FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání diplomové práce

Název:	Návrh a implementace algoritmů pro sběr a analýzu fotodokumentace vozidel s využitím kamery a neuronových sítí
Student:	Bc. Martin Vítek
Vedoucí:	Ing. Lukáš Brchl
Studijní program:	Informatika
Obor / specializace:	Znalostní inženýrství
Katedra:	Katedra aplikované matematiky
Platnost zadání:	do konce letního semestru 2023/2024

Pokyny pro vypracování

Do roku 2030 bude mít globální automobilový průmysl hodnotu kolem 9 bilionů dolarů ročně a není tedy žádným překvapením, že automobilový průmysl je v popředí přijímání nových technologií. Jednou z aktuální oblastí zájmu je prevence před podvodnými praktikami se zastíráním historie vozu a obtížnost v diagnostice skutečné minulosti ojetých automobilů. Novodobé algoritmy strojového učení bazarům a pojišťovacím společnostem umožňují optimalizovat jejich procesy související se sběrem fotodokumentace a její následné vyhodnocení. Cílem práce je prozkoumat tyto algoritmy a implementovat funkční celek, který bude sloužit jako prototyp pro možný rozvoj systémů prověřování původu, originality a historie vozidel v České republice.

- Analyzujte podobná existující řešení a jejich relevanci k řešenému projektu.
- Navrhněte snímací prostředí, související snímací techniku a architekturu + infrastrukturu k běhu potřebných algoritmů.
- Implementujte algoritmy strojového učení, konkrétně homogenizaci osvětlení automobilů a jejich segmentaci v rámci scény.
- Nasnímejte vlastní dataset, na kterém budete provádět experimentální vyhodnocení dosaženýh výsledků. V tomto kontextu také prozkoumejte a zaveďte vhodné obrazové materiky pro vyhodnocení.
- Proveďte zhodnocení výsledků práce a navrhněte možná budoucí rozšíření.

Relevantní literatura:

Elektronicky schválil/a Ing. Magda Friedjungová, Ph.D. dne 3. března 2023 v Praze.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práce

Návrh a implementace algoritmů pro sběr a analýzu fotodokumentace vozidel s využitím kamery a neuronových sítí

Bc. Martin Vítek

Katedra aplikované matematiky
Vedoucí práce: Ing. Lukáš Brchl

4. května 2023

Poděkování

Chtěl bych poděkovat především vedoucímu této práce Ing. Lukáši Brchlovi za veškeré rady a pohotové reakce. Dále bych chtěl poděkovat celé své rodině a všem přátelům za jejich pomoc a 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.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 4. května 2023 . . . . . .

Odkaz na tuto práci

Vítek, Martin. *Návrh a implementace algoritmů pro sběr a analýzu fotodokumentace vozidel s využitím kamery a neuronových sítí.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.

Abstrakt

Automobilový průmysl, který je jedním z nejvlivnějších odvětví světové ekonomiky, je při svém růstu a rozvoji do značné míry závislý na technologických a metodických inovacích. Tato práce zkoumá potenciál pokročilých technik zpracování obrazu jako je homogenizace osvětlení, segmentace vozidla a následné modelováních pomocí technologie NeRF. Hlavním cílem práce je vytvořit „proof-of-concept“ řešení využití nových „state-of-the-art“ technologií pro zpracování a modelování vozidel. Vytvořené řešení ukazuje výhody použití nových technologií ve srovnání s tradičními přístupy.

Klíčová slova zpracování obrazu, modelování vozidla, neuronové sítě, NeRF

Abstract

The automotive industry, a vital and influential sector in the global economy, relies heavily on cutting-edge technological advancements and innovative methodologies to drive its growth and development. This study delves into the potential of advanced image processing techniques, encompassing illumination homogenization, image segmentation, and subsequent vehicle modeling using NeRF technology. The primary objective of this research is to develop a proof-of-concept solution that harnesses state-of-the-art technologies for vehicle processing and modeling, demonstrating their efficacy and superiority over traditional approaches. The developed solution underscores the benefits of utilizing novel technologies, showcasing their potential to transform the automotive industry by offering enhanced accuracy, efficiency, and overall performance compared to conventional methods.

Keywords image processing, vehicle modeling, neural networks, NeRF

Obsah

Seznam obrázků

Úvod

Motivace

Automobilový průmysl je jedním z nejvlivnějších a nejrozsáhlejších odvětví světové ekonomiky s dalekosáhlým dopadem na dopravní infrastrukturu a každodenní život. Pro zajištění jeho dalšího růstu a rozvoje je nezbytné neustále inovovat a zdokonalovat současné technologie a metodiky. Jednou z oblastí zájmu je vývoj a aplikace pokročilých algoritmů pro snímání a analýzu obrazů vozidel, zejména v souvislosti s vylepšením celého procesu nákupu a prodeje ojetých automobilů v autobazarech.

Zpracování obrazu je v tomto ohledu zásadní složkou, protože slouží jako základ pro řadu aplikací, které mohou zefektivnit proces nákupu, vzbudit důvěru zákazníků a podpořit transparentnost transakcí s ojetými vozy. Cílem této práce je prozkoumat nejmodernější metody segmentace a modelování vozidel v rámci trhu s ojetými automobily.

Poměrně velká část této práce se věnuje technologii NeRF [1], která si v posledních letech získala značnou pozornost ve výzkumné i komerční sféře. Nabízí nový způsob modelování a vykreslování složitých 3D scén na základě 2D snímků. Zkoumáním schopností NeRF tato práce identifikuje potenciální přínosy a výhody využití těchto pokročilých metod v oblasti reprezentace vozů.

Cíl práce

Hlavním cílem této práce je vytvořit „proof-of-concept“ řešení modelování a vykreslování automobilů pomocí NeRF na základně snímků. Navrhovaný přístup zahrnuje tři hlavní fáze: homogenizaci světla v rámci snímků, odstranění pozadí pomocí segmentace a modelování pomocí NeRF.

Prozkoumáním nových technologií a metod zpracování obrazu se otevírají možnosti dalšího výzkumu či vývoje komerčního řešení zaměřeného na sjednocení a zjednodušení snímání vozidel pro následnou prezentaci a prodej.

Tato práce se zaměřuje na aktuální „state-of-the-art“ metody a zkoumá nové příležitosti v daném odvětví.

Splněním těchto cílů tato práce poskytne cenné poznatky o potenciálu technologie NeRF a dalších pokročilých technik zpracování obrazu v automobilovém průmyslu. V konečném důsledku se tento výzkum snaží posunout hranice možností v oblasti analýzy obrazu v automobilovém průmyslu a podpořit důvěru, transparentnost a celkově lepší zkušenosti jak pro spotřebitele, tak pro prodejce.

Kapitola 1

Teorie

Tato kapitola poskytuje přehled základních termínů a technologií používaných v rámci této práce, s cílem položit teoretický základ pro následující části práce. Tato úvodní kapitola je strukturována do dvou částí. V první části jsou definovány klíčové pojmy ze strojového vidění a procesu vykreslování virtuálních scén, známého jako rendering. Druhá část kapitoly se zaměřuje na neuronové sítě, které se staly průlomovou technologií v několika oblastech strojového vidění [2–4].

Termíny popsané v této kapitole budou používány v příštích kapitolách při představování konkrétních článků a technologií. Je důležité poznamenat, že tato kapitola nemá za cíl podrobně popsat a vysvětlit veškeré termíny a technologie. Pro konkrétní detaily je třeba prozkoumat citovanou literaturu. Tímto způsobem čtenář získá pouze teoretický základ pro další zkoumání a pochopení praktických aspektů metod a technologií představených v následujících kapitolách.

1.1 Strojové vidění

Strojové vidění je široký obor, který se zabývá ziskem, analýzou a dalším zpracováním obrazu s cílem modelovat jak lidé vnímají okolní tří-dimenzionální svět. Cílem strojového vidění je umožnit počítači získávat, interpretovat a dále zpracovávat vizuální data. Základním cílem je replikovat a v některých případech překonat schopnosti lidského vnímání. Analogem k lidským očím je pro počítače snímací zařízení – kamera, která funguje na podobném principu jako lidské oko. [5] Modelování vizuálního světa v plné míře a své komplexitě je daleko složitější než například modelování hlasového ústrojí, které produkuje mluvené slova. [6] V této části jsou popsány některé termíny z tohoto velmi rozsáhlého oboru.

1.1.1 Obraz

Obraz lze definovat jako dvou-dimenzionální funkci $f(x, y)$, kde x a y jsou prostorové souřadnice a hodnota f na souřadnicích (x, y) se nazývá *intenzita* nebo také *stupeň šedi* obrazu v daném bodě. V případě, že x, y a f mají konečné, diskrétní hodnoty, pak lze funkci nazývat digitálním obrazem (dále jen obraz). Obor digitálního zpracování obrazu se týká zpracování digitálních obrazů prostřednictvím počítače. Digitální obraz se skládá z konečného počtu prvků, z nichž každý má určité umístění a hodnotu. Tyto prvky se nazývají obrazové prvky či pixely. Nejčastěji se však používá termín pixel. Na digitální obraz lze také nahlížet jako na matici, kde prvek matice odpovídá pixelu.

Mezi autory nepanuje obecná shoda v tom, kde končí zpracování obrazu a kde začínají další související oblasti, jako je analýza obrazu a počítačové či strojové vidění. Někdy se rozlišuje to tak, že se zpracování obrazu definuje jako disciplína, v níž jsou vstupem i výstupem procesu obrazy. Domníváme se, že tato hranice je omezující a poněkud umělá. Podle této definice by například ani triviální úloha výpočtu průměrné intenzity obrazu, jejích výsledek je jediné číslo, nebyla považována za operaci zpracování obrazu. Na druhé straně existují obory, jako je počítačové vidění, jejichž konečným cílem je pomocí počítačů napodobit lidské vidění, včetně učení a schopnosti činit závěry a přijímat opatření na základě vizuálních vstupů. Tato oblast je sama o sobě odvětvím umělé inteligence, jehož cílem je napodobit lidskou inteligenci. [7]

Je vhodné zmínit, že digitální obraz nemusí obsahovat informaci o světle, tak jak ho vidí lidé, ale pomocí vhodné aparatury lze zachytit a uložit informaci mimo spektrum viditelného světla. Příkladem mohou být rentgenové snímky nebo snímky pořízené infračervenou kamerou. Dále digitální obraz nemusí být pouze šedotónový – pixel vyjádřený jednou hodnotou, představující světlost pixelu nebo barevný, který se skládá z trojice čísel například v RGB. Existují multispektrální či hyperspektrální obrazy, které zachycují informace z několika frekvenčních pásem elektromagnetického vlnění.

1.1.2 Barevné prostory

Barevný prostor je uspořádání barev, které umožňuje jejich označení a specifikaci. Jeden z nejznámějších a nejpoužívanějších barevných prostorů je založený na modelu RGB. Tento model je založený na Kartézské soustavě souřadnic. Barvě přiřadí souřadnice, trojici čísel, které odpovídají množství zastoupení 3 základních barev – červené, modré a zelené. Tento model využívá aditivního míchání barev. Pokud jsou všechny 3 složky na svém maximu, výsledná barva je bílá. Absence složek odpovídá černé. Rozsah jednotlivých složek je často zvolen jako interval $\langle 0, 1 \rangle$. V počítači jsou RGB barvy nejčastěji reprezentována pomocí 3 osmibitových celých čísel bez znaménka, tedy intenzity jednotlivých kanálů jsou v rozsahu 0-255. Nevýhodou tohoto barevného prostoru je nemožnost zachytit všechny barvy takové, jak je vnímají lidé. Nejen z to-

hoto důvodu existuje mnoho různých variant barevných prostorů založených na modelu RGB jako jsou sRGB, Adobe RGB, Apple RGB. [7]

Dalším barevným prostorem, který se v této práci vyskytuje je barevný prostor CIELAB, též označovaný jako $L^*a^*b^*$. Tento barevný prostor byl vyvinut Mezinárodní komise pro osvětlování (francouzsky Commission internationale de l'éclairage – CIE) v roce 1976 s cílem vytvořit barevný prostor vhodný pro porovnávání rozdílu barev. Cílem bylo vytvořit barevný prostor, který je percepčně uniformní, tedy aby euklidovské vzdálenosti bodů v tomto prostoru odpovídaly vnímání člověka. Na rozdíl od RGB prostoru L*a*b* prostor pokrývá všechny okem viditelné barvy – celý tzv. gamut viditelného spektra. [7] Jak alternativní název $L^*a^*b^*$ napovídá, teto barevný prostor se skládá z 3 složek. První složka představuje světlost (anglicky lightness). Její hodnoty jsou v rozmezí 0–100, kde 0 připadne černé barvě a hodnota 100 označuje bílou. Druhé dvě složky nejsou teoreticky omezeny, ale většinou jsou v rozsahu -128 a 127. Složka a* odpovídá přechodu od zelené (záporné hodnoty) k červené (kladné hodnoty). Složka b* odpovídá přechodu od modré (záporné hodnoty) ke žluté (kladné hodnoty). [8]

1.1.3 Histogram

Nechť r_k, pro $k = 0, 1, \ldots, L-1$, označuje intenzity L-stupňového digitálního obrazu $f(x, y)$. Nenormalizovaný histogram obrazu f je definován jako

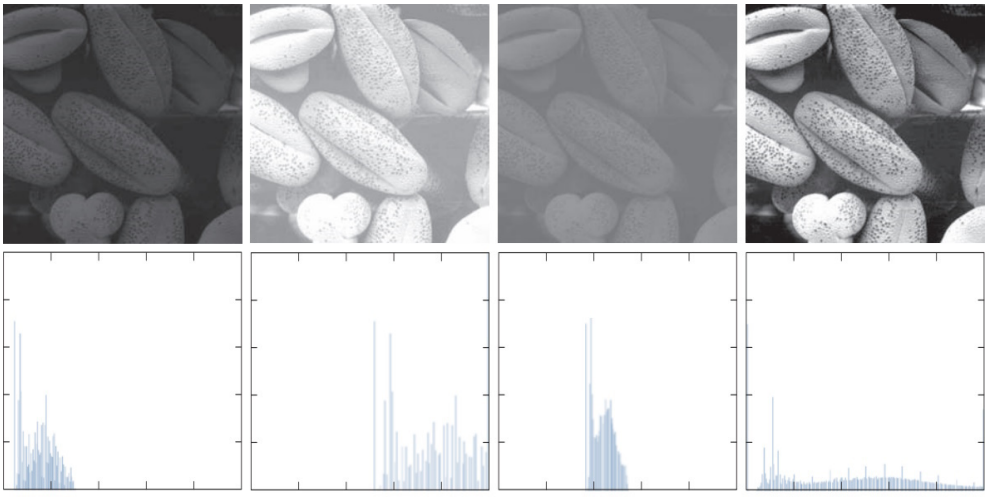
$$h(r_k) = n_k, \quad \text{pro } k = 0, 1, \ldots, L-1,$$

kde n_k označuje počet pixelů v obraze f s intenzitou r_k. Normalizovaný histogram obrazu f je definován jako

$$p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN},$$

kde M a N značí výšku a šířku obrazu, tedy počet řádků a sloupců. Většinou se pracuje s normalizovanými histogramy, které se označují jednoslovně histogramy. Suma $p(r_k)$, přes všechna k je rovna 1. Složky $p(r_k)$ jsou odhady pravděpodobností výskytu dané intenzity v obraze. Histogram lze také označit jako znázornění distribuce jasových hodnot pixelů. Výpočet histogramu je velmi jednoduchý na výpočet a je dobře paralelizovatelný. Z tohoto důvodu jsou techniky zpracování obrazu na základě histogramu populární zvláště v případech, kdy čas zpracování hraje velikou roli. [7]

Prostřednictvím tohoto znázornění lze získat cenné informace o dynamickém rozsahu, kontrastu a dalších vlastnostech obrazu, které mohou být klíčové v různých aplikacích, jako je zlepšení obrazu, segmentace a komprese. Tvar histogramu je úzce spjat s vzhledem obrazu. Tmavé obrazy mají hodnoty častěji se vyskytující se u nižších intenzit, naopak světlé obrazy mají rozdělení posunuté do vyšších intenzit. Další charakteristika, kterou lze jednoduše vyčíst z histogramu je kontrast. Obraz jehož histogram má zastoupení



Obrázek 1.1: Ukázka 4 obrazů s odpovídajícími histogramy. a) Histogram tmavého obrázku. b) Histogram světlého obrázku. c) Histogram obrázku s nízkým kontrastem. d) Histogram obrázku s vysokým kontrastem. Zdroj: [7]

intenzit blízko sebe je méně kontrastní než obraz, u kterého jsou hodnoty $p(r_k)$ více rozprostřené. Tyto vlastnosti jsou zobrazeny na obrázku 1.1. Sledování histogramu je velmi užitečné i při snímání obrazu kamerou. Z histogramu lze vyčíst, zda je obraz podexponován (resp. přeexponován), tedy je-li příliš tmavý (resp. světlý).

Histogram se nemusí počítat pouze u šedotónového obrazu, ale lze ho spočítat i u barevného obrazu. Například u RGB obrazu, nevzniká pouze jeden histogram, ale histogramy tři, pro každou barevnou složku jeden.

1.1.4 Segmentace obrazu

Segmentace obrazu je rozčlenění obrazu na části, které mají podobnou vlastnost, nebo které obsahují nějaké objekty z reálného světa. Například v prvním případě může jít o určení části obrazu, která má stejný jas a v druhém případě by mohlo jít o úlohu nalezení části obrazu, ve které se nachází snímané vozidlo, což bude cílem v této práci. Segmentaci lze chápat jako oddělení objektu od pozadí, ale také jako označení částí obrazu s přiřazením označení, co se v dané oblasti nachází. V této práci půjde hlavně o plošnou segmentaci – označení snímaného vozidla a jeho oddělení od pozadí. Půjde tedy o vytvoření masky, která bude pokrývat daný objekt a oddělí vozidlo od pozadí. Maska se často reprezentuje jako dvouúrovňový, binární, ($L = 2$) obraz. Pozadí je označeno

nulou a objekt je označen jedničkou. Toto označení umožňuje oddělit pozadí pouhým násobením obrazu s maskou po složkách.

Při hodnocení různých metod segmentace je třeba zvolit vhodnou metriku, podle které se hodnotí kvalita segmentace vzhledem k referenční segmentaci – masce. Často používanou takovou metrikou je poměr mezi průnikem výsledné masky a referenční masky a sjednocením těchto masek. Tento poměr se nachází v intervalu $\langle 0, 1 \rangle$, kde 0 vyjde v případě, že masky nemají žádnou společnou část a 1 odpovídá přesné shodě výsledné masky s referencí. Tato metrika se často značí IoU – průnik nad sjednocením (anglicky Intersection over Union) a existuje i její zobecněná varianta GIoU. [9]

Problém segmentace obrazu se dá řešit mnoha tradičními postupy a algoritmy různých složitostí od jednoduchého prahování až po složitější metody. Základní segmentaci lze provést pomocí jednoduchého prahování, které binarizuje obraz na základě porovnání hodnot pixelu se zvoleným prahem. Volba prahu může být ponechána na uživateli nebo může být zvolena automaticky pomocí nějakého kritéria. Například Otsuova metoda volby prahu se snaží zvolit práh tak, aby se maximalizoval rozptyl mezi světlými a tmavými plochami. Další možností je adaptivní prahování, která volí jiný práh pro různé části obrazu. Mezi pokročilejší segmentační techniky jistě patří split & merge segmentace, která rekurzivně dělí obraz na 4 části, kvadranty, na základě kritéria homogenity. Části poté algoritmus spojuje pokud spojení daných oblastí neporušuje podmínku homogenity. V posledních letech se k segmentaci často používají kromě tradičních algoritmů neuronové sítě. Přístup pomocí deep learningových metod nabízí, ve srovnání s tradičními metodami, vyšší přesnost a přizpůsobivost dané úloze.

1.1.5 Obrazové příznaky

Hledání obrazových příznaků je častým dalším krokem ve zpracování obrazu po segmentaci. Cílem je nalézt v obraze určité příznaky a přiřadit jim popisky/označení, které určují jejich vlastnosti. Například jedním hledaným příznakem mohou být rohy, ať už objektů nebo pouze segmentovaných ploch. Nalezeným rohů lze přiřadit popisky, které určují jejich vlastnosti, jako jsou poloha a orientace. [7]

Ačkoli neexistuje žádná všeobecně přijímaná formální definice toho, co je to obrazový příznak, lze intuitivně považovat příznak za charakteristický atribut nebo popis „něčeho“, co chceme označit nebo odlišit. To „něco“ se vztahuje buď k jednotlivým objektům v obraze, k celým obrazům či dokonce k souborům obrazů. O obrazových příznacích tedy uvažujeme jako o atributech, které nám pomohou přiřadit jedinečné označení objektům v obraze nebo mají význam při rozlišování obrazů nebo celých skupin obrazů. [7]

V této práci se obrazové příznaky používají zejména k nalezení korespondujících bodů v různým snímcích. Tyto korespondující příznaky později slouží k odhadu polohy kamery odkud byl snímek pořízen. Jistě by se dali nalézt

příznaky jako jsou kola vozidla, dveře, zpětná zrcátka, jejich polohy, orientace a další informace, ze kterých by později šlo odhadnout polohu celého vozu a kamery v prostoru. Tento postup by byl poměrně složitý, protože vyvíjet algoritmy pro detekci a označení takto konkrétních příznaků by bylo nejen časově náročné, ale také bez záruky funkčnosti bez složitého testování. Pro detekci a následné porovnání se často používají příznaky z algoritmu SIFT, které sice nemají tak přímou interpretaci, ale jde o často používanou technologii, která funguje na veliké škále různých snímků.

SIFT

Algoritmus Scale-Invariant Feature Transform (SIFT) je široce uznávaný a vlivný algoritmus pro detekci a popis popis obrazových v počítačovém vidění a zpracování obrazu. V roce 1999 jej vyvinul David G. Lowe a později jej publikoval ve svém článku "Object Recognition from Local Scale-Invariant Features"[10]. Algoritmus SIFT vyniká v nalezení a popisu lokálních příznaků v obrazech, které jsou invariantní vůči změnám měřítka, rotace a osvětlení, což umožňuje robustní a přesné porovnávání různých pohledů na stejnou scénu nebo objekt.

Mezi hlavní výhody algoritmu SIFT patří jeho invariance vůči změnám měřítka, rotace a osvětlení, jakož i jeho odolnost vůči šumu a malým změnám úhlu pohledu. Díky těmto vlastnostem se dobře hodí pro různé aplikace, jako je rozpoznávání objektů, spojování obrazů a 3D rekonstrukce. Algoritmus SIFT má však i některé nevýhody. Může být výpočetně náročný, zejména při použití na velké obrazy nebo v aplikacích v reálném čase. Navíc algoritmus SIFT není zcela invariantní vůči afinním transformacím nebo výrazným změnám úhlu pohledu, což může v některých případech vést ke snížení výkonu. Navzdory těmto omezením zůstává algoritmus SIFT základní technikou v oblasti počítačového vidění a nadále inspiruje vývoj nových metod detekce a popisu prvků. Podrobný popis algoritmu, jeho několika kroků, lze nalézt v [7] na stranách 881–898.

1.1.6 Ray tracing

Ray tracing je technika pro generování realistických obrazů ve virtuálních scénách pomocí simulace chování světelných paprsků při jejich interakci s objekty a prostředím. Na rozdíl od široce používaného přístupu *rasterizace* ray tracing využívá vrhání paprsků z pozice kamery skrze každý pixel výstupního obrazu do scény. Barva každého pixelu je určena interakcemi mezi paprsky a objekty ve scéně, jako jsou odrazy, lomy, rozptyl nebo absorpce. Díky tomu se ray tracing stal klíčovou složkou pro dosažení realismu a pohlcujících zážitků v aplikacích počítačové grafiky.

Obliba ray tracingu pramení z jeho schopnosti vytvářet fyzikálně přesné a fotorealistické obrazy pomocí složitých světelných efektů a přírodních jevů.

I když je ray tracing v současnosti často diskutovaným tématem zejména v herním průmyslu, první práce na tomto tématu sahají téměř pět desetiletí zpět. [11–13] Od té doby bylo v oblasti počítačové grafiky vyvinuto mnoho technik, které mají za cíl zrychlit výpočty a algoritmy pro výpočetně náročný proces ray tracingu.

Hardwarová akcelerace ray tracingu měla však jen omezený úspěch do té doby, než byla vydána technologie RTX společnosti NVIDIA v jejích grafických procesorech architektury Turing. [14] Tato společnost uvedla, že Turing hardware obsahuje speciální, takzvaná *RT jádra*, která urychlují ray tracing. V oficiálním dokumentu architektury Turing se uvádí, že jádra RT obsahují dvě jednotky, které provádějí testy průniku paprsku s bounding-boxy a s trojúhelníky. [15]

Příbuzná technika ray tracingu je ray marching. Metoda byla poprvé představena jako „sphere tracing“ [16], při které se nepočítá průnik paprsku s objekty, ale postupuje se v prostoru po jednotlivých krocích. Délku kroku určuje takzvaná „signed distance function“ (SDF), která pro každý bod v 3D prostoru určuje vzdálenost k nejbližšímu objektu. Scéna je reprezentována implicitně pomocí této funkce, což umožňuje zajímavé možnosti jako například jednoduchou implementaci měkkých stínů, spojitý přechod mezi objekty a vykreslování fraktálů a objemů. [17, 18]

1.1.7 Image Based Rendering

Image-Based Rendering (IBR) je technika počítačové grafiky pro vytváření nových pohledů nebo obrazů scény na základě sady vstupních obrazů. Hlavní myšlenkou IBR je použití existujících snímků spolu s přidruženými informacemi, například o hloubce nebo poloze kamery, k syntéze nových pohledů nebo obrazů bez explicitní rekonstrukce úplné 3D geometrie scény. Existují i hybridní metody [19], které využívají 3D modelů i obrazových transformací k získání požadovaného výsledku.

Metody IBR se při generování nových pohledů obvykle spoléhají na transformace obrazu, míchání a další techniky, jejichž cílem je poskytnout vizuálně koherentní a realistické výsledky. Tyto metody mohou být užitečné zejména pro virtuální realitu [20], rozšířenou realitu a další aplikace, kde je nutné generovat nové pohledy z existujících dat.

Existují různé přístupy k IBR od interpolace a extrapolace pohledů až po vykreslování světelného pole (anglicky light field nebo také radiance field). Každá z těchto metod má své vlastní výhody a omezení, ale všechny mají společný cíl - syntetizovat nové pohledy na základě existujících obrazových dat. [21] K reprezentaci scény je možné použít „plenoptickou funkci“ [22, 23], což funkce zachycující veškerou informaci o interakci světla se scénou.

1.2 Neuronové sítě

Neuronové sítě představují jeden z možných přístupů umělé inteligence, který byl inspirován způsobem, jakým funguje lidský mozek. Tyto sítě se skládají z umělých neuronů, které jsou propojeny a tvoří hierarchickou strukturu schopnou učení a adaptace. Neuronové sítě se staly základním stavebním kamenem pro řadu úspěšných technik strojového učení, zvláště v oblasti zpracování obrazu, zpracování přirozeného jazyka, doporučovacích systémů a mnoho dalších aplikací.

Jednou z hlavních nevýhod neuronových sítí je jejich obtížná interpretace. Ačkoli neuronové sítě často poskytují lepší výsledky než tradiční metody, zjistit, jak dospěly k daným závěrům, může být náročné. Tento nedostatek vysvětlitelnosti může vést k problémům při identifikaci chyb v rozhodnutích neuronových sítí. Například v oblasti medicíny, kde je důležité porozumět důvodům za diagnózou stanovenou neuronovou sítí, může být nedostatek transparentnosti omezením při získávání důvěry lékařů a pacientů v taková rozhodnutí.

Neuronové sítě lze rozdělit do několika hlavních kategorií na základě jejich architektury a využití. Mezi základní typy patří vícevrstvé perceptrony (MLP), konvoluční neuronové sítě (CNN), rekurentní neuronové sítě (RNN) [24], generativní adversariální sítě (GAN) [25], autoenkodéry (AE) [26] a transformery [27].

Neuronové sítě lze použít například i k zakódovaní dat – obrazu, textu, grafy – do reprezentace pomocí předem definovaného počtu čísel, tzv. „embedding“. Jedná se často o převod z disktrétního prostoru objektů, například slov, do spojitého prostoru vektorů. Typickým požadavkem na embedding je, aby reflektovali některé vlastnosti kódovaných objektů, například podobné slova by měli mít podobné reprezentace. Embedding se většinou získává pomocí aplikace předtrénované neuronové sítě na data a následná extrakce hodnot v předposlení vrstvě. Embeddingy se hojně se využívají ve zpracování přirozeného jazyka a rekomendačních systémech.

Vývoj a výzkum neuronových sítí pokračuje rychlým tempem, což vede k vytvoření nových architektur a technik, které rozšiřují možnosti aplikace neuronových sítí a umělé inteligence obecně v průmyslu a společnosti. Díky pokrokům v hardwaru, zejména v oblasti grafických procesorů (GPU), které urychlují trénink neuronových sítí, se tato oblast stává stále dostupnější a efektivnější.

1.2.1 Vícevrstvý perceptron

Vícevrstvé perceptrony jsou základním typem dopředných neuronových sítí, které lze použít pro různé úkoly, jako je klasifikace, regrese nebo zpracování signálu. Vícevrstvé perceptrony se často také označují jako „vanilla“ neuro-

nové sítě. Toto označení zdůrazňuje fakt, že se jedná o základní typ neuronových sítí.

MLP sítě se skládají ze vstupní vrstvy, jedné či více skrytých vrstev a výstupní vrstvy. Každá vrstva obsahuje sadu neuronů, přičemž vstupní vrstva přijímá data a výstupní vrstva vytváří konečný výsledek. Neurony v každé vrstvě jsou propojeny se všemi neurony v sousedních vrstvách prostřednictvím vážených spojení reprezentovaných maticemi. Váhy mezi neurony jsou doplněny o tzv. bias, který pomáhá modelovat vztahy v datech a vypořádat se s přítomností 0 na vstupu. Na součet skalární součinu hodnot vstupujících neuronů s odpovídajícími vahami je spolu s biasem se aplikuje tzv. aktivační funkce, která do modelu vnáší nelinearitu. Volba aktivační funkce může záviset na pozici neuronu v síti, úloze a dalších faktorech. Během trénování se váhy (a bias) upravují pomocí učícího algoritmu, jako je například zpětné šíření chyby [7, 28] tak, aby se minimalizoval rozdíl mezi předpověďmi modelu a referencí.

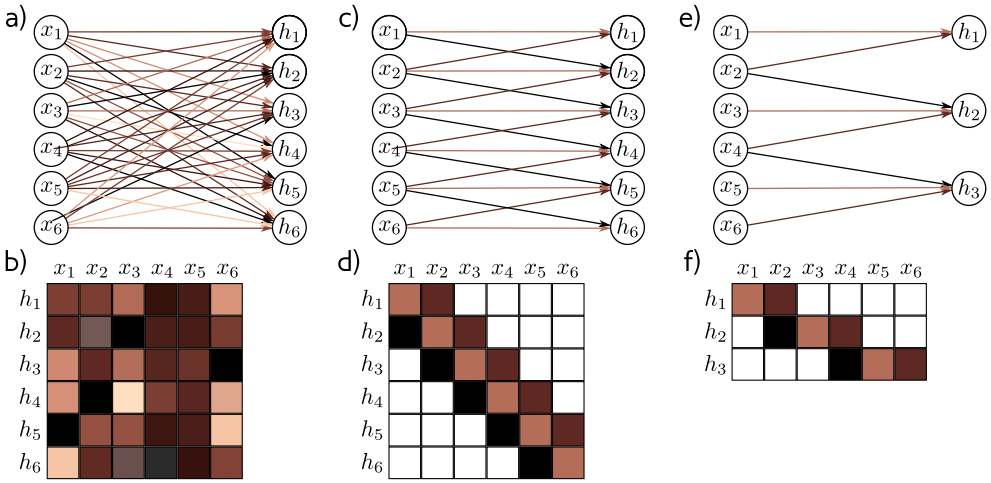
1.2.2 Konvoluční neuronové sítě

Konvoluční neuronové sítě jsou specializovaným typem modelu hlubokého učení určeného ke zpracování dat v mřížce, například obrazů, čímž se liší od tradičních vícevrstvých perceptronů, které zpracovávají vektorová data. Klíčový rozdíl mezi CNN a MLP spočívá v jejich architektuře. CNN se skládají z konvolučních vrstev, tzv. *pooling* vrstev a plně propojených vrstev. Často se používají v úlohách, jako je rozpoznávání obrazů, segmentace obrazu nebo detekce objektů.

Konvoluční vrstvy aplikují na vstupní data učící se filtry v pomocí posuvného okna. Těmito filtry model získává lokální prostorové vzory. Pooling vrstvy zmenšují rozměry map příznaků, které se získali z konvolučních vrstev. Plně propojené vrstvy, obvykle umístěné na konci sítě, se používají pro klasifikační nebo regresní úlohy. Využitím těchto vrstev v hierarchické struktuře se konvoluční neuronové sítě mohou naučit robustní a translačně invariantní funkce pro získání příznaků, což je činí obzvláště vhodnými pro aplikace počítačového vidění a rozpoznávání vzorů.

Jedním z prvních úspěšných aplikací konvolučních neuronových sítí, které využívaly učení založené na zpětném šíření chyby, bylo rozpoznávání ručně psaných číslic. [29] V tomto průkopnickém článku se autoři zaměřili na rozpoznání číslic poštovních směrovacích čísel, která byla používána v Poštovní službě Spojených států amerických. Díky úspěšnému nasazení konvolučních neuronových sítí v této úloze byl položen základ pro další výzkum a vývoj v oblasti počítačového vidění a strojového učení.

Na obrázku 1.2 je vizualizovaný rozdíl mezi plně propojenou vrstvou a konvoluční vrstvou. Z obrázku je patrný velký rozdíl v počtu parametrů, které vrstva obsahuje. U plně propojené vrstvy je počet parametrů závislý na počtu vstupních a výstupních neuronů, kdežto u konvoluční vrstvy je počet parametrů vrstvy závislý pouze na velikosti učících se filtrů. U konvolučních vrstev



Obrázek 1.2: Plně propojená vrstva vs. konvoluční vrstva. a) Plně propojená vrstva s vahami spojující všechny neurony x s neurony h. b) Matice vah plně propojené vrstvy obsahující 36 hodnot. c) Konvoluční vrstva s velikostí filtru 3. d) Zobrazení matice vah jak by vypadala vyobrazena jako v případě b. Matice obsahuje 16 hodnot, ale pouze 3 unikátní. e) Konvoluční vrstva s velikostí filtru 3 a parametrem stride rovným 3. f) Obdoba obrázku d pro jiné parametry konvoluční vrstvy. Obrázek nezobrazuje bias. Zdroj: [28]

jsou matice vah řídké a hodnoty se v nich opakují. V praxi se však tato velká, byť řídká, matice nepoužívá.

1.2.3 U-Net a U2-Net

U-Net je architektura konvoluční neuronové sítě speciálně navržená pro úlohy segmentace obrazu. Poprvé ji představili Olaf Ronneberger, Philipp Fischer a Thomas Brox ve svém článku z roku 2015 s názvem „U-Net: Convolutional Networks for Biomedical Image Segmentation“. [30] Hlavní motivací pro vznik sítě U-Net bylo umožnit přesnou a efektivní segmentaci biomedicínských obrazů, zejména v případech, kdy je množství anotovaných trénovacích dat omezené. Síť U-Net byla původně navržena k řešení úloh segmentace biomedicínských obrazů, ale díky své efektivitě byla tato architektura upravena a použita v různých dalších úlohách segmentace obrazu v různých oblastech.

Architektura sítě U-Net připomíná písmeno „U“, z čehož pochází i její název. Síť U-Net se skládá ze dvou hlavních částí: kodéru (anglicky encoder) a dekodéru (anglicky decoder). Celková struktura sítě je symetrická. Kodér a dekodér jsou propojeny v několika částech. Výstup konvoluční vrstvy kodéru se stává částí vstupu odpovídající vrstvy dekodéru.

U2-Net, též někdy označována U^2-Net, je architektura hlubokého učení určená pro úlohy detekce a segmentace význačných objektů (anglicky salient

object detection). V roce 2020 ji představili Qin et al. v článku s názvem „U^2-Net: Going Deeper with Nested U-Structure for Salient Object Detection“. [31] U2-Net je model pro zpracování obrazu, jehož cílem je identifikovat a zvýraznit vizuálně nejdůležitější nebo výrazné oblasti v obraze a zároveň odstranit méně relevantní pozadí.

Architektura sítě U2-Net je založena na vnořené struktuře odpovídající tvaru písmene „U“. Síť je rozšířením architektury U-Net. Vnořená struktura v U2-Net se skládá z několika malých hierarchicky uspořádaných U-Net sítí, což modelu umožňuje zachycovat a zpracovávat prvky v různých měřítkách a úrovních abstrakce.

Kapitola 2

Související práce

Tato kapitola je zaměřena na prozkoumání a představení klíčových článků a publikací, které souvisí s tématem této práce, aby čtenář mohl získat širší kontext a pochopení souvisejících výzkumných oblastí. Přehled zahrnuje nejen přístupy a techniky, které byly použity v souvisejících oborech, ale také se zaměřuje na aktuální a inovativní technologie, zvláště v oblastech segmentace obrazu a modelování scény.

Kapitola je strukturována do několika sekcí, které postupně pokrývají různé aspekty souvisejících prací. Každá část se zaměřuje na určitý problém, jako je homogenizace a vylepšení obrazu, segmentace obrazu a modelování scény s vozidlem. Tímto způsobem je čtenáři poskytnut koherentní a logický přehled nad souvisejícími pracemi. V závěru této kapitoly je zmíněno několik zajímavých publikací, které jsou blízké tématu této práce, avšak se nepřímo nehodí pro výsledné řešení. Tyto práce jsou zmíněny pro úplnost a podporu dalšího výzkumu.

V následující kapitole budou představené technologie analyzovány, zhodnoceny a jejich relevantní aspekty budou zváženy ve výsledné implementaci, aby bylo dosaženo optimálního řešení problému zkoumaného v této práci.

2.1 NPEA

Článek "Naturalness Preserved Enhancement Algorithm for Non-Uniform Illumination Images"[32] představuje novou techniku vylepšení obrazu s důrazem na zachování přirozeného vzhledu. Na začátku autoři definují metriku, podle které, dle jejich slov, lze objektivně určit zachování přirozenosti vzhledu obrazu. V druhé části jsou pak popsány dvě transformace, které dohromady tvoří algoritmus pro vylepšení snímků. Představený algoritmus NPEA má za cíl zlepšit vzhled u obrazů s neuniformním osvětlením.

Hlavní myšlenka algoritmu je rozdělit vstupní obrázek na dvě složky – osvětlení a odrazivost. Postupnými úpravami se složky vylepší a nakonec se opět sloučí do jednoho obrazu. Algoritmus se skládá z následujících částí:

a) b) c)

Obrázek 2.1: Zobrazení dekompozice obrazu na složku osvětlení a odrazivosti v NPEA. a) původní obrázek b) osvětlení/iluminance L_r c) odrazivost R Zdroj: [32]

1. Získání intenzity v obraze vztahem $L(x, y) = \max_{c \in \{r,g,b\}} I^c(x, y)$. Osvětlení pro daný pixel (x, y) se spočítá jako maximální hodnota přes barevné kanály v daném pixelu obrazu I.

2. Odhad osvětlení (iluminance). V tomto kroku autoři přicházejí s novým přístupem, odlišným od použití Gaussových nebo bilaterárních filtrů. Definují „Bright-Pass Filter“, který, na rozdíl od ostatních filtrů, bere v potaz pouze okolní pixely, které jsou světlejší. Aplikací filtru na intenzitu z prvního kroku vznikne iluminance $L_r = BPF[L(x, y)]$.

3. Získání odrazivosti pomocí získaného osvětlení. Autoři se odkazují na teorii Retinex [33] a vypočítají odrazivost pro každý kanál jako $R^c(x, y) = I^c(x, y)/L_r(x, y)$. Obrázek 2.1 ukazuje možnou dekompozici obrazu.

4. Následná úprava osvětlení pomocí mapování histogramu. Nejprve se na osvětlení aplikuje funkce logaritmus $L_{lg}(x, y) = \log(L_r(x, y) + \varepsilon)$ a poté mapování histogramu na základě statistik o počtu a intenzitách pixelů v obraze. Toto mapování autoři nazývají jako „Bi-Log“ transformaci. Aplikací této transformace se získá iluminance $L_m(x, y)$, která se pak, opět podle teorie Retinex, spojí s odrazivostí $R(x, y)$ a vyjde výsledný vylepšený obrázek.

2.2 LIME

Metoda LIME z článku „LIME: Low-Light Image Enhancement via Illumination Map Estimation“ [34] se soustředí na vylepšení obrazu zachyceného

při nízkých světelných podmínkách. Tento článek opět staví na teorii Retinex. Základní princip je stejný jako u NPEA, tedy nejprve odhad iluminance pomocí maxima přes barevné kanály a následné vylepšení tohoto odhadu.

Autoři navrhují optimalizační problém

$$\min_{\mathbf{T}} \|\hat{\mathbf{T}} - \mathbf{T}\|_F^2 + \alpha \|\mathbf{W} \circ \nabla \mathbf{T}\|_1,$$

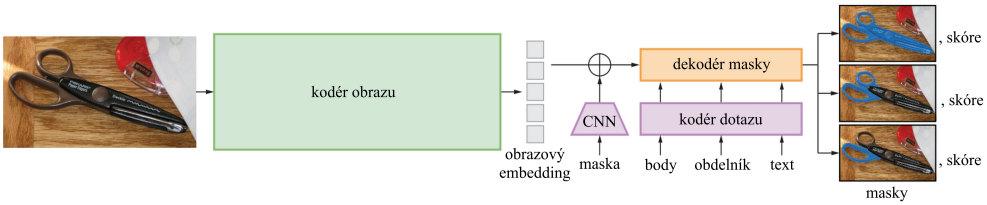
kde $\hat{\mathbf{T}}$ označuje prvotní odhad osvětlení, $\|\cdot\|_F$ a $\|\cdot\|_1$ označují Frobeniovu a ℓ_1 normu, $\mathbf{W}$ je matice vhodně zvolených vah a Řešením tohoto problému se získá „vhodná“ mapa osvětlení $\mathbf{T}$, pomocí které se následně obrázek upraví pouhým dělením po složkách původního obrázku. Autoři navrhují 3 různé strategie volby matice vah $\mathbf{W}$, se kterými následně provádí experimenty. V článku jsou popsány dva algoritmy pro řešení navrženého algoritmu, první přesný a druhý rychlejší využívající aproximací. Přesné fungování metody LIME lze najít v článku nebo také v neoficiální implementaci [35].

Článek „A Spatial-Frequency Domain Associated Image-Optimization Method for Illumination-Robust Image Matching“ [36] staví na metodách NPEA [32] a LIME [35] a následně je porovnává. Hlavním cílem tohoto článku je upravit obrazy s nevhodným osvětlením tak, aby byly vhodné pro následné použití v oblasti fotogrammetrie a konkrétně pro úlohu zvanou „image matching“ – porovnání a registraci obrazů na základě podobnosti obrazových příznaků v nich. Představený algoritmus se skládá ze dvou hlavních částí. Nejprve se u vstupního obrazu sjednotí osvětlení pomocí analýzy v prostorové oblasti obrazu a adaptivní gama korekce, což se jeví jako vhodná metoda [37]. Druhým krokem je dekompozice na osvětlení a odrazivost a následné vylepšení obrazových příznaků pomocí filtrace ve frekvenčním prostoru obrazu. Autoři následně porovnávají představenou metodu s představenými technikami NPEA a LIME, aby demonstrovali účinnost navrženého algoritmu.

2.3 SAM

V oblasti segmentace obrazu přináší čerstvou inovaci článek nazvaný „Segment anything“ [38], ve kterém je prezentován model „Segment anything model“ (zkráceně SAM). Tento článek byl publikován autory z Meta AI Research v dubnu 2023 a představuje novou úlohu, nový model a dataset pro segmentaci obrazu. Autoři čerpali inspiraci z velkých jazykových modelů (anglicky large language model – LLM) [39], které v poslední době zažívají značný rozmach.

Cílem autorů je vytvořit „základní model“ [40] pro segmentaci, který bude předtrénován na široké škále obrazů a umožní segmentovat nové obrázky a generovat pro ně masky na základě uživatelského vstupu. Tento vstup či dotaz (prompt) může nabývat mnoha podob, od jednoho nebo více bodů v obraze, přes ohraničující obdélník vytyčující oblast s objektem, až po specifikaci objektu nebo oblasti určené k segmentaci pomocí textu.



Obrázek 2.2: Přehled modelu SAM. Zdroj: [38]

Dalším požadavkem na model je schopnost vypořádat se s nejednoznačným označením masky. Například pouhý jeden bod na obrázku s člověkem může označovat kus oblečení nebo přímo celou osobu. Model by měl být schopný pracovat s touto nejednoznačností a vrátit správnou masku pro alespoň jednu z možností.

Model SAM se skládá ze tří částí, které se vycházejí z modelů typu transformer [41, 42]. Tyto části zahrnují kodér obrázku, kodér vstupu specifikující požadovanou masku a dekodér masky. Hlavním úkolem kodéru obrázku je vytvořit obrazový embedding, který je dostatečně univerzální a nezávislý na konkrétní výstupní masce. Kodér vstupu generuje embedding pro popis masky zadané uživatelem. Tímto způsobem unifikuje celý proces dotazování, což umožňuje přidání dalších způsobů popisu masky bez nutnosti zásadních změn v zbytku modelu. Poslední částí je dekodér masky, jehož úkolem je rychle vygenerovat odpovídající masku na základě obrazového embeddingu a embeddingu dotazu. Autoři kladou vysoké nároky na rychlost predikce masky. Generování masky by mělo trvat zhruba 50 ms. Hlavní myšlenka spočívá v tom, že obrázek se zpracuje pouze jednou a následně se uživatel může opakovaně dotazovat, čímž se časově rozloží náročnost kódování obrazu. Znázornění modelu a jeho jednotlivých částí lze vidět na obrázku 2.2.

Článek je poměrně rozsáhlý a detailní, avšak zároveň dobře čitelný a srozumitelný. Na úvod představuje problém, který se pokouší řešit, popisuje architekturu modelu a metody získávání dat pro trénink. Autoři se zaměřují na zodpovědný přístup, a proto článek zahrnuje analýzu reprezentace obrázků z hlediska různých aspektů, jako je férové zastoupení společnosti, přičemž berou v úvahu pohlaví, rasu a věk. Dále jsou rozebrány různé zero-shot experimenty, podobných jako u modelu CLIP [43]. Tyto experimenty ukazují zřetelný potenciál modelu SAM v praxi. V příloze jsou detailní informace o implementaci modelu a dalších aspektech článku, které poskytují ucelený a komplexní pohled na dané téma.

2.4 LLFF

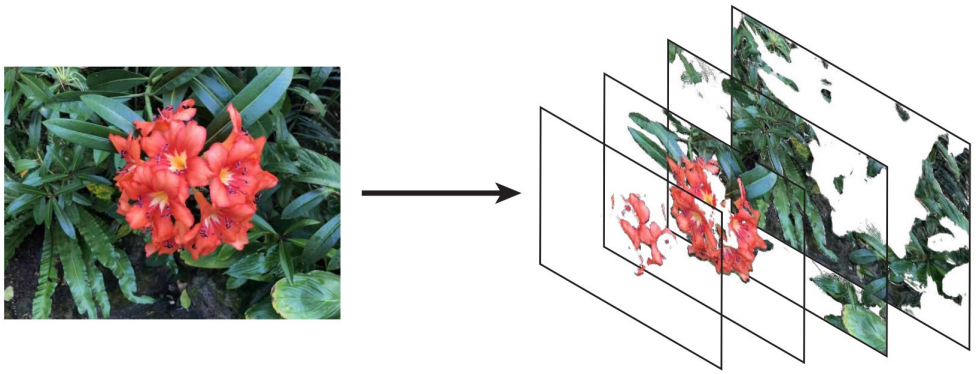
Článek „Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines“ (LLFF) [44] se zaměřuje na zachycení a následné renderování scén z reálného světa, jedná se tedy o IBR. Autoři představují jednoduchou a robustní metodu pro vykreslování nových pohledů scény, která byla zachycena pomocí několika snímků z mobilního fotoaparátu. Další výhodou představeného přístupu je jeho jednoduchost v generování nových pohledů. Po zpracování snímků jsou operace pro generování pohledů velmi jednoduché a proto je lze provádět v reálném čase. Pro účely demonstrace autoři připravili mobilní aplikaci, která uživateli pomáhá správně nasnímat scénu a následně mu umožňuje prohlédnout si nasnímanou scénu po zpracování snímků.

Prvním krokem v procesu zpracování snímků je odhad pozic kamery pro jednotlivé snímky. Autoři v článku využívají software COLMAP, který implementuje algoritmy pro rekonstrukci 3D scény z 2D snímků, které by se dali rozdělit na dvě části. První – „Structure from Motion“ (SfM) – se zabývá výpočtem pozic a paramatrů kamery ze snímků. Výstupem je navíc poměrně řídká reprezentace scény pomocí bodů. Druhá kategorie je „Multi-View Stereo“ (MVS), které bere na vstupu výstup SfM, zahustí reprezentaci body a vznikne tzv. „dense point cloud“. [45, 46]

Po získání informací o polohách kamer snímků je třeba snímky upravit a udělat z nich vícevrstvé obrázky tzv. MPI (anglicky MultiPlane Images). [47] Tyto obrázky jsou reprezentací dané scény pomocí několika obrazů, kde každý odpovídá určité hloubce / vzdálenosti od kamery. K tomu se využívá třídimenzionální konvoluční neuronová síť, díky které se může dynamicky měnit počet vrstev v jednom MPI. Obrázek 2.3 ukazuje možný vstup a výstup tohoto procesu.

Generování nových pohledů z předzpracovaných MPI spočívá kombinací několika MPI a aplikací perspektivních transformací. Nejdříve se na jednotlivé pohledy aplikuje perspektivní transformace a poté jsou sousední MPI míchány tak, že se informace z okolních pohledů doplňuje na místa, která byla v zákrytu pro původní kameru.

Kromě hlavních autorů, Bena Mildenhalla a Pratula Srinivasana z Kalifornské univerzity v Berkeley, se na tomto článku podíleli dva autoři z firmy Fyusion. Tato firma je zmíněna z důvodu svého zaměření a zejména služby Auto3D, která se snaží řešit podobný problém jako je zadání této práce. Tato služba se snaží řešit podobný problém jako je zadání této práce, avšak řešení problému, dle mého průzkumu, je rozdílné. Bohužel se mi nepodařilo zjistit jak přesně služby této firmy fungují.



Obrázek 2.3: Znázornění rozdělení obrazu na vrstvy do MPI pomocí 3D konvoluční neuronové sítě. Zdroj: [44]

2.5 NeRF

Neural radiance fields (NeRFs) představují inovativní přístup k reprezentaci a vykreslování 3D scén, který využívá sílu hlubokého učení k modelování složité struktury světla a geometrie ve scéně. „Zářivé pole“ (anglicky radiance field) je spojitá pěti-dimenzionální funkce, jejímž výstupem je zářivost vyzařovaná v každém směru (θ, ψ) a v každém v bodě (x, y, z) prostoru a hustota, která funguje jako průhlednost, řídí kolik zářivosti se nahromadí v paprsku procházejícím bodem (x, y, z). NeRF se snaží tuto funkci modelovat pomocí plně propojené neuronové sítě (MLP) na základě snímků pořízených z dané scény.

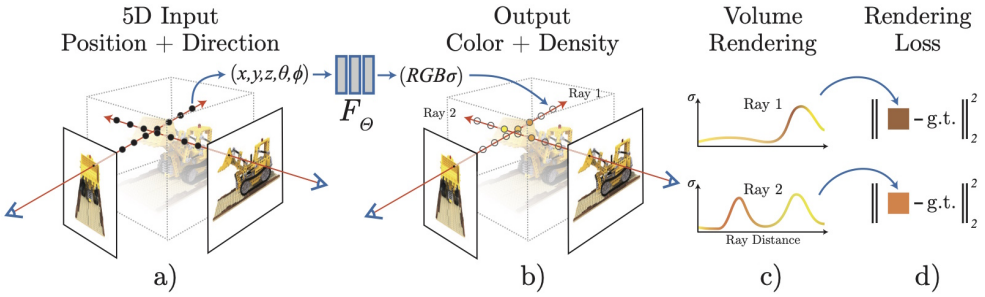
Při vykreslování NeRF z daného bodu se:

1. získá několik bodů v 3D prostoru podél paprsku z kamery pomocí metody ray marching,

2. použijí získané body spolu se směrem pohledu kamery jako vstup do neuronové sítě, které vygeneruje barvy a hustory,

3. na získaná data aplikují techniky vykreslování objemu (anglicky volume rendering) pro získání finálního obrazu.

Protože je tento proces přirozeně diferencovatelný, lze model optimalizovat pomocí metod gradientního sestupu pomocí minimalizace chyby mezi vstupními snímky a jím odpovídajícími vykreslenými obrazy. Minimalizací této chyby přes několik různých pohledů nutí síť modelovat a predikovat koherentní model scény pomocí přiřazování správných hustot a barev na místa, kde se ve skutečnosti vyskytují objekty nasnímané scény. Na obrázku 2.4 je vyobrazený celý proces vykreslení scény a výpočet chyby. [1]

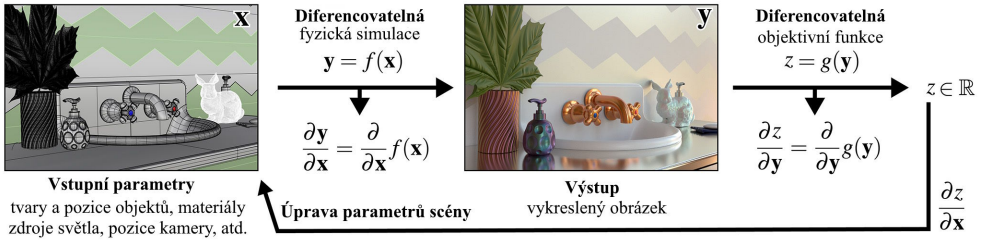


Obrázek 2.4: Ukázka vykreslování NeRF a diferencovatelnosti celého procesu. Obrázky a-c odpovídají bodům 1-3 seznamu pro vykreslení NeRF. Obrázek d je pouze ilustrací výpočtu chyby mezi vykreslenou barvou a skutečností (anglicky ground truth – g.t.) Zdroj: [1]

Tento článek vytvořil se stal inspirací nejen pro mnoho výzkumných skupin, ale také firem, které začaly prozkoumávat možnosti této technologie. Po zveřejnění tohoto článku vzniklo mnoho různých implementací navrženého algoritmu s různými specializacemi. Jedna velmi působivá a oceňovaná implementace přišla od výzkumného oddělení společnosti NVIDIA. Článek „Instant Neural Graphics Primitives with a Multiresolution Hash Encoding“ [48] získala ocenění nejlepšího článku na známé konferenci SIGGRAPH. Tato implementace urychlila učení modelu až o 3 řády. Zrychlení se jim povedlo díky specializovaným algoritmům pro rendering, upravení implementace neuronové sítě přímo pro hardware a vylepšenému kódování vstupních dat. Toto zrychlení umožňuje trénování modelu v řádu minut a následné vykreslování v reálném čase. Kromě výrazného zrychlení modelování i vykreslování autoři přišli i s možností reprezentovat jiné grafická primitiva jako jsou obrázky, SDF nebo objemy pomocí neuronových sítí. Tím se však tato práce nebude zabývat.

Kromě zmíněné implementace od společnosti NVIDIA by bylo vhodné upozornit na několik dalších variant odvozených z původního článku. Článek „NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections“ [49] se zabývá modelováním scén s různými světelnými podmínkami v rámci souboru snímků. Pro řešení tohoto problému autoři zavádějí kódování světelných podmínek, což také umožňuje měnit osvětlení v latentním prostoru během renderování. Díky tomu lze získat snímky stejné scény při různých světelných podmínkách, jako jsou například denní světlo, večerní záře či noční osvětlení.

Další zajímavý článek, který je velmi prakticky zaměřený, je „Block-NeRF: Scalable Large Scene Neural View Synthesis“ [50] od společnosti Waymo, která díky této technologie renderuje celé čtvrti měst pro učení algoritmů autonomních aut. Článek „Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields“ [51] od výzkumníků z Google se zaměřuje na problém anti-aliasingu při renderování v různých měřítkách a rozlišeních



Obrázek 2.5: Ilustrace diferencovatelného renderování. Zdroj: [55]

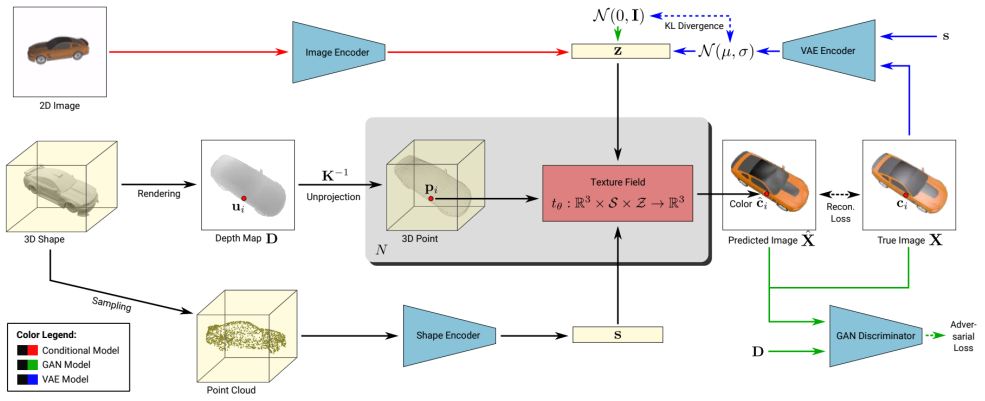
jedné scény. Rozdílná reprezentace „zářivých polí“, která nevyužívá neuronových sítí, je představena v článku „TensoRF: Tensorial Radiance Fields“ [52]. Zde je objem reprezentován pomocí 4D tensoru, pro který je hledán vhodný rozklad na několik tensorů nižších řádů. Dokonce vznikl i open-source projekt „nerfstudio“, který umožňuje použít několik různých implementací a metod NeRF. [53]

2.6 Mitsuba

Mitsuba 3 [54] je vědecky zaměřený renderovací software v Pythonu, který se zaměřuje na fyzikálně založené renderování. Dokáže simulovat polarizaci světla a počítat s barvami jako spektrem elektromagnetického záření místo pouhých složek RGB. Mitsuba implementuje několik různých druhů vykreslování založených na ray tracingu. Díky backendu Dr.Jit může vykreslování běžet buď na CPU i na GPU v závislosti na volbě uživatele. Další schopností tohoto softwaru je automatické derivování při výpočtu. Už tak dost rozsáhlou funkcionalitu lze obohatit pomocí tzv. pluginů.

Právě díky automatického výpočtu derivace a celkové diferencovatelnosti celého procesu vykreslování lze řešit problémy inverzní k renderingu pomocí techniky zvané „diferencovatelné renderování“. Ta popisuje vykreslovací algoritmus jako funkci $f(\boldsymbol{x})$, která převádí vstup $\boldsymbol{x}$ – popis scény – na výstup $\boldsymbol{y}$ – vykreslený obraz. Funkce f je pak derivována pro získání $\frac{d\boldsymbol{x}}{d\boldsymbol{y}}$, čímž se získá aproximace prvního řádu jak pozměnit výstup $\boldsymbol{y}$ pomocí změny vstupních parametrů $\boldsymbol{x}$. Dohromady s diferencovatelnou objektivní funkcí $g(\boldsymbol{y})$ lze použít optimalizační algoritmy využívající gradient k nalezení parametrů scény, které vylepší výsledek objektivní funkce. Lze tedy „end-to-end“ trénovat reprezentaci scény na základě referenčních snímků. [55]

Pokud je prakticky možné scénu reprezentovat pomocí několika parametrů, parametrizovat ji, lze využít algoritmů jako je stochastický gradientní sestup nebo Adam [28] pro nalezení vhodných hodnot. Jednoduchý příklad je odhad orientace objektu na obrázku, kde scénu tvoří objekt a parametry tvoří orientace v prostoru – 4 čísla pro rotaci ve 3D prostoru. Na obrázku 2.5 je znázorněný průběh optimalizace parametrů scény. Není třeba omezovat se



Obrázek 2.6: Přehled modelu pro reprezentaci a generování textury.
Zdroj: [56]

pouze na základní parametry jako polohu, orientaci nebo barvu, ale je možné hledat reprezentaci objemu podobně jako u NeRF nebo optimalizovat povrch skleněné desky pro získání konkrétního tvaru a vzhledu kaustiky při nasvícení. Tento výpočet je ovšem velmi výpočetně náročný, protože už samotné vykreslování je poměrně složitý proces.

2.7 Ostatní práce

Velmi zajímavý článek je „Texture Fields: Learning Texture Representations in Function Space“ [56], který představuje novou techniku reprezentace a generování textur pro 3D objekty. Textury se generují z 3D modelu a latentní reprezentace vzhledu, která může vzniknout například z pouze jednoho snímku modelu v reálném světě.

Článek je velmi zajímavý i z technické stránky, protože se jedná ensemble několika různých neuronových sítí. Reprezentace textury je založena na úpravě architektury konvoluční neuronové sítě ResNet [3]. Při generování nové textury je použit variační autoenkodér [57] a upravená síť GAN [9]. Obrázek 2.6 zobrazuje propojení celého modelu. Výsledné textury při použití pouze jednoho snímku jsou výborné, ale nutnost přesného 3D modelu pro generování textury znemožňuje použití této techniky pro řešení v této práci.

Ekvalizace histogramu obrazu je poměrně populární, jednoduchá a hlavně efektivní technika pro úpravu obrazu. [58] Ekvalizace histogramu je proces, při kterém se mění intenzita pixelů v obraze, tak aby výsledný histogram co nejlépe odpovídal rovnoměrnému rozdělení. V článku [59] autoři zobecňují tuto metodu zvýšení kontrastu pomocí parametrizace.

Podobná technika ekvalizaci histogramu je specifikace histogramu (anglicky histogram matching). U této techniky se neupravuje histogram tak,

aby měl rovnoměrné rozdělení, ale tak aby odpovídal histogramu jiného obrazu. Tato technika se používá na sjednocení částí obrazu nebo dvou obrazů. V článku [60] se autoři autoři vydali opačným směrem a použili tuto techniku pro augmentaci dat pro nesupervizované učení.

Kapitola 3

Analýza

V této kapitole jsou rozebrány představené technologie a metody pro řešení této práce. Jednotlivé metody jsou přiblíženy a doplněny o případné výhody a nevýhody. Úplně prvním krokem pro modelování vozidla, ještě před jakýmkoliv zpracováním, je jeho nasnímání. Kvalita provedení tohoto kroku určuje kvalitu následných částí a celkového výsledku.

Nesprávně provedené snímání scény může významně zkomplikovat či dokonce znemožnit další zpracování a následné modelování vozidla. Je proto důležité prozkoumat vhodné metody pro fotografování vozidla. Při snímání je nutné nastavit vhodné parametry, jako například clonové číslo, ohniskovou vzdálenost, citlivost ISO či expoziční dobu. Další možností snímání scény je nefotit jednotlivé snímky, ale natáčet video, ze kterého se následně získají snímky v daném intervalu. Tato možnost je sice jednodušší na zisk obrazu, ale neumožňuje jednoduchou kontrolu kvality snímků přímo při snímání. Nevhodně vybrané snímky mohou být rozmazané a neostré.

Nastavení parametrů by mělo být stejné pro celý soubor snímků jednoho vozidla. Je tedy důležité nastavit parametry tak, aby snímky nebyly příliš podexponované nebo přeexponované. Správné nastavení parametrů se bude lišit v závislosti na snímaném objektu, okolním prostředí a konkrétním fotoaparátu. Nicméně existuje několik obecných pravidel pro adekvátní snímání. Clonové číslo by mělo být vysoké, což zajistí větší hloubku ostrosti, tedy ostré popředí i pozadí. Pro extrakci příznaků a následný odhad pozice kamery je nežádoucí rozmazané pozadí, ačkoliv snímky mohou esteticky vypadat lépe.

Dalším parametrem pro volbu je ohnisková vzdálenost. Nastavení ohniskové vzdálenosti úzce souvisí s volbou vhodné vzdálenosti kamery od vozidla při snímání. Na obrázku 3.1 je zobrazen vliv změny ohniskové vzdálenosti (a vzdálenosti kamery od objektu) při zachování zdánlivé velikosti objektu v obraze na proporce a celkový vzhled objektu. Lze si všimnout, že obrázek d) vypadá protaženě, zatímco na obrázku a) jsou proporce přirozenější. Tento jev je způsoben různými poměry vzdáleností přední a zadní části vozu od kamery. Obrázek a) byl focen s ohniskovou vzdáleností 75 mm ze vzdálenosti 15 metrů

Obrázek 3.1: Ukázka vlivu různých ohniskových vzdáleností na změnu obrazu. Ohniskové vzdálenosti: a) 75 mm b) 50 mm c) 35 mm d) 24 mm

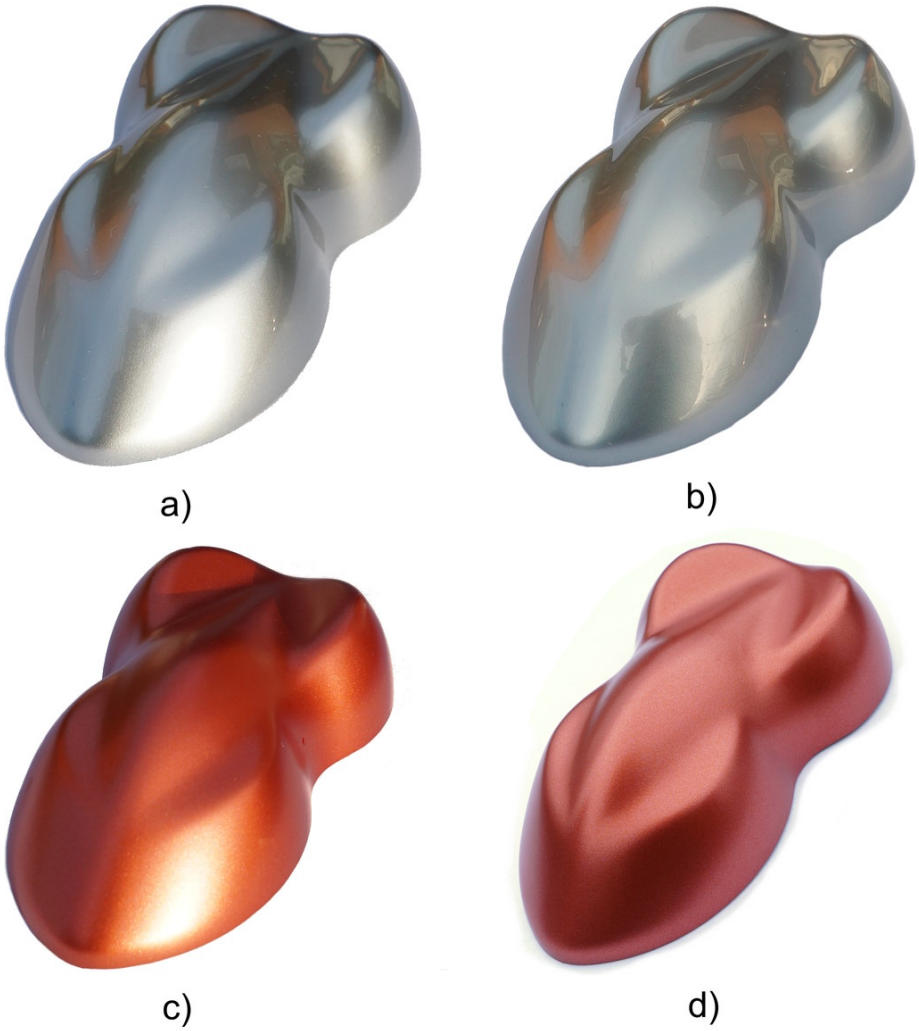
a obrázek d) s ohniskovou vzdáleností 24 mm ze vzdálenosti 5 metrů. To znamená, že na obrázku d) je zadní část téměř 2× dále od kamery než přední část, zatímco u obrázku a) je tento koeficient přibližně 1,2. Problém nastavení velké ohniskové vzdálenosti je však potřeba velkého volného prostoru pro fotografování.

Citlivost ISO by měla být nízká, protože zvyšuje šum v obraze. Je třeba dbát na to, aby intenzita odlesků nebyla příliš silná a naopak některé části vozu nebyly příliš tmavé. Jelikož snímaný vůz je statický, expoziční čas závisí pouze na expozici snímku a schopnosti udržet kameru (nebo ji umístit na stativ) tak, aby snímky nebyly rozmazané. Počet a pozice snímků vozidla jsou také důležité pro správné modelování, což je zkoumáno v následujících kapitolách. Dalším aspektem snímání je osvětlení, které je rozebráno v další části.

3.1 Homogenizace osvětlení

Homogenizace obrazu je proces, při kterém se upravuje nebo transformuje vzhled obrazu takovým způsobem, aby byl jednotnější nebo konzistentnější. Používá se obvykle za účelem snížení rozdílů v osvětlení, barvě nebo textuře. Vhodná homogenizace obrazu může pomoci zlepšit kvalitu následného zpracování obrazu, jako je hledání příznaků, rozpoznávání objektů a segmentace obrazu. V této práci jde hlavně o vylepšení a sjednocení osvětlení v obraze. Výsledné řešení má realisticky zachycovat vzhled vozidla tudíž změna odstínu barvy není žádaná.

Obrázek 3.2 zobrazuje rozdíl ve vzhledu laku při přímém a nepřímém osvětlení. Na obrázku jsou snímky modelu „speedshape“, na které je nanesena barva. Tyto modely se používají jako ukázky laků, protože připomínají tvar vozidla a umožňují posoudit vzhled v různých úhlech pohledů najednou. To je například výhodné u speciálních pigmentů, které mění barvu na základě úhlu pohledu a osvětlení. Levý sloupec obrázku jsou snímky pod přímém osvětlením a pravý sloupec jsou snímky pod nepřímém osvětlení. U přímého osvětlení si lze všimnout výrazných odlesků. Zároveň tyto snímky obsahují vyšší rozdíl mezi nejsvětlejším a nejtmavším. Pro eliminaci odlesků, které



Obrázek 3.2: Porovnání vzhledu autolaků pod přímém a nepřímém osvětlením. a,b) chromový pigment c,d) oranžový pigment

se mění v závislosti na úhlu pohledu je tedy vhodné pořizovat snímky pod rozptýleném světle, v otevřeném prostoru při zatažené obloze.

Existuje mnoho různých typů a metod homogenizace obrazu. Seznam níže ilustruje různorodost přístupů k této problematice.

- Ekvalizace histogramu. Tato metoda je velmi jednoduchá, avšak efektivní. Ekvalizace histogramu se snaží změnit rozdělení intenzit v obraze tak, aby bylo stejné jako uniformní rozdělení. Touto transformací se sjednotí jas a zvýší kontrast obrazu. Pomáhá snížit vliv nevhodných světelných podmínek a zvyšuje viditelnost detailů v obraze. Výhodou

této transformace obrazu je absence parametrů, které je třeba nastavovat nebo měnit. [7]

- Adaptivní ekvalizace histogramu. Adaptivita spočívá v ekvalizaci histogramu v rámci posuvného okna, namísto ekvalizace jednoho globálního histogramu pro celý obraz. Nová hodnota intenzity v pixelu je upravena na základě lokálního histogramu, který je pro každý pixel jiný, čímž se zvyšuje výpočetní náročnost. Parametrem této transformace je velikost posuvného okna. Adaptivní ekvalizace histogramu zvyšuje lokální kontrast, což v některých případech může vypadat nepřirozeně. [59]

- Metody na základě teorie Retinex. [33] Tyto metody se snaží rozdělit obraz na část s iluminancí a odrazivostí. Metody, jako například „single-scale retinex“, „multi-scale retinex“ [61] a již představené NPEA [32] a LIME [34], se snaží odhadnout složku s osvětlením, kterou následně upravují za účelem získání jednotného osvětlení napříč obrazu.

- Homomorfní filtrace. Tato metoda filtruje obraz ve frekvenčním prostoru obrazu a to tak, že tlumí vliv nízkofrekvenčních částí spektra. Tímto lze snížit vliv nerovnoměrného osvětlení napříč obrazem. Homomorfní transformace lze aplikovat i pouze na iluminanci. Jelikož lze obraz rozdělit na součin dvou složek – iluminance a reflektance je třeba obraz před převodem do frekvenčního spektra upravit. Díky aplikaci funkce logaritmus na obraz lze využít využít linearity Fourierovy transformace a tedy použít filtr pouze na složku s iluminancí. [7]

- Metody hlubokého učení. Využití neuronových sítí si našlo své uplatnění i v oblasti vylepšení a homogenizace obrazu. Model z článku [62] staví na teorii Retinex a používají architekturu konvoluční neuronové sítě pro získání modelu, který zlepšuje nízké světelné podmínky v obraze. Autoři z článku [63] řeší stejný problém navíc s odstraněním šumu. Jako řešení navrhli model založený architektuře autoenkodér (AE). Tyto modely se mohou naučit velmi komplexní a nelineární transformace, které mohou efektivně řešit homogenizaci obrazu i v případě velmi složitých podmínek.

Různé metody homogenizace obrazu zahrnují úpravu nebo transformaci obrazu za účelem vytvoření jednotnějšího a konzistentnějšího vzhledu. Kromě představených metod existuje i mnoho dalších jako například gama transformace či roztažení kontrastu [7], avšak u těch není triviální zvolit vhodné hodnoty parametrů.

Experimentování s jednotlivými metodami homogenizace obrazu ukázalo, že základní metody (single-scale a multi-scale retinex) založené teorii Retinex nejsou vhodné pro využití v této práci, jelikož nelze jednoduše najít optimální hranici mezi slabým vylepšením obrazu a zřetelnými artefakty. Podobně dopadlo i homomorfní filtrování, které dokáže odstranit nerovnoměrné osvětlení,

ale při příliš vysokém nastavení propustnosti se začínají objevovat artefakty na hranách objektů.

3.1.1 Operace s histogramem

Techniky založené na operacích s histogramem jsou velmi oblíbené díky své schopnosti jednoduše a účinně upravovat a vylepšovat různé základní charakteristiky obrazu jako je jas, kontrast a celková vizuální kvalita. Manipulací s histogramem mohou tyto transformace přerozdělit intenzitu pixelů a dosáhnout tak požadovaného vizuálního zlepšení. Tyto metody jsou výpočetně efektivní a relativně jednoduché na implementaci, což je činí jako vhodný základ a „benchmark“ při vyvíjení nových, složitějších algoritmů.

Základní metodou pro homogenizaci osvětlení je již zmiňovaná ekvalizace histogramu. Ekvalizace histogramu lze definovat jako transformaci T, která přiřadí pixelu s intenzitou r hodnotu

$$T(r) = L \sum_{s=0}^{r} p(s),$$

kde L značí maximální intenzitu (často 255), M a N značí výšku a šířku obrazu a p je normalizovaný histogram obrazu. Tato definice předpokládá obraz s jedním kanálem, například šedotónový obraz. Pro barevné obrázky je třeba tuto transformaci nějak upravit nebo vhodně zvolit vstupní data.

Jedna z možností je ekvalizovat každý barevný kanál RGB obrazu samostatně, avšak tento způsob může výrazně změnit odstín barvy v daném pixelu a posílit méně barvy zastoupené barvy v obraze, což v některých případech vypadá nepřirozeně. V článku [64] autoři navrhují ekvalizaci pouze poslední složku obrazu v barevném prostoru HSV (Hue Saturation Value). Tento způsob však produkuje obrázky, které jsou v některých oblastech příliš tmavé, což je způsobeno tím jak se mění barva při změně složky „value“. Experimenty ukázaly, že převedením obrazu do barevného prostoru CIELAB a následná ekvalizace první složky „lightness“ – světelnosti – se jeví jako vhodný kandidát pro homogenizaci snímků. Tato varianta ekvalizace histogramu produkuje nejpřirozeněji vypadající snímky z porovnávaných variant.

Adaptivní verze ekvalizace histogramu, například metoda CLAHE [58], zvyšují lokální kontrast, což u vozidel, které obsahují velké, často jednobarevné plochy, způsobuje nepřirozený vzhled. Malé nedokonalosti, například v podobě špatně umyté karosérie, se zvýrazňují a mění vzhled vozu. Vliv nežádoucího zvýraznění lze potlačit vhodnou volbou parametrů, avšak ty se musí ladit zvlášť a tím pádem brání jednoduchému automatickému použití.

Technika známá jako specifikace histogramu (anglicky histogram specification nebo také histogram matching) je v jistém smyslu zobecněním ekvalizace histogramu. Cílem není získat rovnoměrné rozdělení intenzit pixelu, ale obraz transformovat tak, aby výsledný histogram byl podobný referenčnímu

obrazu. Specifikace histogramu lze počítat jako nalezení transformace ekvalizace histogramu pro referenční resp. cílový obraz, $G(r)$ resp. $R(r)$, nalezení inverze G a následným složením $T(r) = G^{-1}(R(r))$. Ve spojitém případě je hledání inverze netriviální úloha avšak v diskrétním a omezeném případě, jako jsou intenzity pixelů digitálního obrazu, lze inverzi spočítat pomocí vyhodnocení funkce v každém bodě, zaokrouhlení hodnot a jejich následné uložení v tabulce. [7]

Podobně jako ekvalizace histogramu, i specifikace histogramu má své adaptivní varianty. Ty jsou vhodné například k sjednocení různých regionů v rámci jednoho obrazu. Nejdříve se zvolí referenční oblast a následně se ostatní oblasti transformují pomocí specifikace histogramu. Tato technika se používá například při snímání a měření vlastností různých materiálů.

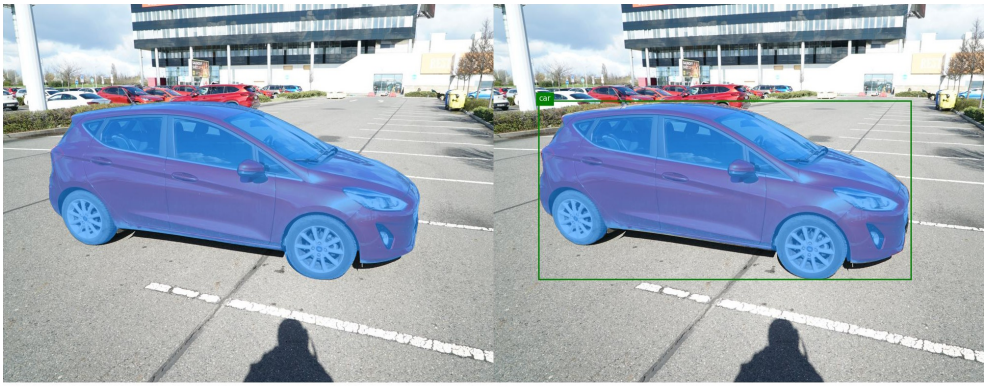
3.2 Segmentace vozidla

Segmentace vozidla z nasnímaných obrázků představuje netriviální úlohu, kterou nelze řešit pomocí jednoduchých tradičních technik pro segmentaci obrazu. Vzhledem k rozmanitosti vozidel různých velikostí, tvarů a barev, světlých i tmavých, oblých i hranatých, se tato úloha stává velmi náročnou pro tradiční metody segmentace. Navíc různé světelné podmínky komplikují situaci ještě více.

Metody segmentace založené na hlubokém učení se ukázaly být robustnější a efektivnější při segmentaci vozidel ve srovnání s tradičními metodami. Tyto metody dokážou lépe zvládnout výzvy spojené se segmentací vozidel, jako je proměnlivost vzhledu, okluze a světelné podmínky, což vede k přesnějším a robustnějším výsledkům segmentace. Automobily a další vozidla jsou běžně zahrnuty v datasetech pro trénování neuronových modelů pro segmentaci, jako je například COCO [65] nebo Open Images [66]. Díky tomu je možné využít předtrénované modely bez dalších úprav.

Při použití předtrénované neuronové sítě U2-Net není třeba specifikovat druh objektu pro segmentaci. Jedná se o model natrénovaný na úloze, ve které je cílem oddělit význačný (anglicky salient) objekt od pozadí obrazu. Tuto úlohu rozvádí autoři U2-Net v článku [67], kde představují nový dataset a model pro tzv. dichotomní segmentaci. Při použití této sítě je tedy třeba dbát na to, aby se na snímcích vyskytovalo pouze dané vozidlo bez dalších v pozadí objektů. Velká výhoda této segmentace je její plná automatizace.

Model SAM vyžaduje vstup ve formě dotazu, který určuje požadovanou výstupní masku. Autoři článku [38] nepředstavují pouze jeden typ dotazu, ale několik, které se před zpracováním dekodérem masky zakódují, čímž se unifikuje reprezentace dotazu. Navrhují různé typy vstupů, jako například jeden nebo více bodů, které specifikují místa, kde maska má či nemá být. Další možností je ohraničující obdélník, který vymezuje oblast s požadovanou



a) b)

Obrázek 3.3: Porovnání masky vozidla při segmentaci. a) Maska vytvořená sítí U2-Net. b) Detekované vozidlo pomocí modelu YOLO a výsledná maska z modelu SAM.

maskou. Zajímavým druhem vstupu je čistý text, který popisuje objekt nebo oblast pro segmentaci.

Bohužel oficiální implementace modelu SAM, v době psaní této práce, nepodporuje text jako možný dotaz pro masku. Z tohoto důvodu je nutné při použití využít ostatních možností dotazů, jako jsou body či ohraničující obdélník. Pro získání ohraničujícího obdélníku lze použít jiný model, například některou z verzí modelu YOLO [4]. Výstup z YOLO modelu je potřeba filtrovat pro odstranění masek jiných objektů než požadovaného vozidla. Následně lze získaný ohraničující obdélník použít jako vstup pro SAM.

Na obrázku 3.3 je ilustrace generování masky. Obrázek a) je maska vygenerovaná přímo z U2-Net pomocí balíčku rembg [68]. Obrázek b) ilustruje postup při generování masky při použití modelu SAM. Nejprve je třeba získat ohraničující obdélník požadovaného vozidla a následně tento obdélník dát jako vstup do SAM. Jak je vidět z obrázku 3.3, masky jsou si velmi podobné, IoU skóre u tohoto snímku vyšlo 0.976 a rozdíly jsou podél okraje vozidla.

3.3 Modelování vozidla

Velmi důležitou částí této práce je modelování nasnímaného vozidla za účelem vytváření 3D vizualizací nebo videí, která zobrazují vozidlo z úhlů, ze kterých nebylo původně nasnímáno. Tuto úlohu lze klasifikovat jako „image-based rendering“ [21] nebo „image-based modeling“ [69], což znamená syntézu obrazu či modelu na základě snímků scény – v případě této práce scény s vozidlem.

Tradiční přístup k řešení takové úlohy spočívá v extrakci geometrického 3D modelu objektu na základě extrakce příznaků a jejich lokalizace ve 3D prostoru. Fotogrammetrie je vědní obor, který se zaměřuje na získávání infor-

mací o reálných objektech ze snímků, jako jsou barva, tvar nebo vzdálenosti mezi objekty. Extrakce 3D modelu je jedna z úloh, kterými se fotogrammetrie zabývá.

Často používaný postup při vytváření 3D modelu se skládá z několika částí. Nejprve se ve snímcích detekují obrazové příznaky, často jsou to příznaky SIFT. Poté se snímky a příznaky porovnají tak, aby byly nalezeny korespondující příznaky napříč snímky. Následně se hledají 3D pozice kamer snímků a nalezených příznaků pomocí algoritmů nazývaných „Structure-from-Motion" (SfM). Tím vznikne množina bodů ve 3D prostoru, která se poté zahustí a použije pro rekonstrukci 3D modelu. Nakonec se získá textura pro daný model. Konkrétní detaily se liší u různých přístupů a softwarů, ale základní kostra postupu zůstává stejná.

Tato práce se však zabývá i jiným způsobem modelování. Namísto explicitní reprezentace vozidla pomocí 3D modelu se u techniky NeRF používá implicitní reprezentace pomocí neuronové sítě nebo pomocí vícevrstvých obrazů u metody LLFF. Odstranění požadavku na explicitní rekonstrukci 3D modelu přináší jisté výhody, jako jsou například nižší nároky na kvalitu snímků a výpočetní výkon. Technologie NeRF přináší mnoho zajímavých možností, kterých je složité dosáhnout pomocí tradičních přístupů.

3.3.1 Fotogrametrie

Fotogrammetrie zahrnuje metody měření a interpretace obrazu, jejichž cílem je odvodit tvar a polohu objektu z jedné nebo více fotografií tohoto objektu. Fotogrammetrické metody lze v zásadě použít v jakékoli situaci, kdy lze měřený objekt fotograficky zaznamenat. Primárním účelem fotogrammetrického měření je trojrozměrná rekonstrukce objektu v digitální podobě (souřadnice a odvozené geometrické prvky) nebo v grafické podobě (obrázky, nákresy, mapy). Fotografie nebo obrázek představují zásobu informací, ke kterým lze kdykoli znovu přistupovat. [70]

Vytvářením 3D modelů částí aut se zabýval článek [71], ve kterém se autoři snažili posoudit využitelnost fotogrammetrie pro digitalizaci dílů, konkrétně karosérie starých aut. Při sběru dat ovšem využívali 3D skeneru. Modelování vozidla pomocí fotogrammetrie z klasických snímků má jistá úskalí na která je si dát pozor. Níže je seznam několika možných problémů, které mohou nastat.

1. Nedostatečný překryv snímků. Jako vhodný překryv snímků pro přesnou rekonstrukci modelu se uvádí být mezi 60 a 80 procenty. [45, 72] Tento poměrně vysoký překryv však zvyšuje redundanci dat a s tím i čas potřebný na celý výpočet.

2. Nekonzistentní osvětlení. Špatné nebo nekonzistentní osvětlení může způsobovat problémy při rekonstrukci. Odlesky a stíny mění obrazové příznaky a tím pádem je nelze správně porovnat s příznaky z ostatních snímků. Při přímém osvětlení se může barva vozidla měnit v závislosti na

úhlu pohledu, což není vhodné pro výpočet textury. Vhodné je pořizovat snímky při difuzním osvětlení. [73]

3. Proměnlivé prostředí. Ačkoliv se snímané vozidlo se mezi snímky nejspíše nebude měnit, je třeba dát si pozor na jeho okolí. Změny v pozadí mohou zmást algoritmus na porovnání obrazových příznaků. [73]

4. Lesklý lak. I když bude při snímání vhodné, rozptýlené, osvětlení, samotný lak může vytvářet problémy podobné jako v bodě 2. Matný lak pro auta existuje, ale není zdaleka tak rozšířen.

5. Hladký povrch. Fotogrammetrie silně závisí na správné detekci obrazových příznaků. Auta jsou však hladká a jejich lak je často uniformní bez jakékoliv textury. Tím se snižuje počet spolehlivých příznaků, které lze identifikovat na vozidle.

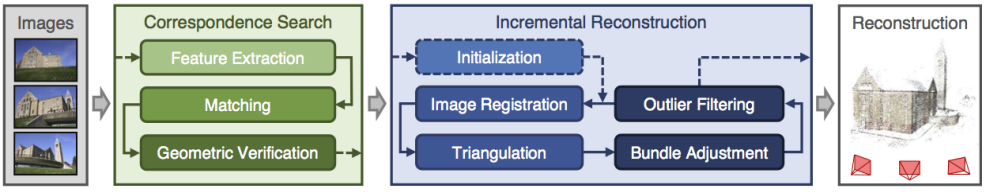
6. Průsvitnost. Okna a jiné průhledné či průsvitné části vozidla je složité správně modelovat. Zvláště okna dělají velké problémy, protože se jedná o velkou plochu, která pod některými úhly světlo propouští a pod jinými úhly ho odráží.

Velikost snímaných obrázků je další aspekt, který je třeba řešit. Nelze říct, že čím vyšší rozlišení snímku, tím lepší bude výsledek. V článku [74], zabývající se fotogrammetrií v topografii, vycházejí z vlastních zkušeností a tvrdí, že snímky s vyšším rozlišením než 12 MP sice přinášejí více detailů, ale jejich zpracování trvá příliš dlouho a nakonec musejí být zmenšeny.

3.3.2 LLFF

Metoda „Local Light Field Fusion“ (LLFF) [44] se může jevit jako vhodný kandidát pro modelování a generování nových pohledů založených na snímcích vozidla. Skutečně, tato metoda poskytuje vysokou úroveň kvality ve formě realisticky vypadajících výstupů, avšak pouze v případě, že se pohled kamery pohybuje v jedné rovině a úhel pohledu se liší pouze o několik stupňů. Nicméně, LLFF čelí problému s neschopností extrapolace dat. Nové pohledy vznikají pomocí interpolace mezi několika, ve výchozím nastavení pěti, nejbližších MPI. Realističnost výstupu je však narušena ve chvíli, kdy se nový pohled dostane mimo snímanou oblast. Ve výsledném obrazu se objevují prázdné mezery a do popředí se dostává pozadí z jiných pohledů.

Kromě problému s rušivě vypadající extrapolací, metoda LLFF vykazuje další nedostatek, který se projevuje při pokusu o zachycení 360° scény. Tento problém popisuje sám Mildenhall (autor článku LLFF) v porovnání NeRF z článku [1] s ostatními přístupy k reprezentaci scény. LLFF využívá ke generování MPI ze snímku předtrénovanou neuronovou síť, která se, na rozdíl od neuronové sítě použité v NeRF, nemění či neadaptuje pro danou scénu. V důsledku toho vznikají nekonzistence mezi jednotlivými pohledy, což se



Obrázek 3.4: Schéma průběhu inkrementálního algoritmu Structure from Motion (SfM). Zdroj: [45]

při generování nových pohledů projevuje jako artefakty v podobě skokových změn. Tato skutečnost snižuje celkovou kvalitu a přesvědčivost výstupů metody LLFF, což může mít negativní dopad na její praktické využití.

3.3.3 NeRF

Technologie NeRF reprezentuje scénu jako objem, zářivé pole (anglicky radiance field), které se při vykreslování vzorkuje podél paprsku z kamery. Na daných pozicích počítá barvu a hustotu objemu pomocí neuronové sítě. Scéna je tak reprezentována natrénovaným modelem neuronové sítě. Trénování modelu probíhá ze snímků scény, které jsou obohaceny o pozice a parametry kamery. Na rozdíl od klasických úloh jako je klasifikace nebo regrese, funguje jeden model pouze pro jednu scénu. Častý požadavek na modely je generalizace. U NeRF model negeneralizuje napříč scénami, ale v rámci scény pro úhly pohledů, které nebyly zahrnuty v trénovacím datasetu snímků. Jednomu modelu odpovídá právě jedna scéna.

V této práci je využita implementace NeRF pojmenovaná Instant-NGP [48, 75], která rozšiřuje myšlenku reprezentace 3D scény pomocí neuronové sítě i na další grafická primitiva jako je například až gigapixelový obraz nebo „signed distance function“ (SDF) [18]. Kromě tohoto rozšíření přináší i řádové zrychlení trénování modelu. To je možné díky optimalizaci architektury neuronové sítě pro grafické karty, přizpůsobení algoritmu pro vykreslování a speciální vstupní kódování pro neuronovou síť. Díky těchto optimalizacím je možné natrénovat model pomocí jedné grafické karty za několik málo minut oproti desítkám minut až hodinám u jiných implementací. Poměrně nízké nároky a rychlé trénování činí tuto implementaci vhodnou pro použití v této práci.

Autoři Instant-NGP [75] popsali vhodné podmínky pro použití technologie NeRF a jejich implementace a případná úskalí, na která je třeba dát si pozor. Vstupem pro trénování modelu je kromě snímků i soubor *transforms.json*, který obsahuje parametry kamery a pro každý snímek matici pozici dané kamery vyjádřenou pomocí matice velikosti 4×4. Jeden soubor *transforms.json* odpovídá jedné stejné kameře, která nasnímala dané obrazy.

Pro odhad pozice kamer používají, stejně jako například LLFF, software COLMAP [45, 46]. Repozitář Instant-NGP obsahuje skript, který pomocí jed-

noho příkazu několikrát spustí COLMAP s různými parametry za účelem hledání obrazových příznaků, jejich porovnání napříč snímky a nakonec odhadu pozic kamer pro jednotlivé snímky.

Software COLMAP je použit pro svou část Structure-from-Motion (SfM) [45]. Je výhodné rozumět, jak tento software funguje, protože správná změna parametrů může vést k lepším, nebo v některých případech vůbec nějakým, výsledkům. Ilustrace průběhu inkrementálního algoritmu SfM je zobrazena na obrázku 3.4. Nejdříve jsou z obrázků získány obrazové příznaky. Tyto příznaky jsou následně porovnány mezi snímky. Lze zvolit strategii pro výběr snímků pro porovnání (anglicky matching), což může výrazně ovlivnit výsledek. Po nalezení párů obrázků s korespondujícími příznaky se ověří, zda snímky obsahují společnou část scény. Následně se spustí inkrementální proces rekonstrukce pozic kamer snímků. Algoritmus se snaží postupně přidávat a upravovat pozice kamer ve scéně na základě nalezených a porovnaných příznaků. Výstupem jsou pozice kamer a body scény v 3D prostoru. Při špatně nasnímané scéně, například s nízkým překryvem, nemusí algoritmus nalézt správné pozice kamer nebo je vůbec nemusí nalézt. Toto je důležité brát v potaz při použití COLMAP pro účely 3D modelování vozidel či jiných objektů. Správná volba parametrů a strategie může vést k lepším výsledkům a přesnější rekonstrukci 3D modelu.

Použití se může zdát jednoduché, ale jak je napsáno v dokumentaci Instant-NGP, jejich implementace je náchylná na nepřesné odhady pozice kamery a pro dosažení dobrého výsledku je třeba získat tyto odhady co nejpřesnější. Pro získání dostatečně dobrého výsledku je třeba správně zvolit parametry pro COLMAP, přestože je většina nastavena předem.

Spolu s nepřesnými odhady pozic kamery jsou rozmazané snímky dalším faktorem, který snižuje kvalitu výstupu. Špatně zaostřené snímky, snímky s nízkou hloubkou ostrosti nebo rozmazané snímky z důvodu pohybu fotoaparátu (motion blur) snižují schopnost správně detekovat, porovnat a registrovat obrazové příznaky, což vede ke špatnému odhadu pozice kamery. Snímky by měli být ostré a s dostatečným překryvem.

Pokud se spustí trénování modelu s špatnými pozicemi nebo parametry kamery, pak model nezkonverguje a zdegeneruje do objemu plného šumu, který jen v pohledech, které byly na vstupu, generuje ucházející obrázky. Model tedy není schopen generalizovat a nereprezentuje skutečnou scénu. Autoři Instant-NGP navrhují jednoduchou heuristiku pro odhad kvality vstupních dat. Pokud model nezačne konvergovat během prvních 20 vteřin trénování, je nepravděpodobné, že se výsledek zlepší s přibývajícím časem. Zároveň navrhují, že učení by nemělo trvat řádově déle než několik minut.

Schopnost zachytit složité tvary a interakci světla s okolím jako je například odraz a lom světla se dá označit za jednu z předností technologie NeRF. Díky této schopnosti reprezentovat jevy závislé na pohledu je NeRF zvláště účinné při vytváření konzistentních a vysoce kvalitních a fotorealistických obrazů z různých úhlů pohledu.

Veškeré interakce a změny v závislosti na pohledu jsou však kódovány v rámci neuronové sítě bez možnosti jejich separace a změny. V článku [49] se snaží oddělit světelné podmínky pro danou scénu do svého vlastního embeddingu. Dále pak v článku [76] se zaměřují na umožnění do scény vkládat nové objekty tak, aby vypadaly realisticky a interagovaly se scénou okolo nich, například vrhaly stíny. Technologie z tohoto článku je součástí „Neural Reconstruction Engine“ společnost NVIDIA, který umožňuje rekonstruovat a následně měnit reálné prostředí pro účely simulace autonomního řízení.

3.3.4 Porovnání

Ačkoliv v této práci je použita fotogrammetrie a technologie NeRF k řešení stejného problému, jsou to dva rozdílné přístupy k jeho řešení. Vzhledem k tomu, že oba přístupy v této práci používají některou implementaci algoritmu SfM pro získání pozic kamer, lze tuto první část označit za shodnou. Avšak následné modelování je už rozdílné.

První rozdíl je v přístupu. Technika NeRF používá metody strojového učení k natrénování neuronové sítě. Tato síť se učí reprezentovat spojitý 3D prostor scény. Vstupem je bod v prostoru a úhel pohledu a výstupem je barva a hustota v daném bodě. Techniky fotogrammetrie používají tradičnější přístup v podobně analyzování a extrakce informace pomocí algoritmů. Tyto algoritmy hledají společné příznaky na více snímcích, ze kterých pak vytvářejí 3D rekonstrukci scény. Fotogrammetrie využívá texturní informace objektů a má problém s rekonstrukcí objektů, které obsahují málo texturní informace [73] (lak vozidla) nebo jsou průhledné/průsvitné (okna vozidla).

Druhý významný rozdíl je ve výstupu. Výstupem u NeRF je neuronový model reprezentující danou scénu jako objem, zářivé pole. Tato reprezentace umožňuje generovat realisticky vypadající snímky z dané scény pro nové pohledy. Výstupem fotogrammetrického softwaru může být množina bodů v 3D prostoru (point cloud) nebo přímo 3D model scény spolu s texturami. Tento model je možné dále upravovat. NeRF neumožňuje přímé získání 3D modelu, i když lze použít techniku zvanou „marching cubes“ [77]. Reprezentace scény u natrénovaného modelu NeRF nelze prakticky upravovat.

NeRF umožňuje realisticky zachycovat světelné podmínky scény a interakci světla s okolím. Obrázky vykreslené z NeRF vypadají realisticky přímo díky reprezentaci scény jako zářivé pole. NeRF modeluje i okolí objektu, což přidává na realističnosti výsledného vzhledu. Výstupem fotogrammetrie je často 3D model, avšak pro získání realisticky vypadající snímek zasazený do scény, je třeba použít další software pro vykresování – renderer.

NeRF a fotogrammetrie jsou tedy odlišné přístupy k 3D rekonstrukci, z nichž každý má své silné stránky a omezení. NeRF využívá hluboké učení k vytváření vysoce kvalitních fotorealistických vykreslení scén, zatímco fotogrammetrie je tradičnější přístup, který přímo rekonstruuje 3D geometrii

z překrývajících se snímků. Z výše uvedených důvodů lze usoudit, že NeRF se zdá být vhodnější technologií pro řešení této práce.

Pro úplnost je zde krátké rozebrání možností softwaru Mitsuba 3 [54]. Použití rendereru Mitsuba 3 pro účely modelování scény s vozidlem je velmi výpočetně náročné a tím pádem nevhodné pro řešení problému této práce. Avšak jeho použití má své výhody, díky fyzikálně založenému rendereru a diferencovatelnému vykreslování. Tyto výhody by se mohly hodit při případném rozšíření této práce. Nabízí získání explicitního a přesnějšího vyjádření scény pomocí technik strojového učení.

Asi největší výhoda spočívá v realitě odpovídající reprezentaci materiálů. Fyzikálně přesná simulace by mohla separovat vlastnosti materiálů ve scéně a jejich interakce se světlem či kamerou. Takovou separabilitu ostatní metody neumožňují. Tímto způsobem by se dali měnit nejen pohledy na model, ale také světelné podmínky a nebo odstín barvy. Tento přístup však není jednoduchý.

Problém složitosti nastává již při sestavení parametrizované scény v reálném prostředí. Pro správné výsledky není třeba mít jen přesný model vozidla, ale také modelovat jeho okolí kvůli interakci světla. Už samotná reprezentace osvětlení není jednoduchá v nekontrolovaných podmínkách jako je exteriér.

Kapitola 4

Design a implementace

Analýza v předchozí kapitole doporučila jak nastavit parametry kamery pro snímání scény s vozidlem. Zbývá určit odkud a jak fotit. V případě modelování vozidla v 360° scéně je třeba vozidlo nasnímat ze všech stran. Počet snímků autoři Instant-NGP omezují na rozsah na 50–150 snímků. U nižších počtů může mít COLMAP problém správně určit pozice kamer a u vyšších počtů se pouze zvyšují paměťové nároky a výpočetní čas bez velkých zisků v kvalitě.

Vozidlo je třeba nafotit ze všech úhlů dokola ve 3 nebo více úrovních výšky. Například může být foceno tak, že během obcházení vozidla v kruhu se vytvoří 24 snímků a tento proces opakovat třikrát. Takto vznikne 72 snímků. Vozidlo by mělo být vidět celé na každém snímku spolu s jeho okolím. Okolí je důležité pro porovnání příznaků a odhad pozic kamer. Kolem vozidla je třeba mít dostatek místa, aby ostatní části scény nebránily pohledu na vozidlo. S tímto zastíněním si NeRF dokáže poradit, protože modeluje celou scénu i s okolím a tím pádem namodeluje i překážku, pokud je překážka vidět z jiných pohledů.

Snímání by mělo probíhat pod rozptýleným nepřímým světlem tak, aby se minimalizovali odlesky, které vadí při odhadu pozic kamer a při použití fotogrammetrie obecně. Při focení venku je optimální, když je zataženo a vozidlo nevrhá stíny. Tento požadavek není tak striktní, jak je ukázáno v příští kapitole u modelování vozidla snímaného za jasného počasí.

4.1 Homogenizace osvětlení

Algoritmy pro homogenizaci osvětlení na základě ekvalizace a specifikace histogramu byly implementovány v Python pomocí knihoven OpenCV [78] a scikit-image [79].

Pokus implementovat algoritmus NPEA v Pythonu podle článku [32] byl neúspěšný. I zdlouhavé debugování nepřineslo žadné ovoce a kontaktování autorů článku bylo neúspěšné. Autoři sice poskytují implementaci v MATLABu, kterou lze spustit a využít, avšak kód je uložen v „.p souborech“, které jsou obfuskovány, čímž je znemožněné jejich postup přepsat do jiného jazyka.

Pro algoritmus LIME existuje neoficiální implementace v Pythonu [35], kterou lze použít a případně upravit. Podobný případ jako u kódu NPEA je i u článku [36], který vylepšuje obrázky pro následnou extrakci obrazových příznaků a jejich porovnání. Zdrojový kód je opět ve formě „.p souborů“. Z článku není jasné nastavení některých proměnných, což velmi znesnadňuje vlastní implementaci.

4.2 Odstranění pozadí

Pro odstranění pozadí ze snímků s vozidlem se v této práci používají dvě různé technologie – U2-Net a SAM. Výstupy dvou nezávislých postupů lze porovnávat a získat tak více informace o kvalitě masek obrazu.

První metodou odstranění pozadí je využití předtrénované sítě U2-Net pomocí nástroje rembg [68]. Rembg je Python balíček a zároveň nástroj pro příkazovou řádku, který usnadňuje odstranění pozadí z obrazu. Sjednocuje několik různých předtrénovaných modelů různých typů i zaměření. Tento nástroj je open-source a v době psaní této práce je stále ve vývoji. Nedávno byla přidána možnost využít SAM přímo přes rozhraní tohoto nástroje.

Druhou metodou segmentace vozidla je použití modelu SAM. V této práci jsem nevyužil nástroje rembg pro použití modelu SAM, ale použil jsem implementaci přímo autorů článku [38]. Implementace je v dostupná v Pythonu jako balíček, který je třeba nainstalovat ze zdrojového kódu.

Model SAM je v této práci použit, protože má velký potenciál a je velmi pravděpodobné, že se tato technologie bude nadále vyvíjet. Další vývoj by mohl přinést lepší výsledky a nové možnosti vstupu. Lze očekávat například přidání možnosti textového dotazu určující masku jako například „modré vozidlo“, „největší vozidlo v obraze“ nebo pouze „vozidlo“. Tímto by se odstranila závislost na modelu YOLO, který „pouze“ nalezne vozidlo v obraze a vrátí ohraničující obdélník.

4.3 Modelování

Důležitou částí této práce je modelování vozidla na základě snímků. Hlavní zaměření této práce je na NeRF, ale pro srovnání a nalezení silných a slabých stránek NeRF je zde použit i fotogrammetrický software, ze kterého lze získat 3D model vozidla.

4.3.1 Meshroom

Pro srovnání technologie NeRF s „tradiční“ fotogrammetrií jsem zvolil software Meshroom [80]. Meshroom je uživatelsky přívětivý fotogrammetrický software s otevřeným zdrojovým kódem, který umožňuje uživatelům vytvářet

3D rekonstrukce modelů na základě snímků. Vyvíjí jej společnost AliceVision, společný projekt několika výzkumných institucí (spolu se zastoupením z ČVUT) a firem, který se zaměřuje na počítačové vidění a 3D rekonstrukci. Meshroom je založen na frameworku AliceVision a využívá fotogrammetrii k proměně obrázků na detailní 3D modely.

Meshroom nabízí jednoduché grafické rozhraní typu „drag-and-drop“ a je tedy přístupný uživatelům s různými úrovněmi zkušeností. Software uživatele provede celým procesem od importu snímků až po vygenerování finálního 3D modelu spolu s texturami. Pro správu různých úloh využívá systém založený na uzlech, který uživatelům poskytuje jasný přehled o každém kroku v procesu 3D rekonstrukce.

Uvnitř softwaru Meshroom se používají vlastní implementace algoritmů SfM a MVS k rekonstrukci 3D geometrie objektů ze vstupních snímků. Tyto algoritmy spolupracují při extrakci charakteristických bodů, odhadu polohy kamery a generování hustých bodů v prostoru (anglicky dense point cloud), které se následně změní na 3D polygonovou síť a texturovaný model.

Jako software s otevřeným zdrojovým kódem je Meshroom volně dostupný a může být upravován nebo rozšiřován komunitou uživatelů. Je však třeba zmínit, že existuje více podobných programů, které dokáží dosáhnout lepších výsledků, jako například Agisoft Metashape nebo Reality Capture s českými kořeny, ale ty jsou často placené a jejich uživatel musí být zkušený. Jistá výhoda tohoto programu spočívá v jeho jednoduchosti na použití pro úplného začátečníka. Celý proces je velmi automatizovaný, ale zkušenějšímu uživateli dovoluje měnit přednastavené volby různých parametrů v rámci jednotlivých kroků. Použití COLMAP pro zisk 3D modelu vyžaduje použití dalšího softwaru jako je například Meshlab. Meshroom je tedy nejjednodušší způsobem pro zisk 3D modelu s texturou ze snímků.

4.3.2 Instant NGP

V předchozí kapitole byly zmíněny důvody k výběru implementace NeRF nazvané Instant-NGP. Hlavními důvody byly rychlost a poměrně nízké nároky na hardware. Zdrojový kód je open-source, je stále udržován a přijímá vylepšení od autorů i uživatelů.

Prvním pilířem modelování vozidla pomocí NeRF je odhad pozic a parametrů kamer pomocí softwaru COLMAP. Skript `colmap2nerf.py` vygeneruje ze snímků na vstupu soubor `transforms.json`. Tento soubor obsahuje veškeré potřebné informace o snímcích a kameře, která je snímala. Skript obsahuje několik parametrů, které mění běh COLMAP, čímž mohou výrazně ovlivnit výsledný odhad pozic kamer. Důležitý parametr, určující jakým způsobem má COLMAP porovnávat nalezené příznaky, se jmenuje `colmap_matcher`. Hodnota `exhaustive` bere v potaz všechny možné dvojice snímků, zatím co `sequential` porovnává snímky, které byly pořízeny za sebou. Dále je možné určit druhy objektů, které se mají ze snímků odstranit. Tím se snaží odstranit

části obrazu, které se mezi snímky pohybují, jako například lidé nebo auta. Tato možnost však v této práci není využita, protože odstranění nežádoucích objektů či částí scény se provádí separátně.

Druhou částí je samotné trénování neuronové sítě. Instant-NGP umožňuje sledovat, jak se model učí v reálném čase díky aplikaci s grafickým rozhraním. Tato aplikace nabízí kromě přehledu aktuálního modelu a přehledu o průběhu trénování i mnoho různých nastavení jak pro trénování, tak i pro renderování. Natrénovaný model lze uložit do souboru pro případné další použití. Aplikace umožňuje vygenerovat 3D model pomocí algoritmu „marching cubes“, avšak výsledky nebývají dobré. Jsou velmi zašuměné a ztrácí se informace o vlastnostech při různých úhlech pohledu. Po natrénování modelu lze renderovat jednotlivé snímky nebo přímo video.

Poměrně vlivný parametr na průběh učení a výsledný vzhled scény je `aabb_scale`. Tento parametr určuje velikost oblasti, ve které se bude scéna modelovat. Čím vyšší hodnota parametru (maximálně až 128), tím větší prostor se vzorkuje a pozadí či okolí scén vypadá lépe. Pro segmentované snímky je vhodnější volba nižších hodnot okolo 2. Nižší hodnota parametru pro segmentované scény snižuje počet artefaktů a zrychluje učení.

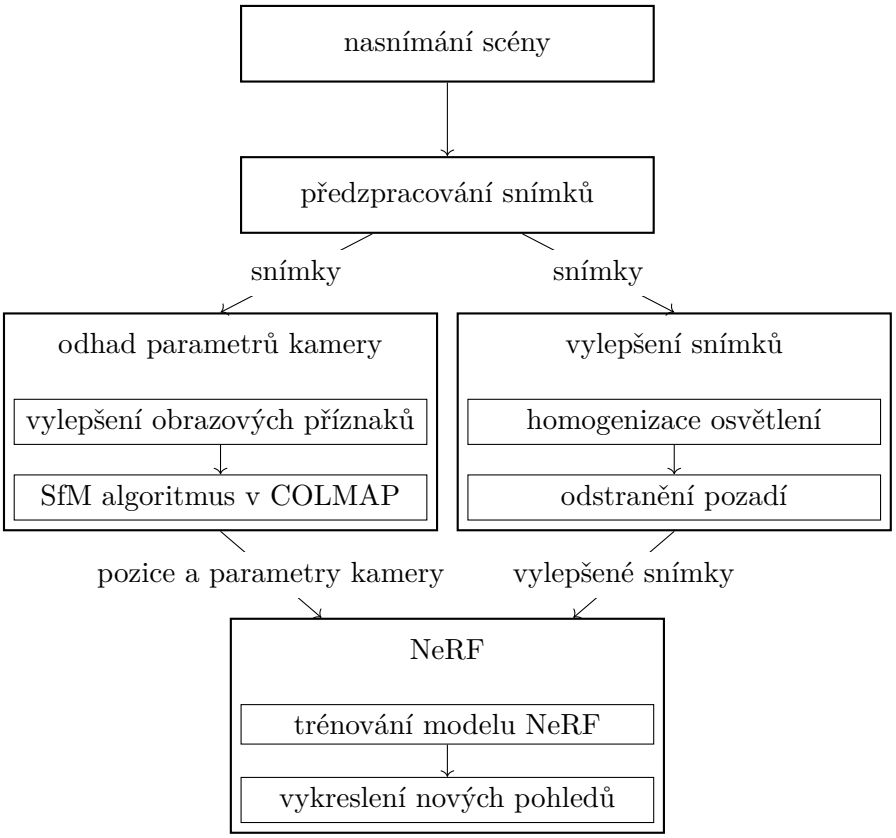
Segmentace vozidla v obraze má příznivý vliv na učení v několika ohledech jak v rychlosti, tak i v kvalitě výsledku. Kromě samotného odstranění okolí z výsledného modelu také snižuje výpočetní čas a zároveň zvyšuje kvalitu modelu. Neuronová síť se nemusí snažit modelovat okolí a veškerou svou kapacitu tak věnuje čistě vozidlu, což má pozitivní vliv kvalitu výsledného modelu.

4.3.3 Proces modelování ze snímků

Celý proces od nasnímání po natrénování výsledného modelu lze rozdělit do několika kroků. Po nasnímání scény s vozidlem proběhne homogenizace osvětlení a další úpravy snímků, odhad pozic a parametrů kamery, případná segmentace a nakonec trénování modelu NeRF. Je důležité dodržet provádět segmentaci až po odhadu pozic, protože blízké okolí vozidla a pozadí je velmi důležité pro odhad pozic kamer pomocí softwaru COLMAP.

Zisk pozic kamer a úprava snímků lze separovat a provádět v opačném pořadí pouze za předpokladu, že úpravy nijak nedeformují výsledné obrázky. Například jednoduchá úprava jasu, změna barev, nebo segmentace při zachování rozměrů původního obrázku nezmění odhad parametrů kamery.

Na obrázku 4.1 je zobrazený diagram procesu modelování vozidla. Nejdříve je třeba nasnímat scénu s vozidlem. Druhým krokem je předzpracování snímků, které může měnit odhady parametrů kamery jako je například změna rozlišení, oříznutí, translace nebo transformace pro odstranění distorze. Následně ze snímků získají parametry kamery pomocí SfM algoritmu ze softwaru COLMAP. Před tímto odhadem lze obrázky upravit pomocí algoritmů pro vylepšení obrazových příznaků jako je popsán v [36]. Výstupem jsou pozice a parametry kamery pro jednotlivé snímky. Vedle odhadu parametrů lze separátně



Obrázek 4.1: Proces zpracování snímků scény

vylepšit snímky pomocí homogenizace osvětlení a následného odstranění pozadí. Tyto dvě větve se následně spojí před trénováním NeRF v souboru `transforms.json`. Soubor bude obsahovat parametry kamery a její pozice pro jednotlivé snímky z části s odhadem parametrů a cesty k vylepšeným snímkům z druhé větve. Nakonec se natrénuje model NeRF a případně vegenerují nový pohledy pro další použití.

Kapitola 5

Experimenty a výsledky

Tato kapitola obsahuje zhodnocení metod a technologií použitých v této práci. Jsou zde vyhodnoceny experimenty, které vznikaly v průběhu práce. V této kapitole jsou ukázky výstupů metod homogenizace osvětlení snímků a konečného trénování modelu NeRF. Kapitola taktéž obsahuje rozbor úskalí modelování pomocí NeRF. Na konci této kapitoly je krátké shrnutí dalších možností a postupů pro navázání a rozšíření této práce.

5.1 Data

V průběhu této práce vzniklo několik různých sad snímků a videí, které se používaly při experimentování a prozkoumávání možností řešení této práce. Vznikly dvě sady snímků zachycující vozidlo z 360°, které se podařilo následně modelovat. Jedna sada byla focena běžným mobilním telefonem a druhá fotoaparátem SONY A7S. Dále vznikla virtuální scéna umožňující testování různých pozic kamer.

Během zkoušení různých přístupů focení vyšlo najevo, že pro správný odhad pozic kamer je velmi důležité blízké okolí vozidla. Lesklý lak, průhledná okna a praktická absence textury na velkých plochách vozidla snižuje počet příznaků, které jsou správně porovnány a projektovány do 3D prostoru. Vizualizace v softwaru Meshroom ukázaly, že body nalezené po běhu algoritmu SfM jsou hlavně na blízkém okolí vozidla. Ty body, které jsou skutečně na vozidle se nejčastěji vyskytují v okolí SPZ a kol.

V průběhu testování se sice nepodařilo najít jeden způsob snímání scény, který by zaručil dobré výsledky, ale podařilo se identifikovat několik tipů, které pomáhají zvýšit šance na úspěch. Dále v této kapitole jsou rozebrány další možnosti snímání, které by mohly pomoci zkvalitnit zejména odhady pozic kamery.

Testy byly prováděny na laptopu s 6 jádrovým procesorem AMD Ryzern 5 5600H, 16 GB operační paměti RAM a grafickou kartou NVIDIA GeForce RTX 3060 Laptop GPU s 6 GB paměti. V průběhu testování se ukázalo,

že trénování NeRF je poměrně náročné na paměť grafické karty, protože je třeba všechny snímky nahrát do paměti grafické karty. Z tohoto důvodu byly některé experimenty s modelováním prováděny v polovičním rozlišení. Pro odhad velikosti potřebné paměti lze využít online kalkulačku [81].

5.2 Experimenty

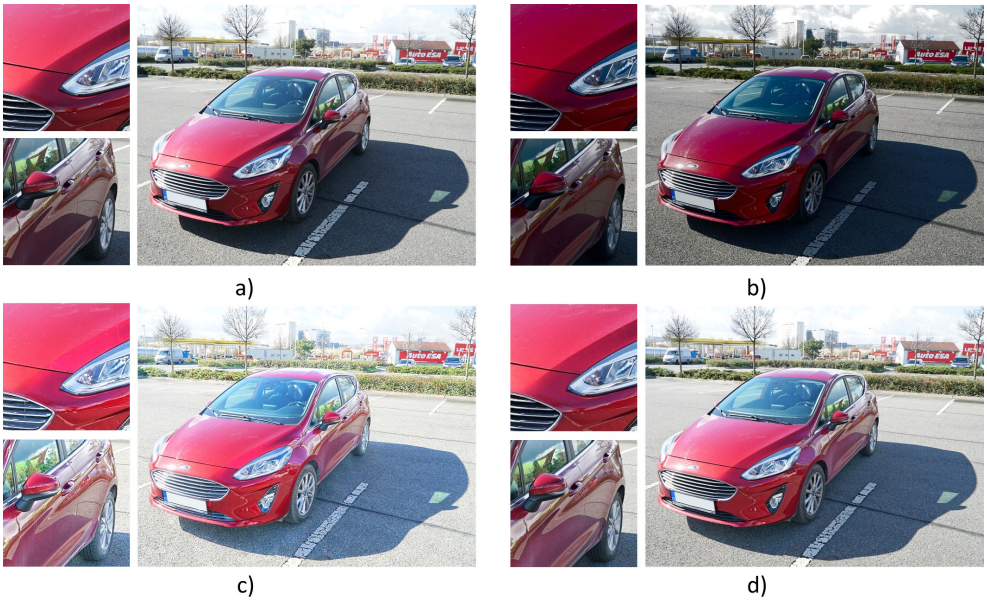
Při porovnávání fungování metod homogenizace obrazu jako ekvalizace nebo specifikace histogramu či algoritmů NPEA a LIME nelze použít klasické metriky jako MSE (mean squared error) nebo PSNR (peak signal-to-noise ratio), protože vyžadují existenci referenčního obrazu, který u těchto úloh často k dipozici není. Úloha zabývající se hodnocením kvality obrazu bez reference se také označuje jako NR-IQA (no-reference image quality assessment) [82]. Metriky spadající do této kategorie jsou například VIF [83], BRISQUE [82, 84] nebo CNNIQA [85], která využívá neuronové sítě k určení kvality snímku. Při porovnávání snímků před a po úpravě osvětlení lze podle návrhu z článku [32] počítat LOE (light order error), tedy chybu v uspořádání jasu ve snímku.

První velký rozdíl mezi jednotlivými metodami je v době běhu. Pro 3 MP obrázek je doba běhu ekvalizace či specifikace histogramu v řádu desítek milisekund, NPEA pro stejný obrázek běží přibližně 70 s a LIME spočítá vylepšený obrázek za přibližně 650 s. Experimenty ukázali, že NPEA i LIME škálují lineárně s počtem pixelů, což by znamenalo, že LIME je prakticky nepoužitelný pro scény se snímky s rozlišením vyšším jak 12 MP.

Metody byly porovnány i pomocí obrazových metrik. Hodnoty metrik byly měřeny u původního a upraveného obrázku. Vyhodnocována byla pak změna hodnoty po upravení. Zajímavé je, že se metriky v některých případech neshodují o tom, jestli byl původní snímek kvalitnější či nikoliv. Například metoda LIME zlepšila metriky BRISQUE téměř ve všech případech, ale metrika CNNIQA neindikovala výrazné zlepšení ani zhoršení. U metody NPEA vyšlo hodnocení podle BRISQUE špatné, ale změna metriky CNNIQA ukazovala částečné vylepšení snímků. Metrika LOE, která se zaměřuje na přirozenost osvětlení scény, vyšla u NPEA v průměru 2,5× nižší než pro metodu LIME. V případě ekvalizace histogramu se LEO velmi měnilo pro různé snímky, což je způsobeno citlivostí na změny světelných podmínek v pozadí.

Na obrázku 5.1 jsou zobrazené výstupy metod homogenizace osvětlení. Obrázek a) obsahuje původní neupravený snímek a obrázek b) výstup po ekvalizaci světlosti – první složky v L*a*b* prostoru. Obrázek c) je původní snímek po zpracování pomocí metody LIME a d) je po aplikaci NPEA. Na první pohled je vidět, že ekvalizace světlosti snížila jas vozidla, zatímco LIME některé části zesvětlila. Na přiblížených snímcích si lze všimnout různých chování na kapotě, zadním kole, či odraze v okénku.

Závěr experimentování s metodami homogenizace je takový, že při nutnosti zpracování v reálném čase je třeba využít metod založených na operacích



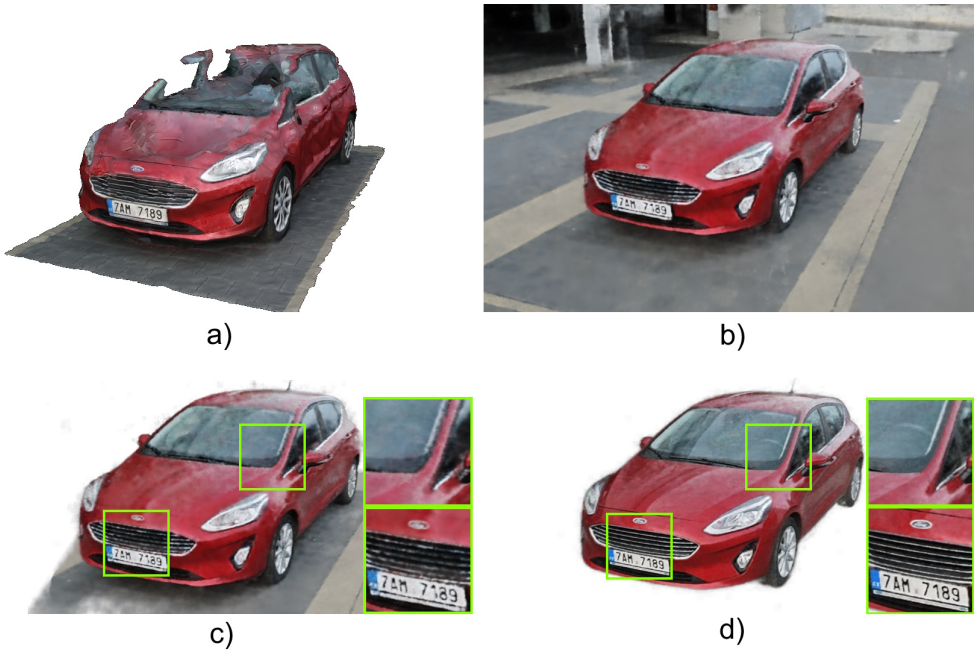
Obrázek 5.1: Porovnání výstupů algoritmů pro homogenizaci osvětlení. a) původní snímek b) ekvalizace histogramu světlosti c) LIME d) NPEA

s histogramem. Při relaxaci takových požadavků je výhodné použít sofistikovanější, avšak stále upočítatelnou, techniku NPEA. Metoda LIME, kvůli své výpočetní náročnosti a parametrizaci, není vhodnou volbou pro řešení této práce.

Segmentace snímků pomocí U2-Net je v některých případech nedokonalá. Některé snímky obsahují části, které jsou sice blízko vozidla, ale nejsou jeho součástí. Tento problém je způsoben tím, že neuronová síť neví, že objekt pro segmentaci je právě vozidlo. Tyto artefakty, které se vyskytují zřídka, ovšem nečiní problém při trénování modelu NeRF. Neuronová síť tyto často nahodilé nepřesnosti nemodeluje a výsledný model je tak velmi dobrý.

Při použití modelu SAM je třeba využít i nějaký další model či algoritmus pro označení ohraničujícího obdélníku vozidla. K tomu byl použit model YOLO, který vrátil obdélníky spolu s typy objektů. Tyto obdélníky byly následně filtrovány pro odstranění nevhodných tříd objektů a následně se vybral takový obdélník, který měl největší plochu. Tento postup vrací správné ohraničení za předpokladu, že vozidlo na snímku je největším objektem na snímku. Model SAM je schopný generovat několik různých masek pro jeden dotaz, ale při použití ohraničujícího obdélníku stačí generovat jedinou masku a výsledek je správný.

Na obrázku 5.2 jsou zobrazeny 4 různé výstupy při modelování vozidla ze stejných snímků. Při modelování této scény byly všechny pozice kamer správně odhadnuty, takže lze vyloučit to, že deformace a nedostatky jsou způsobené



Obrázek 5.2: Porovnání různých výstupů modelování ze stejného souboru snímků. a) 3D model s texturou z softwaru Meshroom b) výstup z NeRF trénovaného na snímcích bez segmentace c) výstup z NeRF trénovaného na snímcích bez segmentace s oříznutím prostoru vykreslování d) výstup z NeRF trénovaného na segmentovaných snímcích bez pozadí

špatným odhadem pozic a parametrů kamer. Na obrázku a) je 3D model vozidla s texturou. Lze si všimnout chybějícího předního skla, částí sloupků a zdeformované střechy a kapoty. Kvalitě vzhledu výrazně pomáhá ostrá textura a to hlavně v místech, kde je model správný. Obrázky b) a c) jsou výstupy z modelu NeRF, který byl trénovaný na snímcích bez segmentace. Obrázek c) se liší tím, že je v něm kvádrem omezený prostor, ve kterém se vzorkuje NeRF, což zapříčiní odstranění pozadí. Obrázek d) je výstup modelu NeRF trénovaného na segmentovaných snímcích. Díky odstranění pozadí ještě před samotným trénováním NeRF se zvýšila kvalita modelu. Model na obrázku d) je ostřejší a obsahuje méně šumu. Na dvou menších obrázcích s přiblížením si lze všimnout ostřejší SPZ a u d) patrný volant na rozdíl od obrázku c).

Implementace Instant-NGP používá jednoduchý model kamery pro vykreslování scény, který nebere v potaz distorzní koeficienty. Z tohoto důvodu nelze správně registrovat vstupní a výstupní snímky pro porovnání pomocí klasických metrik jako je MSE. Kvalitu modelu lze částečně usuzovat podle hodnot loss funkce. Pro scénu ze snímků b) a c) je po 10000 krocích trénování hodnota loss funkce 0,003411 (24.7 dB), u scény ze snímku d) je po stejném počtu trénovacích kroků hodnota loss funkce rovna 0.000689 (31.6 dB), tedy

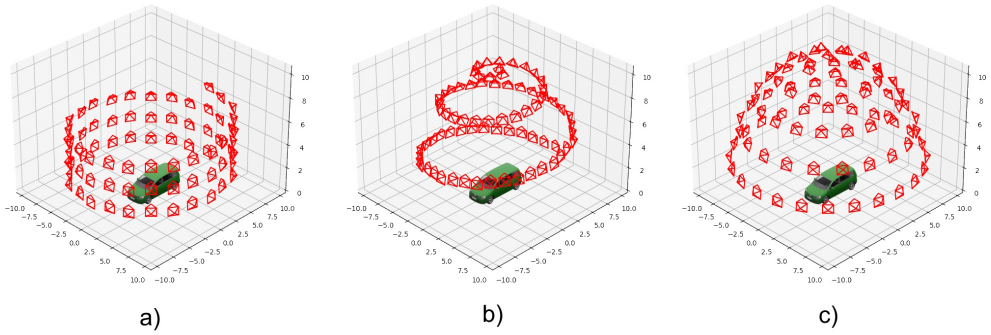
Obrázek 5.3: Ukázka artefaktů v natrénovaném modelu v podobě poletujících obláčků (anglicky floater)

téměř o řád lepší. Odstranění pozadí před trénováním se zdá být velmi velkou pomocí při trénování.

Složité scény a scény s nesprávnými odhady pozic kamer vykazují různé artefakty. Asi nejzřetelnějším artefaktem jsou obláčky ve scéně (anglicky floaters). Na obrázku 5.3 jsou vidět takové obláčky. Nejlépe jsou vidět proti černému pozadí či červenému vozidlu, protože barva takových obláčků je často bílá nebo šedá. Tyto artefakty vznikají i při nekonzistentním nastavení expozice nebo při příliš malé hodnotě parametru `aabb_scale`.

Nastavení parametru `aabb_scale` neovlivňuje pouze přítomnost artefaktů, ale má velký vliv na celkový proces modelování. Hodnota, která se nastavuje v mocninách 2 až do 128, určuje jak velký prostor se vzorkuje při modelování a vykreslování scény. Jedná se v podstatě o velikost scény. Vyšší hodnoty umožňují lépe modelovat okolní scénu a vzdálené pozadí za cenu vyššího výpočetního času. Při modelování segmentovaných snímků je naopak výhodná nízká hodnota – přibližně 2. Hodnota musí být dostatečně velká, aby se tam vešlo celé modelované vozidlo. Menší prostor zabraňuje „halucinacím“ neuronové sítě mimo vozidlo a zároveň zvyšuje rychlost a přesnost modelování.

Kromě parametru `aabb_scale` je možné v souboru `transforms.json` nastavit další parametry, které mohou přispět k lepším výsledkům. Parametr `scale` mění škálu celé scény. Jeho hodnoty nejsou omezeny na mocniny 2 a lze ho tedy volit tak, aby vozidlo co nejvíce naplňovalo prostor scény. Druhý parametr upravující učení je `n_extra_learnable_dims`, který nastavuje dimenzi latentního vektoru pro každý vstupní obraz. Přidáním této latentní reprezentace pomáhá při vypořádávání se s nekonzistentními snímky. Oba popsané



Obrázek 5.4: Konfigurace pozic kamery při snímání

parametry pomáhají k lepšímu modelování, ale jejich vliv není moc velký. Zásadnější jsou dobré snímky a správný odhad pozic a parametrů kamery.

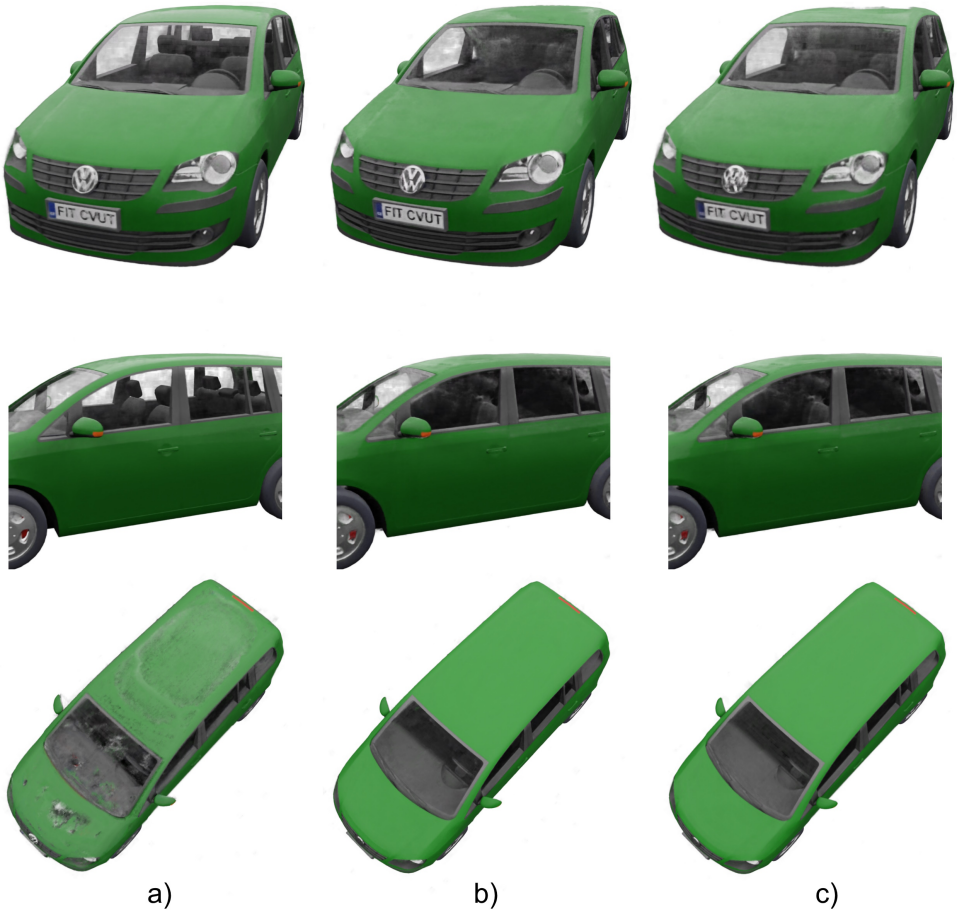
5.2.1 Virtuální scéna

Jeden z prvních experimentů se zaměřil na určení vhodný pozic pro snímání vozidla. V rámci stejné scéně bylo vozidlo fotografováno různými způsoby, které byly analyzovány na základě výsledných modelů. Zjistilo se, že pro jednu rotaci kolem vozidla je třeba alespoň 16 snímků a optimální počet se pohybuje někde mezi 20 a 25 snímky. Dále bylo zkoumáno, kolik úrovní/výšek snímání je potřebných. Výsledky ukázaly, že pouze jedna nebo dvě úrovně snímání neposkytují dostatečný překryv a pokrytí vozidla pro uspokojivý výsledek. Nejnižší úroveň snímání by měla být zhruba v úrovni kol, zatímco nejvyšší úroveň by měla být nad úrovní střechy vozidla.

Pro umožnění testování různých umístění kamery při snímání byla vytvořena virtuální scéna v 3D modelovacím softwaru Blender [86]. S využitím definic křivek v prostoru nebo pomocí vlastní naprogramované funkce je možné určit pozice kamer, z nichž se má scéna snímat, a poté vykreslit jednotlivé pohledy. Tento přístup umožňuje rychlé iterování různých konfigurací snímání a testování i těch, které by vyžadovaly speciální zařízení, například dron s kamerou.

Díky virtuální scéně je možné extrahovat přesné pozice a parametry kamery, čímž se eliminuje potřeba spoléhat na odhad z dalších programů. Je možné měnit i jiné parametry kamery či přímo celé prostředí scény. Připravený skript vytvoří soubor `transforms.json` s jednotlivými parametry kamery. Případně i vykreslí snímky virtuální scény. Tím se připraví vše potřebné pro přímé trénování modelu NeRF.

Na obrázku 5.4 jsou zobrazené tři různé konfigurace pozic kamery. Pro všechny tyto konfigurace platí, že kamera se vždy natočí tak, aby mířila přímo na jeden zvolený bod, který je uprostřed modelu. Na obrázku a) je zobrazené rozmístění kamer, které kopíruje spirálu s konstantní horizontální vzdáleností



Obrázek 5.5: Vykreslené pohledy pro virtuální scénu s různými konfiguracemi kamer. Pozice kamer odpovídají těm v obrázku 5.4.

od středu, u které se zvedá výška o 2 metry s každou rotací. Na obrázku b) jsou kamery umístěny podél poloviny sférické spirály s poloměrem 10 m. Na obrázku c) je zobrazení rozložení 81 pozic kamer v 6 různých elevacích tak, aby bylo pokrytí snímky sféricky uniformní. Podobné snímání se používá například u měření vzhledu materiálů.

Obrázek 5.5 ilustruje výhody a nevýhody jednotlivých rozmístění kamer z obrázku 5.4. Například model s konfigurací a) se naučil reprezentovat nejen vnější vzhled, ale také průhlednost oken a velmi dobře modeluje interiér vozu. Problém mu ovšem dělají pohledy kolmo dolů, protože v těch musí výrazně extrapolovat a vznikají tak artefakty na střeše a kapotě. Naopak modely s konfiguracemi b) a c) tyto kolmé pohledy zvládají dobře, ale problém jim dělají nízké pohledy nebo zmíněný interiér. Výhoda konfigurace c) oproti b) je sy-

metrie v úhlech pohledů. Výsledný model u b) se chová trochu jinak na levé a pravé straně.

5.2.2 Problém odhadu parametrů kamery

Během experimentování s modelováním se vyskytl jeden zásadní problém, kterým byl nesprávný odhad pozic kamer v softwaru COLMAP. V některých situacích iterativní část algoritmu SfM nezkonvergovala a tím pádem ani neodhadla žádné polohy kamer. Tento problém může být způsoben například rozmazanými snímky nebo jejich nedostatečným překryvem. V jiných případech došlo k vyřazení některých nekvalitních snímků, což vedlo k absenci některých pozic kamer na výstupu. Hlavním problémem zůstávají špatně odhadnuté pozice kamer. Model NeRF následně zohledňuje i tyto nesprávné pohledy, což vede k vizualizacím, které sice mohou být zajímavé, ale působí nerealisticky. Častým jevem je, že odhady kamer se podaří získat, ale výsledné pozice jsou pouze na jedné straně vozidla.

Experiment ukázal, že snímky pořízené s delší ohniskovou vzdáleností jsou náchylnější k nesprávnému odhadu pozic kamer. Důvodem je pravděpodobně větší proměnlivost pozadí snímku, což vede k nedostatečnému překryvu mezi snímky. Pokud jde o samotné trénování modelu NeRF, zdá se, že rozdíly mezi ohniskovými vzdálenostmi nemají významný vliv, neboť tyto hodnoty jsou odhadovány při odhadu pozic a parametrů kamery a následně používány během trénování.

Další problém, který způsobuje nesprávnost odhadu pozic kamer, jsou opakující se obrazové příznaky. Vozidlo je často téměř dokonale symetrické, zvláště když se vezme v potaz pouze exteriér vozidla. Jak již bylo zmíněno, velká část registrovaných bodů na vozidle se vyskytuje v oblasti kol a SPZ. Algoritmus pro porovnávání obrazových příznaků je tedy zmaten a přiřadí nekorespondující příznaky k sobě. Například při zmatení u SPZ se může stát, že snímky zadní části vozidla jsou umisťovány po boku snímků předku vozidla. Výsledný model tak interpoluje mezi zadní a přední částí, což úplně naboural realističnost vzhledu. Jedním z částečných řešení je zakrytí jedné z SPZ před snímáním nebo jejích odstranění během předzpracování.

Problém se scénami s vyskytujícími se opakujícími se objekty/příznaky a vysokou symetrií se snažil řešit v rámci svého projektu Lixin Xue. [87] V tomto projektu implementoval dvě různé metody, které se zaměřují na řešení problémů SfM algoritmů s porovnáváním příznaků. První článek je „Distinguishing the indistinguishable: Exploring structural ambiguities via geodesic context“ [88] zaměřující se na nejednoznačnosti v obraze. Druhá implemetovaná metoda, se kterou se experimentovalo, je popsána v „Global structure-from-motion by similarity averaging“ [89]. Xue experimentoval i s metodou implementovanou v „Improving structure from motion with reliable resectioning“ [90]. Xue bohužel došel k závěru, že žádná z těchto metod není dostatečně

robustní na to, aby fungovala pro většinu scén bez nastavování parametrů pro každou scénu zvlášť.

Autoři implementace Instant-NGP navrhují možné použití aplikace Record3D, NeRFCapture [91] nebo jiných aplikací založených na frameworku ARKit. Aplikace už při snímání odhaduje pozici kamery pro každý snímek. Tyto odhady lze vylepšit pomocí COLMAP tak, že pozice se použijí jako apriorní odhady. COLMAP využije tyto odhady pro incializaci a následně odhady pozic vylepší. Tímto by měl vzniknout robustnější způsob odhadování. Tato technologie je však dostupná pouze pro některá zařízení od společnosti Apple. Alternativou je multiplatformní ARCore od společnosti Google, ale tato možnost zatím nebyla dostatečně otestována a integrována do existující implementace.

Podobný přístup k inicializaci pozic kamer pro jednotlivé snímky je využití geotaggingu u jednotlivých snímků. Při experimentování byly obrázky obohaceny o metadata ve formě souřadnic zeměpisné šířky a délky s nadmořskou výškou. Tyto pokusy byly však neúspěšné a nepřinášely lepší výsledky při odhadu pozic.

V článku [92] autoři představují novou verzi NeRF, která při trénování nepotřebuje pozice a parametry kamery. Poloha v 3D prostoru a parametry kamery jsou přidány do souboru parametrů, které se neuronová síť učí. Dále ukazují, že tento přístup má srovnatelné výsledky s klasickou verzí NeRF. Tato modifikace funguje pouze na scény snímané z jedné strany, podobně jako LLFF. Z tohoto důvodu nelze tuto alternativu použít.

5.3 Možné úpravy a rozšíření

Při implementaci a experimentování bylo nalezeno několik nedostatků, které brání využití potenciálu technologie NeRF v plné míře. Zde je přehled několika myšlenek, které by mohly napomoci robustnějšímu a vizuálně lepšímu zpracování scény.

Pro snížení dopadu odlesků na celý proces homogenizace osvětlení a odhadů pozic kamery lze použít polarizační filtr při snímání vozidla. Při odrazu světla od lesklého laku dochází k polarizaci světla. Použitím polarizačního filtru lze částečně nebo v některých případech úplně odstranit výrazný odlesk.

Přetrénování sítě U2-Net speciálně pro účely segmentace vozidel ze scény by mohlo zvýšit přesnost a kvalitu výsledné masky. Trénováním na datasetu pouze s vozidly by se mohlo dosáhnout odstranění artefaktů v podobě částí objektů v okolí vozidla, které se na snímcích občas vyskytují.

Hlavním problémem celého zpracování je nesprávný odhad pozic a parametrů kamery na základě snímků. Možné řešení je definovat pozice kamery při snímání. Po nasnímání scény by se definované pozice použili jako apriorní odhad, který by se pomocí algoritmu SfM mohl dále zlepšovat. Případně za-

vedením vlastních implementací částí algoritmu SfM, jako například „feature matching“, by se dal algoritmus přizpůsobit konkrétně scénám s vozidlem.

Dalším možným řešením je opuštění použití algoritmů SfM pro odhad pozic kamery a využít například registrační značky, které by byly umístěny na přesně definovaných místech. Takové registrační značky by mohly být umístěny na plachtě, na kterou by vozidlo najelo. Tento způsob by však nejspíš vyžadoval použití dalších algoritmů pro zpřesnění a odstranění malých nedokonalostí v pozicích. Zajímavým způsobem vylepšení této práce by bylo využití frameworku ARCore pro získání apriorního odhadu pozic kamery.

Firma RECON Labs se zabývá ziskem 3D modelů na základě krátkého videa objektu. Zmiňují použití technologie NeRF spolu s následným odhadem SDF, které vylepšuje výsledný 3D model. Jedná se spojení klasického fotogrammetrického postupu s využitím NeRF pro augmentaci dat. SDF používají k vyhlazení výsledného modelu. Pro odhad pozic kamery používají cloudové řešení Agisoft Metashape. Jejich služba 3Dpresso umožňuje vyzkoušet jejich technologii na vlastních datech. [93]

Závěr

Tato práce zkoumala potenciál pokročilých technik zpracování obrazu, a to konkrétně neuronových zářivých polí (NeRF) a metod hlubokého učení pro segmentaci U2-Net a SAM, v kontextu snímání vozidel. Tento výzkum byl zaměřen na prozkoumání možných řešení pomocí „state-of-the-art“ metod segmentace a modelování scény a jejich následné porovnání s klasickými metodami.

Hlavním cílem této práce bylo vytvořit „proof-of-concept“ řešení modelování a vykreslování vozidel na základě pořízených snímků. Za tímto účelem se práce zaměřila na tři hlavní fáze: homogenizaci světla, odstranění pozadí pomocí segmentace a následné modelování pomocí NeRF. Výsledné řešení demonstruje výhody nových technologií při zpracování obrazu vozidel.

Tato práce poskytla cenné poznatky o potenciálních aplikacích a výhodách technologie NeRF a dalších pokročilých technik zpracování obrazu v automobilovém průmyslu, konkrétně na trhu ojetých automobilů. Zjištění naznačují, že navrhovaný přístup může vést k přesnějším a vizuálně přitažlivějším reprezentacím vozidel. V rámci této práce byla vytvořena virtuální scéna v modelovacím softwaru Blender, která umožňuje vytvářet umělé datové sady pro testování různých postupů.

Kromě toho tato práce identifikovala oblasti pro další zlepšení a inovace a zdůraznila potřebu pokračovat ve výzkumu a vývoji s cílem zdokonalit a rozšířit možnosti NeRF a souvisejících technik. Posouváním hranic možností v oblasti analýzy obrazu v automobilovém průmyslu může budoucí výzkum přispět k pokračujícímu růstu a rozvoji automobilového průmyslu jako celku.

V důsledku tato práce slouží jako odrazový můstek k porozumění nových technologií jako je SAM nebo neuronová reprezentace scény pomocí NeRF. Tyto technologie jsou stále ve vývoji a mají velký potenciál. Výsledné řešení ukázalo, že tyto technologie jsou vhodnou cestou pro další výzkum či tvorbu komerčního řešení.

Bibliografie

[1] Ben Mildenhall et al. „Nerf: Representing scenes as neural radiance fields for view synthesis“. In: *Communications of the ACM* 65.1 (2021), s. 99–106.

[2] Ross Girshick et al. „Rich feature hierarchies for accurate object detection and semantic segmentation“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, s. 580–587.

[3] Kaiming He et al. „Deep residual learning for image recognition“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, s. 770–778.

[4] Joseph Redmon et al. „You only look once: Unified, real-time object detection“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, s. 779–788.

[5] Milan Sonka, Vaclav Hlavac a Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[6] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[7] Rafael C Gonzales a Paul Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[8] Christine Connolly a Thomas Fleiss. „A study of efficiency and accuracy in the transformation from RGB to CIELAB color space“. In: *IEEE transactions on image processing* 6.7 (1997), s. 1046–1048.

[9] Hamid Rezatofighi et al. „Generalized intersection over union: A metric and a loss for bounding box regression“. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, s. 658–666.

[10] D.G. Lowe. „Object recognition from local scale-invariant features“. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Sv. 2. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410.

[11] Arthur Appel. „Some Techniques for Shading Machine Renderings of Solids“. In: *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference*. AFIPS '68 (Spring). Atlantic City, New Jersey: Association for Computing Machinery, 1968, s. 37–45. ISBN: 9781450378970. DOI: 10.1145/1468075.1468082. URL: https://doi.org/10.1145/1468075.1468082.

[12] Turner Whitted. „An Improved Illumination Model for Shaded Display“. In: *Commun. ACM* 23.6 (led. 1980), s. 343–349. ISSN: 0001-0782. DOI: 10.1145/358876.358882. URL: https://doi.org/10.1145/358876.358882.

[13] James T. Kajiya. „The Rendering Equation“. In: *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '86. New York, NY, USA: Association for Computing Machinery, 1986, s. 143–150. ISBN: 0897911962. DOI: 10.1145/15922.15902. URL: https://doi.org/10.1145/15922.15902.

[14] Nvidia. *Nvidia Turing GPU architecture*. [cit. 2023-04-05]. URL: https://images.nvidia.com/aem-dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf (cit. 05. 04. 2023).

[15] VV Sanzharov et al. „Examination of the Nvidia RTX“. In: *CEUR Workshop Proceedings*. Sv. 2485. 2019, s. 7–12.

[16] John C Hart. „Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces“. In: *The Visual Computer* 12.10 (1996), s. 527–545.

[17] John C Hart, Daniel J Sandin a Louis H Kauffman. „Ray tracing deterministic 3-D fractals“. In: *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*. 1989, s. 289–296.

[18] Inigo Quilez. *SDF and ray marching articles*. URL: https://iquilezles.org/articles/ (cit. 20. 04. 2023).

[19] Paul E Debevec, Camillo J Taylor a Jitendra Malik. „Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach“. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996, s. 11–20.

[20] Shenchang Eric Chen. „Quicktime VR: An image-based approach to virtual environment navigation“. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 1995, s. 29–38.

[21] Harry Shum a Sing Bing Kang. „Review of image-based rendering techniques“. In: *Visual Communications and Image Processing 2000*. Sv. 4067. SPIE. 2000, s. 2–13.

[22] Edward H Adelson, James R Bergen et al. „The plenoptic function and the elements of early vision“. In: *Computational models of visual processing* 1.2 (1991), s. 3–20.

[23] Leonard McMillan a Gary Bishop. „Plenoptic modeling: An image-based rendering system“. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 1995, s. 39–46.

[24] Jeffrey L Elman. „Finding structure in time“. In: *Cognitive science* 14.2 (1990), s. 179–211.

[25] Ian Goodfellow et al. „Generative adversarial networks“. In: *Communications of the ACM* 63.11 (2020), s. 139–144.

[26] Geoffrey E Hinton a Ruslan R Salakhutdinov. „Reducing the dimensionality of data with neural networks“. In: *science* 313.5786 (2006), s. 504–507.

[27] Ashish Vaswani et al. „Attention is all you need“. In: *Advances in neural information processing systems* 30 (2017).

[28] Simon J.D. Prince. *Understanding Deep Learning*. MIT Press, 2023. URL: https://udlbook.github.io/udlbook/.

[29] Yann LeCun et al. „Handwritten digit recognition with a back-propagation network“. In: *Advances in neural information processing systems* 2 (1989).

[30] O. Ronneberger, P.Fischer a T. Brox. „U-Net: Convolutional Networks for Biomedical Image Segmentation“. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Sv. 9351. LNCS. (available on arXiv:1505.04597 [cs.CV]). Springer, 2015, s. 234–241. URL: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a.

[31] Xuebin Qin et al. „U2-Net: Going deeper with nested U-structure for salient object detection“. In: *Pattern recognition* 106 (2020), s. 107404.

[32] Shuhang Wang et al. „Naturalness Preserved Enhancement Algorithm for Non-Uniform Illumination Images“. In: *IEEE Transactions on Image Processing* 22.9 (2013), s. 3538–3548. DOI: 10.1109/TIP.2013.2261309.

[33] Edwin H Land a John J McCann. „Lightness and retinex theory“. In: *Josa* 61.1 (1971), s. 1–11.

[34] Xiaojie Guo, Yu Li a Haibin Ling. „LIME: Low-light image enhancement via illumination map estimation“. In: *IEEE Transactions on image processing* 26.2 (2016), s. 982–993.

[35] Souhaib Attaiki. *Low light Image Enhancement*. [GitHub repo]. 2020. URL: https://github.com/pvnieo/Low-light-Image-Enhancement.

[36] Chun Liu et al. „A spatial-frequency domain associated image-optimization method for illumination-robust image matching“. In: *Sensors* 20.22 (2020), s. 6489.

[37] Michela Lecca, Alessandro Torresani a Fabio Remondino. „On image enhancement for unsupervised image description and matching“. In: *Image Analysis and Processing–ICIAP 2019: 20th International Conference, Trento, Italy, September 9–13, 2019, Proceedings, Part II 20*. Springer. 2019, s. 82–92.

[38] Alexander Kirillov et al. „Segment anything“. In: *arXiv preprint arXiv:2304.02643* (2023).

[39] Tom Brown et al. „Language models are few-shot learners“. In: *Advances in neural information processing systems* 33 (2020), s. 1877–1901.

[40] Rishi Bommasani et al. „On the opportunities and risks of foundation models“. In: *arXiv preprint arXiv:2108.07258* (2021).

[41] Yanghao Li et al. „Exploring plain vision transformer backbones for object detection“. In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Springer. 2022, s. 280–296.

[42] Alexey Dosovitskiy et al. „An image is worth 16x16 words: Transformers for image recognition at scale“. In: *arXiv preprint arXiv:2010.11929* (2020).

[43] Alec Radford et al. „Learning transferable visual models from natural language supervision“. In: *International conference on machine learning*. PMLR. 2021, s. 8748–8763.

[44] Ben Mildenhall et al. „Local light field fusion: Practical view synthesis with prescriptive sampling guidelines“. In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), s. 1–14.

[45] Johannes Lutz Schönberger a Jan-Michael Frahm. „Structure-from-Motion Revisited“. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, s. 4104–4113.

[46] Johannes Lutz Schönberger et al. „Pixelwise View Selection for Unstructured Multi-View Stereo“. In: *European Conference on Computer Vision (ECCV)*. 2016, s. 501–518.

[47] Tinghui Zhou et al. „Stereo magnification: Learning view synthesis using multiplane images“. In: *arXiv preprint arXiv:1805.09817* (2018).

[48] Thomas Müller et al. „Instant Neural Graphics Primitives with a Multiresolution Hash Encoding“. In: *ACM Trans. Graph.* 41.4 (čvc. 2022), 102:1–102:15. DOI: 10.1145/3528223.3530127. URL: https://doi.org/10.1145/3528223.3530127.

[49] Ricardo Martin-Brualla et al. „NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, s. 7210–7219.

[50] Matthew Tancik et al. „Block-nerf: Scalable large scene neural view synthesis“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, s. 8248–8258.

[51] Jonathan T Barron et al. „Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields“. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, s. 5855–5864.

[52] Anpei Chen et al. „Tensorf: Tensorial radiance fields“. In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer. 2022, s. 333–350.

[53] Matthew Tancik et al. „Nerfstudio: A Modular Framework for Neural Radiance Field Development“. In: *arXiv preprint arXiv:2302.04264* (2023).

[54] Wenzel Jakob et al. *Mitsuba 3 renderer*. Ver. 3.0.1. https://mitsuba-renderer.org. 2022.

[55] Wenzel Jakob et al. *Mitsuba 3 documentation*. [cit. 2023-04-15]. URL: `https://mitsuba.readthedocs.io/en/stable/index.html` (cit. 15. 04. 2023).

[56] Michael Oechsle et al. „Texture fields: Learning texture representations in function space“. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, s. 4531–4540.

[57] Diederik P Kingma a Max Welling. „Auto-encoding variational bayes“. In: *arXiv preprint arXiv:1312.6114* (2013).

[58] Etta D Pisano et al. „Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms“. In: *Journal of Digital imaging* 11 (1998), s. 193–200.

[59] J Alex Stark. „Adaptive image contrast enhancement using generalizations of histogram equalization“. In: *IEEE Transactions on image processing* 9.5 (2000), s. 889–896.

[60] Can Yaras et al. „Randomized histogram matching: A simple augmentation for unsupervised domain adaptation in overhead imagery“. In: *arXiv preprint arXiv:2104.14032* (2021).

[61] Zia-ur Rahman, Daniel J Jobson a Glenn A Woodell. „Multi-scale retinex for color image enhancement“. In: *Proceedings of 3rd IEEE international conference on image processing*. Sv. 3. IEEE. 1996, s. 1003–1006.

[62] Liang Shen et al. „Msr-net: Low-light image enhancement using deep convolutional network“. In: *arXiv preprint arXiv:1711.02488* (2017).

[63] Kin Gwn Lore, Adedotun Akintayo a Soumik Sarkar. „LLNet: A deep autoencoder approach to natural low-light image enhancement“. In: *Pattern Recognition* 61 (2017), s. 650–662.

[64] Partha Pratim Banik, Rappy Saha a Ki-Doo Kim. „Contrast enhancement of low-light image using histogram equalization and illumination adjustment“. In: *2018 international conference on electronics, information, and communication (ICEIC)*. IEEE. 2018, s. 1–4.

[65] Tsung-Yi Lin et al. „Microsoft coco: Common objects in context“. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, s. 740–755.

[66] Alina Kuznetsova et al. „The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale“. In: *International Journal of Computer Vision* 128.7 (2020), s. 1956–1981.

[67] Xuebin Qin et al. „Highly accurate dichotomous image segmentation“. In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*. Springer. 2022, s. 38–56.

[68] Daniel Gatis. *rembg*. [GitHub repo]. URL: `https://github.com/danielgatis/rembg`.

[69] Long Quan. *Image-based modeling*. Springer Science & Business Media, 2010.

[70] Thomas Luhmann et al. *Close range photogrammetry: principles, techniques and applications*. Sv. 3. Whittles publishing Dunbeath, 2006.

[71] Andrea Petruccioli, Francesco Gherardini a Francesco Leali. „Assessment of close-range photogrammetry for the low cost development of 3D models of car bodywork components“. In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 16.2 (2022), s. 703–713.

[72] F Wang et al. „Optimal UAV Image Overlap for Photogrammetric 3D Reconstruction of Bridges“. In: *IOP Conference Series: Earth and Environmental Science*. Sv. 1101. 2. IOP Publishing. 2022, s. 022052.

[73] Johannes Lutz Schönberger. *COLMAP documentation*. [cit. 2023-04-15]. URL: `https://colmap.github.io/` (cit. 15. 04. 2023).

[74] Matthew J Westoby et al. „‘Structure-from-Motion’photogrammetry: A low-cost, effective tool for geoscience applications“. In: *Geomorphology* 179 (2012), s. 300–314.

[75] NVLabs. *Instant NGP*. URL: `https://github.com/NVlabs/instant-ngp`.

[76] Zian Wang et al. „Neural Fields meet Explicit Geometric Representations for Inverse Rendering of Urban Scenes“. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Čvn. 2023.

[77] William E Lorensen a Harvey E Cline. „Marching cubes: A high resolution 3D surface construction algorithm“. In: *ACM siggraph computer graphics* 21.4 (1987), s. 163–169.

[78] G. Bradski. „The OpenCV Library“. In: *Dr. Dobb's Journal of Software Tools* (2000).

[79] Stéfan van der Walt et al. „scikit-image: image processing in Python“. In: *PeerJ* 2 (čvn. 2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453. URL: https://doi.org/10.7717/peerj.453.

[80] AliceVision. *Meshroom*. Ver. 2023.1.0. URL: https://alicevision.org/#meshroom.

[81] Michael Rubloff. *VRAM Calculator*. [cit. 2023-04-19]. URL: https://neuralradiancefields.io/vram-calculator/ (cit. 19. 04. 2023).

[82] Anish Mittal, Anush Krishna Moorthy a Alan Conrad Bovik. „No-reference image quality assessment in the spatial domain“. In: *IEEE Transactions on image processing* 21.12 (2012), s. 4695–4708.

[83] Hamid R Sheikh, Alan C Bovik a Gustavo De Veciana. „An information fidelity criterion for image quality assessment using natural scene statistics“. In: *IEEE Transactions on image processing* 14.12 (2005), s. 2117–2128.

[84] Anish Mittal, Anush K Moorthy a Alan C Bovik. „Blind/referenceless image spatial quality evaluator“. In: *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*. IEEE. 2011, s. 723–727.

[85] Le Kang et al. „Convolutional neural networks for no-reference image quality assessment“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, s. 1733–1740.

[86] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2022. URL: http://www.blender.org.

[87] Lixin Xue. *SfM disambiguation with COLMAP*. [cit. 2023-04-16]. URL: https://github.com/cvg/sfm-disambiguation-colmap (cit. 16. 04. 2023).

[88] Qingan Yan et al. „Distinguishing the indistinguishable: Exploring structural ambiguities via geodesic context“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, s. 3836–3844.

[89] Zhaopeng Cui a Ping Tan. „Global structure-from-motion by similarity averaging“. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, s. 864–872.

[90] Rajbir Kataria, Joseph DeGol a Derek Hoiem. „Improving structure from motion with reliable resectioning“. In: *2020 international conference on 3D vision (3DV)*. IEEE. 2020, s. 41–50.

[91] Jad Abou-Chakra. *NeRFCapture: A tool for streaming posed images*. Ver. 1.0.0. URL: `https://github.com/jc211/NeRFCapture`.

[92] Zirui Wang et al. „NeRF–: Neural radiance fields without known camera parameters“. In: *arXiv preprint arXiv:2102.07064* (2021).

[93] RECON Labs. [cit. 2023-04-25, dostupné po přihlášení]. URL: `https://www.nvidia.com/en-us/on-demand/session/gtcspring23-s51547/` (cit. 25. 04. 2023).

Příloha **A**

Seznam použitých zkratek

CNN Convolutional Neural Network

GAN Generative Adversarial Network

GPU Graphics Processing Unit

HSV Hue Saturaion Value

IBR Image Based Rendering

IoU Intersection over Union

LLM Large Language Model

LOE Light Order Error

MLP Multi-Layer Perceptron

MP Mega Pixel

MPI MultiPlane Image

MVS Multi-View Stereo

NGP Neural Graphics Primitives

NPEA Naturalness Presesrved Enchancement Algorithm

RGB Red Green Blue

RNN Recurrent Neural Network

SAM Segment Anything Model

SDF Signed Distance Function

SfM Structure from Motion

SIFT Scale-Invariant Feature Transform

SPZ Státní Poznávací Značka

YOLO You Only Look Once

Příloha B

Obsah přílohy