduž osi dispariteta.

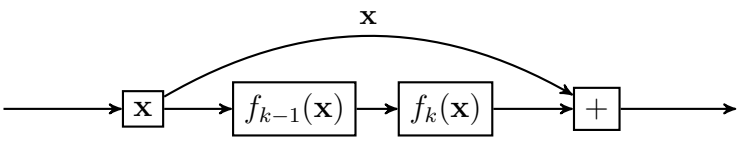
$$softargmin = \sum_{d=0}^{d=D} d \cdot \sigma(-c_d) \tag{4.2}$$

Prednost ovako definiranog izlaza dubokog modela za stereoskopsku rekonstrukciju nad operacijom $argmin$ leži u činjenici da je operacija $softargmin$ derivabilna funkcija, što omogućuje učenje dubokog modela s kraja na kraj. Također vrijedi naglasiti kako izlazi mogu biti realni brojevi, što modelu omogućuje da nauči rekonstrukcije ispod preciznosti piksela, čime se anulira uzorkovanje kamerom. Arhitektura modela u potpunosti je prikazana tablicom 4.1.

Model sadrži rezidualne veze [7] za koje se pokazalo da ubrzavaju učenje i omogućuju da modeli imaju više slojeva, efektivno imajući šire receptivno polje, te postižu bolje rezultate. Rezidualna veza je operacija zbroja dvaju izlaznih tenzora slojeva po elementima na način da se operacijom zbroja preskače sloj među njima. Argument za korištenje rezidualne veze je slijedeći: za distribuciju podataka koja se želi naučiti, $h(\mathbf{x})$, uz rezidualnu vezu, izlaz sloja koji se tada želi naučiti je $f(\mathbf{x}) - \mathbf{x}$. U slučaju da je $h(\mathbf{x})$ identitet, odnosno $h(\mathbf{x}) \equiv \mathbf{x}$, lakše je optimirati funkciju $f(\mathbf{x})$ da na izlazu daje $\mathbf{0}$ kada postoji rezidualna veza, nego da se izravno uči $f(\mathbf{x}) \equiv h(\mathbf{x})$. Pokazalo se da rezidualne veze omogućuju bolju propagaciju gradijenata k početnim slojevima modela,

Tablica 4.1: Arhitektura dubokog modela za stereo rekonstrukciju

Sloj	Opis	Dimenzije izlaznog tenzora
	Ulazna slika	HxWxC
1	Konvolucija 5x5, 32 značajke	H/2 x W/2 x F
2	Konvolucija 3x3, 32 značajke	H/2 x W/2 x F
3	Konvolucija 3x3, 32 značajke	H/2 x W/2 x F
	Rezidualna veza sloja 1 i 3	
4-17	Ponovljeni slojevi 2 i 3 te rezidualna veza	H/2 x W/2 x F
18	Konvolucija 3x3, 32 značajke bez BN	H/2 x W/2 x F
	Volumen cijene	D/2 x H/2 x W/2 x 2F
19 - 20	3D kovolucija 3x3x3, 32 značajke	D/2 x H/2 x W/2 x F
	Rezidualna veza slojeva 19 i 20	
21 - 23	3D kovolucija 3x3x3, 64 značajke	D/4 x H/4 x W/4 x 2F
	Rezidualna veza slojeva 21 i 23	
24 - 26	3D kovolucija 3x3x3, 64 značajke	D/8 x H/8 x W/8 x 2F
	Rezidualna veza slojeva 24 i 26	
27 - 29	3D kovolucija 3x3x3, 64 značajke	D/16 x H/16 x W/16 x 2F
	Rezidualna veza slojeva 27 i 29	
30 - 32	3D kovolucija 3x3x3, 128 značajki	D/32 x H/32 x W/32 x 4F
33	3D dekonvolucija 3x3x3, 64 značajke	D/16 x H/16 x W/16 x 2F
	Rezidualna veza slojeva 29 i 33	
34	3D dekonvolucija 3x3x3, 64 značajke	D/8 x H/8 x W/8 x 2F
	Rezidualna veza slojeva 26 i 34	
35	3D dekonvolucija 3x3x3, 64 značajke	D/4 x H/4 x W/4 x 2F
	Rezidualna veza slojeva 23 i 35	
36	3D dekonvolucija 3x3x3, 32 značajke	D/2 x H/2 x W/2 x F
	Rezidualna veza slojeva 20 i 36	
37	3D dekonvolucija 3x3x3, 1 značajka	D x H x W x 1
	$softargmin$	H x W



Slika 4.3: Rezidualna veza prikazana izračunskim grafom

što je razlog zašto je moguće ostvariti dublje modele dobrih performansi. Izračunski graf za slojeve s rezidualnim vezama prikazan je na slici 4.3

Model se trenira na način da se minimizira srednja apsolutna pogreška izračunatih dispariteta i stvarnih dispariteta:

$$L = \frac{1}{N} \sum_{n=0}^{N} |h(\mathbf{I_L}, \mathbf{I_D}|\theta)_n - d_n| \tag{4.3}$$

gdje je d točan disparitet, $h(\mathbf{I_L}, \mathbf{I_D}|\theta)$ je izlaz modela uz parametre θ. Vrijednost gubitka se usrednjuje s obzirom na broj poznatih dispariteta, što je bitno za bolju procjenu gradijenata kada su poznati dispariteti rijetki. Gubitak srednje kvadratne pogreške tipično se koristi prilikom regresijskih problema te se pokazalo kako se bolji rezultati postižu kada se stereoskopska rekonstrukcija formulira kao regresijski problem, a ne kao problem klasifikacije cjelobrojnog dispariteta [10].

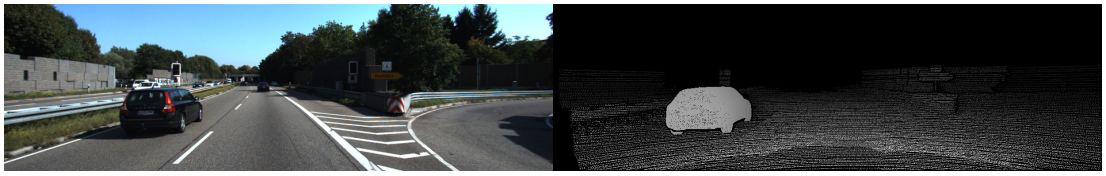
5. Podatkovni skupovi za učenje postupka stereoskopske rekonstrukcije

Modeli opisani u prethodnom poglavlju uče se nadzirano. Duboki modeli najčešće zahtijevaju velike podatkovne skupove za učenje kako bi imali dobru mogućnost generalizacije. S tim ciljem stvoren je niz podatkovnih skupova slika koji sadrže precizne disparitete za vrednovanje rekonstrukcijske točnosti. Za naučene rekonstrukcijske postupke ti dispariteti mogu se iskoristiti u procesu učenja. U ovom poglavlju opisana su dva takva podatkovna skupa, KITTI 2015 [14] i SceneFlow Driving [16]. Opisane su i specifičnosti prilikom učenja te postupci pripreme podataka.

5.1. Podatkovni skup KITTI 2015

Podatkovni skup KITTI sadrži 200 parova slika za koje postoje mape dispariteta mjerene laserskim uređajem. Stereo par kamera nalazi se na krovu automobila te slike prikazuju scene iz vožnje u gradskim sredinama. Za oko 30% piksela u podatkovnom skupu su poznati stvarni dispariteti. Stereo par je kalibriran te rektificiran. Ovaj podatkovni skup primamljiv je zbog sve veće prisutnosti računalnog vida u području autonomne vožnje.

Iako su mape laserski dobivenih dispariteta rijetke, u post obradi su dispariteti na automobilima progušćeni ubacivanjem 3D modela automobila u odgovarajući oblak točaka. Ovo implicira kako je ključno da stereo postupak bude precizan na automobilima kako bi ostvario visoku točnost. Automobili najčešće imaju refleksivne površine, što predstavlja problem rekonstrukcijskim algoritmima pošto ista 3D točka refleksivne površine može izgledati drugačije promatrajući ju iz svake kamere stereo para. Rijetke mape dispariteta imaju vrijednost nula na mjestima gdje nisu poznati dispariteti. Ranije je spomenuto kako je disparitet 3D točke u beskonačnosti također jednak nuli što znači da modeli neće biti učeni na udaljenim objektima, poput neba. Slika 5.1 pokazuje jednu od slika iz podatkovnog skupa te pripadajući disparitet.



Slika 5.1: Primjer iz KITTI skupa. Lijevo referentna slika, desno odgovarajući disparitet.

5.2. Podatkovni skup SceneFlow Driving

Duboki modeli imaju velik kapacitet, stoga je potreban velik skup podataka za treniranje kako bi modeli dobro generalizirali. Ručno pribavljanje slika s preciznom informacijom o disparitetima je dugotrajan proces, stoga je stvoren podatkovni skup SceneFlow. Podatkovni skup SceneFlow Driving sadrži 4400 slika dobivenih iz 3D virtualnog prostora te postoje gusti dispariteti za svaki stereo par. Guste mape dispariteta omogućuju modelima da nauče rekonstrukcije na disparitetu 0, što se manifestira točnim rekonstrukcijama na primjerice nebu. Primjer iz podatkovnog skupa prikazan je na slici 5.2.

Iako je modele velikog kapaciteta korisno trenirati na velikim podatkovnim skupovima s ciljem bolje generalizacije, bitno je postaviti pitanje hoće li razlika između slika pribavljenih kamerom i slika pribavljenih umjetno iz virtualnog prostora imati utjecaj na rekonstrukcijsku točnost postupka. Svakako je potrebno vrednovati postupak sa i bez prethodnog treniranja na umjetnom podatkovnom skupu.

Slika 5.2: Primjer iz SceneFlow skupa. Lijevo referentna slika, desno odgovarajući disparitet.

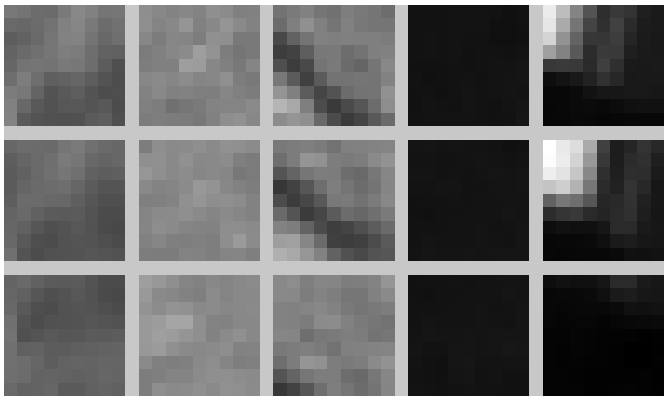
5.3. Učenje modela za ugrađivanje okana u metrički prostor za ostvarivanje korespondencije

Duboki model za korespondencijsku metriku učen je na podatkovnom skupu KITTI. Slike su obrađene na način da su nad čitavim podatkovnim skupom izračunate sred-

nja vrijednost μ i standardna devijacija σ te su pikseli $I(x, y)$ svedeni na normalnu razdiobu $\mathcal{N}(0, 1)$:

$$I'(x, y) = \frac{I(x, y) - \mu}{\sigma} \tag{5.1}$$

Iz ovako obrađenih slika izdvajaju se primjeri za učenje, te ih na KITTI skupu za učenje ima 17 milijuna. Na slici 5.3 prikazani su primjeri za učenje dubokog modela. Gornji red pokazuje okna referentne (lijeve) slike, srednji red pokazuje pozitivne primjere a donji negativne. Zadaća modela je naučiti ugrađivanje u visokodimenzionalni prostor u kojem je vektor značajki referentnog okna bliži vektoru značajki pozitivnog okna u odnosu na negativno po kosinusnoj sličnosti. Može se primijetiti kako slikovno okno dimenzije 9x9 ne nosi veliku količinu informacije te da postoje negativni primjeri koji su u velikoj mjeri slični referentnima, što učenje korespondencijske metrike otežava. Usprkos tome, pokazuje se da postupak daje dobre rezultate.



Slika 5.3: Primjeri za učenje dubokog modela za korespondencijsku metriku slikovnih okana.

Učenje traje 14 epoha te se kao algoritam učenja koristi ADAM [11] uz stopu učenja $1e^{-3}$ do 11. epohe, zatim $1e^{-4}$. Jedan korak učenja radi se u mini grupama od 128 primjera. Ne koristi se rano zaustavljanje zato što funkcija gubitka modela definirana formulom 4.1 nije izravno povezana s rekonstrukcijskom točnosti, što znači da malena vrijednost gubitka na validacijskom skupu ne donosi nužno bolji rezultat. Ova tvrdnja potvrđena je eksperimentom u kojem se trenira model koji koristi normalizaciju po grupi i ostvaruje manju vrijednost gubitka no konačna rekonstrukcijska točnost nije veća.

5.4. Učenje integriranog rekonstrukcijskog model koji se uči s kraja na kraj

Duboki model za rekonstrukciju kao ulaz prima par slika te na izlazu daje konačnu mapu dispariteta. Svaki sloj modela je derivabilan što omogućuje optimizacijski postupak s kraja na kraj. Model se uči na način da se minimizira gubitak definiran u 4.3. Zbog velikog kapaciteta modela, predtreniranje modela radi se na skupu SceneFlow, zatim 200 epoha na skupu Kitti. Predtreniranjem se omogućuje da model ima sposobnost bolje generalizacije, no kasniji eksperimenti pokazali su da treniranje isključivo na skupu Kitti također ostvaruje dobre rezultate. Algoritam optimizacije je ADAM uz stopu učenja $1e^{-3}$ na SceneFlow skupu, na skupu Kitti $1e^{-3}$ do 50-te epohe, $5e^{-4}$ do 80-te epohe i $1e^{-4}$ do 200-te epohe. Korak učenja vrši se nad nasumičnim isječkom jedne slike veličine 256x512. Kao regularizacijska mjera korišteno je rano zaustavljanje te normalizacija grupe. Rano zaustavljanje napravljeno je na način da se iz podatkovnog skupa izdvojio podskup nad kojim model nije učio. Konačni parametri modela su oni za koje je gubitak na skupu za rano zaustavljanje najmanji. Pokazalo se da se najbolji rezultati postižu kada isječci na validacijskom skupu nisu nasumični, već isti za svaku epohu. Fiksiranjem isječaka na slikama za validaciju preciznije se može mjeriti generalizacijska pogreška postupka te je time informacija o gubitcima na validacijskim skupovima korisnija za praćenje tijeka učenja.

6. Eksperimentalno vrednovanje rekonstrukcijske točnosti

Kvalitetu stereo postupka potrebno je ocijeniti na slikama za koje postoje točni disparitеti. Podatkovni skupovi koji pružaju tu mogućnost opisani su u prethodnom poglavlju. Mjera kvalitete postupka je točnost rekonstrukcije svakog piksela s poznatim disparitetom. Svi eksperimentalni rezultati mjereni su na podatkovnom skupu KITTI 2015, za koji se postavlja da je piksel točno rekonstruiran ako je razlika točnog i rekonstruiranog disparitеta manja od 3. Od 200 slika iz podatkovnog skupa, 80% je izdvojeno za treniranje modela dok ostatak služi za validaciju.

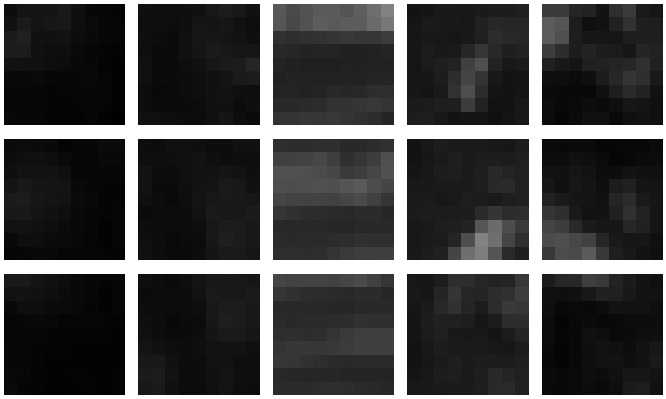
6.1. Rekonstrukcijska točnost modela za ugrađivanje slikovnih okana

Učenje modela traje 10-ak sati. Eksperimenti su napravljeni kako bi se ustanovio utjecaj boje na kvalitetu korespondencijske metrike te odredio utjecaj normalizacije grupa. U tu svrhu, napravljena su 4 eksperimenta: po jedan za sivu sliku te sliku u boji te po jedan s i bez normalizacije po grupama. Dobiveni rezultati prikazani su tablicom 6.1.

Tablica 6.1: Izmjerene rekonstrukcijske točnosti modela za ugrađivanje slikovnih okana. BN označava normalizaciju po grupama. Najbolji rezultat je podebljan.

Model	Točnost - treniranje	Točnost - testiranje
Sive ulazne slike	84.99%	82.32%
Sive ulazne slike - BN	74.51%	71.61%
Ulazne slike u boji	$\mathbf{85.50}\%$	$\mathbf{82.84}\%$
Ulazne slike u boji - BN	74.40%	71.33%

Najbolji rezultat daje model koji ne koristi normalizaciju po grupama te kao ulaz

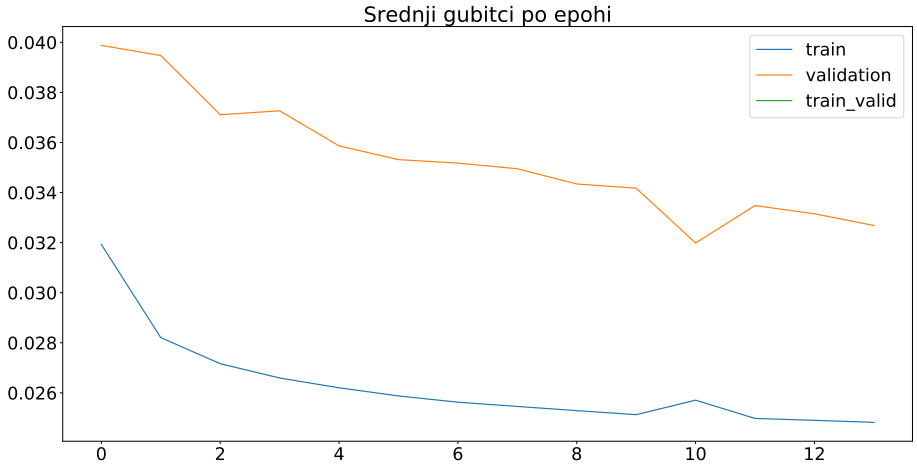


Slika 6.1: Primjeri iz skupa za učenje za koje su negativni primjeri sličniji pozitivnim primjerima po izlazu naučenog modela. Gornji red prikazuje isječke iz desne slike, srednji red prikazuje korespondentne isječke na točnom disparitetu dok donji red prikazuje negativne primjere na pomaknutom disparitetu.

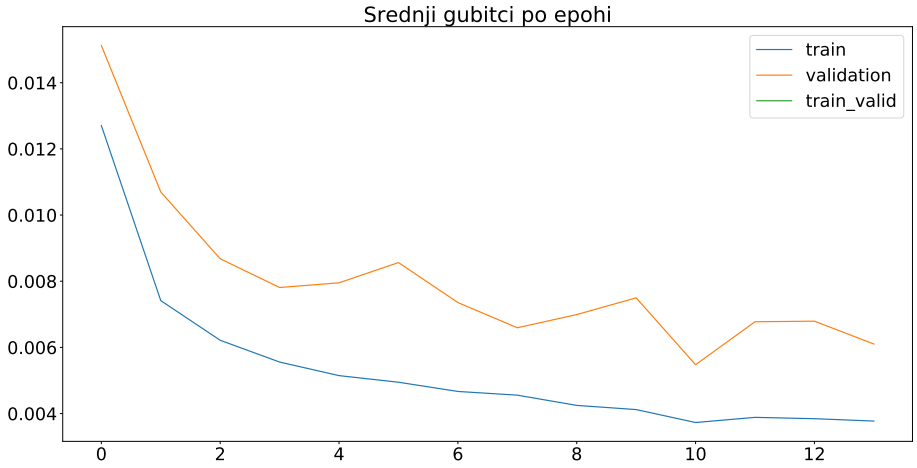
dobiva slike u boji. Zanimljivo je kako normalizacija po grupama znatno smanjuje rekonstrukcijsku točnost modela, iako praksa govori kako normalizacija po grupama može znatno poboljšati rezultate dubokog modela. Razlog padu točnosti leži u činjenici što funkcija gubitka modela korištena prilikom učenja nije izravno povezana s rekonstrukcijskom točnosti. Na slici 6.2 prikazano je kretanje srednjih gubitaka po epohi na skupovima za validaciju i treniranje za model s i bez normalizacije po grupama. Vidljivo je kako model s normalizacijom po grupama brže minimizira pogrešku te je razlika gubitaka na skupovima za treniranje i validaciju puno manja nego kod modela koji ne koristi normalizaciju po grupama. Normalizacija po grupama izlaze slojeva svodi na razdiobu $\mathcal{N}(0, 1)$, što na metrički prostor u koji model ugrađuje slikovna okna utječe na način da su sva ugrađivanja međusobno bliža, te postupak pronalaženja najsličnijih okana za posljedicu ima pad rekonstrukcijske točnosti.

Slika 6.1 prikazuje primjere iz skupa za učenje za koje model nije uspio naučiti ugrađivanje koje će dati željeni rezultat. Većina primjera su dijelovi scene bez teksture, većinom s vrijednosti piksela jednakoj nuli.

Na slici 6.3 prikazane su rekonstrukcije za neke od slika iz podatkovnog skupa KITTI 2015. Zbog činjenice da u model nije ugrađen kriterij glatkoće vidljivi su skokovi u susjednim disparitetima. Također valja primijetiti kako su područja na kojima metoda griješi refleksivne podloge automobila te područja bez kontrasta (preeksponirani dijelovi ceste). Zbog činjenice da je podatkovni skup rijedak te korespondencijska metrika nije učena na područjima slike s disparitetom jednakim nuli, vidljivo je kako je rekonstrukcija loša na dijelovima slike gdje se nalazi nebo.

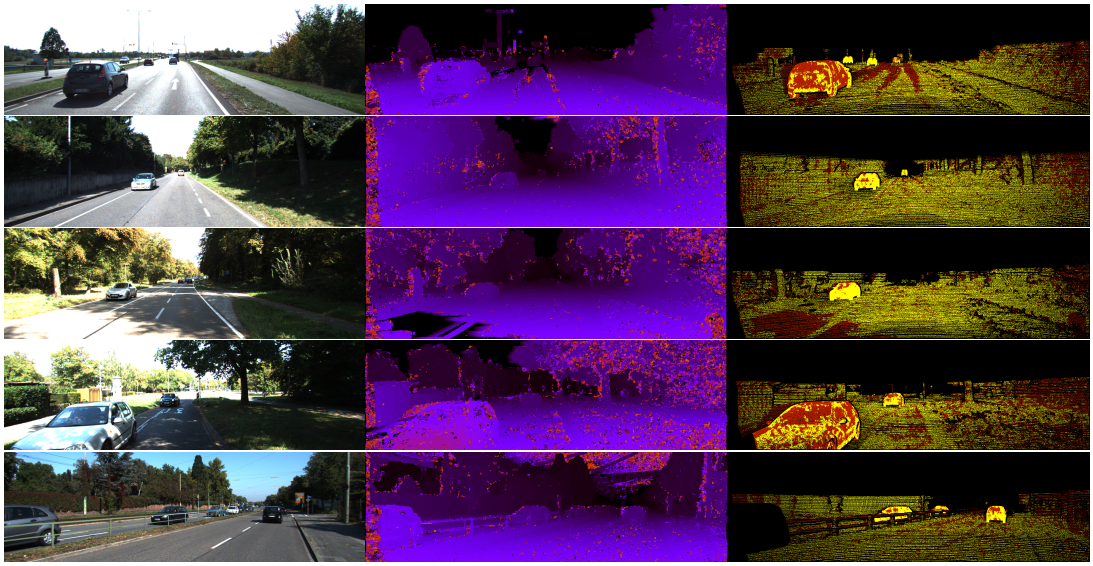


(a) Srednji gubitci modela po epohi učenja modela bez normalizacije po grupama



(b) Srednji gubitci modela po epohi učenja modela s normalizacijom po grupama

Slika 6.2: Srednji gubitci modela po epohi učenja. Plavom bojom označen je gubitak na skupu za testiranje dok narančasta boja označava gubitak na skupu za validaciju. Slika 6.2a prikazuje učenje modela bez normalizacije po grupama, a slika 6.2b prikazuje učenje modela koji koristi normalizaciju po grupama u svim slojevima osim zadnjeg.



Slika 6.3: Rekonstrukcijska točnost modela za ugrađivanje slikovnih okana. Lijeva slika prikazuje snimku iz referentne kamere. Slika u sredini prikazuje rekonstrukciju modela. Desna slika prikazuje rekonstrukcijsku točnost: crvenom bojom označeni su pogrešno a žutom bojom točno rekonstruirani pikseli dok za crne piksele ne postoje laserski dobiveni dispariteti.

6.2. Točnost integriranog rekonstrukcijskog modela

Integrirani rekonstrukcijski model koji se uči s kraja na kraj kao ulaz prilikom učenja dobiva stereo par. Za izračun gubitka potrebna je i informacija o točnoj mapi dispariteta. Zbog memorijskih zahtjeva, model se uči na isječcima slika veličine 256×512 piksela te se položaj isječka određuje nasumično. Izračuni gubitaka na skupu za validaciju i skupu koji se koristi za rano zaustavljanje imaju fiksiran položaj isječka slike kako bi se kvalitetnije mogla ocijeniti dinamika učenja.

Uz osnovni model opisan u odjsečku 4.2, implementiran je i model koji prilikom izgradnje volumena cijene ne vrši konkatenaciju unarnih značajki lijeve i desne slike, već po elementima radi razliku tenzora značajki lijeve i desne slike. Izlazni tenzor sadrži dvostruko manje elemenata, čime se postiže memorijska ušteda s obzirom na to da je tenzor koji izlazi iz volumena cijene najveći u čitavom modelu. Značajnu uštedu u memoriji može se postići modelom koji prilikom izgradnje volumena cijene vrši normalizaciju vektora uzduž osi značajki te skalarni umnožak normaliziranih vektora.

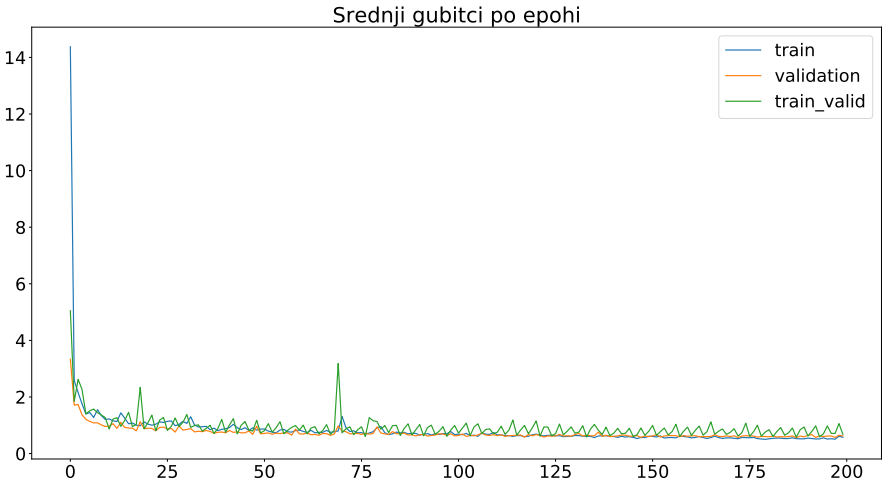
Podatkovni skup SceneFlow Driving korišten je na način da se model trenirao na slikama iz podatkovnog skupa u svrhu inicijalizacije modela. Treniranje na skupu SceneFlow trajalo je 10 epoha. Ovako naučeni parametri korišteni su kao inicijalizacija za treniranje na skupu KITTI 2015. Model je treniran i bez prethodne inicijalizacije

kako bi se utvrdio utjecaj predtreniranja na konačnu točnost postupka.

Točnosti postupaka na podatkovnom skupu KITTI 2015 prikazane su tablicom 6.2. Prikazani su rezultati oba modela i to s i bez prethodnog treniranja na podatkovnom skupu SceneFlow Driving. Vidljivo je kako model koji konkatenira unarne značajke prilikom izgradnje volumena cijene postiže najbolje rezultate te da predtreniranje pridonosi porastu preformansi. Doduše, predtreniranje ne donosi napredak kod modela koji prilikom izgradnje volumena cijene koristi razliku unarnih značajki. Ovaj model u oba eksperimenta pokazuje sličnije performanse na skupovima na treniranje i testiranje uz uštedu memorije. Model koji prilikom izgradnje volumena cijene koristi skalarni umnožak te kao takav postiže značajnu uštedu memorije ostvaruje lošije rezultate od prethodna dva modela, no pokazuje se kako regresija dispariteta, koja nosi većinu memorijskih zahtjeva modela, nema velik utjecaj na performanse.

Tablica 6.2: Izmjerene točnosti integriranog rekonstrukcijskog modela. Najbolji rezultat je podebljan.

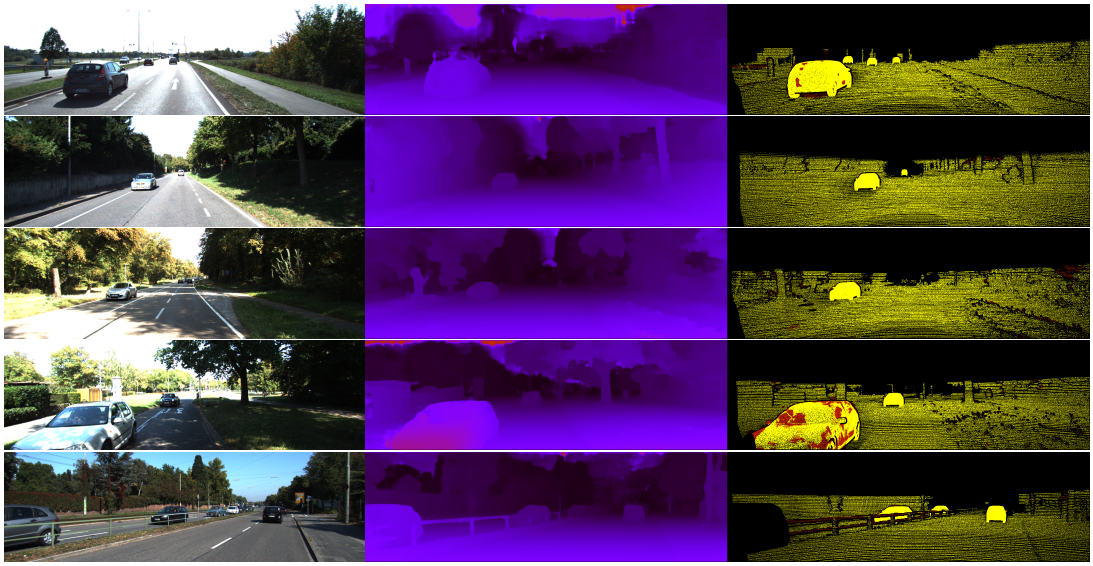
Model	Točnost - treniranje	Točnost - testiranje
Osnovni model	**97.98%**	**97.23%**
Razlika unarnih značajki	97.62%	97.17%
Skalarni umnožak unarnih značajki	97.95%	97.07%
Osnovni model - predtreniranje	**98.06%**	**97.31%**
Razlika unarnih značajki - predtreniranje	97.94	97.18



Slika 6.4: Srednja vrijednost gubitka po epohi učenja za 3 izdvojena skupa: plavom bojom označen skup za učenje, narančastom validacijski skup a zelenom skup korišten za rano zaustavljanje.

Slika 6.4 prikazuje dinamiku učenja dubokog modela na podatkovnom skupu KITTI bez prethodnog treniranja. Sve tri krivulje prikazuju srednji gubitak po epohi: plava na skupu za treniranje, zelena na skupu korištenom za rano zaustavljanje i narančasta za gubitak na skupu na validaciji. Vidljivo je kako optimizacijski postupak ADAM uspijeva u svega nekoliko epoha drastično minimizirati funkciju gubitka. Sve tri krivulje gubitaka kreću se zajedno te su bliske prilikom konvergencije, što govori kako model ima dobru sposobnost generalizacije.

Slika 6.5 prikazuje rekonstrukcije integriranog modela. Model je uspješno naučio glatku rekonstrukciju visoke točnosti. Može se primijetiti kako rekonstrukcije na područjima slike gdje se nalazi nebo nisu precizne. Razlog tome je činjenica što model nije učio na tim slikovnim elementima. Također je vidljivo kako postupak griješi na refleksivnim područjima poput automobila. Pogreške su vidljive i na objektima male širine, poput ograda ili stupova.



Slika 6.5: Točnost integriranog rekonstrukcijskog modela. Lijeva slika prikazuje snimku iz referentne kamere. Slika u sredini prikazuje rekonstrukciju modela. Desna slika prikazuje rekonstrukcijsku točnost: crvenom bojom označeni su pogrešno a žutom bojom točno rekonstruirani pikseli dok za crne piksele ne postoje laserski dobiveni dispariteti.

7. Programska izvedba i vanjske biblioteke

7.1. Biblioteka Tensorflow

Tensorflow je biblioteka otvorenog koda [1] koja omogućava izražavanje numeričkih postupaka uz pomoć računskih grafova a posebno je pogodna za izradu modela strojnog učenja. Najviše je korišten pri izradi dubokih modela gdje podrška za grafičke procesore omogućava brže učenje modela. Tensorflow nudi podršku za automatsku diferencijaciju, što korisnicima omogućuje stvaranje novih modela bez potrebe za pisanjem izračuna gradijenata. Slojevi u Tensorflowu pisani su u jeziku C++ te je podržano izvođenje na arhitekturi CUDA. Tensorflow pruža sučelje prema programskom jeziku Python u kojem se model može definirati kao izračunski graf te se zbog prirode jezika Python tipično postiže jednostavnija implementacija.

7.2. Ostale korištene biblioteke

Za pripremu podataka korištena je Python biblioteka NumPy koja pruža podršku za efikasno rukovanje višedimenzionalnim poljima. Njegova je implementacija u programskom jeziku C, te koristi biblioteku BLAS koja omogućuje paralelizaciju matričnih operacija na CPU. Za vizualizaciju podataka korištene su biblioteke Matplotlib i SciKit Learn, a za pohranu metrika učenja relacijska baza podataka MySQL.

8. Zaključak

Stereoskopska rekonstrukcija jedan je od dugovječnijih problema u računalnom vidu. Postupak rekonstrukcije 3D scene iz para kamera koje nude 2D sliku izazovan je zadatak. Rješenje tog problema nudi značajnu informaciju za razumijevanje scene, što je temeljni zadatak računalnog vida. Računalni vid doživio je velik napredak pojavom dubokog učenja i danas duboki modeli postižu najbolje rezultate u rješavanju svih problema pa tako i stereoskopske rekonstrukcije.

U okviru ovog rada opisan je problem stereoskopske rekonstrukcije, geometrija prostora i kalibracija stereo para. Opisan je i algoritam stereoskopske rekonstrukcije te su definirani njegovi ključni dijelovi s naglaskom na korespondencijsku metriku slikovnih okana koja se koristi u postupku.

Obje metode korištene za ostvarivanje stereo-rekonstrukcije zasnivaju se na dubokom učenju. Zadaća prve metode je ugrađivanje slikovnih okana u visokodimenzionalni metrički prostor. Ugrađivanje je naučeno na velikom podatkovnom skupu te je postignuta metrika robusna i efikasna za korištenje u postupcima stereoskopske rekonstrukcije, određivanja vlastitog gibanja kamere ili računanju optičkog toka. Druga metoda duboki je model za stereoskopsku rekonstrukciju učen s kraja na kraj. Za ulazni stereo par, model na izlazu daje glatku mapu dispariteta visoke rekonstrukcijske točnosti.

Opisani su korišteni podatkovni skupovi s preciznom informacijom o dubini te je vrednovana rekonstrukcijska točnost oba postupka. Također je dan naglasak na potrebi dubokih modela za velikim podatkovnim skupovima za učenje s ciljem dobre generalizacije postupka. Opisani su postupci učenja i validiranja te su navedeni izazovi prilikom učenja dubokih modela.

Dobiveni rezultati potvrđuju uspjeh dubokog učenja u računalnom vidu. Vidljiv je potencijal za mnoge primjene, gdje je jedna od njih primjena u autonomnoj vožnji, koje je trenutno industrijski zanimljivo područje, a senzorika dobivena računalnim vidom neizbježan je čimbenik.

LITERATURA

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, i Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016. URL `http://arxiv.org/abs/1603.04467`.

[2] J. C. A. Barata i M. S. Hussein. The Moore-Penrose Pseudoinverse: A Tutorial Review of the Theory. *Brazilian Journal of Physics*, 42:146–165, Travanj 2012. doi: 10.1007/s13538-011-0052-z.

[3] Aleksandar Botev, Guy Lever, i David Barber. Nesterov's accelerated gradient and momentum as approximations to regularised update descent. abs/1607.01981, 2016. URL `https://arxiv.org/abs/1607.01981`.

[4] John Duchi, Elad Hazan, i Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. 2011. URL `http://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf`.

[5] Ian Goodfellow, Yoshua Bengio, i Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[6] R. I. Hartley i A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, i Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[8] The MathWorks Inc. im2col. URL https://www.mathworks.com/help/images/ref/im2col.html.

[9] Sergey Ioffe i Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

[10] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, i Adam Bry. End-to-end learning of geometry and context for deep stereo regression. *CoRR*, abs/1703.04309, 2017. URL http://arxiv.org/abs/1703.04309.

[11] Diederik P. Kingma i Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

[12] Dino Kovač. Gusta stereoskopska rekonstrukcija scene predstavljene ravninskim odsječcima. Magistarski rad, Sveučilište u Zagrebu, Fakultet Elektrotehnike i Računarstva, 2015. URL http://www.zemris.fer.hr/~ssegvic/project/pubs/kovac15ms.pdf.

[13] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, 1981. URL https://cseweb.ucsd.edu/classes/fa01/cse291/hclh/SceneReconstruction.pdf.

[14] Moritz Menze i Andreas Geiger. Object scene flow for autonomous vehicles. U *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[15] David Nister. An efficient solution to the five-point relative pose problem. *IEEE-CVPR*, 2003, 2003. URL https://pdfs.semanticscholar.org/c288/7c83751d2c36c63139e68d46516ba3038909.pdf.

[16] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, i T.Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. U *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16. arXiv:1512.02134.

[17] Daniel Scharstein i Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. URL http://vision.middlebury.edu/stereo/taxonomy-IJCV.pdf.

[18] Jure Zbontar i Yann LeCun. Computing the stereo matching cost with a convolutional neural network. *CoRR*, abs/1409.4326, 2014. URL http://arxiv.org/abs/1409.4326.

[19] Jan Šnajder i Bojana Dalbelo Bašić. *Strojno učenje*. 2012.

Učenje korespondencijske metrike za gustu stereoskopsku rekonstrukciju

Sažetak

Stereoskopska rekonstrukcija je važno područje primjene računalnog vida. Vrlo važan korak pri rješavanju tog problema jest ostvarivanje korespondencije piksela lijeve i desne slike. Ta korespondencija može se ostvariti analizom udaljenosti deskriptora dobivenih ugrađivanjem slikovnih okana u visokodimenzionalni metrički prostor. Ugrađivanje se tipično provodi dubokim modelom naučenim na stereoskopskim slikama s poznatom dubinskom informacijom. U okviru rada, proučeni su postupci stereoskopske rekonstrukcije temeljene na naučenim metrikama. Preuzeti su javno dostupni parametri ugrađivanja i vrednovana je postignuta točnost rekonstrukcije na prikladnoj kolekciji stereoskopskih slika. Naučena je vlastita korespondencijska metrika te su provedene usporedbe s rezultatima dobivenima javnom parametrizacijom. Opisani su postupci učenja i validiranja hiperparametara. Prikazani su i ocijenjeni ostvareni rezultati. Predloženi su pravci budućeg razvoja.

Ključne riječi: Duboko učenje, stereoskopska rekonstrukcija, računalni vid.

Learning the correspondence metrics for dense stereoscopic reconstruction

Abstract

Stereoscopic reconstruction is an important field of computer vision application. One of the main steps to solving this problem is computing the correspondence of left and right image pixels. Such correspondence may be achieved by distance analysis of image patch embeddings to a high dimensional metric space. Such embedding is typically achieved using deep learning trained on stereoscopic image pairs with ground truth depth information. In course of this thesis, learned metric stereo reconstruction algorithms have been studied. Publicly available embedding parameters have been evaluated on suitable stereoscopic datasets. Own correspondence metric was trained and compared against existing results. Training and hyperparameter validation steps are described. Achieved results are presented and further work is proposed.

Keywords: Deep learning, stereoscopic reconstruction, computer vision.