



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Vizualizace interiéru pro interaktivní prezentaci
Student:	Tomáš Bilák
Vedoucí:	Ing. Radek Richtr
Studijní program:	Informatika
Studijní obor:	Web a multimédia
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2019/20

Pokyny pro vypracování

Analyzujte základní postupy a techniky zpracování vizuálních dat a možnosti jejich zobrazení ve virtuální realitě. Porovnejte možnosti zobrazení na různých platformách/zařízeních. Navrhněte prototyp (demonstrátor) na základě analýzy ve webovém rozhraní. Prototyp podrobte vhodným testům. Výsledky testů vyhodnoťte.

1. Analyzujte základní postupy a techniky zpracování vizuálních dat.
2. Na základě analýzy zvolte vhodné techniky pro vytvoření modelu interiéru
3. Analyzujte možnosti zobrazení vizualizace ve webovém rozhraní a VR
4. Navrhněte prototyp (demonstrátor) na základě analýzy
5. Prototyp podrobte vhodným testům a vyhodnoťte dosažené výsledky

Seznam odborné literatury

Literatura:

1. Žára, J., VRML 97 - Laskavý průvodce virtuálními světy. Computer Press, Brno, 1999. ISBN: 80-7226-143-6
2. Kraus, T., Accurate 3D model of DCGI corridor, Praha, 2014
3. Žára, J., Jazyky pro popis virtuální reality. Skripta ČVUT, Praha, 2000.
4. Langweilův model Prahy, www.langweil.cz

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 2. března 2018



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalárska práca

Vizualizace interiéru pro interaktivní prezentaci

Tomáš Bilák

Katedra softwárového inženýrství
Vedúci práce: Ing. Radek Richtř

10. januára 2019

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 10. januára 2019

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2019 Tomáš Bilák. Všechny práva vyhrazené.

Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Bilák, Tomáš. *Vizualizace interiéru pro interaktivní prezentaci*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Cieľom tejto bakalárskej práce je vytvoriť čo možno najvernejší model vybraného interiéru fakulty s ohľadom na použitie vo webovom rozhraní. Práca sa zaoberá známymi technikami spracovania vizuálnych dát do 3D prostredia virtuálnej reality. Ďalej je v práci priblížený základ 3D grafiky a spracovania 3D objektov. V analytickej časti práce je tiež rozbor vhodných možností zobrazenia vizualizácie vo webovom rozhraní. Výsledkom praktickej časti práce je prototyp interaktívnej prezentácie vybraného interiéru fakulty, ktorý je otestovaný záťažovými testami na rôznych zariadeniach.

Kľúčová slova A-Frame, VR, AR, Vizualizácia

Abstract

The purpose of this bachelor thesis is to create as authentic model of chosen interior as possible with regard to use in web interface. Thesis describes established techniques of visual data processing for use in 3D virtual reality environment. Further, it outlines basics of 3D graphics and handling of 3D objects. In thesis' analytical part is also breakdown of convenient possibilities of displaying visualization in web interface. Outcome of thesis' practical part is prototype of interactive presentation of chosen faculty interior, which is tested on variety of devices.

Keywords A-Frame, VR, AR, Visualization

Obsah

Úvod	3
1 Cieľ práce	5
1.1 Rozbor zadania	5
2 Analýza možností spracovania obrazových dát	7
2.1 Techniky spracovania	7
2.1.1 3D skenovanie	8
2.1.2 Ručné modelovanie	14
2.2 Porovnanie techník	16
3 Možnosti zobrazenia modelu vo webovom rozhraní	19
3.1 Formáty 3D modelov	19
3.2 Low-level	20
3.3 High-level	21
3.4 ECS	22
3.4.1 A-Frame	22
3.5 Zhrnutie možností	23
4 Návrh a realizácia vizualizácie	25
4.1 Príprava a podklady	25
4.2 Použité techniky	26
4.2.1 Modelovanie	26
4.2.2 Textúrovanie	27
4.2.3 Export	28
4.2.4 VR prezentácia	29
4.2.5 AR prezentácia	29
4.3 Výsledok práce	29
5 Testovanie	33

5.1	Rýchlosť vykresľovania	33
5.2	Dátová náročnosť	34
Záver		35
	Zhodnotenie práce	35
	Možnosti ďalšieho rozšírenia práce	35
Literatúra		37
A Zoznam použitých skratiek		39
B Obsah priloženého CD		41

Zoznam obrázkov

2.1	Ukážka lietajúcich pixelov [1]	10
2.2	Princíp triangulácie [2]	10
2.3	Sken pískačej kačičky spracovaný skenerom Konica Minolta Vivid 9i	11
2.4	Princíp štrukturovaného svetla [3]	12
2.5	Ukážka spracovania skenu štrukturovaným svetlo pre potreby 3D tlače	12
2.6	Vľavo vidíme skladanie hĺbky ostrosti, v strede referenčnú fotogra- fiu, v pravo výsledný model [4]	13
2.7	Ukážka spracovania dát fotogrametriou: a. výstup rekonštrukcie, b. vyčistený a opravený model	14
2.8	Ukážka použitia formátu .GLTF s prehľadom použitých textúr [5]	15
2.9	Ukážka práce na 3D modeloch z rôznych predmetov: a. MGA, b. BLE, c. 3DT	16
4.1	Referenčné obrázky stoličky: a. z pravej strany, b. zo spodu, c. z predu, d. z hora	25
4.2	Pôdorys 14. poschodia budovy A	26
4.3	a. Albedo dlaždica, b. AO textúra, c. Difúzna textúra	28
4.4	Ukážka glTF materiálu pre export	28
4.5	Zobrazenie modelu stoličky vo formáte .GLTF	30
4.6	Výsledná podoba prototypu VR aplikácie	30
4.7	Výsledná podoba prototypu AR aplikácie	31

Zoznam tabuliek

2.1	Hodnotenie techník	16
5.1	Porovnanie FPS na rôznych zariadeniach	34

Zoznam kódov

3.1	Jednoduchá AR scéna[6]	23
-----	----------------------------------	----

Úvod

Vizualizácia sveta, tak ako ho vnímame je téma stará už od nepamäti. Medzi prvé techniky patria praveké jaskynné maľby, či obrúsené horniny, ktoré zobrazovali väčšinou človeka alebo nejakú zver. Neskôr sa ľudia zdokonalili ako v sochárstve, tak aj v maliarstve, vďaka používaniu rôznych nástrojov. Dodnes tak môžeme obdivovať mnoho zachovaných diel ako napríklad, vrcholné dielo renesancie, Michelangelovu sochu Davida. V 19. storočí prichádzajú začiatky analógovej fotografie spojením princípu projekcie obrazu cez malý otvor (camera obscura) a využitia fotosenzitívnych vlastností niektorých látok. Druhá polovica 20. storočia, obdobie mnohých významných objavov, zaznamenáva zrodenie digitálnej fotografie. Po prvýkrát v histórii vieme zachytiť obraz reálneho sveta a zobraziť tieto vizuálne dáta na počítači. Až rok 1992, v ktorom Sir Tim Berners-Lee nahral prvý obrázok¹ na web, znamenal prístupnosť a zdieľanie vizuálnych dát pre všetkých.

V súčasnosti sa k nám denne dostanú stovky megabytov multimediálneho obsahu. Už to však dávno nie je iba statický obraz. Často sledujeme rôzne videá alebo dokonca živé prenosy. Další rozmer a dynamiku stránkam vedia pomocou animácii dodať kaskádové štýly či javascript. Ako je to ale s 3D scénou? Dokážeme vo webovom prostredí zobraziť virtuálnu realitu v reálnom čase? Ako zachytiť scénu okolo nás, tak aby sme ju dokázali zobraziť v limitovanom prostredí webových prehliadačov? Čím sme obmedzovaní? Práve týmito otázkami sa zaoberá táto práca.

¹Obrázok kapely Les Horribles Cernettes

Cieľ práce

Práca rozoberá tému pretvárania reálneho sveta do digitálnej podoby. Analyzuje problémy spracovania obrazových dát a ich reprezentáciu vo virtuálnom prostredí so zameraním na potreby prezentácie vo webovom rozhraní. Ďalej analyzuje formáty uchováajúce 3D modely a technológie umožňujúce ich zobrazenie vo webovom prehliadači. Výsledkom práce je návrh a podrobný popis techník použitých na vytvorenie prototypu zobrazujúceho scénu vybraného interiéru. Na záver je prototyp podrobený testom na rýchlosť vykresľovania a dátovú náročnosť.

1.1 Rozbor zadania

Zo zadanie vyplývajú jasné otázky. Ako pretvoríme objekty z reálneho sveta do toho virtuálneho? Na túto otázku zodpovieme v analytickej časti 2.1, ktorá sa zaoberá technikami spracovania obrazových dát. Ako zobrazíme spracované dáta vo webovom rozhraní pre všetkých? Tejto otázke sa bude venovať kapitola 3, zameraná na analýzu súčasných riešení. Ďalej navrhne spôsob ako vytvárať scény od prázdnej scény v 3D editore až po výslednú scénu zobraziteľnú v prehliadači. Nakoniec vytvorenú scénu otestujeme na rôznych zariadeniach.

Analýza možností spracovania obrazových dát

Táto časť práce sa venuje súčasným spôsobom zachytávania vizuálnych dát a nasledným technikám úpravy získaných dát. Každá podkapitola sa venuje samostatnému spôsobu a jeho prípadným variáciám. Podkapitoly sú obsahovo rozdelené nasledovne:

Princíp získavania dát Popis technologického postupu danej možnosti spracovania

Príklady použitia technológie Poznatky z použitia vybranej technológie z rozsiahlejších projektov v akademickej, či komerčnej sfére

Zhrnutie Rozbor technológie, analýza výhod a nevýhod

V závere kapitoly venujeme jednu podkapitolu porovnaniu jednotlivých technologických postupov s ohľadom na potreby našej práce.

2.1 Techniky spracovania

Existuje niekoľko rôznych techník spracovania obrazu do digitálnej 3D reprezentácie. Musíme si dobre premyslieť akú techniku vyberieme, pretože to so sebou vždy prináša radu výhod, no i nevýhod. Zvolenie nespravnej techniky môže pre projekt znamenať zbytočné plýtvanie časových prostriedkov, finančných prostriedkov alebo až nerealizovateľnosť. Základnými faktormi pre voľbu techniky sú:

Faktory priamo závislé od objektu

- Veľkosť
- Materiál

- Mobilita

Faktory nezávislé od objektu

- Osvetlenie/Počasie
- Okolie objektu
- Prístupnosť

Najväčšou výzvou digitalizácie objektov je zachytiť ich hĺbku na celom povrchu. Každé riešenie má odlišnú mieru škaloateľnosti a flexibility vzhľadom na vyššie uvedené faktory. Pri materiály nás hlavne zaujíma to ako reaguje na osvetlenie. Optické metódy sú veľmi citlivé na príliš reflektívne či priehľadné materiály. Mobilitou myslíme to, keď máme predmet, ktorým nemôžeme hýbať, pretože je veľký alebo nachylný na prenos, máme obmedzené možnosti zbierania dát. Vtedy môžu sofistikované riešenia veľmi ľahko zlyhať.

Faktory nezávislé od objektu sú predovšetkým založené na našej prípave. Či dokážeme zabezpečiť dostatok osvetlenia v prípade, že v bežných podmienkach onoho objektu nie je.

2.1.1 3D skenovanie

Analyzovanie objektu alebo prostredia z hľadiska jeho vzhľadu a zozbieranie dát o jeho tvare, hĺbke a prípadne aj farbe. Aj takto by sme mohli definovať 3D skenovanie. Výsledkom tohto procesu je zväčša mračno bodov, ktoré sa následne rekonštruuje do siete bodov, tvorenej z menších častí, polygónov. Takto vzniknutou sústavou polygónov dostaneme model. V závislosti na technike a zozbieraných dátach, môžeme na tento model priamo naniesť textúru. Na trhu je dostupných množstvo zariadení od ručných skenerov, cez mobilné skenovacie stanice, až po špecifické riešenia pre výrobné linky, či skenovanie lomov. Mohli by sme ich rozdeliť do dvoch kategórií na:

- Kontaktné
- Bezkontaktné

Pre potreby tejto práce môžeme kategóriu kontaktných skenerov vynechať, pretože v praxi sa väčšinou využívajú na meranie, kontrolu a testovanie výsledných produktov. Ďalej však môžeme uvažovať o rozdelení bezkontaktných skenerov na aktívne a pasívne. S výnimkou fotogrametrie spomíname pasívne 3D skenovanie v tejto práci len pre úplnosť. Tieto technológie nevyžarujú na skenovaný objekt žiadnu formu elektromagnetického vlnenia ale naopak iba zachytávajú prichádzajúce žiarenie, najčastejšie viditeľné žiarenie či blízke IR žiarenie, preto sú ich výsledkom často iba aproximácie hĺbky objektu. Pre potreby reprodukcie objektu do virtuálneho sveta sú často nedostačujúce. Napokon sa dostávame ku kategórii aktívnych skenerov. Tá obsahuje niekoľko

rôznych prístupov zbierania dát. Existujú skenery, ktoré získavajú dáta bod po bode, ďalej poznáme skenery, ktoré laserom prechádzajú objekt po pruhoch a napokon tie, ktoré nasvietia objekt nejakým vzorom a analyzujú premietnutý/odrazený vzor na objekte.

Čas letu

Tento princíp spočíva v tom, že vyžiaríme laserom impulz svetla a meriame čas, za ktorý sa nám odrazí naspäť. Keďže poznáme rýchlosť svetla $c = 299\,792\,458\text{ m s}^{-1}$, tak dosadením zmeraného času t do rovnice dostávame vzdialenosť $D = c\frac{t}{2}$. Čas, ktorý sme zmerali, musíme vydeliť 2, pretože ide o čas, za ktorý pulz precestuje od diódy po objekt a naspäť k snímaču.

Rôzne odvetvia využívajú princíp ToF, vďaka tomu, že je veľmi dobre škálovateľný. Dá sa použiť na vzdialenosti niekoľkých kilometrov pri meraní lomů či pohoria, desiatok metrov pri meraní budov, jednotky metrov pri sledovaní pohybu ľudí, či desiatky centimetrov napríklad pre reprodukciu objektu pomocou 3D tlačiarne. Táto technológia je často využívaná v asistovaných službách automobilov, napríklad pri dodržiavaní odstupu od vozidla pred nami. K tejto technológii prešla aj spoločnosť Microsoft v zariadení Kinect pre Xbox One a tiež spoločnosť Apple, ktorá ToF využíva od modelu iPhone X na rozpoznávanie tváre. Tieto použitia sú v systémoch, ktoré pracujú v reálnom čase. Rozlíšenie spracovávaného obrazu je väčšinou pomerne nízke, preto je to vhodné riešenie pre takéto systémy.

Princíp ToF dosahuje veľmi dobre presnosti na stredné a dlhé vzdialenosti, no na menšie vzdialenosti sa presnosťou dostane prinajlepšom do rozmedzia jednotiek milimetrov. [7] Výhodou je rýchla akvizícia dát, bohužiaľ výmenou za menšiu presnosť. Mračno bodov vzniknuté touto technikou navyše často obsahuje artefakty, ktoré vyžadujú rozsiahly post-processing a zložitú rekonštrukciu. Pre mnoho skenovacích zariadení platí, že farba skenovaného objektu má vplyv na presnosť nameraných hodnôt. Takisto veľké odchylky sa objavujú pri skenovaní veľmi lesklých povrchov, či konkávných povrchov. [1] Pri náhlych veľkých rozdieloch v hĺbke sa v nameraných hodnotách objavujú tzv. lietajúce pixely. 2.1 Napokon posledným bežným problémom je šum v dátach vzniknutý prirodzeným osvetlením, či zachytenie zvyškov viacnásobných odrazov svetla od iných častí predmetu.

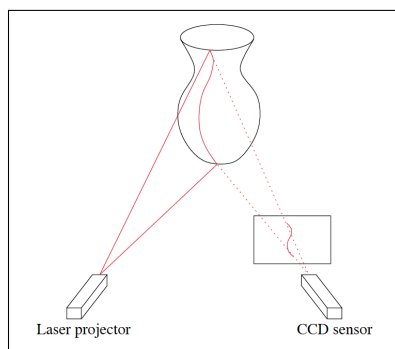
Triangulácia

Základom triangulácie je, ako to z názvu vyplýva, trojuholník. Trojuholník tvorený kamerou, laserom a snímaným predmetom. V tomto trojuholníku poznáme vzájomné pozície kamery a lasera a ich orientáciu. Z nasnímaného obrazu vieme aký uhol zvierajú trojuholník pri kamere. [2] Podľa vety USU, máme všetky informácie potrebné pre dopočítanie hĺbky. Skenery, ktoré využívajú trianguláciu väčšinou premietnu na objekt čiaru, pre urýchlenie procesu získavania



Obr. 2.1: Ukážka lietajúcich pixelov [1]

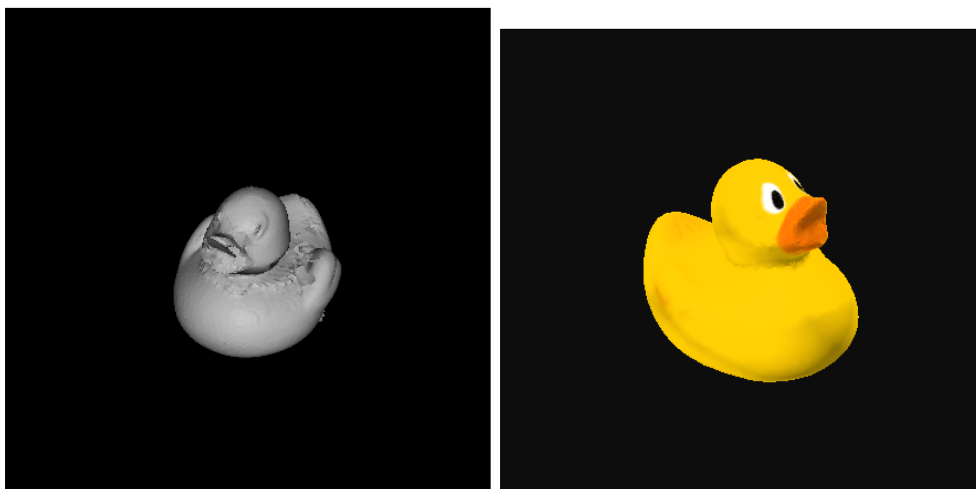
hlbkovej mapy. Prechádzajú predom definovanú plochu. Po nasnímaní celej plochy je potrebné objekt otočiť a snímať znova. Táto časť je často automatizovaná motorizovanou podložkou, čím sa môže výrazne vylepšiť výsledok digitalizácie. V prípade potreby je objekt otočený ručne aby došlo k skenovaniu predtým zakrytých plôch. Ako aj pri predchádzajúcej technológii, tak aj pri triangulácii vznikjú mračná bodov, ktoré je potrebné vyčistiť, zarovnať a zlúčiť.



Obr. 2.2: Princíp triangulácie [2]

V roku 1999 sa vybrala skupina vedená Marcom Levoyom do Florencie s projektom The Digital Michelangelo Project. Vybavení špeciálnym laserovým skenerom začali skenovať sochu Davida. Trvalo to 4 týždne a celkovo urobili 480 individuálnych skenov sochy, z ktorých neskôr vznikol model pozostávajúci z 940 miliónov polygónov. Model bol nameraný s presnosťou na 0.25 mm. [8] Počas štúdia sme tiež mali možnosť vyskúšať si akvizíciu a spracovanie dát touto technológiou so skenerom Konica-Minolta Vivid 9i, v rámci predmetu 3D modelování a virtuální realita vyučovanom na FEL. [9]

Výhodou laserových skenerov je, že ponúkajú lepšie rozlíšenie často až



Obr. 2.3: Sken pískej kačičky spracovaný skenerom Konica Minolta Vivid 9i

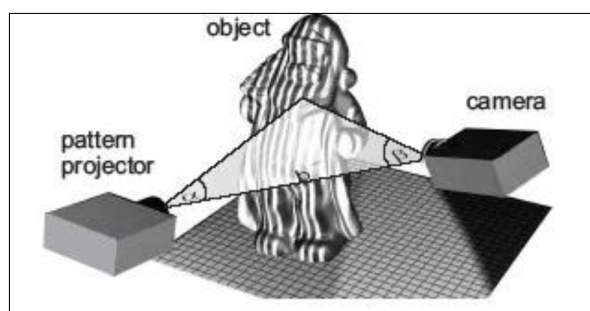
v ráde desiatok μm . Navyše pri skenovaní premietaním čiary na objekt, získavame odhad o tom, ako budú steny orientované. Stačí vypočítať vektorový súčin jednotlivých bodov s ich susednými bodmi, čím sa dá zefektívniť rekonštrukcia objektu.[2] Napriek obmedzenému rozsahu princípu triangulácie, dokážeme vďaka kompaktným a mobilným skenerom získať dáta aj o väčších objektoch, než je skenovacia plocha. Na to je avšak nevyhnutný pomerne drahý hardvér a softvér, ktorý musí sledovať posun skenera a dokáže jednotlivé body priebežne zarovnať a zlúčiť. Niektoré skenery zachytávajú iba informácie o tvare objektu, no nevedia snímať jeho farbu. To je pre potreby tejto práce nevyhnutné, pretože nanášanie textúr na komplikovaný predmet môže byť časovo veľmi náročné a výsledný produkt nemusí vôbec splňať požiadavky.

Štrukturované svetlo

3D skenery využívajúce štrukturované svetlo pozostávajú z kamery (prípadne sústavy kamier) a zdroja svetla, väčšinou projektora. Projektor nasvieti na povrch objektu nejaký špecifický vzor, často pripomínajúci mriežku. Kamera, posunutá o konštantnú vzdialenosť od zdroja, zachytí vzor z inej perspektívy. Následne analyzujeme deformáciu vzoru na zachytenom obraze, z čoho získavame hĺbkove informácie. Vďaka nasvieteniu väčšej plochy vzorom a pomerne zložitému algoritmu dokážeme v krátkom čase získať veľké množstvo dát. [3] Po nasnímaní celého objektu obdobne ako pri predošlej metóde je nevyhnutný rozsiahly post-processing.

Prvá generácia zariadenia Kinect pre Xbox 360 využívala tento princíp. Primárne Kinect nebol vyrobený pre 3D skenovanie, no po sprístupnení Kinect for Windows SDK to už bolo možné. Stal sa tak lacnou alternatívou

2. ANALÝZA MOŽNOSTÍ SPRACOVANIA OBRAZOVÝCH DÁT



Obr. 2.4: Princíp štrukturovaného svetla [3]

k inak značne drahším konkurenčným alternatívam. Samozrejme presnosťou sa Kinect pohybuje, pri meraní na vzdialenosť 1 m, na úrovni 3 mm oproti profesionálnym riešeniam, ktoré dosahujú presnosť pod $50\ \mu\text{m}$. [3, 10] Skenovanie s Kinectom sme si mohli vyskúšať počas cvičení predmetu 3D tisk. Výsledný sken bol spracovaný a vytlačený, môžeme ho vidieť na obrázku 2.5.



Obr. 2.5: Ukážka spracovania skenu štrukturovaným svetlo pre potreby 3D tlače

Pri skenovaní so štrukturovaným svetlom vieme dosiahnúť vysokú presnosť aj vďaka tomu, že na skenovanie má prirodzené osvetlenie nižší vplyv. Skenovanie celej osvetlenej plochy naraz, znamená vysokú rýchlosť v porovnaní s ostatnými spomínanými možnosťami. Medzi nevýhody rozhodne patrí cena profesionálnych zariadení, ktorá sa pohybuje v stovkách tisícov korún. Zároveň sú však stále závislé na vhodných svetelných podmienkach a limitované materiálom skenovaného objektu.

Fotogrametria

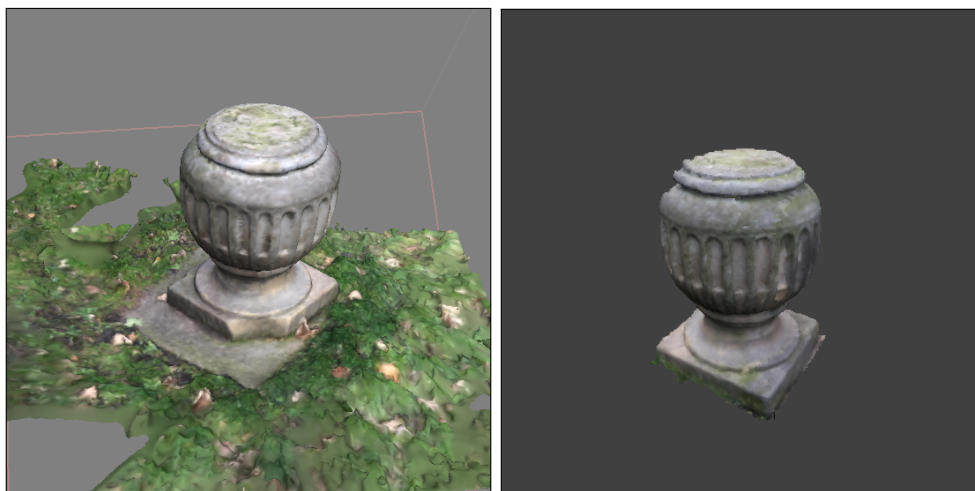
Základom fotogrametrie je meranie vzdielonsti jednotlivých bodov na fotografii. Na to aby sme zrekonštruovali objekt potrebujeme veľké množstvo fotografií, ktoré sa od seba líšia len malým posunom či zmenou uhla pohľadu. Potrebujeme zistiť vzťah medzi jednotlivými obrazmi. Musíme nájsť minimálne 8 významných bodov vo fotografii na zkalibrovanie kamier. Významné body sú body, ktoré sú jednoznačne odlišiteľné od okolia. Keďže je potrebné ich nájsť opakovane v rôznych fotografiách, mali by byť stabilné z hľadiska lokálnych i globálnych deformácií. Často sú to rohy či hrany predmetov alebo zmeny textúr. Na detekovanie významných bodov sa v praxi využívajú algoritmy napríklad SIFT či SURF. Po zarovnaní fotografií získame mračno bodov, z ktorého interpoláciou dostaneme súvislý tvar objektu. Následne z fotiek modelu priradíme textúry.[11]

Dnes sa fotogrametria využíva v mnoho odvetviach. V hernom priemysle pomáha pri prieskume lokácii či vytváraní objektov. Často je využívaná pri mapovaní miest.[12] Langweilov model Prahy bol digitalizovaný pomocou fotogrametrie a špecifických softvérových riešení. Značné komplikácie spôsobilo niekoľko faktorov. Ručne kreslený model dosahuje na malých plochách veľmi vysoké rozlíšenie. V spojení s veľmi obmedzenými svetelnými podmienkami, aby sa predišlo stratám na farebnosti modelu, sa zdala digitalizácia takmer nemožnou. Tím však pozostával z odborníkov z rôznych oblastí a tak prišli so zaujímavými riešeniami. Samotné fotografovanie si vyžadovalo makro fotosústavu, ktorá bola automatizovaná špeciálnym robotom. Mnohokrát bolo nevyhnutné z jedného pohľadu nafotiť niekoľko záberov s rôznou hĺbkou ostrosti a následne fotky zložiť. 2.6 Ak nebolo možné dostať sa k nejakej časti modelu robotom, bol využitý boroskop. [4]



Obr. 2.6: Vľavo vidíme skladanie hĺbkovej ostrosti, v strede referenčnú fotografiu, v pravo výsledný model [4]

S fotogrametriou sme sa stretli aj počas štúdia v predmete 3D modelování a virtuální realita. Vyskúšali sme si tak výhody i úskalia tejto techniky. Presvedčili sme sa, že je možné získať relatívne presný model použitím mobilného telefónu. Model sochy z obrázku 2.7.a vznikol približne z 50 fotografií a po vyčistení a vyplnení dier pozostával z 5597 vrcholov.



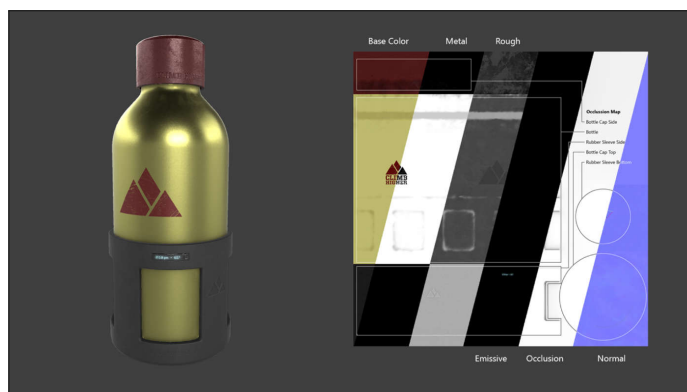
Obr. 2.7: Ukážka spracovania dát fotogrametriou: a. výstup rekonštrukcie, b. vyčistený a opravený model

Fotogrametria prináša jednoduchý a lacný spôsob 3D skenovania. Nie je presná ako aktívne formy 3D skenovania, no zo získaných dát sa ľahšie dajú vyrobiť dobre otexturované modely, čo vie vizuálne pomôcť detailnosti. Dá sa tiež škálovať pre potreby projektu. Nevýhodou je väčšia náchylnosť na chyby v meraní pri zlých svetelných podmienkach či nevyhovujúcich materiáloch. Čo sa týka softvérov na spracovanie fotografií, väčšinou sú pomerne drahé. Existujú dostupné softvéry zdarma, no na výsledkoch to je poznať.

2.1.2 Ručné modelovanie

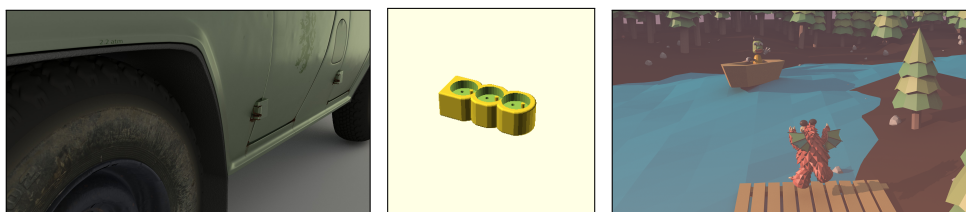
Bez ohľadu na program, v ktorom model vytvárame, si musíme najprv pripraviť vhodné podklady. Je potrebné nafotiť objekt z rôznych strán aby sme mali všetky referenčné informácie. Potom na základe referenčných fotografií začneme vytvárať model zo základných primitívnych tvarov ako napríklad kváder, guľa, štvorec, kruh či trojuholník. Následne model upravujeme pridávaním ďalšej geometrie a jednotlivé vrcholy, hrany či steny transformujeme základnými operáciami posunu, škálovania a rotácie. Modelovacie prostredia ponúkajú mnoho nástrojov na úpravu vytvorenej geometrie. Z jednoduchej kostry, ktorá pripomína modelovaný predmet vieme ľahko vytvoriť podrobnú sieť bodov automatickým rozčlenením povrchu. Môžeme definovať hrany, ktoré majú zostať ostré a takisto tie, ktoré sa majú vyhladiť. Takto podrobnú geometriu môžeme ďalej upravovať napríklad sochárskymi nástrojmi. Aj napriek tomu, že príliš detailny model pravdepodobne nevyužijeme ako finálny produkt, je vhodné ho vytvoriť kvôli zapekaniu textúr na menej detailny model. Po vytvorení modelu podľa predlohy je potrebné nastaviť vlastnosti materiálu. Ďalším krokom

je rozbalenie 3D modelu do 2D mapy pre potreby textúrovania. Keď máme vytvorenú tzv. UV mapu môžeme z referenčných fotiek na model nanášať textúry. Môžeme využiť niekoľko druhov textúr na vylepšenie vzhľadu modelu. Základom je textúra pre difúznu zložku materiálu. Tá však môže pozostávať z dvoch textúr. Jednou z textúr je albedo textúra, ktorá popisuje farebnosť objektu bez ohľadu na osvetlenie. Ďalšou textúrou je AO textúra, ktorá definuje predpočítané tieň, ktoré vrhá objekt sám na seba v malých záhyboch. Pre nepohyblivé časti a pri konštantnom osvetlení môžeme tieň predpočítať aj pre objekty navzájom. Často využívané normálové textúry pomáhajú opticky vylepšiť komplikovanosť predmetu vďaka vypočítavaniu osvetlenia pre každý pixel. Dokážeme tak pridať veľa detailov bez pridávania geometrickej zložitosti, nevýhodou však je, že tento efekt funguje iba pre určité uhly pohľadu. Ak sa pozrieme na plochu z boku, uvidíme, že jej povrch sa reálne nezmenil. Ďalšou možnou textúrou, ktorá sa často využíva pri tvorení PBR fotorealistických scén, je textúra drsnosti. V nej definujeme zmeny na mikroplochách materiálu, ako napríklad odtlačky prstov na lesklom materiáli či mikrodeformácie. Objekt môže mať aj plochu, ktorá žiari, čo definujeme mapou vyžarovania. No a nakoniec môžeme definovať či je na povrchu niektorá časť z kovu. Pretože nekovové časti odrážajú v spekulárnej zložke farbu svetla narozdiel od kovových, ktoré odrážajú farbu svojho povrchu. Výsledný produkt môže vyzeráť takto. 2.8



Obr. 2.8: Ukážka použitia formátu .GLTF s prehľadom použitých textúr [5]

V priebehu štúdia sme sa zaoberali ručným modelovaním na niekoľkých predmetoch. V predmete Multimediálni a grafické aplikácie sme získali základný prehľad práce s modelovacími nástrojmi. Na tieto základy sme mohli ďalej stavať v predmete Blender. Na predmete 3D tisk sme zas skúšali prístup konštruovaním pevnej geometrie z geometrických primitív a množinovými operáciami medzi nimi.



Obr. 2.9: Ukážka práce na 3D modeloch z rôznych predmetov: a. MGA, b. BLE, c. 3DT

	Presnosť	Kontrola	Škálovateľnosť	Dostupnosť	Skúsenosť	Celkovo
ToF	4	5	1	4	5	5
Triangulácia	2	4	4	5	4	4
Štrukturované svetlo	1	3	5	3	3	3
Fotogrametria	3	2	3	2	2	2
Ručné modelovanie	5	1	2	1	1	1

Tabuľka 2.1: Hodnotenie techník

2.2 Porovnanie techník

V predchádzajúcich podkapitolách sme si rozobrali jednotlivé prístupy a ich technologické variácie. Táto podkapitola ich zhrnie a ohodnotí jednotlivé techniky s ohľadom na potreby našej práce. Výsledným produktom majú byť modely zobrazené vo webovom prostredí, teda musíme balansovať na hrane detailu a komplexnosti s kompaktnosťou a efektivitou. Základnými požiadavkami sú:

- Malá veľkosť
- Realistický vzhľad

Potrebuje vytvoriť malé modely aby ich používatelia mohli pohodlne a rýchlo načítať i na mobilných zariadeniach. Preto potrebujeme veľkú kontrolu nad počtom vrcholov, plôch a detailnosťou textúr. Realistický vzhľad väčšinou dosiahneme presným opakom predošlej požiadavky, pretože na to aby sme zobrazili každý detail objektov, potrebujeme detailny model a textúry s vysokým rozlíšením. Musíme využiť čo najviac optimalizačných metód a efektov syntetizujúcich detailnosť.

Pri výbere techniky zohľadňujeme niekoľko aspektov. 1. je presnosť, s ktorou pri hodnotení počítame v ideálnych podmienkach. Presnosť a rýchlosť skenovaním sa teda nemôže rovnať modelovaniu človekom pri komplikovaných predmetoch. Druhým parametrom je kontrola. Tým máme na mysli kontrolu nad komplexnosťou výsledného modelu i jeho priebežnej podoby. Tu má zas umelec navrh oproti skenovacím zariadeniam. Tie často ponúkajú rôzne miery presnosti skenovania, no pri modelovaní postupne pridávame detaily a tak

máme úplnu kontrolu nad LOD výsledného objektu. Škálovateľnosť je z uvedených faktorov najmenej podstatná. Zaujímame sa o virtualizáciu interiéru, teda nemáme veľký rozptyl vo veľkostiach objektov. Dostupnosť hodnotíme všeobecne vzhľadom na hardvér i softvér potrebný pre danú techniku. Nakoniec hodnotíme našu skúsenosť s jednotlivou technikou. Podľa tabuľky 2.1 môžeme vidieť, že pre potreby tejto práce je najvhodnejšou technikou ručné modelovanie. Podľa individuálnych potrieb však môžeme využiť i kombináciu metód. Mohli by sme vytvoriť zložitý model fotogrametriou a modelovaním vytvoriť jeho zjednodušenú podobu, na ktorú by sme potom zapiekli detaily.

Možnosti zobrazenia modelu vo webovom rozhraní

V nasledujúcej kapitole sa budeme venovať formátom vhodným na zobrazenie 3D modelov vo webovom rozhraní a súčasným technológiám schopným zobraziť tieto modely. Najprv venujeme podkapitolu formátom a neskôr v nej zhrnieme poznatky a vyberieme jeden najvhodnejší formát z jednotlivých možností. Ďalej sa budeme v niekoľkých podkapitolách venovať jednotlivým prístupom a technológiám a nakoniec zhrnieme zistenia v podkapitole o ich porovnaní.

Jednotlivé prístupy k zobrazovaniu dát, ktorým sa budeme ďalej v tejto kapitole venovať sú:

- Low-level
- High-level
- ECS

3.1 Formáty 3D modelov

Väčšina modelovacích nástrojov má svoj proprietárny formát, preto nie je možné ich využívať v ďalších aplikáciách od iných vývojárov. Potrebujeme preto zvoliť vhodný formát, ktorý je široko podporovaný a zároveň kompaktný. Okrem toho by mal zvládať popisovanie materiálu a uchovávanie viacerých textúr. To však väčšina formátov nepodporuje, pretože definovanie materiálov veľmi úzko súvisí so samotným vykresľovaním. Ktoré textúry by sa mali v takých formátoch nachádzať a ako by sa mali volať, aby fungovali všeobecne s akýmkoľvek vykresľovacím jadrom? Nedáva to veľmi zmysel a preto sa tým formáty nezaoberajú alebo sú nekonzistentné. Prípadne existujú samostatné formáty popisujúce materiály ako napríklad .MTL. Dnes je pri vytváraní 3D aplikácií, vykresľujúcich v reálnom čase, pomerne často zaužívaný pracovný

postup s ohľadom na PBR. I keď neide o nejaký štandard a existujú rôzne implementácie vykresľovania založeného na fyzikálnom modeli, skupina Khronos vytvorila formát .GLTF, ktorý v sebe môže uchovávať informácie o celej scéne. Vráťane informácií o kamerách, svetlách, ba dokonca v sebe môže niesť dáta o animovaní modelov. Formát má podporu mnohých veľkých spoločností a pri jeho špecifikácii stála i spoločnosť Microsoft.[5] Facebook obohatil svoju sociálnu sieť možnosťou pridávania 3D obsahu pomocou .GLTF súborov do príspevkov.[13]

Kedysi sa o niečo podobné pokúšal formát .WRL s jazykom VRML.[14] Jeho problém však bol, že predbehol svoju dobu. Hardvér ani ľudia ešte neboli pripravení na virtuálnu realitu v prehliadači. Neskôr sa naň pokúsil nadviazať formát .X3D, no nemal o mnoho väčší úspech. [15] Dnes sme ale o kus ďalej. Takmer každý má vo vrecku telefón, ktorý je schopný zobrazovať aspoň jednoduchú scénu.

V júni 2018 predstavila spoločnosť Apple nový formát .USDZ podporujúci popis scény zameraný primárne na AR aplikácie. Ide o archív založený na otvorenom formáte .USD spoločnosti Pixar. Vďaka tomu, že je založený na otvorenom formáte, je dostupný aj iným aplikáciám a operačným systémom. Spoločnosť Adobe bola súčasťou vytvárania tohto formátu a informovala i o vývoji prvých nástrojov na úpravu a výrobu prvkov pre AR aplikácie.

Zo získaných informácií hodnotíme, že najvhodnejším formátom bude zrejme .GLTF, pretože vznikol s ohľadom na rovnaké parametre, aké vyžadujeme i v tejto práci. Navyše je pomerne dobre rozšírený a podporovaný rôznymi webovými frameworkmi. Formát .USDZ vyzerá sľubne, no momentálne je zobraziteľný iba pomocou ARKitu pre macOS a iOS zariadenia v prehliadači Safari.

3.2 Low-level

Tento prístup je založený na využívaní WebGL API. WebGL 2.0 stojí na základoch OpenGL ES 3.0 a pomocou shaderov prináša 3D priamo do prehliadačov bez potreby dodatočných zásuvných modulov. Podľa skratky ES vidíme, že ide o OpenGL pre vstavané systémy. Nad WebGL je Javascriptový riadiaci kód. Je podporovaný všetkými populárnymi prehliadačmi. [16] Dáva nám pomerne veľkú slobodu a kontrolu nad výsledkom aplikácie zobrazujúcej vizualizáciu, no za relatívne vysokú cenu. Obyčajná scéna s jednoduchou osvetlenou kockou a nejakým materiálom znamená desiatky volaní WebGL API.

WebGL využíva HTML 5 `canvas` element, vďaka ktorému dokáže vykresľovať naše modely v prehliadači. Pre vykreslenie však aplikácia musí urobiť niekoľko krokov:

1. Vytvoriť `canvas` element
2. Získať kontext `canvas` elementu

3. Inicializovať `viewport`
4. Vytvoriť aspoň jeden buffer s dátami na vykreslenie
5. Definovať aspoň jednu maticu pre transformáciu vrcholov do obrazového priestoru
6. Vytvoriť aspoň jeden shader implementujúci vykresľovací algoritmus
7. Inicializovať shader s parametrami
8. Vykresliť

Podrobný popis jednotlivých krokov nájdeme v knihe Tonyho Parisiho [17] ale i v množstve online kurzov zameraných na WebGL.

3.3 High-level

Pre zjednodušenie komplikácií vyplývajúcich z predošlého prístupu vznikli veľké množstvá knižníc poskytujúcich istý stupeň abstrakcie nad WebGL. Jedným z najpoužívanejších riešení je Three.js. Ďalšími knižnicami podobnými Babylon.js. Výhodami používania knižnice sú napríklad:

- Rozšíriteľnosť
- Overená stabilita a výkon komunitou
- Podpora interakcií
- Zjednodušenie najpoužívanejších matematických operácií
- Podpora množstva formátov

Tieto výhody majú prakticky všetky súčasné knižnice. Mnoho knižníc má veľmi dobre spracované dokumentácie vrátane príkladov použitia z komunity. Oproti low-level prístupu sa môžeme sústrediť na podstatnú časť našej úlohy a značne si tak uľahčiť prácu. Pri tomto prístupe potrebujeme na vykreslenie scény tieto kroky:

1. Vytvoriť vykresľovacie jadro
2. Vytvoriť scénu
3. Vytvoriť kameru
4. Načítať geometriu modelu
5. Načítať materiál (jednu či viac textúr)
6. Vytvoriť model a priradiť mu geometriu a materiál

7. Pridať model do scény

8. Vykresliť

Takto zipsaný číslovaný zoznam má síce rovnaký počet bodov ako bodový zoznam pri low-level princípe, no tu každý bod naozaj predstavoval prevažne jeden riadok kódu. Kód je vďaka tomu prehľadnejší a menej náchylný na chyby.

3.4 ECS

Vo vývoji VR aplikácií sa veľakrát využíva práve ECS architektúra. Architektúra je zameraná na kompozíciu objektov oproti princípu dedenia a hierarchie. Medzi jej výhody patrí:

- Flexibilita pri definovaní objektov a miešaní opakujúcich sa častí
- Čistejší dizajn vďaka odstráneniu väzieb, zapúzdrovaniu, modularizácii a znovupoužitelnosti
- Odstránenie problémov pri dlhých reťaziach dedičností s prepojenými komplexnými funkcionalitami
- Rozšíriteľnosť o nové funkcionality a zdieľanie komunitných componentov

3.4.1 A-Frame

Práve touto architektúrou je modelovaná knižnica A-Frame, ktorá sa zameriava špecificky na 3D a VR aplikácie vo webovom rozhraní. Jadro knižnice A-Frame je postavené na Three.js, čím získava všetky výhody poskytujúce knižnicou Three.js, ktoré ďalej rozširuje. V knižnici sú jednotlivé časti ECS architektúry využívané nasledovne:

Entity Reprezentované `<a-entity>` elementom a prototypmi

Komponenty Reprezentované HTML atribútmi na `<a-entity>` elementoch. Ďalšie componenty môžeme registrovať cez `AFRAME.registerComponent(name, definition)` API

Systémy Reprezentované atribútmi `<a-scene>` elementu. Sú podobné komponentom a môžeme ich registrovať cez `AFRAME.registerSystem(name, definition)` API

Výhodou deklaratívneho popisu je prehľadné a ľahko pochopiteľné HTML. Ďalšiu výhodu vidíme v možnosti ľahko generovaných scén.

AR.js

Jerome Etienne vytvoril AR.js komponent pre A-Frame založený na knižniciach Three.js a artoolkit. Knižnica artoolkit vyhledáva v obraze snímanom kamerou známu značku, ktorú môžeme definovať vzorom či 2D čiarovým kódom, a na ňu premietne 3D model. Silou AR.js je to, že ide o multiplatformný nástroj a vôbec nemá vysoké hardvérové nároky. V tejto ukážke kódu 3.1 môžeme vidieť ako jednoducho sa dá pár riadkami pridať AR komponentu do A-Frame scény. Stačí pridať do scény systém `arjs` a doplniť scénu o entitu `<a-marker>`, v ktorej definujeme scénu, ktorá sa má na značke zobrazit'. Nakoniec pridáme entitu s komponentom `camera` a základná scéna je hotová.

```

1 <!doctype HTML>
2 <html>
3 <script src="https://aframe.io/releases/0.6.1/aframe.min.js"></script>
4 <script src="https://cdn.rawgit.com/jeromeetienne/AR.js/1.5.0/aframe/build/aframe-ar.js"></script>
5 <body style='margin : 0px; overflow: hidden;'>
6   <a-scene embedded arjs>
7     <a-marker preset="hiro">
8       <a-box position='0 0.5 0' material='color: black;'></a-box>
9     </a-marker>
10    <a-entity camera></a-entity>
11  </a-scene>
12 </body>
13 </html>

```

Kód 3.1: Jednoduchá AR scéna[6]

3.5 Zhrnutie možností

V predchádzajúcich podkapitolách sme analyzovali aké formáty sú vhodné na použitie vo webovom rozhraní. Sekcia 2.2 sa venovala najvhodnejšiemu spracovaniu dát a načrtli sme tam podstatné požiadavky na výsledné modely. Pre optimalizovanie dátového toku i záťaže servera sme rozoberali hlavne formáty, ktoré v sebe nesú informácie o objekte a jeho vzhľade, či dokonca celej scéne. Formát .GLTF skupiny Khronos je otvorený, štandardne využívaný v rôznych multiplatformných webových aplikáciách a bol vyvinutý práve na tento účel. Jeho jediným ekvivalentným konkurentom je formát .USDZ, ktorý však v súčasnosti je podporovaný iba platformami spoločnosti Apple. Jednoznačne nám teda vychádza .GLTF ako najvhodnejší formát.

Pokračovali sme rozborom vývoja prezentačnej aplikácie pre webové rozhranie. Definovali sme tri prístupy vývoja. Low-level prístup využívajúci

3. MOŽNOSTI ZOBRAZENIA MODELU VO WEBOVOM ROZHRAŇÍ

WebGL API a riadiaci Javascriptový kód. Tento spôsob je trochu ako objavovanie kolesa. Na niektoré špecifické problémy je určite vhodný, no pre potreby tejto práce ho môžeme zavrhnúť. Ďalším prístupom bolo využitie nejakej knižnice, ktorá abstrahuje od konkrétností rozhrania WebGL. Takýchto knižníc existuje veľa a ponúkajú zhruba podobné možnosti. Prístup je to vhodný, no odporúčame 3. možnosť, využitie ECS architektúry. Táto práca rozhodne nevyužije plný potenciál tohto návrhového vzoru, no v sekcii 5.2 popíšeme aké ďalšie využitie si vieme predstaviť.

Návrh a realizácia vizualizácie

Kapitola popisuje postup zvolený pri vytváraní vizualizácie. Na začiatku sa budeme v jednej podkapitole venovať prípravám na projekt. Popíšeme aké podklady sme si pripravili, či mali k dispozícii. V nasledujúcej podkapitole rozoberieme ako sme vytvárali modely a scénu. Posledná podkapitola prezentuje výsledok vizualizácie, a popisuje krátke zhrnutie autora.

4.1 Príprava a podklady

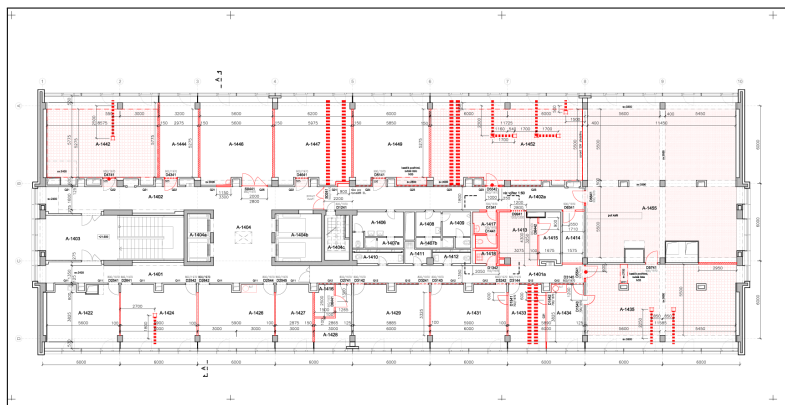
V analýze v sekcii 2.2 sme zistili, že pre potreby tejto práce je najvhodnejšou technikou spracovania ručné modelovanie. Výnimkou by mohli byť veľmi komplexné tvary. V tom prípade by sme mohli zvoliť fotogrametriu a následne model rekonštruovať a postupne ho decimovať až do prijateľnej podoby. Vybrali sme sa na obhliadku miestností A1446. Zistili sme, že v miestnosti je pomerne mnoho jednoduchých až stredne zložitých predmetov. Začali sme si vytvárať referenčné obrázky. Niektoré predmety boli dobre dostupné a tak sme boli schopní vytvoriť si výborné referenčné obrázky 4.1. Po získaní dostatočných obrazových predlôh sme sa mohli pustiť do ďalšej fáze. Namerali sme si niektoré hodnoty



Obr. 4.1: Referenčné obrázky stoličky: a. z pravej strany, b. zo spodu, c. z predu, d. z hora

4. NÁVRH A REALIZÁCIA VIZUALIZÁCIE

Ako podklady pre vytvorenie samotnej miestnosti sme využili okrem fotiek využili pôdorys, ktorý nám dodal Ing. Chludil. Na základe pôdorysu 4.2 máme presné údaje o rozmeroch miestnosti.



Obr. 4.2: Pôdorys 14. poschodia budovy A

4.2 Použité techniky

4.2.1 Modelovanie

Na modelovanie sme použili Blender. Pri vytváraní modelov začneme pripevnením referenčných fotografií na jednotlivé im odpovedajúce pohľady v ortografickom zobrazení (3D view > Properties N > Background image). Potom vložíme do scény jedno zo základných modelovacích primitív (add <shift>+A), podľa toho, ktoré sa nám na daný objekt najviac hodí. Pokračujeme upravovaním jednotlivých bodov, hrán či plôch podľa predlohy. Na úpravu výberu bodov väčšinou používame 3 základné transformácie, ktorými sú posun (translate G), rotácia (rotate R) a škálovanie (scale S). Ďalším využívaným nástrojom je vytlačanie (extrude E). Tieto nástroje sa úpravy zo začiatku opakujeme často po sebe až pokiaľ objekt nezačne pripomínať požadovaný tvar. Ak má objekt v sebe nejakú dieru, ako napríklad operadlo stoličky, môžeme doň vyrezať dieru nožom (knife K). Pri operadle sme využili to, že drevo má všade rovnakú hrúbku, preto sme vytvorili iba plochu jednej strany a pridali jej hĺbku modifikátorom spevnenia (solidify). Po vytvorení jednoduchého modelu si model duplikujeme, jednu verziu skryjeme a pokračujeme v pridávaní detailov na druhej. Detaily väčšinou pridávame zhusťovaním a spresňovaním pomocou nástroja pridania hrán po obvodu (loop cut and slide <ctrl>+L) a rozdelenia plochy na podplochy (subdivide). Na hranách môžeme definovať hodnotu záhybu (crease <shift>+E), ktorá sa pohybuje v rozmedzí $(-1, 1)$ a definuje či má byť hrana zaoblená alebo ostrá. Po nastavení hrán podľa potrieb môžeme použiť modifikátor rozdelenia plochy (subdivision surface <ctrl>+0).

Výhodou modifikátora je, že môžeme ďalej upravovať jednoduchý model a sledovať zmeny modifikátora. To nám pomáha rýchlo vymodelovať veľký a komplexný model.

4.2.2 Textúrovanie

Ďalším krokom je pridávanie textúr na modely. Prvý krok, ktorý musíme urobiť je rozbalenie (Edit > unwrap U) modelu. Niekedy stačí model jednoducho rozbaľiť. To je ale veľmi výnimočné, stačí to zväčša iba pri jednoduchých tvaroch. Najprv si teda vyberieme hrany a označíme švy (mark seam). Pri výbere hrán sa snažíme predstaviť si model z papiera a hľadáme hrany tak aby sme po rozstihnutí hrany vedeli model rozprestrieť do roviny. Po označení hrán a rozbaľení modelu môžeme zapieť normálovú textúru podľa komplikovaného modelu na ten jednoduchý. Musíme vykonať tieto kroky:

1. Vytvoríme si nový obrázok v editore blenderu (Image editor > New image <alt>+N)
2. Pridáme tento obrázok do materiálu modelu v uzlovom editore (Node editor > Add <shift>+A > Image texture)
3. Označíme pridaný uzol
4. Označíme komplexný model v grafe scény (Outliner)
5. Spolu s komplexným modelom označíme i ten zjednodušený (držaním tlačidla <shift>)
6. V nastaveniach vykresľovania, v časti zapekania, zvolíme zapieť z objektu s vysokým rozlíšením (Render > Bake > Bake with Multires)
7. Typ zapekania zvolíme pre normálovú mapu
8. Stlačíme zapieť

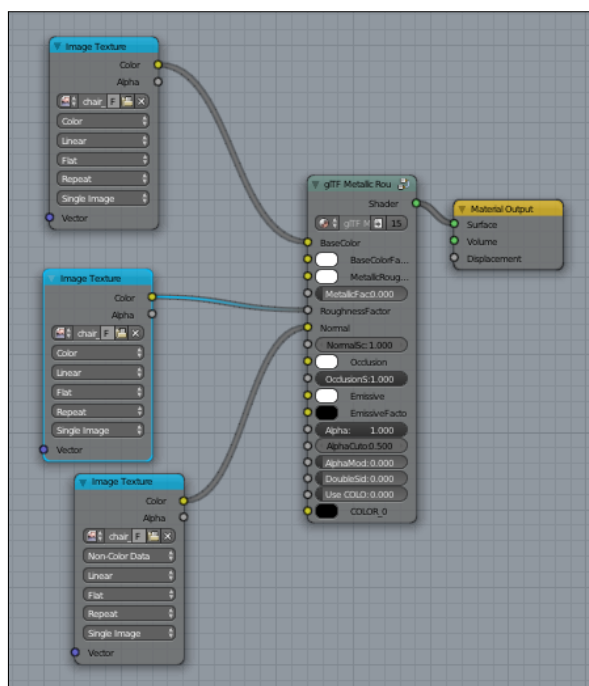
Ďalšou textúrou, ktorú môžeme vytvoriť zapečením, je AO textúra. Tu nám stačí pridať ďalší obrázok, nastaviť materiál na bielu farbu a v nastaveniach sveta zaškrtnúť AO. Pokračujeme vytvorením albedo textúry. Tú si môžeme naniesť na model priamo v editore obrázkov, na náš rozložený model alebo môžeme kresliť priamo na model v 3D pohľade v móde kreslenia textúr. Ďalším spôsobom je exportovať UV textúru a upraviť ju v inom editore. Ako príklad uvedieme vytvorenie textúry na zemi. Z referenčného obrázka sme si v rastrovom editore vystrihli časť podlahy. Upravili sme farebnosť obrázka aby sme sa zbavili odleskov, či silných tieňov. Potom sme obrázok rozstihli na štvrtiny a dvojice po diagonálach navzájom prehodili. Klonovacím nástrojom sme vyhladili spojené čiary v strede. Takto sme si vytvorili dlaždicu. Tú sme potom vložili do Blendera a nastavili opakovanie textúry. Zapiekli sme aj AO textúru a spojili vo výslednú difúznú textúru 4.3.c.



Obr. 4.3: a. Albedo dlaždica, b. AO textúra, c. Difúzna textúra

4.2.3 Export

Po vytvorení modelov a nachystaní textúr sa môžeme pustiť do exportovania modelov. Využívame prídavný komponent pre Blender na export do .GLTF formátu. Pre Blender 2.79 je potrebné priložiť komponent priamo do zložky komponentov a potom v projekte pridať odkaz na materiál, ktorý exportér využíva. Materiál vložíme v uzlovom editore (Node editor > Add <shift>+A > Group > glTF Metallic Roughness) a spojíme s textúrami 4.4.



Obr. 4.4: Ukážka glTF materiálu pre export

4.2.4 VR prezentácia

Najprv si pridáme do A-Frame scény všetky potrebné modely ako elementy `<a-asset-item>` pod element `<a-assets>`. Pre každý element `<a-asset-item>` je zavolaný `three.js FileLoader` aby zaistil načítanie súborov. Scéna sa začne vykresľovať, keď loader vráti pre každý prvok návratovú hodnotu úspechu či neúspechu. Ďalej pridáme do scény entitu s komponentom osvetlenia `light` a nastavíme jednotlivé hodnoty. Pridáme na scénu vytvorené modely `<a-gltf-model>` a nastavíme im zdroj na id svojho prvku z elementu `a-asset`. Pridáme im komponent tieňa a nastavíme ho pre prijímanie i vrhanie. Nakoniec pridáme na scénu kameru a pridáme jej komponenty pre pohyb klávesami `wasd-controls` a interakciu na základe pohybu zariadenia `look-controls`.

4.2.5 AR prezentácia

Postupujeme obdobne ako v predošlom príklade, no navyše ešte pridáme skript pre `AR.js`. Pridáme do scény systém pre `ar.js`. Ďalej pridáme našu vlastnú značku `<a-marker>` a vložíme doň scénu. Nakoniec pridáme kameru, no tentoraz bez komponentov pre pohyb, pretože kamerou hýbeme rukami a o synchronizáciu scény a kamera sa stará `AR.js` pomocou značky.

4.3 Výsledok práce

Táto sekcia je venovaná obrazovým ukázkam jednotlivých častí práce. V obrázku 4.5 uvádzame príklad výstupu z exportu modelu do `.GLTF` formátu. Nasleduje ukážka z VR aplikácie na obrázku 4.6. No a na záver ukážka zobrazenia AR aplikácie na mobilnom zariadení na obrázku 4.7.

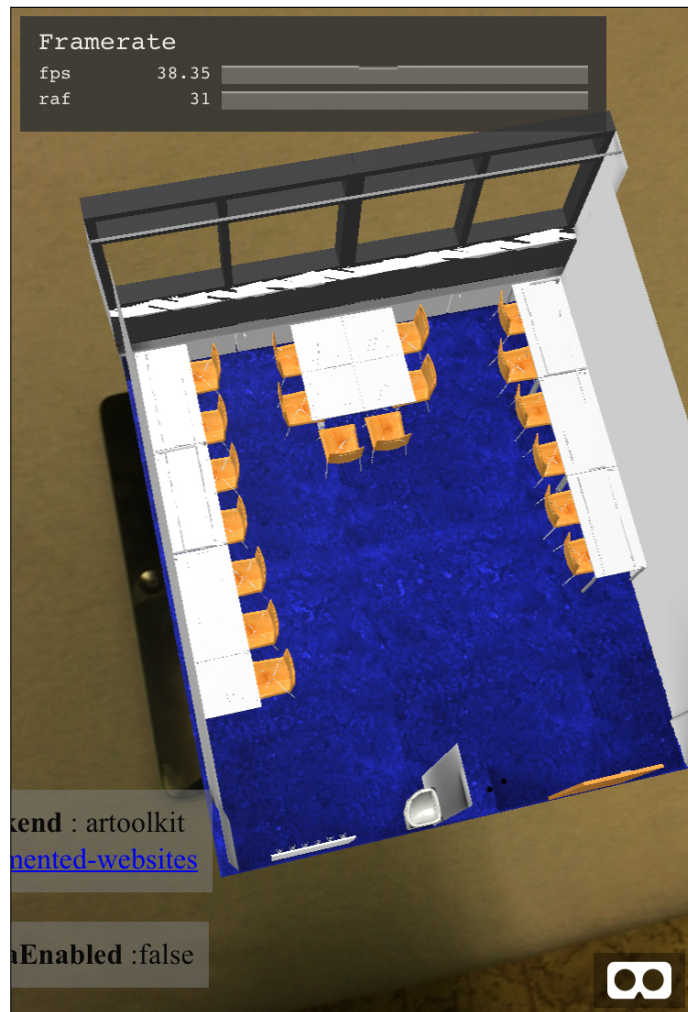
4. NÁVRH A REALIZÁCIA VIZUALIZÁCIE



Obr. 4.5: Zobrazenie modelu stoličky vo formáte .GLTF



Obr. 4.6: Výsledná podoba prototypu VR aplikácie



Obr. 4.7: Výsledná podoba prototypu AR aplikácie

Testovanie

V tejto kapitole sa budeme venovať testovaniu výslednej aplikácie a jej použitiu na rôznych zariadeniach a platformách. Najprv zhodnotíme aplikácie výkonnostne a následne ich rozoberiem na základe dátovej náročnosti.

5.1 Rýchlosť vykresľovania

Vykresľovanie hodnotíme na základe vykreslených obrazov za sekundu (FPS). Aplikácie by mali byť dostupné pre širokú verejnosť. Budeme ich testovať na starších zariadeniach konkrétne:

- iPhone 6
- iPad (4th gen.)
- Macbook Pro 13 (Unibody, Early 2011)

Každé zariadenie najprv reštartujeme, necháme ho pár minút bežať a potom spustíme jeden z troch najpoužívanejších prehliadačov, v ktorom si zobrazíme scénu:

- Google Chrome
- Mozilla Firefox
- Apple Safari

Výsledky testov sme zhrnuli v tabuľke 5.1. Zistili sme, že staršie zariadenia sú vhodné na prezeranie virtuálnych scén, no neodporúčame ich na použitie vo VR cardboard. Trochu prekvapivé sú rozdiely vo výsledkoch medzi mobilným telefónom a tabletom. Hardvérovými špecifikáciami sú si podobné. Kľúčové rozdiely budú zrejme zapríčinené tým, že daný tablet má 32 bitovú architektúru, teda tiež nemá najnovší operačný systém. Ďalším faktorom bude rozdiel v rozlíšení displejov.

	Chrome	Firefox	Safari
iPhone 6	~ 28	~ 27	~ 27
iPad (4th gen.)	~ 6	~ 6	~ 6
Macbook Pro 13	~ 19	~ 30	~ 15

Tabuľka 5.1: Porovnanie FPS na rôznych zariadeniach

5.2 Dátová náročnosť

Dátovú náročnosť hodnotíme s ohľadom na celkovú veľkosť dát potrebných na zobrazenie kompletnej scény. Celková scéna vrátane Javascriptových knižníc nepresiahne 10 MB, čo hodnotíme ako uspokojivé a dostupné pre mobilné zariadenia.

Záver

Zhodnotenie práce

Práca podľa nás naplňa podstatu zadania. Najprv podrobne rozoberá možnosti digitalizácie dát, neskôr zas možnosti zobrazovania dát vo webovom rozhraní. Časť o modelovaní popisuje dnešné praktické prístupy pri vytváraní 3D obsahu. Ďalej je popísaný postup a zvolená technológia pre výsledný prototyp prezentácie. Na záver je prevedený test rýchlosti vykresľovania ako kontrola kompatibility s bežnými zariadeniami.

Možnosti ďalšieho rozšírenia práce

Vidíme v navrhnutom postupe veľký potenciál PR nástroja, ktorý by sa dal aplikovať pre FIT. Predmety ako Blender, či Multimediálne grafické aplikácie by mohli študentom na cvičeniach dávať ako zadanie vizualizáciu nejakých priestorov či nábytkov. Tie by sa mohli agregovať a následne by z toho mohla vzniknúť knižnica, z ktorej by dali jednoducho generovať scény jednotlivých miestností, či celých poschodí.

Literatúra

- [1] Wasenmüller, O.; Stricker, D.: Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision. In *Computer Vision – ACCV 2016 Workshops*, editace C.-S. Chen; J. Lu; K.-K. Ma, Cham: Springer International Publishing, 2017, ISBN 978-3-319-54427-4, s. 34–45.
- [2] Bernardini, F.; Rushmeier, H.: The 3D Model Acquisition Pipeline. *Computer Graphics Forum*, ročník 21, č. 2: s. 149–172, doi:10.1111/1467-8659.00574, <https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.00574>. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.00574>
- [3] Georgopoulos, A.; Ioannidis, C.; Valanis, A.: ASSESSING THE PERFORMANCE OF A STRUCTURED LIGHT SCANNER. 01 2010.
- [4]
- [5] Inc., T. K. G.: glTF - Runtime 3D Asset Delivery. Apr 2013. Dostupné z: <https://www.khronos.org/glTF/>
- [6] Etienne, J.: Augmented Reality for the Web. Dostupné z: <https://jeromeetienne.github.io/AR.js/>
- [7] Remondino, F.: Heritage Recording and 3D Modeling with Photogrammetry and 3D Scanning. *Remote Sensing*, ročník 3, č. 6, 2011: s. 1104–1138, ISSN 2072-4292.
- [8] Levoy, M.: The Digital Michelangelo Project. Jul 2009. Dostupné z: <https://accademia.stanford.edu/mich/>
- [9] Sedláček, D.: 4. Cvičení - 1. model do galerie - Laser Scan. Dostupné z: <https://cw.fel.cvut.cz/old/courses/a7b39mvr/tutorials/04b/start>

- [10] Smisek, J.; Jancosek, M.; Pajdla, T.: *3D with Kinect*. 01 2013, ISBN 978-1-4471-4639-1, s. 3–25.
- [11] Sedláček, D.: 3D REKONSTRUKCE.
- [12] Friends, N. a.: Apr 2017. Dostupné z: https://www.youtube.com/watch?v=suo_aUTUpps
- [13] Gönen, A.: Richer 3D Posts on Facebook and New Ways to Share. Dostupné z: <https://developers.facebook.com/blog/post/2018/02/20/3d-posts-facebook/>
- [14] Žára, J.: *VRML 97*. Praha: Computer Press, vyd. 1 vydání, 1999, ISBN 80-7226-143-6.
- [15] Web3D Consortium Open Standards for Real-Time 3D Communication. Dostupné z: <http://www.web3d.org/x3d/what-x3d>
- [16] WebGL - OpenGL ES for the Web. Jul 2011. Dostupné z: <https://www.khronos.org/webgl/>
- [17] Parisi, T.: *WebGL*. Sebastopol, Calif.: O'Reilly, první vydání, c2012, ISBN 978-1-449-32357-8.

Zoznam použitých skratiek

- 3D** Trojdimenzionálny
- AO** Ambient Occlusion
- API** Application Programming Interface
- AR** Augmentovaná realita
- ECS** Entity component system
- FEL** Fakulta elektrotechnická
- FIT** Fakulta informačných technológií
- FPS** Frames per second
- HTML** Hypertext Markup Language
- IR** Infračervený
- LOD** Level of detail
- PBR** Physically based rendering
- SDK** Software development kit
- ToF** Time of flight
- VR** Virtuálna realita
- VRML** Virtual reality modeling language

Obsah priloženého USB

	readme.txt.....	stručný popis obsahu USB kľúča
	src	
	impl	zdrojové kódy implementácie
	thesis	zdrojová forma práce vo formáte \LaTeX
	text	text práce
	thesis.pdf	text práce vo formáte PDF