

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Odhad změny vlastností obrazu v závislosti na světelných podmínkách

Šimon Pavlín

Vedoucí: Ing. Otakar Jašek
Květen 2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Pavлін** Jméno: **Šimon** Osobní číslo: **470321**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra kybernetiky**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Odhad změny vlastností obrazu v závislosti na světelných podmínkách

Název bakalářské práce anglicky:

Data-Driven Model Estimating Changes of Image Properties According to Illumination

Pokyny pro vypracování:

Pro mnoho search & rescue misí je potřeba uvažovat případ, kdy se rapidně změní osvětlení oproti předpokládanému stavu, např. v důsledku výpadku elektřiny. V takových situacích je potřeba, aby robot byl na takovou situaci připraven. Proto je důležité vědět, jak se změní obrazová data z kamer robota v případě, že se robot dostane do prostředí bez globálního osvětlení a je nucen si svítit pouze bodovým světlem umístěným na konstrukci robota.

1. Nastudujte aktuální odbornou literaturu, [1,2,3] možno použít jako výchozí články.
2. Vytvořte dataset zachycující RGBD data několika stejných scén za různého osvětlení (bodové vs globální).
3. Naučte neuronovou síť zachycující transformaci RGBD dat z globálního osvětlení do bodového.
4. Případně naučte neuronovou síť schopnou transformace RGBD obrazu v závislosti na pozici bodového osvětlení.
5. Přegenerujte veřejně dostupný dataset (např. [4]) pomocí vytvořené neuronové sítě tak, aby realisticky působil jako osvětlen pomocí bodového osvětlení.

Seznam doporučené literatury:

- [1] Barron, Jonathan T., and Jitendra Malik. "Intrinsic scene properties from a single rgb-d image." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013.
- [2] Hold-Geoffroy, Yannick, et al. "Deep outdoor illumination estimation." IEEE International Conference on Computer Vision and Pattern Recognition. Vol. 2. 2017.
- [3] Wang, Tuanfeng, Tobias Ritschel, and Niloy Mitra. "Joint material and illumination estimation from photo sets in the wild." 2018 International Conference on 3D Vision (3DV). IEEE, 2018.
- [4] Ammirato, Phil, et al. "A dataset for developing and benchmarking active vision." Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Otakar Jašek, vidění pro roboty a autonomní systémy FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **06.02.2019**

Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **20.09.2020**

Ing. Otakar Jašek
podpis vedoucí(ho) práce

doc. Ing. Tomáš Svoboda, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Otakarovi Jaškovi za ochotu, cenné rady a pevné nervy při vedení této bakalářské práce. Zároveň bych chtěl poděkovat i rodině a přátelům za veškerou podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 22. května 2019

.....

Abstrakt

Při search & rescue misích se robot často pohybuje v rapidně odlišném osvětlení, než na které je primárně naučen. Příkladem může být navigace robota v dole, kdy je robot nucen si přisvicovat vlastním zdrojem světla. V těchto případech je užitečně vědět, jak se podoba objektů ve scéně změní oproti podobě objektů ve standardním globálním osvětlení.

Tato práce se zabývá návrhem algoritmu, který odhaduje tuto změnu osvětlení ve scéně. Cílem práce se předpokládá algoritmus, který ze znalosti obrazu v globálním osvětlení a odpovídajících hloubkových datech vygeneruje obraz scény, osvětlené jediným bodovým světlem umístěným na robotu.

Klíčová slova: odhad osvětlení, osvětlovací model, bodový zdroj světla

Vedoucí: Ing. Otakar Jašek

Abstract

In search & rescue missions, a robot is located in places with extremely different lighting conditions than those it's primarily trained for. An example of this is navigation in a mine, where the robot must use its own light source. In these cases, it's useful to know how image properties in a scene change compared to a standard, globally-illuminated environment.

This thesis focuses on the development of an algorithm which predicts this change of illumination in a scene. Given an image of a globally illuminated scene and corresponding depth data, the algorithm generates an image as it would appear if the scene were illuminated by a point source of light mounted on the robot.

Keywords: prediction of illumination, reflection model, light source

Title translation: Data-Driven Model Estimating Changes of Image Properties According to Illumination

Obsah

1 Úvod	1	5.2 Neuronová síť	26
1.1 Seznámení s problematikou	1	5.2.1 Vytvořená trénovací data	26
1.2 Návrh řešení	1	5.2.2 Naučená neuronová síť	27
2 Teorie	3	6 Závěr	33
2.1 Osvětlovací model	3	Literatura	35
2.1.1 Intenzita osvětlení	3		
2.1.2 Obousměrná odrazová distribuční funkce	3		
2.1.3 Phongův osvětlovací model	4		
3 Související práce	7		
3.1 Odhad osvětlení	7		
3.2 Odhad světelných vlastností povrchů	7		
3.3 Zlepšení hloubkové mapy	8		
3.4 Shrnutí	8		
4 Implementace	9		
4.1 Zvolené technologie	9		
4.1.1 PyTorch	9		
4.1.2 Intel RealSense	9		
4.1.3 OpenCV	9		
4.2 Struktura souborů	10		
4.3 Trénovací data	10		
4.3.1 Vytvoření datasetu	10		
4.4 Úprava vstupních dat	11		
4.4.1 Předzpracování hloubkové mapy	12		
4.4.2 Bilaterální filter	13		
4.4.3 Mrak bodů z hloubkové mapy	13		
4.4.4 Zarovnání hloubkových dat a RGB obrázku	14		
4.4.5 Konečná úprava dat	15		
4.5 Normálové vektory	15		
4.6 Neuronová síť	16		
4.6.1 Vstupní a výstupní data	16		
4.6.2 Návrh	17		
4.6.3 Ztrátová funkce	19		
4.6.4 Optimalizační algoritmus	20		
4.7 Implementace osvětlovacího modelu	21		
4.7.1 Vstupní parametry	21		
4.7.2 Způsob implementace	21		
4.7.3 Úpravy oproti tradičnímu modelu	22		
5 Experimenty	25		
5.1 Osvětlovací model	25		

Obrázky

2.1 Význam jednotlivých vektorů použitých ve vztazích pro výpočet jednotlivých složek odražené intenzity	5
4.1 Porovnání hloubkové mapy před a po procesu dopočítávání	13
4.2 Porovnání normál při filtrování a nefiltrování hloubkových dat	14
4.3 Použitá struktura konvoluční neuronové sítě pro získání světelných vlastností povrchů ve scéně	17
4.4 Schéma konvolučních vrstev uvnitř neuronové sítě pro získání světelných vlastností bodů ve scéně	18
4.5 Schéma procesu učení neuronové konvoluční sítě	19
5.1 Obrázky demonstrující funkci osvětlovacího modelu	26
5.2 Scény s velmi podobným výchozím (první řádek) a vzorovým obrázkem (druhý řádek)	27
5.3 Průběh hodnoty ztrátové funkce během učení. Světle růžová značí naměřenou hodnotu ztrátové funkce a tmavě růžová je vyhlazený průběh	28
5.4 Porovnání výchozích obrázků (levý sloupec), vygenerovaných obrázků (prostřední sloupec) a vzorových obrázků (pravý sloupec)	30
5.5 Příklad vstupních a výstupních dat při získávání světelných vlastností a aplikování osvětlovacího modelu	31

Kapitola 1

Úvod

1.1 Seznámení s problematikou

V praxi je často třeba robota navigovat v odlišných světelných podmínkách, než ve kterých je primárně provozován. Tato situace nastává především při search & rescue misích, kdy se robot pohybuje v prostředí bez světelného zdroje. Příkladem mohou být zavalené nebo zřícené prostory bez funkční elektrické sítě, kdy je robot nucen si přisvicovat vlastním světlem. I v takovýchto případech by robot měl být schopen rozeznat základní objekty. Aby byl robot na tuto situaci připraven, je potřeba dopředu odhadnout, jak se vzhled okolního prostředí změní.

Hlavní motivací pro zaobírání se touto problematikou byla úloha "DARPA Subterranean Challenge: Teams of Robots Compete to Explore Underground Worlds", kterou se tým z ČVUT zabýval. V této úloze bylo nutné navigovat robota v důlních prostorech a mimo jiné jej naučit rozeznávat různé objekty (např. hasicí přístroje). V dnešní době se pro rozeznávání objektu v obraze hojně používá konvoluční neuronová síť. Pro naučení těchto sítí je nutné mít velký dataset fotografií daných objektů. Přes internet není obtížné nalézt tyto datasety v globálním (venkovním) osvětlení. Jelikož si ale robot v dolech musí sám přisvicovat, může se díky tomu podoba objektů rapidně změnit. Nasnímání velkého množství fotografií v pozměněných podmínkách nemusí být vždy možné či vytvoření datasetu by bylo značně zdlouhavé.

Již existují postupy, které změni osvětlení ve scéně. Tyto postupy jsou ale zpravidla založeny na znalosti celé scény. Celá scéna je následně vymodelována a je vypočten obraz v novém osvětlení například pomocí ray-tracking metody. Toto je však značně časově velmi náročné a v praxi málokdy známe celou scénu.

1.2 Návrh řešení

Cílem této práce je odhadnout změnu obrazu při předem známé změně osvětlení. Jako vstup našeho algoritmu je RGB obraz v globálním osvětlení (bez jiného znatelného zdroje světla) a hloubková mapa ze 3D senzoru. Výstupem se předpokládá RGB obraz okolní scény, která je osvětlena jediným bodovým

světlem umístěném na rámu robota.

Prvním krokem navrhovaného řešení je výpočet normálových vektorů k povrchu z hloubkové kamery. Následně se použije neuronová síť, která z obrázku v globálním osvětlení a normálových vektorů odhadne světelné vlastnosti jednotlivých bodů ve scéně. Odhadnuté vlastnosti se použijí jako vstupní parametry osvětlovacího modelu. Jelikož předem víme, že světlo bude na rámu robota, známe pozici jediného zdroje světla ve scéně. Proto můžeme pomocí osvětlovacího modelu vygenerovat výsledný RGB obraz.

Kapitola 2

Teorie

2.1 Osvětlovací model

Jedním ze základních prvků této práce je osvětlovací model. Osvětlovací model má za cíl přiřadit k jednotlivým bodům ve scéně intenzitu světla pozorovanou pozorovatelem.[1]

2.1.1 Intenzita osvětlení

Uvažujme případ, kdy ve scéně je pouze jeden bodový zdroj světla. Pak je intenzita osvětlení dopadající v bodě x určena vztahem

$$I_0 = \frac{L}{d^2}, \quad (2.1)$$

kde L značí svítivost bodového zdroje světla a d je vzdálenost bodu x od zdroje světla. Je patrné, že v reálném prostředí není ve scéně pouze jedno světlo a ani žádné světlo není dokonalé bodové světlo. Pro naše účely však tato aproximace je vyhovující. Intenzita světla, kterou pozorovatel vnímá, se od této hodnoty zpravidla liší. Závisí na vlastnostech povrchu, na který světlo dopadá.

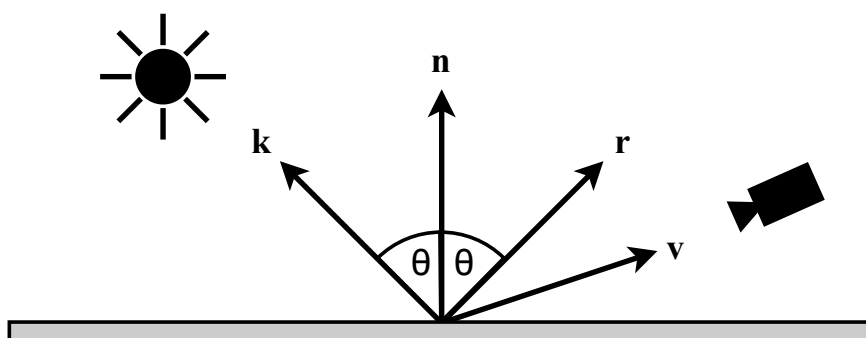
2.1.2 Obousměrná odrazová distribuční funkce

Obousměrná odrazová distribuční funkce (ang. BRDF - Bidirectional Reflectance Distribution Function) je funkce udávající pozorovanou (odraženou) intenzitu světla v závislosti na pozici pozorovatele a zdroje světla. Podoba takovéto funkce může být různorodá. Pro většinu povrchů je možné tuto funkci aproximovat na základě následující představy.

Při dopadu světelného paprsku na povrch předmětu se paprsek rozptýlí do všech směrů. Díky tomu, že reálné povrchy nejsou dokonale hladké, dochází na nich i k několikanásobným odrazům světelných paprsků od mikročástic.

Způsob odrazu světelného paprsku můžeme rozdělit na dvě možnosti. Paprsek se od povrchu odrazí buď zrcadlově či dojde k několikanásobnému odrazu. Proto můžeme obecně napsat, že intenzita odraženého světla I v určitém bodě je dána vztahem

$$I = I_d + I_s, \quad (2.2)$$



Obrázek 2.1: Význam jednotlivých vektorů použitých ve vztazích pro výpočet jednotlivých složek odražené intenzity

■ Ambientní složka

Nejjednodušší složkou intenzity je ambientní složka, která představuje globální osvětlení scény. Narozdíl od ostatních složek odražené intenzity hodnota ambientní složky nezáleží na vzdálenosti od zdroje světla, jelikož kompenzuje nepřímá osvětlení objektů. Pro ambientní složku platí vztah, že

$$I_a = C_a C_d k_a, \quad (2.4)$$

kde C_a je barva ambientního světla, C_d je barva difuzního povrchu v bodě a $k_a \in \langle 0; 1 \rangle$ je koeficient ambientního odrazu.

■ Difuzní složka

Jak již bylo avizováno výše, difuzní složka závisí pouze na pozici zdroje světla a normále k povrchu. Difuzní materiál představuje materiál, který rozptýlí dopadající paprsek světla do všech směrů rovnoměrně. Z této vlastnosti je patrné, že difuzní odraz světla nezávisí na pozici pozorovatele. Platí, že

$$I_d = I_0 \cdot \left[C_l C_d k_d \max(0, \mathbf{n}^T \mathbf{k}) \right], \quad (2.5)$$

kde I_0 je intenzita dopadajícího záření, C_l je barva světla, C_d je barva difuzního povrchu, $k_d \in \langle 0; 1 \rangle$ je koeficient difuzního odrazu, \mathbf{n} je normálový vektor v daném bodě k povrchu a \mathbf{k} je směrový vektor od bodu na povrchu ke světlu. Poznamenejme, že v celé práci uvažujeme sloupcové vektory. Pro usnadnění budeme dále v textu pod pojmem *barva povrchu* rozumět barvu difuzního povrchu.

Přímé světlo by nemělo osvětlovat povrchy, které jsou odkloněné od zdroje světla. Pro povrchy takto odvrácené od bodového zdroje světla je skalární součin $\mathbf{n}^T \mathbf{k} < 0$. Díky použití funkce $\max(0, \mathbf{n}^T \mathbf{k})$ je difuzní složka těchto povrchů nulová.

■ Zrcadlová složka

Zrcadlová složka představuje složky intenzity odraženého světla, která závisí na pozici pozorovatele. Vliv této složky je možné vidět na hladkých, lesklých materiálech, kde se světlo odráží především dle zákona odrazu. Ve Phongově osvětlovacím modelu je tato složka určena vztahem

$$I_s = I_0 \cdot \left[C_l C_s k_s \max(0, \mathbf{v}^T \mathbf{r})^h \right], \quad (2.6)$$

kde I_0 je opět intenzita dopadajícího záření, C_l je barva světla, C_s je barva zrcadlového povrchu, $k_s \in \langle 0; 1 \rangle$ je koeficient zrcadlového odrazu světla, \mathbf{v} je směrový vektor mezi bodem na povrchu a pozorovatelem, \mathbf{r} je směrový vektor paprsku dokonale odraženého od povrchu podle zákona odrazu a $h \in \langle 1; \infty \rangle$ je ostrost zrcadlového odrazu (míra lesklosti). Funkce \max má u zrcadlové složky podobný význam jako funkce \max u difuzní složky.

Kapitola 3

Související práce

V této kapitole bychom chtěli představit různé dosavadní výzkumy a projekty, na které jsme více či méně navazovali. Společným prvkem, kterému se budeme věnovat, je osvětlení scén.

3.1 Odhad osvětlení

Pro mnoho aplikací je třeba znát, jaké je ve scéně osvětlení. Příkladem takové aplikace může být snaha přidat do scény objekt tak, aby vypadal realisticky. Výstupem algoritmu, které osvětlení předpovídají, se rozumí funkce, která udává intenzitu přicházejícího světla v závislosti na směru.

Návrh algoritmu se zpravidla liší pro venkovní a vnitřní scény. U venkovních scén se funkce intenzity povětšinou určuje z osvětlení od oblohy.[4] Existují však algoritmy, které výslednou funkci venkovního osvětlení určují pomocí více znalostí. Kromě vzhladu oblohy mohou tyto algoritmy být také založeny na vržených stínech na zemi či na stínování vertikálních stěn.[5]

Odlišnost v postupu návrhu algoritmů odhadujících vnitřní osvětlení oproti venkovnímu osvětlení spočívá především v tom, že ve venkovním prostředí je v zásadě jediný dominantní zdroj světla - Slunce. Tato skutečnost může být použita při návrhu algoritmů pro venkovní osvětlení a značně zlepšit výsledky. Tento předpoklad bychom také mohli použít ve vnitřních scénách, avšak existují scény, ve kterých by předpoklad jediného zdroje světla přinesl velké nepřesnosti. Proto jsou navrženy algoritmy zaměřující se na odhad vnitřního osvětlení.[6, 7]

V mnoha aplikacích je též třeba odhadnout osvětlení v reálném čase. Proto jsou vyvíjeny algoritmy, které toto umožňují.[8, 9] Aplikace jsou především zábavného charakteru. Příkladem takové aplikace může být přidání objektů do reálných scén ve virtuální realitě či virtuální přidání modních doplňků na obličeje.

3.2 Odhad světelných vlastností povrchů

Každý materiál má zpravidla specifické světelné vlastnosti. Existuje mnoho prací, které se snaží tyto vlastnosti odhadnout. Velká část z nich je zaměřena

na odhad světelných vlastností jediného objektu. [10][11] Pak je totiž možné často předpokládat, že celý objekt je tvořen jediným materiálem, a proto má i každý bod na objektu stejné světelné vlastnosti. Vstupem do těchto algoritmů buď je i vyznačení té části obrázku kde je daný objekt či je použit další algoritmus určující pixely, na kterých se objekt vyskytuje. [9] Kromě jediného vstupního obrázku může být vstupem i více obrazů v odlišných podmínkách. Každý ze vstupních obrazů může být vyfocen z jiného úhlu [12][13] nebo může být pozorován pod jiným osvětlením. [8]

Samozřejmě už existují i algoritmy odhadující světelné vlastnosti rozmanitých scén. [14][15] Zásadní rozdíl oproti výše diskutovanému případu je ve výstupu. Výstupem algoritmu pro rozmanité scény nemůže být pouze informace o jediném materiálu, ale každému pixelu na vstupním obrázku mohou být přiřazeny odlišné světelné vlastnosti.

Ne vždy je možné algoritmy dělit dle toho, jestli odhadují osvětlení ve scéně či jestli odhadují světelné vlastnosti povrchů. Některé algoritmy zahrnují obě tyto disciplíny. [7][16]

3.3 Zlepšení hloubkové mapy

Hloubkové senzory zpravidla vrací data značně zašumněná. V praxi je proto často potřeba zaznamenaná hloubková data vylepšit. Za tímto účelem bylo provedeno mnoho studií zabývajících se touto problematikou. [17][18][19]

Zajímavou práci v tomto oboru ukázali Jonathan T. Barron a Jitendra Malik. [20] Jejich algoritmus přijímá jeden RGB-D obraz určité scény. Výstupem je zahlušená hloubková mapa a normálové vektory k povrchu. Jako vedlejší produkt tohoto algoritmu je i obrázek s odrazivostmi a stínováním. K získání těchto parametrů používají množinu různých tvarů a osvětlení, ze kterých se následně snaží sestavit původní obrázek. Postup vede na řešení minimalizační úlohy.

3.4 Shrnutí

Již řada lidí odhadovala světelné vlastnosti povrchů ve scénách a následně předpovídali vzhled scény v jiném osvětlení. V našem případě musíme vytvořit algoritmus, který získává světelné vlastnosti rozmanité scény, a proto nemůžeme užít postupy pro odhad světelných vlastností na objektu z jediného materiálu.

Zároveň nepotřebujeme v navrhovaném algoritmu odhadovat osvětlení ve scéně jelikož předpokládáme, že osvětlení ve scéně je fixní - globální směrové osvětlení shora. Známé jsou i pozice světla ve výstupním obraze, díky čemuž výsledný obraz můžeme při znalosti vlastností povrchu vygenerovat.

Kapitola 4

Implementace

4.1 Zvolené technologie

4.1.1 PyTorch

Pro tento projekt jsme zvolili programovací jazyk Python. Jelikož bylo zřejmé, že bude nutné užít *Deep learning*, museli jsme vybrat vhodnou knihovnu. V Pythonu přicházejí do popředí dvě nejvíce používané knihovny pro práci s neuronovými sítěmi - PyTorch a TensorFlow. Zvolili jsme PyTorch, protože narozdíl od TensorFlow jsem s ním již předtím pracoval.

Open-source knihovna PyTorch je zaměřena na strojové učení. V této knihovně není těžké navrhnout jednoduchou neuronovou síť a následně ji naučit. Výhodou užití podobné knihovny je značné usnadnění při zpětné propagaci gradientů v trénovacím cyklu. Pro zpětnou propagaci gradientů je potřeba v neuronových sítích určit derivace jednotlivých operací. Při užití knihovny toto určení je prováděno samo automaticky.

4.1.2 Intel RealSense

Pro záznam RGB-D obrázků jsme použili senzor Intel RealSense Depth Camera D435. Data z tohoto senzoru jsou získávána pomocí knihovny PyRealSense. Knihovna poskytuje i obslužné funkce, jako je například transformace hloubkových dat na množinu prostorových bodů.

Hloubkový senzor, který Intel RealSense obsahuje, rozhodně není nejpřesnější hloubkový senzor na trhu. Existují senzory jak s lepší hloubkovou mapou, tak s lepším obrazem z kamery. My však chceme navrhnout řešení se senzorem, který je jak snadno, tak levně dostupný.

4.1.3 OpenCV

Při práci se zpracováním obrazových dat je vhodná jejich vizualizace. Pro tento účel se osvědčila knihovna OpenCV. Pro ještě přívětivější užívání jsme vytvořili třídu `Visualizer`. Po inicializaci objektu této třídy funkčnost nabízí především funkce `showData`. Při opakovaném volání této funkce s jinými

obrazovými daty se nově vytvářená okna řadí dlaždicově za sebe. Toto předchází manuálnímu vypočtu pozice jednotlivých oken při zobrazování nových obrazů.

4.2 Struktura souborů

Celý algoritmus je rozdělen na 3 dílčí programy. Pro každý podprogram je vytvořen zvláštní spustitelný soubor. Název a popis těchto podprogramů je v následujícím seznamu:

- *main_capture.py* – Zajišťuje záznam trénovacích dat.
- *main_convert.py* – Převádí zaznamenaná data na data, která jsou vhodná pro neuronovou síť. Jelikož toto předzpracování (výpočet normál, zarovnání dat z kamery a hloubkového senzoru apod.) je časově náročné, je pro to vyhrazen zvláštní spustitelný program. Tento podprogram je možné spustit ve dvou různých módech. Uživatel si buď sám přepíná mezi jednotlivými snímky a sleduje jejich předzpracování nebo se provede automatické předzpracování a uložení všech snímků.
- *main_net.py* – Provádí jak trénování neuronové sítě, tak její následné otestování. Proto i tento program je možné spouštět ve dvou základních módech - natrénování sítě na předzpracovaných trénovacích datech a otestování naučené sítě na zaznamenaných datech.

4.3 Trénovací data

Pro naučení jakékoli neuronové sítě, jsou vyžadovaná trénovací data. V této práci je potřeba dvojice snímků z kamery a hloubkového senzoru. Prvním obrázkem ve dvojici je vstupní (výchozí) obrázek osvětlen globálně. Druhý obrázek je osvětlen bodovým světlem z rámu robota a představuje výstup algoritmu v ideálním případě - vzorový obrázek.

4.3.1 Vytvoření datasetu

Jelikož potřebujeme data ve dvou takto specifických světlených podmínkách, bylo nutné si vytvořit vlastní dataset. V obou odlišných osvětleních zaznamenáváme tyto snímky:

- *Barevná fotografie* z kamery RealSense senzoru
- *Hloubková mapa* získaná z RealSense senzoru

Nyní zde budou popsány jednotlivé situace při zanaménávání snímků pro jednotlivé světelné podmínky.

■ Globální osvětlení

Při tomto druhu osvětlení budeme simulovat situaci, kdy osvětlení ve scéně připomíná venkovní osvětlení. S velkým zjednodušením můžeme obecně říct, že osvětlení bude globální bez znatelného bodového osvětlení a ve scéně bude nanejvýš směrové osvětlení shora. Toto očividně není zcela přesné, ale jako aproximace osvětlení na většině snímkách to postačí.

Uvědomme si, že při vytváření datasetu potřebujeme dvojice snímků, které odpovídají jedné scéně ve dvou rozdílných osvětleních. Vytvářeli bychom snímky ve venkovním prostředí, bylo by nutné první snímek nasnímat za světla a bez pohybu kamerou nasnímat stejný záběr v noci s rozsvíceným pomocným světlem. Je zřejmé, že toto není praktické řešení.

Pro snadnější vytváření datasetu jsme vyzorovali, že velmi podobné světelné podmínky jsou i v místnosti s vysokým stropem osvětlené zářivkami. Abychom předešli nechtěnému osvětlení z venku, snímání jsme prováděli již za tmy. Zvolíme trénovací scénu a se zapnutými zářivkami můžeme vytvořit snímek pro globální osvětlení.

V algoritmu se tento obrázek bude používat jako vstupní obraz do algoritmu, ze kterého budeme vycházet. V této práci budeme pod pojmem *výchozí obraz* (angl. initial image) rozumět právě tento obraz z kamery, který je vyfocený v globálním osvětlení.

■ Osvětlení bodovým světlem připevněném na konstrukci robota

Protože jsme si vhodně zvolili způsob snímání dat v globálním osvětlení, není obtížné nasnímat snímky ani pro toto osvětlení. V tomto případě nejprve potřebujeme, aby ve scéně nebyl žádný zdroj světla. Při splnění této podmínky budeme následně přisvícovat pomocným světlem. Světlo pro demonstraci umístíme těsně nad senzor, pomocí kterého snímky zaznamenáváme.

První druh osvětlení zaznamenáváme v místnosti již v době, kdy venkovní osvětlení je nepatrné. Proto není nic snadnějšího, než vypnout zdroj světla v místnosti a rozsvítit pomocná světla u senzoru. Tuto konfiguraci můžeme považovat za totožnou s případem, kdy se robot pohybuje v prostoru bez zdroje světla a musí si přisvícovat sám.

Abychom dále v práci nemuseli zdlouhavě popisovat, že se jedná o obraz v tomto osvětlení, zavedeme si pojem *vzorový obraz* (ang. ground truth image). Tímto pojmem je míněn obraz z kamery, který je nasnímán ve světlených podmínkách popsanych v této kapitole. Přídavné jméno "vzorový" je užito z důvodu, že později při učení konvoluční neuronové sítě bude tento obraz považován za vzor, jemuž by se měl náš výstupní obraz co nejvíce podobat.

■ 4.4 Úprava vstupních dat

Jelikož získáváme data ze dvou různých senzorů je nutné jejich obrazy převést na obrazy ze stejného úhlu pohledu. Zároveň použitý hloubkový senzor není nejpřesnější, a proto bylo třeba se vypořádat i s tímto nedostatkem.

4.4.1 Předzpracování hloubkové mapy

Rozlišení relevantních hodnot

Hloubkový senzor promítá vzory infračervených bodů na předměty před kamerou. Tyto vzory jsou následně pozorovány ze dvou kamer a pomocí vypočítaných pozic jsou určeny hloubky na jednotlivých pixelech. Ne vždy se ale tímto postupem určí hloubka správně. Proto je potřeba pro vytvoření konzistentní hloubkové mapy neznámé body dopočítat.

Hloubkový senzor vrací hodnoty v milimetrech v intervalu $\langle 0; 65535 \rangle$. Senzor však dokáže dostatečně spolehlivě měřit vzdálenosti zhruba od 26cm do 5m. Proto jsme všechny hodnoty mimo tento interval považovali za neplatné a pracovali s nimi, jako bychom je neznali.

Odhad neznámých hloubkových dat

Jak již bylo řečeno, je třeba přibližně odhadnout i hloubky na místech, kde údaj o hloubce neznáme. Proces dopočítání neznámých mezihodnot se skládá z následujících kroků:

1. Hloubkovou mapu označme jako `depth_map`

2. Vytvoření masky platných a neplatných dat

K vytvoření těchto masek jsem užil funkci `cv2.threshold`. Prvním parametrem je hloubková mapa `depth_map` a druhým parametrem je číselná hodnota. Funkce vrací 0 u pixelů, jejichž hodnota je nižší než druhý číselný parametr, jinak hodnotu 1. Zavoláním dvakrát této funkce a následným vynásobením těchto masek je možné, získat masku hodnot, které jsou větší než 26 centimetrů a menší než 5 metrů. Nazvěme tuto masku `mask` a `mask_inv` nazvěme inverzní masku, kde jsou 0 a 1 prohozeny.

3. Nastavení počáteční hodnoty u pixelů s neplatnými daty

U pixelů, které jsou označeny jako neznámé, je jejich hodnota přenastavena na náhodné číslo v rozmezí $\langle 0; 5 \rangle$.

4. Vytvoření kopie hloubkové mapy `depth_map`

5. Rozostření pomocné kopie hloubkové mapy

I zde jsem použil funkci knihovny OpenCV. Tentokrát funkci `cv2.filter2D`, která na vstupní mapě pomocí zadaného konvolučního jádra provede konvoluci. Pro rozostření jsem zvolil jako jádro matici $\mathbf{M} \in \mathbb{R}^{10 \times 10}$, kde $M_{i,j} = \frac{1}{100}$. Tímto získáme mapu `depth_map_blured`.

6. Aktualizování hloubkové mapy

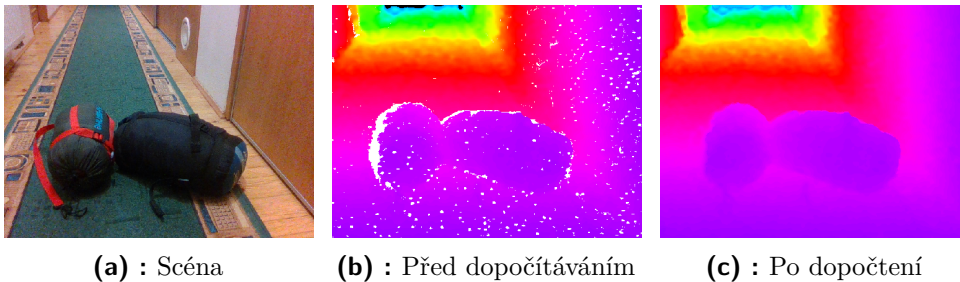
Hloubková mapa `depth_map` je aktualizována dle následujícího předpisu:

$$\text{depth_map} = \text{depth_map} * \text{mask} + \text{depth_map_blured} * \text{mask_inv},$$

kde symbol $*$ představuje složkové násobení. Takto aktualizovaná hloubková mapa nepozměňuje hodnoty u známých pixelů, ale pouze v pixelech s předem neznámou hodnotou se jí snaží interpolovat.

7. Tyto kroky jsou znovu od kroku 4 iterativně volány dokud neproběhne 300 iterací.

Na obrázku 4.1 je možné vidět porovnání hloubkové mapy získané ze senzoru a hloubkové mapy s dopočítanými hodnotami pomocí výše uvedeného postupu. Bíle jsou značena místa, kde neznáme hloubky.



Obrázek 4.1: Porovnání hloubkové mapy před a po procesu dopočítávání

4.4.2 Bilaterální filter

Hloubková data ze senzoru jsou značně zašuměná. Na obrázku 4.1c sice toto zašumění není příliš patrné, ale v případě, že si určíme normálové vektory pomocí hloubkových dat, je toto zašumění výrazné (viz obrázek 4.2a).

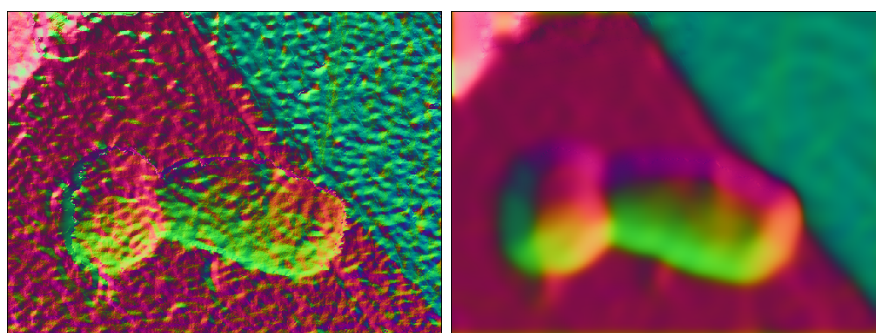
Aby byl tento nechtěný šum potlačen, bylo by možné užít klasický filtr na rozostření obrazu. Tímto postupem bychom však přišli o ostré hrany, který jsou pro naši práci potřebné. Ukázalo se, že použitelným řešením tohoto problému je *bilaterální filtr*. Tento filtr značně potlačí šum avšak zanechává ostré hrany. Pro samotné užití bilaterálního filtru jsme použili funkci `bilateralFilter` z knihovny OpenCV.

```
out_depth_map = cv2.bilateralFilter(in_depth_map, 50, 0.5, 10)
```

Experimentálně bylo zjištěno, že funkce volána s výše uvedenými parametry vrací nejspokojivější výsledky. Na obrázku 4.2 jsou porovnány normálové mapy vypočtené z filtrovaných a nefiltrovaných hloubkových dat. Můžeme vidět, že se však nepovedlo zcela potlačit šum a stále je na rovných plochách znatelné zvlnění. Tento nedostatek se však nepovedlo ani přes značnou snahu odstranit.

4.4.3 Mrak bodů z hloubkové mapy

Samotný formát hloubkové mapy není pro naše účely vyhovující. Jelikož chceme osvětlovat scénu, je vhodnější místo hloubkových dat pracovat s body



(a) : Normály určené z nefiltrovaných hloubkových dat

(b) : Normály určené z hloubkových dat po filtraci

Obrázek 4.2: Porovnání normál při filtrování a nefiltrování hloubkových dat

v prostoru. Tento formát je i vhodnější pro správné určení normálových vektorů.

Z RealSense senzoru je možné pomocí knihovny *Pyrealsense* získat zobrazovací parametry kamery i hloubkového senzoru. Tato knihovna zároveň obsahuje i funkce pro transformace. Použitím funkce `rs2_deproject_pixel_to_point` je možné ze znalosti hloubky, souřadnic pixelu a parametrů senzoru získat bod v prostoru vzhledem k danému senzoru (v našem případě hloubkového senzoru). Aplikujeme-li tuto funkci na všechny pixely v hloubkové mapě, získáme 3D body odpovídající jednotlivým pixelům na hloubkové mapě.

4.4.4 Zarovnání hloubkových dat a RGB obrázku

Pozice hloubkového senzoru a kamery není stejná. Proto je nutné namapování dat z hloubkového senzoru na data z kamery. Jelikož je pro nás vhodné pracovat především s body v prostoru potřebujeme přiřadit každému pixelu v RGB obrázku odpovídající bod v prostoru.

Nejprve je nutné převést získaný bod v prostoru ze souřadného systému hloubkového senzoru do souřadného systému kamery. Za tímto účelem byla užitá funkce `rs2_transform_point_to_point`, která přijímá jako parametr informaci o transformaci mezi dvěma souřadnými systémy a transformovaném 3D bodu.

Nyní určíme na jaký pixel v RGB obrázku se daný 3D bod promítne. Knihovna *Pyrealsense* obsahuje funkci `rs2_project_point_to_pixel`, která tuto akci provede za nás. Vstupem je promítaný bod v prostoru a zobrazovací informace senzoru - kamery. Tímto postupem můžeme namapovat celou mapu bodů v prostoru na RGB obrázek. Poznamenejme, že původní hloubková mapa má rozměry 1280×720 a rozměry RGB obrázku jsou 640×480 . Proto při zarovnávání bylo změněno i rozlišení hloubkové mapy na rozlišení výchozího obrázku. Nyní pro každý pixel v barevném obrázku známe jeho barvu a odpovídající bod v prostoru.

4.4.5 Konečná úprava dat

Při transformaci může dojít k tomu, že v zarovnané mapě 3D bodů neznáme souřadnice pro každý pixel. Toto může nastat například v případě, že je stěna některého předmětu kolmo k hloubkovému senzoru. Tato stěna však může být viditelná kamerou, a tak při transformaci dat ze souřadného systému hloubkového senzoru na souřadný systém kamery nedokážeme určit hloubková data pro takovou stěnu, ačkoliv ji na RGB obrázku vidíme. Proto může nastat, že RGB pixel nemá korespondenci v hloubkové mapě.

Z tohoto důvodu je opět ze zarovnaných bodů v prostoru vytvořena hloubková mapa, u které jsou neznámá data dopočtena znovu pomocí postupu uvedeného výše. Z dopočítané hloubkové mapy jsou znovu určeny 3D body se kterými budeme pracovat.

4.5 Normálové vektory

Nechť \mathbf{P} je mapa 3D bodů. Zvolili jsme výpočet normálových vektorů pomocí čtyř okolních bodů. Normálový vektor $\mathbf{n}_{x,y}$ v bodě na souřadnicích x, y lze pak určit vztahem

$$\mathbf{n}_{x,y} = \mathbf{u}_{x,y} \times \mathbf{v}_{x,y},$$

kde

$$\mathbf{u}_{x,y} = \frac{\mathbf{P}_{x+1,y} - \mathbf{P}_{x-1,y}}{2}, \quad \mathbf{v}_{x,y} = \frac{\mathbf{P}_{x,y+1} - \mathbf{P}_{x,y-1}}{2}$$

Je očividné, že i přes dělení koeficientem 2 nemusí být délka takového normálového vektoru nutně 1. Pomocí vztahu

$$\hat{\mathbf{n}}_{x,y} = \frac{\mathbf{n}_{x,y}}{\|\mathbf{n}_{x,y}\|}$$

převeďme normálové vektory $\mathbf{n}_{x,y}$ na jednotkové normálové vektory $\hat{\mathbf{n}}_{x,y}$.

Aby výpočet byl rychlejší bylo potřeba používat funkce, které pracují s celou mapou, nikoli jen volat funkce na jednotlivé body. Za tímto účelem jsme opět použili funkci `cv2.filter2D`. Použitím následujících konvolučních jader bylo možné, získat například mapu vektorů $\mathbf{u}_{x,y}$ a $\mathbf{v}_{x,y}$.

$$\mathbf{C}_u = \begin{bmatrix} 0 & 0 & 0 \\ -0.5 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{C}_v = \begin{bmatrix} 0 & -0.5 & 0 \\ 0 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}$$

Nyní je třeba provést vektorový součin. Jelikož kvůli rychlosti chceme provádět operace pouze na celých mapách, běžné funkce pro výpočet vektorového součinu nepřicházejí v úvahu. Proto jsme výpočet vektorového součinu implementovali sami. Při výpočtu se pracuje zvlášť s jednotlivými souřadnicemi

3D bodů. Použijeme-li výše uvedenou konvoluční funkci `filter2D` na mapy jednotlivých souřadnic, získáme následujících šest matic.

$$U_x, U_y, U_z, V_x, V_y, V_z,$$

kde U_x představuje matici x -ových souřadnic vektorů $\mathbf{u}_{x,y}$, matice V_z je matici y -ových souřadnic vektorů $\mathbf{v}_{x,y}$ a podobně si lze intuitivně domyslet význam zbylých matic.

S tímto značením pak můžeme vektorový součin určit vztahem

$$N_x = U_y \circ V_z - U_z \circ V_y,$$

$$N_y = U_z \circ V_x - U_x \circ V_z,$$

$$N_z = U_x \circ V_y - U_y \circ V_x,$$

kde symbol \circ značí násobení dvou matic prvek po prvku. Nyní již stačí jen převést normálové vektory na jednotkové normálové vektory.

4.6 Neuronová síť

V této části se budeme věnovat návrhu a implementaci konvoluční neuronové sítě pro získání jednotlivých složek světelných vlastností bodů ve scéně.

4.6.1 Vstupní a výstupní data

Učelem navrhované konvoluční neuronové sítě je určit světelné vlastnosti jednotlivých pixelů v obrázku. Jako vstup do této neuronové sítě uvažujme barevný obrázek scény v globálním osvětlení a mapa normálových vektorů. Z hloubkové mapy je již mapa normálových vektorů předpočítána, a proto vstupem do neuronové sítě již samotná hloubková mapa není.

Jak již bylo zmíněno v předešlém odstavci, výstupem by měly být informace o světelných vlastnostech jednotlivých pixelů. Budeme-li následně generovat obraz pomocí Phongova osvětlovacího modelu, je vhodné, aby výstupní mapy z neuronové sítě obsahovaly pro každý pixel následující informace:

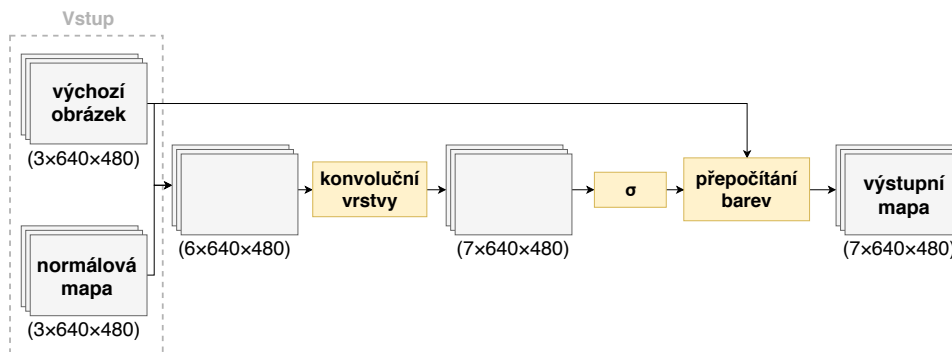
- Barva povrchu
- Koeficient ambientního odrazu
- Koeficient difuzního odrazu
- Koeficient zrcadlového odrazu
- Koeficient míry lesklosti

Je možné si povšimnout, že v teoretické části o Phongově osvětlovacím modelu (kapitola 2.1.3) je při výpočtu zrcadlové složky přítomen i parametr C_s , který značí barvu zrcadlového povrchu. V naší práci však budeme uvažovat, že barva zrcadlového povrchu u všech materiálů je bílá. Tento předpoklad

jsme učinili kvůli tomu, abychom měli méně stupňů volnosti, a proto i méně parametrů, které je třeba zjistit. Zároveň se domníváme, že tento předpoklad nepřináší pro každodenní scény velkou chybu.

4.6.2 Návrh

V této podkapitole bude představeno jak a podle čeho byla konvoluční neuronová síť navržena. Schématický návrh této neuronové sítě je možné vidět na obrázku 4.3.



Obrázek 4.3: Použitá struktura konvoluční neuronové sítě pro získání světelných vlastností povrchů ve scéně

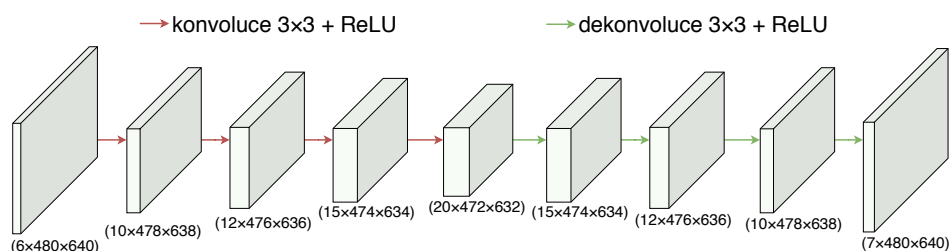
Rozměry

Již jsme si představili, jaké budou vstupní a výstupní argumenty navrhované neuronové sítě. Nyní se zaměříme na to, jaké rozměry mají jednotlivé mapy.

Na vstupu je barevný obrázek a mapa normálových vektorů. Každý z těchto vstupních argumentů má 3 kanály - R, G, B složky u barevného obrázku, x, y, z souřadnice u mapy normálových vektorů. Celkem je tedy na vstupu 6-kanálová mapa. Senzor RealSense vrací obraz z kamery v rozlišení 640×480 , a proto jsme toto rozlišení zvolili jako "hlavní" rozlišení, ve kterém budeme se všemi mapami pracovat. Hloubková mapa získaná z hloubkového senzoru má sice rozlišení 1280×720 , avšak pro lepší práci s daty, ji při zarovnávání s obrazem z kamery (kapitola 4.4.4) převádíme do rozlišení 640×480 . Vstupní mapa do neuronové sítě má tedy rozměry $6 \times 640 \times 480$.

Na výstupu očekáváme mapu představující světelné vlastnosti jednotlivých bodů v obraze. Jelikož barva povrchu obsahuje tři složky (R, G, B) a dále uvažujeme čtyři charakteristické koeficienty (ambientní, difuzní, zrcadlový, míra lesklosti) je pro každý pixel určováno 7 hodnot. Výstupní mapa má stejnou výšku a šířku jako barevný obraz, mapa normálových vektorů i mapa prostorových bodů. Toto je zvoleno především kvůli snadné práci s mapami uvnitř osvětlovacího modelu. Rozměry mapy, která je na výstupu navrhované neuronové sítě, jsou $7 \times 640 \times 480$.

Konvoluční vrstvy



Obrázek 4.4: Schéma konvolučních vrstev uvnitř neuronové sítě pro získání světelných vlastností bodů ve scéně

Návrh naší konvoluční neuronové sítě je inspirován podobou *U-Net* sítě.[21] Schéma samotných vrstev navržené neuronové sítě je na obrázku 4.4. Na obrázku je možné vidět 4 konvoluční a 4 dekonvoluční vrstvy. Počet kanálů je z 6 zvýšen na 20 kanálů a následně snížen na 7, které potřebujeme na výstupu. Bylo by možné výsledky neuronové sítě značně zlepšit tím, kdybychom více zvýšili počet kanálů uvnitř neuronové sítě.) Jestli bychom zvýšili počet kanálů, zvýšil by se i počet parametrů, které by se neuronová síť musela naučit. Z toho důvodu jsme počet kanálů omezili, aby učení neuronové sítě netrvalo příliš dlouho.

Sigmoida

Na obrázku 4.3 si je možné všimnout funkce sigmoidy (σ). Funkce sigmoidy převádí jakékoliv reálné číslo na číslo v intervalu $\langle 0; 1 \rangle$. Předpis této funkce je dán vzorcem

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (4.1)$$

Tuto funkci jsme užili kvůli tomu, že z Phongova osvětlovacího modelu víme, že hodnota téměř všech parametrů je v intervalu $\langle 0; 1 \rangle$. Záporné hodnoty a hodnoty větší než 1 nejsou vyhovující a přinášely by nevhodné výsledky. Jediným koeficientem, který se nepohybuje v tomto intervalu je koeficient h představující míru lesklosti. Obecně je uvažováno, že $h \in \langle 1; \infty \rangle$. Hodnotu tohoto koeficientu vystupující z neuronové sítě však necháme stále na pracovním intervalu $\langle 0; 1 \rangle$ stejně jako ostatní parametry. Převod na potřebný interval bude proveden až uvnitř samotné funkce generující obraz dle Phongova osvětlovacího modelu.

Barva povrchu

Na schématu navržené neuronové sítě (obr. 4.3) také vidíme, že na konci sítě je aplikována závěrečná funkce s názvem **přepočítání barev**. Jak je znázorněno na obrázku, je do této funkce přiveden i nezměněný výchozí obrázek. Tato funkce upravuje jen výstupní barvu povrchu a ostatní světelné parametry vrací nezměněné.

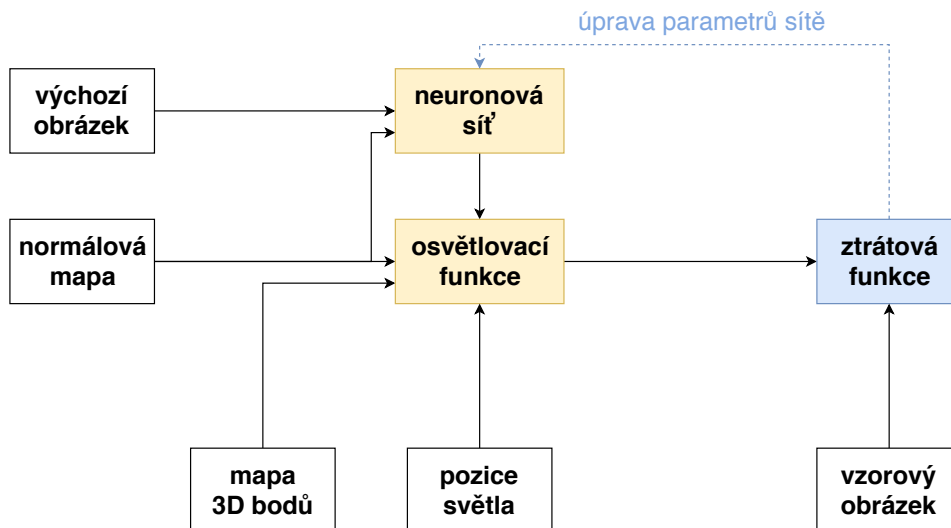
Tato funkce je přidána, jelikož očekáváme, že barva povrchu se nebude výrazně lišit od barev na výchozím obrázku. Necht x představuje vstupní hodnoty této funkce příslušící k barvám povrchu. Pak je výsledná barva povrchu vystupující z funkce **přepočítání barev** určena vzorcem

$$C_d = C + C_x, \quad (4.2)$$

kde C_d značí barvu (difuzního) povrchu, C je barva bodu ve výchozím obrázku a $C_x = 2x - 1$. Výpočet C_x je zvolen tak, aby hodnoty z intervalu $\langle 0; 1 \rangle$ byly převedeny na interval $\langle -1; 1 \rangle$.

Tato úprava značně zkrátí dobu učení, jelikož již od začátku učení se barva povrchu pohybuje blízko hodnot, které jsou získány z obrazu v globálním osvětlení (výchozího obrazu).

4.6.3 Ztrátová funkce



Obrázek 4.5: Schéma procesu učení neuronové konvoluční sítě

Aby se neuronová konvoluční síť byla schopná učit, je třeba definovat ztrátovou funkci. Návrhovou hodnotou ztrátové funkce je jediné skalární číslo, které určuje, jak dobrý je výsledek ze sítě. Optimalizační postupy pak upravují parametry neuronové sítě tak, aby hodnota ztrátové funkce byla co nejnižší. Na obrázku 4.5 je vidět celý cyklus učení implementované neuronové sítě. Neuronová síť přijme vstupní data a vygeneruje mapu světelných vlastností bodů v obraze. Tato mapa je spolu s dalšími informacemi o scéně a pozici světla poslána do osvětlovací funkce, která vygeneruje příslušný obrázek. Vygenerovaný obrázek je porovnán se vzorovým obrázkem pomocí ztrátové funkce. Podle hodnoty ztrátové funkce jsou vhodně poupraveny parametry neuronové sítě a celý cyklus se opakuje.

V našem případě budeme ztrátovou funkci určovat mezi dvěma obrázky - vzorového obrázku a obrázkem vygenerovaného osvětlovacím modelem. Ztráto-

vou funkci \mathcal{L} jsme definovali jako lineární kombinaci dvou ztrátových podfunkcí. Platí, že

$$\mathcal{L} = 3 \cdot \mathcal{L}_1 + \mathcal{L}_2, \quad (4.3)$$

kde \mathcal{L}_1 a \mathcal{L}_2 jsou jednotlivé ztrátové funkce. Koefficienty jsme volili podle toho, že více apelujeme na podobnost obrázků v jednotlivých bodech než na celkový dojem z obrázku. Samotný koeficient 3 jsme zjistili experimentálně na základě několika testů. Nyní si postupně popíšeme obě ztrátové podfunkce.

Ztrátovou funkci \mathcal{L}_1 jsme definovali jako funkci `L1Loss` z knihovny *PyTorch*. Jedná se o funkci, která je definována následujícím vzorcem

$$\mathcal{L}_1 = L1Loss = \sum_{i=0}^n |x_i - y_i|, \quad (4.4)$$

kde x_i jsou postupně všechny hodnoty pixelů ve vygenerovaném obrázku osvětlovacím modelem a y_i představuje jednotlivé hodnoty pixelů ve vzorovém obraze. Bylo by možné použít i ztrátové funkce vyšších řádů. Například bychom ztrátovou funkci mohli definovat následovně

$$L4Loss = \sum_{i=0}^n (x_i - y_i)^4. \quad (4.5)$$

Kdybychom však použili ztrátové funkce vyšších řádů, tak by jediný pixel, který se bude výrazně lišit od ideálního případu, celý obrázek značně penalizoval. To by mohlo zapříčinit to, že by se neuronová síť snažila zmenšit chybu na daném pixelu i na úkor toho, že bude zvětšena chyba všech ostatních. Toto však nechceme. Snažíme se dosáhnout toho, aby většina obrázku odpovídala realitě a spokojíme se s tím, že pár pixelů nebude zcela odpovídat.

Ztrátová funkce \mathcal{L}_2 by se měla postarat o to, aby celý vygenerovaný obraz vypadal podobně jako vzorový obraz. Pro tuto funkci platí, že

$$\mathcal{L}_2 = |\bar{R}_{GT} - \bar{R}_{gen}| + |\bar{G}_{GT} - \bar{G}_{gen}| + |\bar{B}_{GT} - \bar{B}_{gen}|, \quad (4.6)$$

kde \bar{R}_{GT} , \bar{G}_{GT} a \bar{B}_{GT} značí průměrnou hodnotu vzorového obrázku na R,G,B kanálech a \bar{R}_{gen} , \bar{G}_{gen} a \bar{B}_{gen} představuje průměrnou hodnotu jednotlivých R,G,B kanálů vygenerovaného obrázku.

■ 4.6.4 Optimalizační algoritmus

Jako metodu pro zlepšování parametrů neuronové sítě dle hodnoty ztrátové funkce jsme použili optimalizační algoritmus Adam [22]. Algoritmus jsme aplikovali pomocí funkce `optim.Adam` z knihovny *PyTorch*. Potom, co jsou vypočteny jednotlivé gradienty pro zpětnou propagaci, jsou parametry neuronové sítě upraveny právě pomocí této optimalizační metody. Při optimalizaci je rychlost jejich změny v jednom kroku optimalizace určena vstupním argumentem `learning_rate`. Volbu `learning_rate = 0.0001` jsme učinili experimentálně na základě testování několika různých hodnot.

4.7 Implementace osvětlovacího modelu

Nyní se zaměříme na implementovaný osvětlovací model. Od osvětlovacího modelu předpokládáme, že ze znalostí informací o scéně a pozici světla vygeneruje odpovídající RGB obraz. Za tímto účelem jsme implementovali funkci s názvem `getComposedBatch`. Neuronová síť nepracuje pouze se snímkem jediné scény, ale zpravidla najednou zpracovává několik těchto snímků různých scén. Proto, jak je také z názvu funkce patrné, naimplementovaná funkce pracuje s celou sadou snímků několika scén.

4.7.1 Vstupní parametry

Funkce `getComposedBatch` přijímá čtyři vstupní parametry:

- `decomposed_batch` – Tento parametr představuje mapu světelných vlastností bodů ve scéně. Jedná se o 7 kanálů. Popis jednotlivých kanálů je v předešlé kapitole 4.6.1, která se věnuje návrhu neuronové sítě. Je nutné připomenout, že i vstupní parametr představující míru lesklosti je prozatím na intervalu $\langle 0; 1 \rangle$.
- `normal_batch` – Parametr reprezentující mapu normálových vektorů k povrchu ve scénách.
- `point_batch` – Reprezentuje mapu bodů v prostoru popisující geometrii povrchu
- `light_point` – Udává pozici světla ve scéně. Jedná se o 3D bod. Tento parametr bychom ani nemuseli potřebovat, jelikož požadovanou pozici světla známe. Při testování správnosti implementace této funkce se však ukázalo, že interaktivní změna pozice světla je užitečná.

4.7.2 Způsob implementace

Při celé implementaci této se funkce se pracuje s objekty knihovny *PyTorch*. Toto je nutné z toho důvodu, aby automatická zpětná propagace gradientů fungovala.

Pro Phongův osvětlovací model je kromě normálových vektorů potřeba ještě určit směrové vektory mezi body na povrchu a pozici světla a pozici kamery. Za tímto účelem jsem implementoval dvě funkce `getLightVectorBatch` a `getCameraVectorBatch`. Tyto funkce vracejí mapy daných normalizovaných vektorů.

Při výpočtu budeme nutné provést skalární součiny vektorů. Kvůli tomuto pracujeme s každou souřadnicí zvlášť a výpočet skalárního součinu provádíme přímo dle vzorce

$$U \cdot V = U_x \circ V_x + U_y \circ V_y + U_z \circ V_z,$$

kde U a V jsou mapy vektorů a $U_x, U_y, U_z, V_x, V_y, V_z$ jsou mapy (matice) jednotlivých souřadnic použitých pracovních vektorů.

Pro zaimplementování funkce $\max(\mathbf{0}, \mathbf{x})$ do modelu jsme použili funkci *ReLU*.

■ 4.7.3 Úpravy oproti tradičnímu modelu

Při implementaci bylo nutné udělat určitá omezení či rozdíly oproti klasickému Phongově osvětlovacímu modelu.

■ Barva povrchu

Jak již bylo v textu zmíněno, každý materiál může mít odlišnou zrcadlovou barvu. V této práci uvažujeme, že barva zrcadlového povrchu je bílá, což se ukazuje jako dobrá aproximace. Jestliže máme určenou barvu zrcadlového povrchu, není důvod ji pomocí neuronové sítě určovat, a proto ji v osvětlovacích parametrech ze sítě nenajdeme.

■ Barva světla

Volba barvy světla ve scéně není posílána do funkce jako parametr. Je však možné v souboru *config.py* tuto barvu zvolit. Je zvolen tento způsob, jelikož se nepředpokládá dynamická změna světla při běhu programu.

■ Intenzita světla

Stejně jako u barvy světla je možné základní svítivost světla měnit pouze v souboru *config.py*. Tato svítivost číselně odpovídá intenzitě světla dopadající na objekt ve vzdálenosti jeden metr od zdroje světla.

■ Koeficient ambientního odrazu

Pro následující úpravu si připomeňme, co vlastně představuje ambientní složka odražené intenzity světla. Ve scéně nejsou povrchy osvětlovány pouze přímo, ale jsou také osvětleny nepřímým světlem odraženým od ostatních objektů.

Ambientní koeficient vstupující do naší osvětlovací funkce nebyl použit jako koeficient ambientního odrazu ve Phongově modelu. Ve výpočtu ambientní složky, byl koeficient ambientního odrazu určen dle vzorce

$$\tilde{k}_a = k_a k_d, \quad (4.7)$$

kde \tilde{k}_a je koeficient ambientního odrazu použitý ve Phongově osvětlovacím modelu, k_a je ambientní koeficient vstupující do naší osvětlovací funkce a k_d je koeficient difuzního odrazu.

Tato úprava je i založena na skutečnosti, že i v případě, že je objekt osvětlen nepřímým osvětlením, musí odražená intenzita tohoto světla být značně ovlivněna difuzní složkou materiálu. Tato změna oproti tradičnímu Phongově modelu byla implementována také kvůli zlepšení učení neuronové

sítě. Jelikož takto definovaný ambientní koeficient je přímo svázán s difuzním, je nemožné, aby nastala situace kdy

$$k_a \rightarrow 1, \quad k_d \rightarrow 0. \quad (4.8)$$

Při trénování konvoluční neuronové sítě tato anomálie často nastávala. Důvodem pravděpodobně byla špatná volba datasetu, kdy neuronová síť se naučila, že pouhé "zkopírování" vstupního obrázku na výstup je nejlepší. Této situaci jsme se snažili vyvarovat i úpravou datasetu (viz kapitola 5.2.1).

■ Zrcadlová složka

Při výpočtu zrcadlové složky jsme provedli dvě úpravy oproti Phongově osvětlovacímu modelu popsaného v kapitole 2.1.3. Ve vzorci pro výpočet zrcadlové složky intenzity odraženého světla (viz vzorec 2.6) se objevuje výraz

$$\max(0, \mathbf{v}^T \mathbf{r})^h, \quad (4.9)$$

na kterém lze demonstrovat obě úpravy.

Koeficient h ve výše zmíněném výrazu značí koeficient lesklosti a obecně platí, že $h \in \langle 1; \infty \rangle$. Abychom hodnoty toho koeficientu udrželi v rozmezí, se kterým lze dobře pracovat, omezili jsme h na interval $\langle 1; 50 \rangle$. Jelikož však do osvětlovací funkce vstupuje mapa koeficientů lesklosti, která má hodnoty v intervalu $\langle 0; 1 \rangle$, jsou hodnoty z intervalu $\langle 0; 1 \rangle$ lineárně převedeny do intervalu $\langle 1; 50 \rangle$ až uvnitř osvětlovací funkce `getDecomposedBatch`.

Bez druhé změny u výpočtu zrcadlové složky by nebylo možné síť rozumně učit. Při učení konvolučních neuronových sítí je nutné aby veškeré použité operace byly diferencovatelné. Výše uvedený výraz 4.9 je možné vnímat jako funkci $f(c, x) = c^x$. Platí, že

$$\frac{\partial f(c, x)}{\partial x} = c^x \ln c. \quad (4.10)$$

Může nastat situace, kdy $c = 0$. Tehdy je $\ln c$ nedefinované. Z tohoto důvodu byl výpočet mírně poupraven a funkce byla volána ve tvaru $f(c + \epsilon, x)$, kde v našem případě $\epsilon = 0.001$. Díky tomuto je i pro $c = 0$ parciální derivace podle x definována. Zároveň je ze vztahu 4.9 patrná nerovnost $c \geq 0$.

Kapitola 5

Experimenty

Tato kapitola se bude věnovat ověření správnosti implementovaného osvětlovacího modelu a předvedení výsledných obrázku z naučené neuronové sítě.

5.1 Osvětlovací model

Výstupní obrázky z osvětlovacího modelu, které v této podkapitole prezentujeme, jsou generovány s ručně zvolenými konstantními světelnými koeficienty pro celou scénu. Jako barvy povrchů jsou zvoleny stejné barvy jako na výchozím obrázku.

Na obrázku 5.1 je uvedeno více podobrázků, na kterých je demonstrována funkčnost implementovaného osvětlovacího modelu. Na většině těchto obrázcích je zelený křížek. Světlo je pak ve scéně umístěné 50 centimetrů nad místem, které křížek vyznačuje.

Obrázek 5.1a je výchozí obrázek, který je zároveň použit jako barva povrchu v jednotlivých bodech.

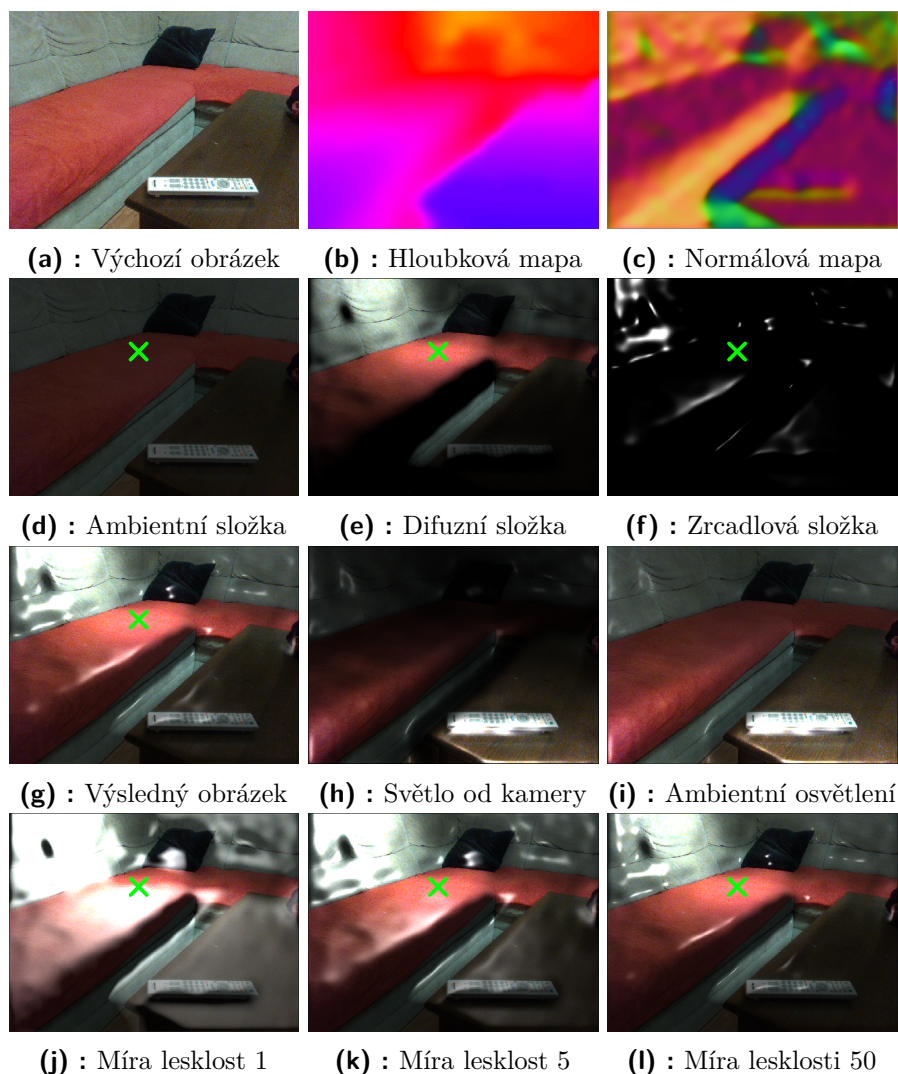
Obrázky 5.1b a 5.1c reprezentují geometrii scény. Na obrázku normálové mapy je vidět, že i povrchy, které jsou v skutečnosti rovné, mají v obrázku viditelné zvlnění. Hlavním důvodem těchto nedokonalostí je nepřesný hloubkový senzor.

Výsledný obrázek 5.1g byl spočten pomocí geometrie scény a světelných vlastností. Obrázek lze rozložit na tři složky: ambientní, difuzní a zrcadlovou složku. Tyto složky lze vidět na obrázcích v druhém řádku.

Ambientní složka intenzity odraženého světla je na obrázku 5.1d. Tato složka kompenzuje nepřímé osvětlení. Při porovnání obrázků 5.1h a 5.1i je možné vidět vliv ambientní složky. Zatímco na jednom obrázcích jsou vzdálenější povrchy výrazně tmavé, na druhém obrázku lze rozeznat objekty v celé scéně.

Na obrázku 5.1e je zobrazena difuzní složka intenzity. Je možné vidět, že povrchy, které jsou blíže ke světlu, jsou více osvětlené. Zároveň je možné na obrázcích 5.1h vidět, že povrchy, které jsou více kolmé ke světlu, jsou více osvětleny (zde se jedná například o záhyb pohovky).

Zrcadlová složka generuje v obraze odlesky (viz obrázek 5.1f) na místech, kde se paprsky od zdroje světla dle zákona odrazu odráží (téměř) do kamery. Koeficient míry lesklosti určuje, jak moc jsou vygenerované odlesky ostré. Toto lze sledovat na obrázcích 5.1j, 5.1k a 5.1l.



Obrázek 5.1: Obrázky demonstrující funkci osvětlovacího modelu

5.2 Neuronová síť

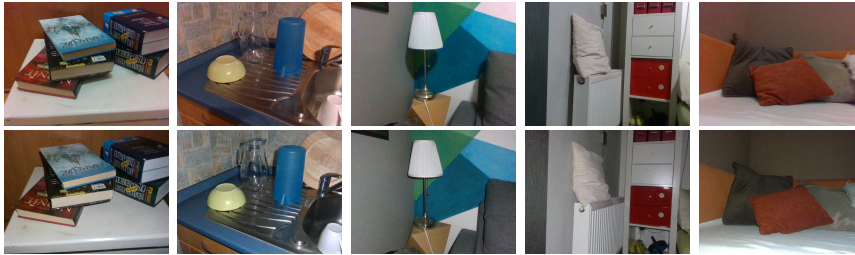
V této sekci předvedeme výsledky související s neuronovou sítí. Zaměříme se také na zaznamenaná trénovací data neuronové sítě a problémy spojené s jejich nasnímáním.

5.2.1 Vytvořená trénovací data

Pro prvotní testování funkčnosti neuronové sítě jsme nasbírali zhruba 6 odlišných scén. Naučená neuronová síť pro tyto trénovací data byla uspokojivá, a proto jsme se rozhodli nasnímat větší dataset. Nasníмали jsme celkem přes 150 různých scén.

Při snaze trénovat síť jsme narazili na problém, že značně nejvyšší roli zde hrál koeficient ambientního odrazu, který byl téměř roven jedné. Koeficient

difuzního odrazu a koeficient zrcadlového odrazu se blížily nule. Nejpravděpodobnější důvod, proč toto nastávalo, bylo, že většina výchozích a vzorových obrázků byla značně podobná (viz obr. 5.2). Podobnost spočívá v tom, že kamera nebyla nasměrována do prostoru, ale všechny objekty ve scéně byly ve relativní blízkosti kamery. To zapříčinilo to, že ve výchozím obrázku nebyla místa, ve kterých by bylo značně nižší osvětlení. Proto se i neuronová síť naučila, že nejvýhodnější je jen s malou změnou barev výchozí obrázek vrátit na výstup.



Obrázek 5.2: Scény s velmi podobným výchozím (první řádek) a vzorovým obrázkem (druhý řádek).

■ Třídění obrázků

Kvůli tomuto neočekávanému problému jsme nasnímaný dataset roztrídili na snímky, které mají a které nemají značnou odlišnost výchozího a vzorového obrázku. Toto třídění nám dataset značně zredukovalo, a proto se síť učila pouze z 50 scén.

Kromě třídění byla kvůli tomuto problému do algoritmu přidána i souvislost mezi koeficientem ambientního odrazu a koeficientem difuzního odrazu. Tato závislost je popsána v kapitole 4.7.3.

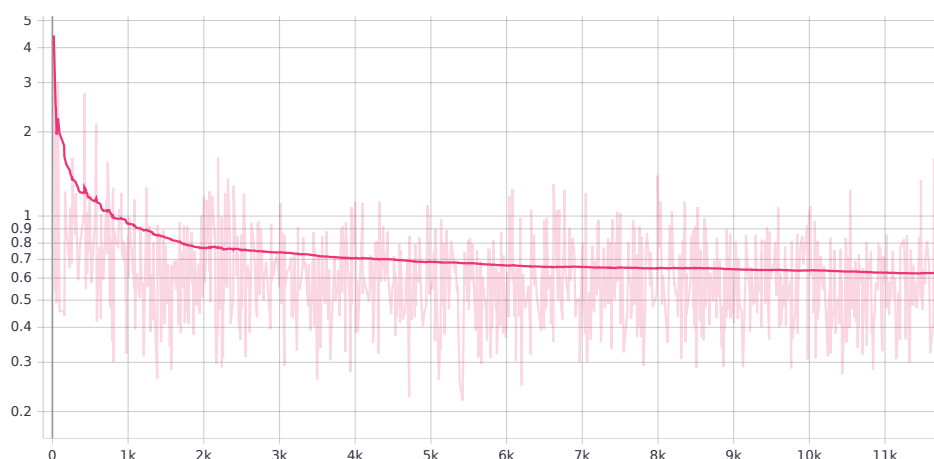
■ 5.2.2 Naučená neuronová síť

■ Učení

Neuronová síť se učila více než 9 hodin. Hodnotu ztrátové funkce během učení jsme sledovali pomocí nástroje *TensorBoard*. Na obrázku 5.3 je možné vidět její hodnotu v čase. Na vertikální ose je hodnota ztrátové funkce a na horizontální je číslo iterace. Ztrátová funkce má stejný průběh, který bychom očekávali. Se zvyšující se dobou trénování se ztrátová funkce čím dál tím více blíží k ideální nulové hodnotě.

■ Výsledky

Na obrázcích 5.4 je uvedeno několik scén, kde je vždy uveden výchozí obrázek v globálním osvětlení, vzorový obrázek osvětlen bodovým světlem od kamery a obrázek vygenerovaný osvětlovacím modelem se světelnými parametry z naučené neuronové sítě.



Obrázek 5.3: Průběh hodnoty ztrátové funkce během učení. Světle růžová značí naměřenou hodnotu ztrátové funkce a tmavě růžová je vyhlazený průběh.

Je možné vidět, že nepřesně určené normálové vektory přinášejí do výsledných obrazů značnou chybu. Nejen, že aplikování osvětlovacího modelu s nepřesně určenými normálovými vektory přinese nerealistický výsledek, ale ani neuronová síť pro získání světelných parametrů nedokáže vhodně využít znalost těchto nepřesných normálových vektorů. To vede k výběru nedokonalých světelných parametrů, což opět způsobí nevyhovující výsledky osvětlovacího modelu.

V prvním řádku můžeme vidět, že vygenerovaný obrázek je skutečně tmavší v místech, která jsou více vzdáleny od kamery. Je vidět, že "nohy" v obrázku správně více září, než tráva ve stejných místech. Zároveň je možné pozorovat značné zkreslení vzhledu dlaždic.

Druhý řádek ukazuje scénu, na které byla demonstrována funkce osvětlovacího modelu (viz obr. 5.1) Algoritmus správně odhadl, že svislá stěna je více osvětlena než vodorovná. Zároveň vidíme, že se algoritmus snaží zesvětlit ovladač. Problémem jsou opět normály, které tuto úlohu značně ztěžují.

Třetí a čtvrtá scéna demonstruje, že je zvláště obtížné určovat normály na tenkých objektech (nohách židlí, kmenu). Můžeme si všimnout, že program i vhodně změnil odstín barvy ubrusu, který se jeví více podobný odstínu na vzorovém obrázku než na výchozím.

Poslední dvě řádky na obrázku 5.4 ukazují případy, kdy je vidět značná chyba. Zvlněná stěna přináší v obou scénách výrazné artefakty. Špatně určené normály přinášejí nerealistický vzhled a dokonce i téměř zabraňují rozeznat druhy objektů.

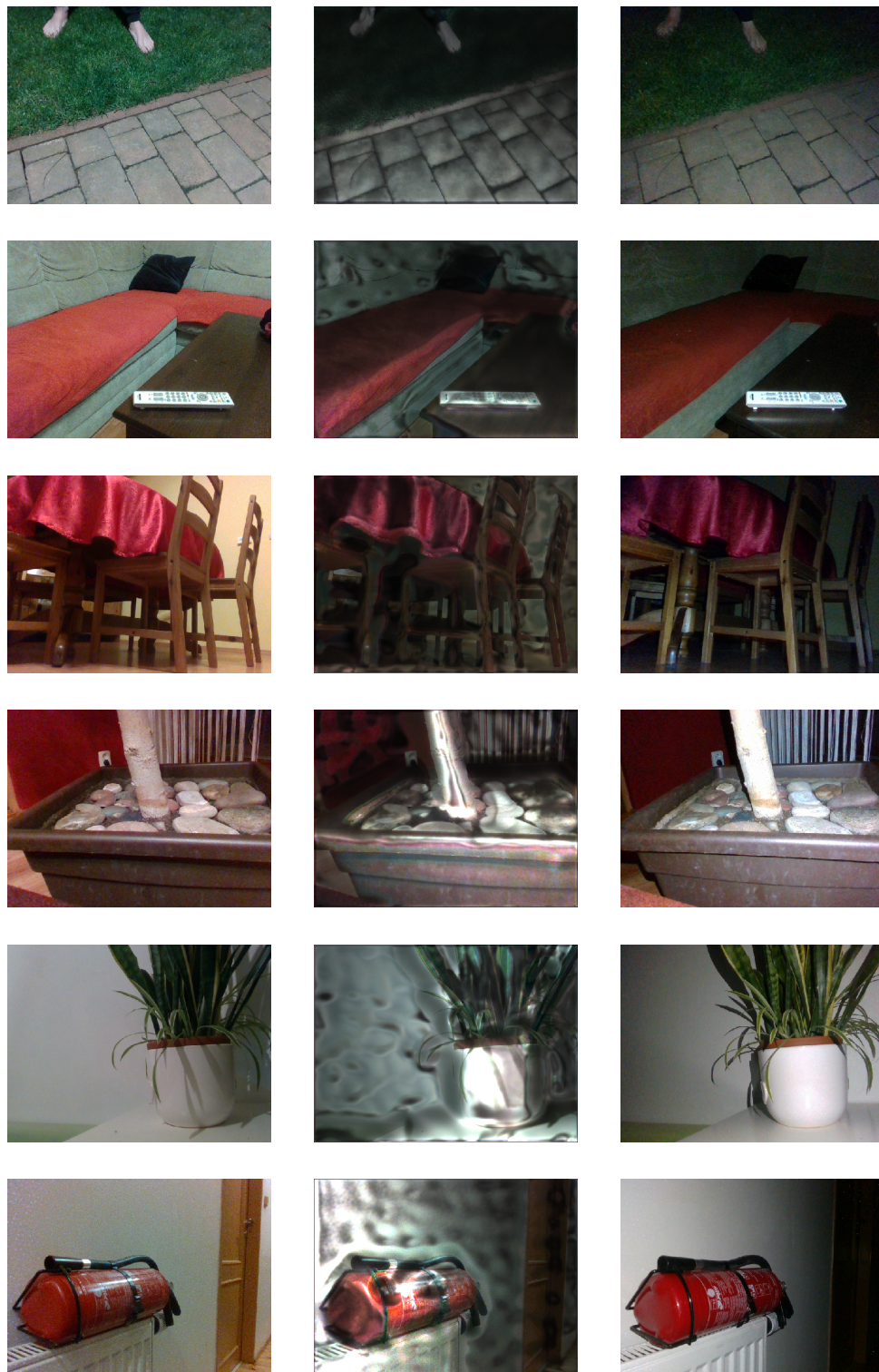
■ Světelné parametry

Zde předvedeme světelné parametry získané pomocí naučené neuronové sítě. Tyto parametry vstupující do osvětlovacího modelu, který generuje výsledný obrázek. Na skupině obrázků 5.5 můžeme vidět jak vstupní mapy do neuronové sítě (výchozí obraz a normálová mapa), tak světelné parametry na výstupu.

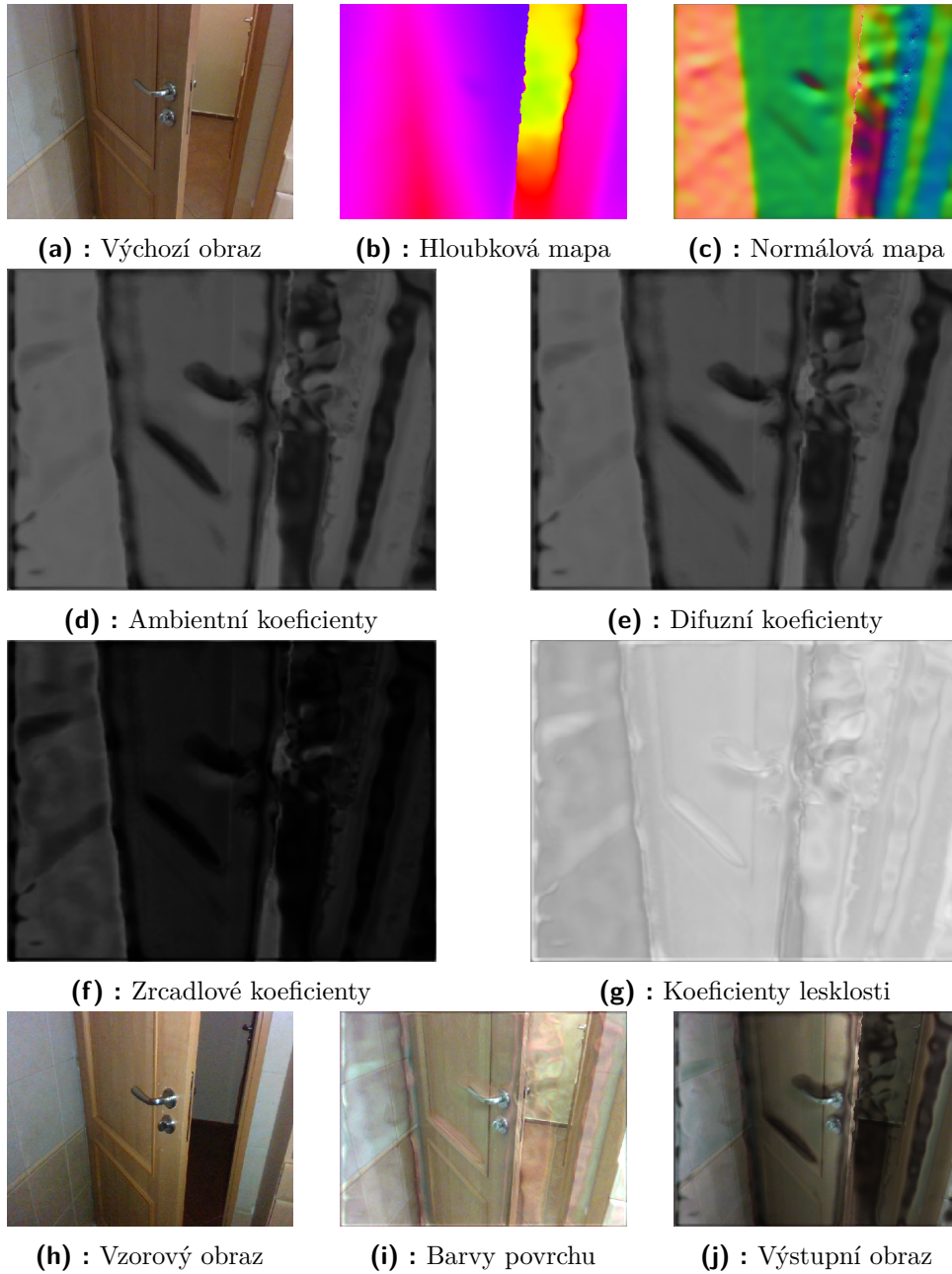
Opět můžeme vidět značné nerovnosti v normálové mapě, které nám způsobují značně zkreslený výsledek.

Při testování bylo vyzorováno, že se algoritmus snaží zesvětlit stěny tím více, čím více jsou rovnoběžné se směrem pohledu (zde se jedná o stěnu na levo). Zároveň můžeme vidět, že ambientní a difuzní mapa (koeficienty) jsou téměř totožné. To nastalo hlavně z toho důvodu, že jsme zavedli závislost mezi těmito dvěma koeficienty (kapitola 4.7.3). Jelikož se ambientní koeficient vycházející s neuronové sítě pohybuje v intervalu $\langle 0; 1 \rangle$, může skutečný koeficient ambientního odrazu vstupující do osvětlovacího modelu nabýt nejvyšší hodnoty koeficientu difuzního odrazu. Neuronová síť se snažila nastavit co nejvyšší ambientní koeficienty, a proto jsou (téměř) rovny koeficientům difuzním.

U zrcadlových koeficientů můžeme vidět, že na stěně jsou značně vysoké, zatímco na dveřích nikoli. U koeficientů míry lesklosti není vidět žádná výrazná korelace.



Obrázek 5.4: Porovnání výchozích obrázků (levý sloupec), vygenerovaných obrázků (prostřední sloupec) a vzorových obrázků (pravý sloupec).



Obrázek 5.5: Příklad vstupních a výstupních dat při získávání světelných vlastností a aplikování osvětlovacího modelu

Kapitola 6

Závěr

Tato práce se zabývá návrhem algoritmu pro odhad změny obrazu při změně osvětlení. Výsledný algoritmus je složen ze dvou hlavních částí. Nejprve jsou ze znalosti obrázku v globálním osvětlení a normálové mapy získány světelné vlastnosti povrchů v obrázku. Tyto vlastnosti jsou získány pomocí naučené neuronové sítě. Světelné vlastnosti popisují nejen skutečnou barvu povrchů, ale také určují, jak moc je povrch materiálu difuzního či zrcadlového charakteru. Následně byl implementován osvětlovací model, který pomocí světelných vlastností, pozice světla a geometrie scény vygeneruje obraz ve zvoleném osvětlení.

Bylo by možné navrhnout neuronovou síť, která se rovnou bude snažit ze vstupních dat vygenerovat obraz v jiném osvětlení. Rozdělení algoritmu na dvě části je však zároveň i výhodou. Získáme-li správné světelné vlastnosti povrchů, není nutné obraz osvětlovat pomocí předem daného osvětlení, ale lze snadno scénu osvětlit dle vlastní volby.

Značnou chybu do algoritmu přináší nepřesné určení hloubkových dat a normálových vektorů. Nepřesnosti vznikly kvůli zvolenému hloubkovému senzoru, kterým jsme data nasníмали. Pro získání o něco lepších dat byl použit *bilaterální filtr*. Existují i sofistikovanější postupy, jak data z hloubkového senzoru více napravit. Tato práce však pojednává o změně osvětlení ve scéně, a tak nebyla tato problematika příliš prozkoumávána.

Pro zlepšení výsledných obrázků by bylo vhodné zdokonalit hloubková data, přeučit neuronovou síť na větším počtu trénovacích dat či přidat do konvoluční neuronové sítě znalost o kvalitě jednotlivých vzdáleností v hloubkové mapě. Zároveň by pravděpodobně bylo přínosem více prozkoumat algoritmy, odhadující směr a druh osvětlení ve scéně. To by vedlo k tomu, že bychom nemuseli předpokládat, že vstupní (výchozí) obrázek je v globálním osvětlení.

I přesto, že výstupní obrázky z vytvořeného algoritmu neodpovídají přímo realitě, je vidět, že další výzkum tímto směrem může přinést lepší výsledky.



Literatura

- [1] M. Dohnal, “Osvětlovací modely v počítačové grafice,” 2008.
- [2] B. T. Phong, “Illumination for computer generated pictures,” *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, 1975.
- [3] J. Bittner, “Materiály, textury, shadery,” 2018.
- [4] Y. Hold-Geoffroy, K. Sunkavalli, S. Hadap, E. Gambaretto, and J.-F. Lalonde, “Deep outdoor illumination estimation,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Honolulu, HI), pp. 2373–2382, IEEE, July 2017.
- [5] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, “Estimating natural illumination from a single outdoor image,” in *2009 IEEE 12th International Conference on Computer Vision*, (Kyoto), pp. 183–190, IEEE, Sept. 2009.
- [6] M.-A. Gardner, K. Sunkavalli, E. Yumer, X. Shen, E. Gambaretto, C. Gagné, and J.-F. Lalonde, “Learning to predict indoor illumination from a single image,” *ACM Transactions on Graphics*, vol. 36, 03 2017.
- [7] S. Jiddi, P. Robert, and E. Marchand, “Reflectance and illumination estimation for realistic augmentations of real scenes,” in *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pp. 244–249, IEEE, 2016.
- [8] Z. Hui and A. C. Sankaranarayanan, “Shape and Spatially-Varying Reflectance Estimation from Virtual Exemplars,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2060–2073, Oct. 2017.
- [9] A. Meka, M. Maximov, M. Zollhoefer, A. Chatterjee, H.-P. Seidel, C. Richardt, and C. Theobalt, “Lime: Live intrinsic material estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6315–6324, 2018.
- [10] K. Hara, K. Nishino, and K. Ikeuchi, “Multiple light sources and reflectance property estimation based on a mixture of spherical distributions,” vol. 2, pp. 1627– 1634 Vol. 2, 11 2005.

- [11] R. O. Dror, E. H. Adelson, and A. S. Willsky, “Recognition of surface reflectance properties from a single image under unknown real-world illumination,” 2001.
- [12] J. Lu and J. Little, “Reflectance function estimation and shape recovery from image sequence of rotating object,” *Computer Vision, IEEE International Conference on*, vol. 0, p. 80, 06 1995.
- [13] K. Nishino, Z. Zhang, and K. Ikeuchi, “Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis,” in *Proceedings Eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1, pp. 599–606, IEEE, 2001.
- [14] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar, “Fast separation of direct and global components of a scene using high frequency illumination,” in *ACM Transactions on Graphics (TOG)*, vol. 25, pp. 935–944, ACM, 2006.
- [15] Y. Yu, P. Debevec, J. Malik, and T. Hawkins, “Inverse global illumination: Recovering reflectance models of real scenes from photographs,” in *Siggraph*, vol. 99, pp. 215–224, 1999.
- [16] T. Y. Wang, T. Ritschel, and N. J. Mitra, “Joint material and illumination estimation from photo sets in the wild,” in *Proceedings of International Conference on 3D Vision (3DV)*, 2018. selected for oral presentation.
- [17] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, “GeoNet: Geometric Neural Network for Joint Depth and Surface Normal Estimation,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (Salt Lake City, UT, USA), pp. 283–291, IEEE, June 2018.
- [18] M. Zollhöfer, P. Stotko, A. Görnitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb, “State of the art on 3d reconstruction with rgb-d cameras,” in *Computer Graphics Forum*, vol. 37, pp. 625–652, Wiley Online Library, 2018.
- [19] Y. Zhang and T. Funkhouser, “Deep Depth Completion of a Single RGB-D Image,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (Salt Lake City, UT, USA), pp. 175–185, IEEE, June 2018.
- [20] J. Barron and J. Malik, “Intrinsic scene properties from a single rgb-d image,” *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 38, 06 2013.
- [21] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing*

and Computer-Assisted Intervention – MICCAI 2015 (N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds.), vol. 9351, pp. 234–241, Cham: Springer International Publishing, 2015.

- [22] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2015.