

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MOŽNOSTI A CHARAKTERISITIKA SKENERA
CRUSE A VYUŽITIE LOKÁLNYCH TIEŇOV
NA ANALÝZU SKENOVANÝCH OBJEKTOV
BAKALÁRSKA PRÁCA

2019
MIKULÁŠ POLÁK

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MOŽNOSTI A CHARAKTERISITIKA SKENERA
CRUSE A VYUŽITIE LOKÁLNYCH TIEŇOV
NA ANALÝZU SKENOVANÝCH OBJEKTOV
BAKALÁRSKA PRÁCA

Študijný program: Informatika
Študijný odbor: 2508 Informatika
Školiace pracovisko: Katedra algebry, geometrie a didaktiky matematiky
Školiteľ: RNDr. Róbert Bohdal, PhD.

Bratislava, 2019
Mikuláš Polák



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Mikuláš Polák
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Možnosti a charakteristika skenera CRUSE a využitie lokálnych tieňov na analýzu skenovaných objektov
Possibilities and Features of CRUSE Scanner and use of local shadows to analyze scanned objects

Anotácia: Charakterizácia riadiacich parametrov skenera CRUSE (technické vlastnosti skenera CRUSE, riadenie skenovania a spracovanie skenov. Skenovacie režimy, osvetlenie a lokálne tieňe v digitálnom obraze). Vytváranie reliéfu, resp. poľa výškových bodov pomocou analýzy obrazov získaných skenovaním.

Cieľ: Cieľom projektu je naštudovať a opísať technické a informatické vlastnosti skenera CRUSE, navrhnúť súbor experimentov a využiť informáciu o lokálnych tieňoch na vytvorenie reliéfu, t.j. priestorovej geometrie naskenovaného originálu. Inými slovami, z geometricky a rádiometricky presných 2D skenov hľadáme výškovú informáciu. Navrhnuté algoritmy ohodnotíme aj z hľadiska presnosti a výpočtovej zložitosti.

Literatúra: CRUSE Scanner. 2018. Firemná dokumentácia. Cruse software CSx 3.9. MANUAL, tutorialy...
HUGHES, J.F. et al. 2013. Computer Graphics. Third edition. Addison Wesley.
RADZIM, P. 2018. Poznámky k školeniu CRUSE. Bratislava, FMFI UK.
Vybrané články z vedeckých konferencií.

Kľúčové slová: digitálna fotografia, skenovanie, spracovanie obrazu, 3D rekonštrukcia

Vedúci: RNDr. Róbert Bohdal, PhD.

Katedra: FMFI.KAGDM - Katedra algebry, geometrie a didaktiky matematiky
Vedúci katedry: doc. RNDr. Pavel Chalmovianský, PhD.

Spôsob sprístupnenia elektronickej verzie práce:
bez obmedzenia

Dátum zadania: 26.10.2018

Dátum schválenia: 27.10.2018

doc. RNDr. Daniel Olejár, PhD.
garant študijného programu

Podakovanie: Ďakujem vedúcemu bakalárskej práce RNDr. Róbertovi Bohdalovi, PhD. za odbornú pomoc pri jej vypracovávaní.

Abstrakt

V práci skúmame algoritmus využívajúci analýzu tieňov pre rekonštrukciu povrchu 3D objektov. Naším cieľom je vytvorenie programu, ktorý bude spracovávať výstupné obrázky zo skenera CRUSE a na základe tieňov skonštruuje 3D model skenovaného objektu. Pre plné využitie potenciálu vysokoformátového skenera najprv opíšeme vlastnosti a spôsoby skenovacích procesov a jednotlivých parametrov a preferencií nastavitelných pri používaní skenera. Následne vykonáme sériu skenovaní rôznych typov objektov pri rôznom osvetlení a implementujeme algoritmus na spracovanie obrázkov. Celý program prispôbíme na konkrétne zariadenie, opíšeme fungovanie jednotlivých tried a zhodnotíme efektívnosť a presnosť celého programu.

Kľúčové slová: digitálna fotografia, skenovanie, spracovanie obrazu, 3D rekonštrukcia

Abstract

In this work we analyze algorithm for 3D surface reconstruction using object's shadow. Our goal is to create a program that will process the output images from the CRUSE scanner and build a 3D model of the scanned object based on shadows. To take full advantage of the high-quality scanner's capabilities, we will first describe the features and methods of scanning processes and individual parameters and settings that can be applied when using the scanner. Subsequently, we perform a series of scans of different types of objects under different lighting and implement an image processing algorithm. We will adjust the whole program to a specific device, describe the operation of individual classes and evaluate the effectiveness and accuracy of the entire program.

Keywords: digital photography, scanning, image processing, 3D reconstruction

Obsah

Úvod	1
1 Skener CRUSE	2
1.1 Geometrické usporiadanie	3
1.2 Špecifikácia skenera	4
1.3 Pracovný postup pri skenovaní objektov	6
2 Tiene	8
2.1 Geometrické vlastnosti tieňov	9
2.2 Bodové a plošné svetlo	9
2.3 Snímanie tieňov	10
2.4 3D model a terén	12
2.5 Rekonštrukcia objektu z tieňov	12
3 Pozorovania a merania	14
3.1 Priebeh skenovania	14
3.2 Umiestnenie zdroja svetla	14
3.3 Použité nástroje	16
3.4 Typy skenovaných objektov	17
3.5 Červený bod	17
3.6 Výstupy zo skenera	18
4 Implementácia	20
4.1 Implementácia algoritmu	20
4.2 Ďalšie vlastnosti kódu	24
5 Výsledky	26
5.1 Presnosť programu	26
5.2 Zložitosť programu	27
5.3 Netriviálne objekty	27
5.4 Súradnicové sústavy	28
Záver	29
Prílohy	31

Úvod

Počítačové videnie a počítačová grafika zaznamenali v posledných desaťročiach veľký pokrok v snahe správne analyzovať a zobrazovať reálny svet na počítačoch. Jedným z prúdov počítačového videnia je rekonštrukcia 3D objektov na základe rôznych typov dát o daných objektoch. Jednou z možností ako určiť 3D súradnice objektov je využiť ich tieň. Práve na tento spôsob rekonštrukcie objektov sme sa zamerali aj my a využili sme algoritmus popísaný v [4], aby sme vytvorili program vhodný na spracovanie obrázkov zo skenera CRUSE.

Napriek tomu, že v dnešnej informačnej spoločnosti existuje množstvo rôznych programov riešiacich mnoho typov problémov, je stále potrebné vyvíjať alebo aspoň upravovať hotové programy na konkrétne požiadavky zákazníkov a zariadení. V našom prípade sa požiadavky na algoritmus a implementáciu odvíjali od špecifikácie zariadenia CRUSE. Samotný skener je natoľko presný, že sa radí medzi špičku snímacích zariadení v dnešnej dobe. Tento veľký potenciál nesie so sebou aj isté odlišnosti od ostatných podobných zariadení, a preto pre získanie doplňujúcich dát o skenovaných objektoch je nutné adaptovať známe riešenia na parametre daného skenera. V našom prípade sú týmito doplňujúcimi dátami výškové súradnice jednotlivých pixlov. Program, ktorý popisujeme v tejto práci, je vhodný na získanie týchto dát a je prispôbený konkrétne na zariadenie CRUSE, čím umožňuje vo väčšej miere využiť potenciál tohoto skenera.

V prvých dvoch kapitolách sa venujeme základným pojmom a postupom z oblasti skenovania a z oblasti tieňov ako takých. Spomíname zloženie a prácu s vysokoformátovým skenerom CRUSE ako aj jeho výhody a špecifikáciu. Zároveň opíšeme vlastnosti a vznik tieňov, ich snímanie a možnosti rekonštrukcie objektov na základe ich tieňov. V tretej kapitole opíšeme pozorovania a merania, ktoré sme uskutočnili na skeneri CRUSE, opíšeme objekty, ktoré sme skenovali a dôvody, prečo sme si tieto objekty zvolili. Zhrnieme aké dáta a výstupy sme z jednotlivých skenovaní dostali a aké parametre sme si na nich všimli. V štvrtej kapitole opíšeme celú implementáciu nášho programu, popis a fungovanie jednotlivých tried. Vo výsledkoch porovnáme výstupy nášho programu s pôvodnými objektami. Na záver zhrnieme, čo sme urobili a navrhujeme ďalšie možné postupy a využitia implementácie.

Kapitola 1

Skener CRUSE

V tejto kapitole opíšeme skener CRUSE, jeho špecifikáciu a pracovný postup pre prácu s ním.

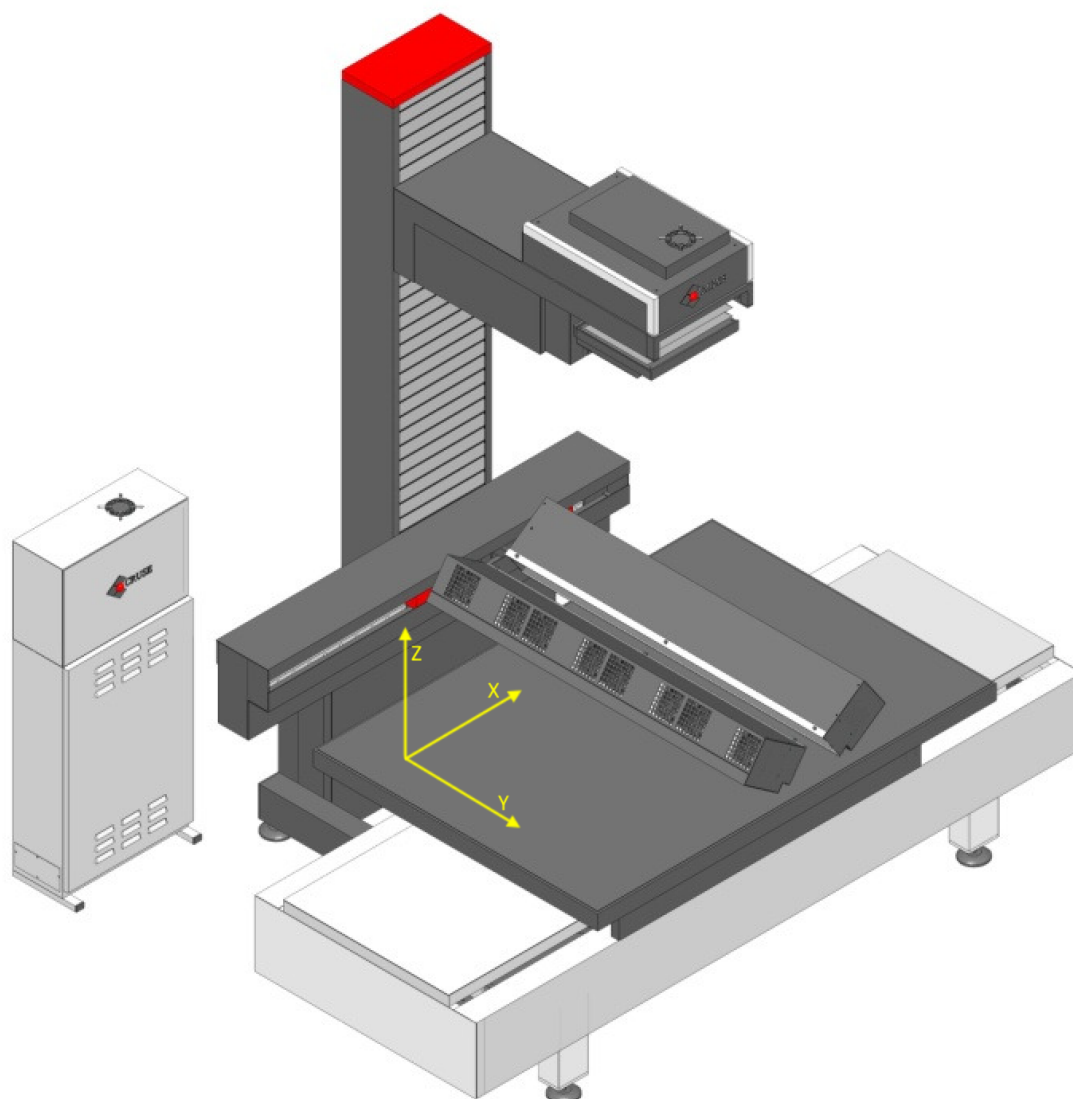
Základom digitálnej archivácie a reprodukcie je vysoko kvalitná digitalizácia. Digitalizácia obrazov, starých dokumentov, grafiky, fotografií a máp býva často veľmi náročná. Výhodou plošných skenerov je ich vysoké rozlíšenie a pracovná prístupnosť. Napríklad skenovanie jednotlivých stránok je možné iba na plošných skenoch, keďže ťahanie medzi valčekmi by mohlo viesť k uviaznutiu, pokrčeniu alebo sa cenné stránky môžu dokonca roztrnúť. Vďaka nastaviteľnej výške skenovacej hlavy a osvetlenia je možné skenovať aj priestorové objekty, ako sú zarámované obrazy, hrubé knihy, kamenné dosky a podobné.

Moderná replikácia fotografií vyžaduje špeciálne vybavené štúdio a profesionálny prístup. Homogénne osvetlenie, filtrovanie odtieňov olejových farieb a rozlišovanie farebných vrstiev často znamená neriešiteľnú úlohu pre fotografa alebo hľadanie kompromisov pre zákazníka. Mierka zväčšenia je často definovaná (a väčšinou obmedzená) rozlíšením digitálnych fotoaparátov a kamier.

Spomínaným riešením digitalizácie objektov je plošný skener. Zariadenie CRUSE je veľkoformátový typ skenovacieho zariadenia vhodný na digitalizáciu cenných artefaktov z múzejných alebo súkromných zbierok a archívov. Skenovacie zariadenie je určené na snímanie všetkých typov objektov s maximálnymi rozmermi 120x180cm. Avšak je treba si uvedomiť, že takýto typ skeneru je určený predovšetkým na profesionálne účely, a preto môže byť pre niektorých užívateľov nepraktický. Aby mohlo skenovanie prebehnúť na tak presnej a detailnej úrovni, ako to skener CRUSE ponúka, je treba akceptovať vedľajšie parametre skenovania. Medzi ne patrí najmä dlhý čas spúšťania a automatické kalibrovanie skeneru, dĺžka skenovacieho procesu, citlivé podmienky okolitého osvetlenia a veľkosť výstupných súborov. Je teda vidieť, že skener CRUSE nie je určený na rýchle a jednoduché používanie, ale jeho význam sa prejaví až pri detailnom reprodukování vzácnych, cenných či zaujímavých objektov.

1.1 Geometrické usporiadanie

Skenovacie zariadenie (obrázok 1.1) sa skladá zo 4 hlavných komponentov. Základným prvkom je skenovací stôl s posuvnou doskou vybavenou vákuovým prisávaním. Pohyb tejto dosky v horizontálnom smere určuje presnosť skenovania objektov. Doska má rozmery 180x120cm a hmotnosť skenovaných objektov môže dosahovať až 100kg. Vďaka vákuovému prisávaniu spodnej časti skenovaných objektov sa materiály ako papiera, látky či fólie prirodzeným spôsobom vyrovnajú a skener tak môže zachytiť objekty bez deformácií. Podkladová doska je tmavej farby, takže nedochádza k sekundárnemu osvetleniu objektov a vo výsledných obrázkoch je jednoduché oddeliť skenovaný objekt od podkladovej dosky. Na obrázku 1.1 sme pridali žltou farbou súradnicové osi, podľa ktorých budeme určovať polohu objektov na pracovnom stole. Počiatočný bod súradnicovej sústavy je bližšie opísaný v kapitole 3.



Obr. 1.1: Skenovacie zariadenie CRUSE [8]

Nad pohyblivým stolom je svetelný zdroj skladajúci sa zo 4 žiarivkových svetiel. Svetlá sú umiestnené vľavo (L), vpravo (R), vpredu (F) a vzadu (B). Skenovať dokumenty je možné pri akejkolvek kombinácií svetiel, ale najčastejšie sa preferuje kombinácia LRFB, LR, prípadne LFB. Nesymetrické osvetlenie ako LFB, prípadne LF, sa používa na zachytenie 3D štruktúry objektu. Vo všeobecnosti platí, že čím silnejšie osvetlenie skenovaného objektu použijeme, tým rýchlejšie prebehne skenovací proces. Preto sa vo väčšine prípadov, kde nepotrebujeme reprodukovat 3D štruktúru objektov, používa osvetlenie zo všetkých štyroch strán (LRFB).

Celú zostavu svetiel je možné nastaviť do potrebnej výšky. Najčastejšie sa osvetlenie umiestňuje čo najbližšie (najnižšie) k skenovaným objektom. Maximálna výška, do ktorej je možné osvetlenie zdvihnúť je 30cm, takže aj výška skenovaných objektov môže byť až do hodnoty 30cm.

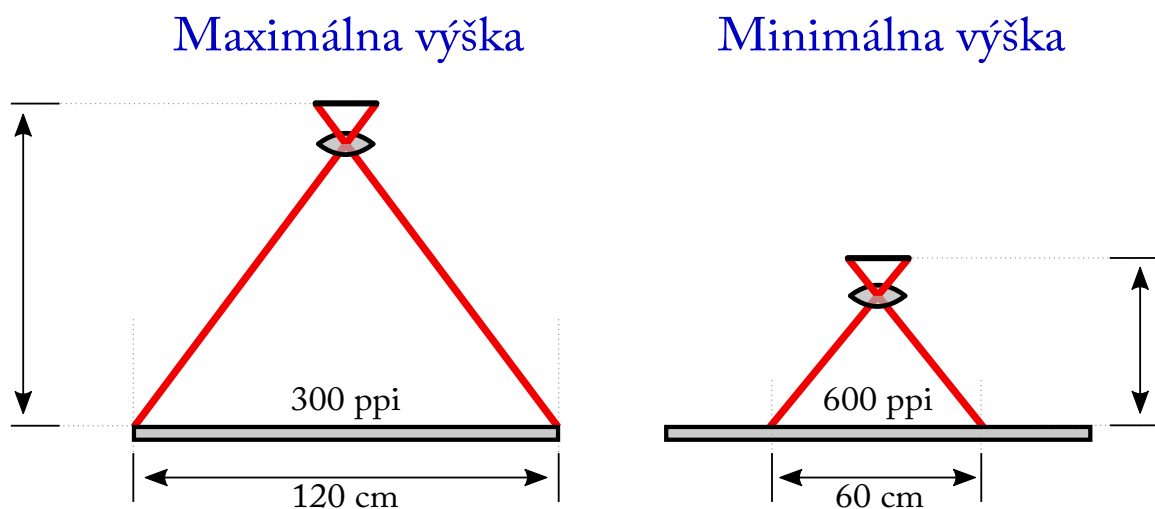
V horizontálnom smere je možné osvetlenie umiestniť na dve pozície. Prvá – základná – je umiestnenie priamo pod kamerou. Alternatívou je posunutie celej svetelnej zostavy do boku tak, aby skenovacia hlava mohla snímať objekty pod uhlom 15° .

Nad všetkým je umiestnená skenovacia hlava s kamerou. Správna kalibrácia a nastavenie objektívu je kľúčovým aspektom vysokého rozlíšenia výstupných dokumentov. Hlava sa nachádza na posuvnom ramene, ktoré je možné posunúť vo vertikálnom smere na zvolenú výšku.

Poslednou časťou skenera je elektrická skriňa umiestnená v blízkosti skenovacieho stola, v ktorej je zabudovaný riadiaci systém zariadenia. Skriňa obsahuje tlačidlo na zapnutie/vypnutie a tlačidlo na núdzové zastavenie skenera. S touto časťou sa vo väčšine prípadov nijak nenarába, keďže kompletne ovládanie skenera prebieha pomocou počítača.

1.2 Špecifikácia skenera

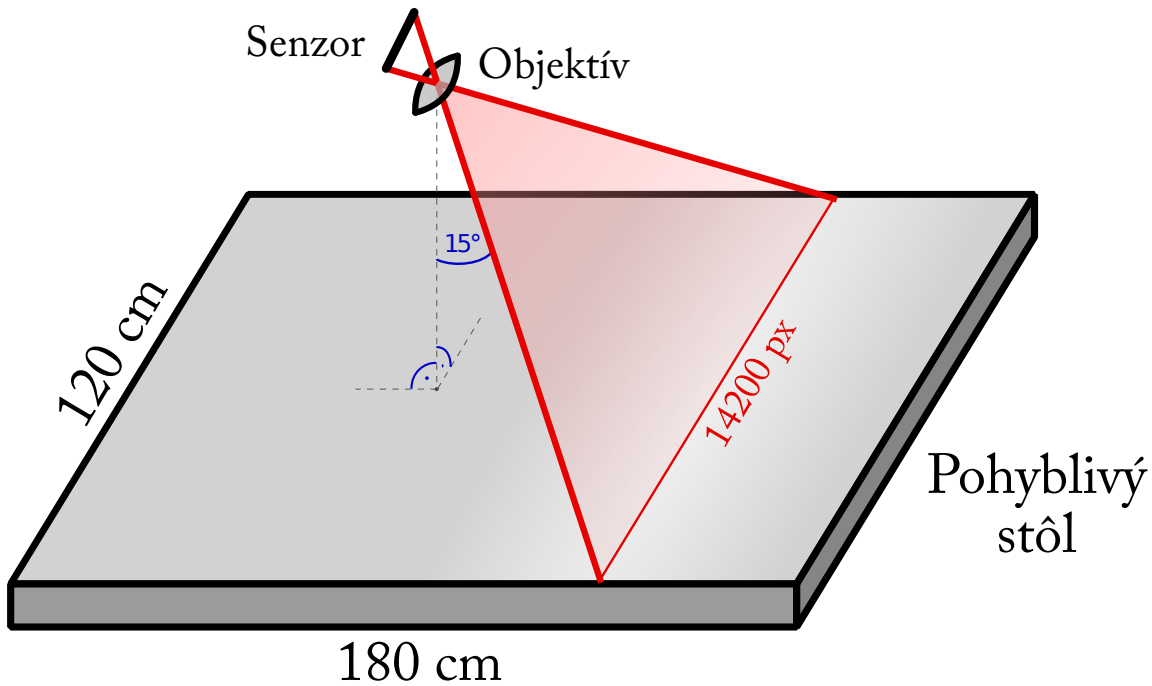
Kvalita skenovacieho zariadenia je charakterizovaná predovšetkým rozlíšením, farebnou hĺbkou a priestorovou deformáciou výstupných obrázkov. Skener CRUSE je z tohto hľadiska vybavený veľmi kvalitnými komponentmi. Rozlíšenie skenera je závislé od veľkosti skenovaného objektu. Na rozdiel od bežných fotoaparátov, ktoré majú senzor pozostávajúci z matice pixlov a v jednom momente vytvoria dvojrozmerný obrázok, skener CRUSE má iba jeden pás senzorov s 14200 pixlami [7]. Výsledné rozlíšenie teda závisí od vertikálneho umiestnenia skenovacej hlavy. Pri najmenej nastavenej výške je možné skenovať objekty so šírkou 60cm, a to až pri rozlíšení 600ppi. Toto nastavenie sa používa pre objekty menších rozmerov, ako napríklad papiere, knihy, menšie obrazy alebo mapy. Pri najvyššej výške je maximálna šírka objektov 120cm a rozlíšenie je 300ppi. Maximálna dĺžka skenovaných objektov je pre všetky rozlíšenia 180cm. Vďaka



Obr. 1.2: Parametre skenov pri maximálnej a minimálnej výške

nastaviteľnej výške skenovacej hlavy je teda možné využiť skener na skenovanie menších objektov pri vysokom rozlíšení, ale aj na skenovanie veľkorozmerných objektov pri nižšom rozlíšení. Na obrázku 1.2 je zobrazený fakt, že ak meníme vzdialenosť senzoru od stola, mení sa tým hustota snímaných bodov.

Okrem rozlíšenia je možné nastaviť aj už spomínané osvetlenie na jednu z kombinácií LRFB svetiel. Pre skenovanie s cieľom zachytiť výraznejší 3D efekt objektov je ideálne nastaviť posun skenovacej hlavy, ktorý umožňuje snímať pod 15° uhlom, keďže skenovacia hlava má dva módy snímania: uhol 0° a uhol 15° . Pri oboch módoch platia rovnaké možnosti nastavenia osvetlenia a rozlíšenia. Uhol 15° sa využíva na zachytenie veľmi jemných textúr, ktoré by inak neboli zosnímané. Alternatívnym využitím tohto módu je skenovanie priestorových objektov z bočných strán. Poslednou možnosťou modifikácie skenovania je pridanie 45° rozptylovača svetla. Tento rozptylovač sa využíva pre verné zachytenie odtieňov textúry povrchu. Vďaka usmerňovaniu svetelných lúčov do rovnobežného smeru sa tak do šošovky objektívu dostane odrazený lúč iba z jedného smeru. V kombinácii s nesymetrickým osvetlením a naklonením hlavy o 15° je tak výsledný efekt veľmi autentický a pôsobí priestorovým dojmom. Avšak pri skenovaní objektov niektorým z uvedených spôsobov vytvárajúcich trojrozmernú ilúziu vďaka tieňom je potrebné dávať pozor na smer, z ktorého svetlo prichádza. Ak je objekt skenovaný z „neprirodzeného“ smeru, tak výsledkom je obraz s negatívnym 3D efektom. To znamená, že výstupný obrázok síce tvorí dojem trojrozmerného objektu (vďaka tieňom, ktoré je na obrázku vidieť), ale keďže svetlo prichádzalo z inej strany ako je v bežnom svete zvykom, tak pre ľudského pozorovateľa sa zdá byť 3D efekt neprirodzený. Pre ľudí je prirodzené, že svetlo prichádza zhora (slnko, lampy) a teda objekty sú hore svetlejšie a dole tmavšie. To znamená, že objekty ako obrazy, knihy,



Obr. 1.3: Skenovanie pri naklonení hlavy o 15°

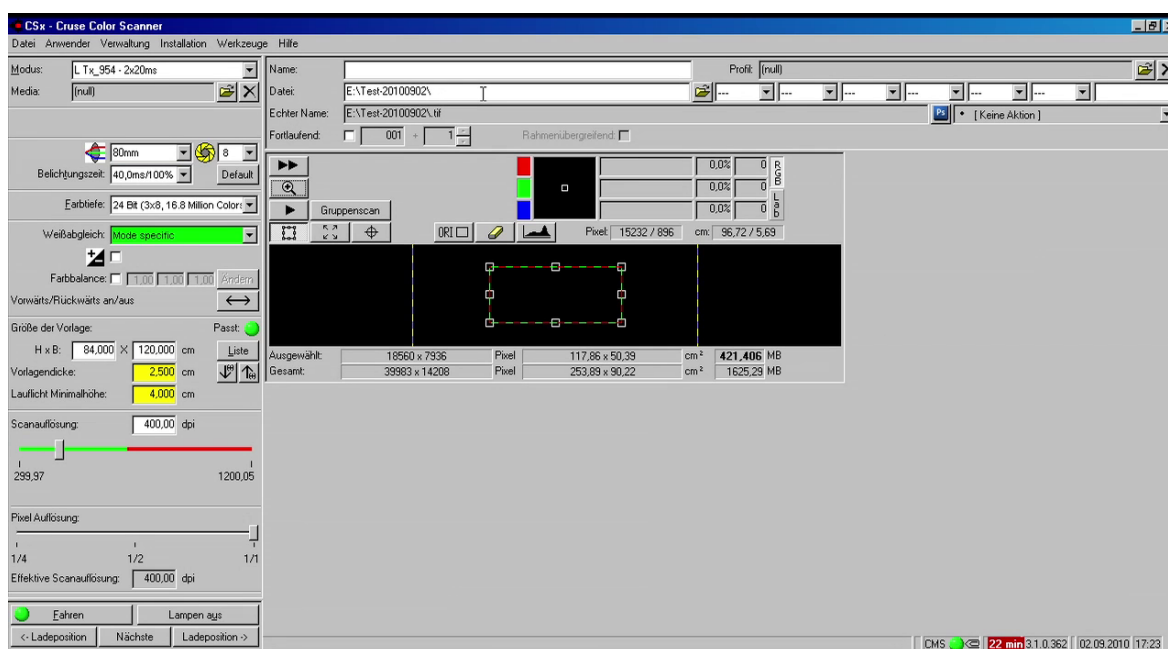
dosky alebo iné predmety, ktoré majú vo svete ustálené vertikálne otočenie, je vhodné skenovať pri osvetlení „zhora“.

Na obrázku 1.3 je načrtnuté skenovanie pri využití módu s naklonením hlavy o 15° , čo umožňuje zachytiť detailnejší 3D efekt.

1.3 Pracovný postup pri skenovaní objektov

Ovládanie skenera je možné iba cez softvér CSx, ktorý je nainštalovaný na pripojenom počítači. Po zapnutí zariadenia a spustení programu sa otvorí hlavné menu (obrázok 1.4). To umožňuje prístup ku všetkým nastaveniam, operáciám a riadeniu skenovacieho systému. Dialógové okno nás upozorní, že po spustení systému je potrebné skener resetovať. Po automatickom resetovaní je systém pripravený na skenovanie objektov.

Pred samotným skenovaním je potrebné nastaviť parametre skenera pre plánované snímanie objektu. V hlavnom menu sú všetky nastavenia umiestnené v ľavej časti obrazovky. Je tu možnosť nastavenia režimu osvetlenia, farebnej hĺbky, rozlíšenia, určenie výšky skenovacej hlavy a výšky skenovaného objektu. Pre každý skenovaný objekt je dôležité dôkladné nastavenie jednotlivých parametrov pre aktuálny zámer. V prípade, že nepotrebujeme, aby výstupný súbor mal maximálnu autenticitu skenovaného objektu, ktorú skener ponúka, je možné znížiť niektoré parametre, a tak napríklad urýchliť skenovací proces alebo zmenšiť veľkosť výstupného súboru. Napríklad pre osvetlenie platí,



Obr. 1.4: Uživatelské rozhranie programu CSx

že čím vyššiu intenzitu zvolíme, tým bude proces skenovania trvať kratšie. Taktiež platí, že nastavením nižšieho rozlíšenia, sa zmenší veľkosť výstupného súboru. Po zvolení všetkých položiek v ľavom menu spustíme samotné skenovanie. Následne prebehne skenovací proces a po dokončení je výsledný súbor uložený vo formáte .tiff do vopred vybraného priečinku. Pri ukladaní súborov je vhodné dbať na pomenovanie obrázkov podľa konvenčných pravidiel.

Pri voľbe objektov na skenovanie a pri ich umiestňovaní na pracovný stôl je dôležité riadiť sa určitými zásadami. Medzi základné pravidlá patrí umiestňovanie objektov na ich najväčšiu plochu, použitie kontrastnej podložky a dôkladné očistenie materiálov a lesklých plôch. Nie je vhodné skenovať vlhké alebo zašpinené objekty, keďže sa tak nebude môcť zachovať ich originálny vzhľad.

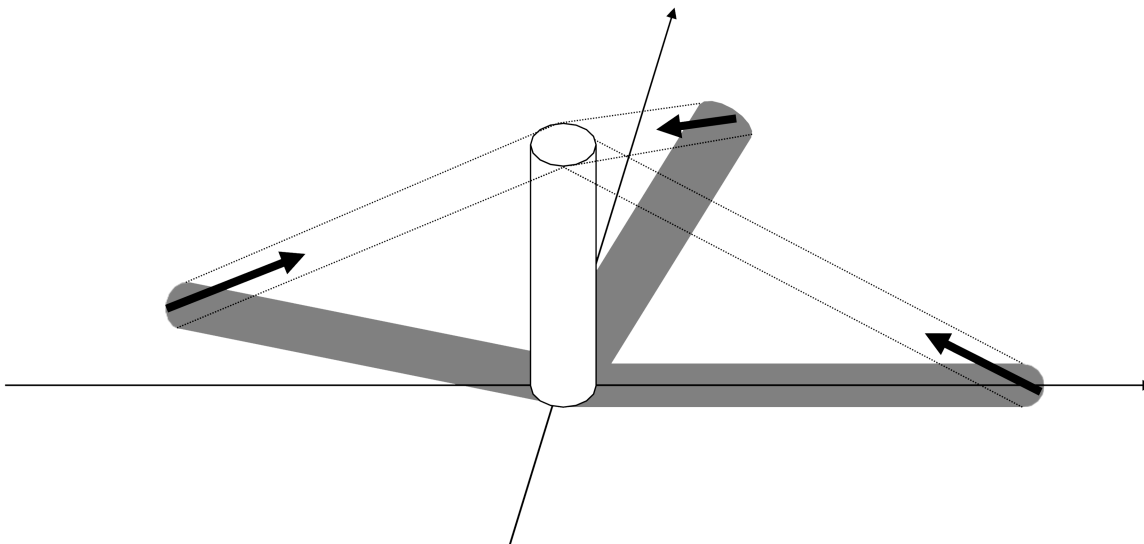
Pre presné a autentické vyhotovenie snímok je taktiež potrebné prispôbiť okolie skenovaciemu procesu. Základným predpokladom je eliminácia všetkých nežiadúcich svetelných zdrojov alebo odrazov v miestnosti, čo sa dosiahne úplným zatemnením okien a použitím tmavých materiálov v miestnosti. Keďže skenovanie prebieha na neobvykle detailnej úrovni, je dôležité taktiež redukovať prachu v miestnosti vysávaním, keďže prach by mohol mať negatívny vplyv na výsledné dokumenty.

Kapitola 2

Tiene

V tejto kapitole opíšeme problematiku tieňov z teoretického hľadiska.

Vznikom tieňov, ich orientáciou, rozostrením, veľkosťou a farbou sa zaoberalo množstvo výskumov. Ukázalo sa, že vďaka tieňom vieme z obrázkov a plôch získať množstvo cenných informácií o tvare a povrchu objektu. Vzniklo niekoľko štúdií zaoberajúcich sa 3D rekonštrukciou objektov a scén z ich tieňov. Medzi ne patria napríklad aj práce [5] a [2]. Postupy založené na skúmaní tieňov majú výhody hlavne v skúmaní jednotlivých objektov. Pri komplikovaných scénach obsahujúcich viacero objektov sa často vyskytujú príliš veľké chyby a odchýlky v meraniach na to, aby sa dali tieto výsledky považovať za presné. Avšak pri pozorovaní jednotlivých objektov bez rušivých elementov sa dá dosiahnuť vysoká presnosť výsledných dát.



Obr. 2.1: Rekonštrukcia povrchu valca na základe jeho tieňov [4]

Základnou metódou je skúmanie takých tieňov jednotlivých objektov, ktoré sa premietajú na rovinu kolmú na pohľad kamery. Inými slovami kamera sa pozerá „zhora“

a všetky tieňe sú vytvárané iba na podložke ležiacej v rovine. Pri pozorovaniach je objekt nasvietený zo známeho zdroja svetla a je snímaný zo známej polohy fotoaparátu. Pri opakovaní merania sa mení umiestnenie zdroja svetla, zatiaľ čo šošovka zostáva vzhľadom na skenovaný objekt na tom istom mieste. Tak sa vytvorí séria obrázkov, ktoré zobrazujú statický objekt, ale líšia sa vrhnutým tieňom. Keďže vieme pre každý obrázok povedať, kde sa nachádzal zdroj svetla, tak pri dostatočnom počte rôznych umiestnení zdroja svetla, je možné zrekonštruovať 3D reliéf objektu. Na obrázku 2.1 je znázornená rekonštrukcia povrchu valca na základe jeho tieňov.

2.1 Geometrické vlastnosti tieňov

Existuje viacero spôsobov, ako môžeme z matematického hľadiska vnímať tieň. Jeden zo spôsobov (podľa [2]) je nasledujúci:

Definícia 1. *Tieň je taká časť povrchu, v ktorej každý bod spojený so zdrojom svetla má prienik so samotným povrchom.*

Táto definícia je dostatočná v prípade, že osvetľujeme scénu iba jedným bodovým zdrojom svetla. Vtedy sa dá naozaj rozdeliť celý povrch na časti, kde sa tieň nachádza a časti, kde sa tieň nenachádza. Tento prístup budeme využívať aj my pri rozoznávaní tieňa na obrázkoch vytvoreného bodovým zdrojom svetla.

Problémy však nastanú, ak zdroj svetla nebude jeden alebo nebude bodový. V tom prípade totiž bude mať tieň rôzne úrovne a vznikne takzvaný polotieň. V tom prípade sa dá upraviť pôvodná definícia na:

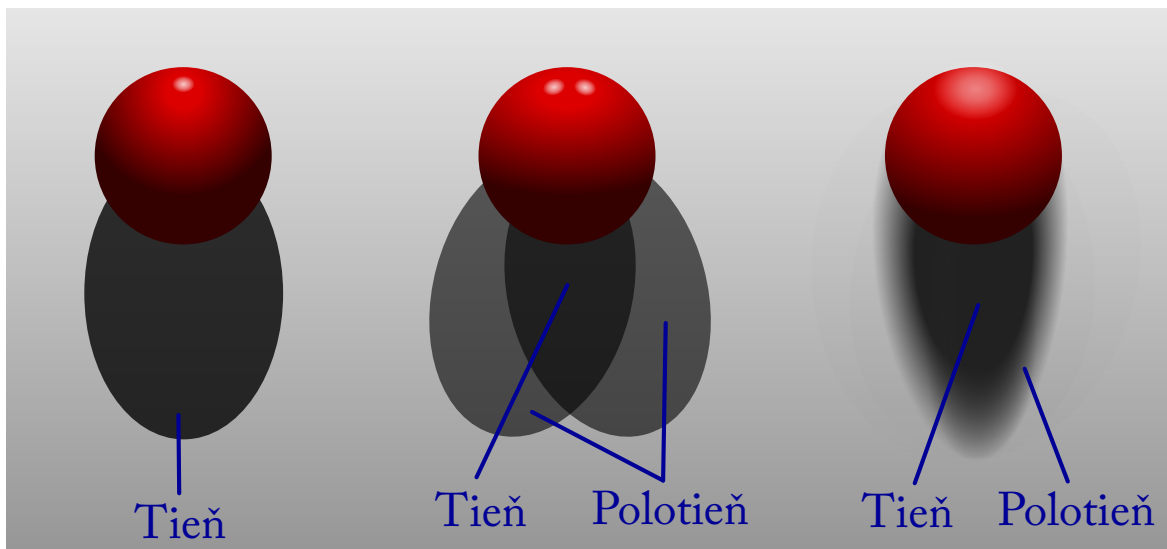
Definícia 2. *Polotieň je taká časť povrchu, v ktorej každý bod spojený s aspoň jedným zdrojom svetla má prienik so samotným povrchom.*

Napriek tomu, že polotiene vznikajú, je niekedy nutné rozdeliť povrch na oblasť s tieňom a bez tieňa. V tom prípade je stále možné použiť definíciu 1 a tak každý polotieň bude považovaný za oblasť bez tieňa.

2.2 Bodové a plošné svetlo

Medzi základné typy zdrojov svetla patrí bodový a plošný zdroj. Ako už samotný názov hovorí, bodový zdroj svetla tvorí jediný bod v priestore a vyžaruje svetelné lúče do všetkých smerov. Takto vzniknuté lúče sa šíria radiálne a všetky sa pretínajú iba v jednom bode – v bode odkiaľ sú vyžarované.

Plošný zdroj svetla je taký zdroj, ktorý na rozdiel od bodového zdroja má nenulové rozmery. Typickým príkladom plošného zdroja svetla je osvetlenie v kancelárii.



Obr. 2.2: Prítomnosť tieňov a polotieňov pri rôznom osvetlení

Svetlo prichádza z obdĺžnikovej plochy a smeruje dolu do miestnosti. Pri takomto type osvetlenia vznikajú okrem tieňov aj polotiene. Polotiene sa vyskytujú aj pri bodovom osvetlení ak je v scéne prítomných viac zdrojov svetla.

Na obrázku 2.2 je znázornená prítomnosť tieňov a polotieňov pri bodovom osvetlení, pri osvetlení s 2 bodovými zdrojmi svetla a pri osvetlení plošným zdrojom. V našich pozorovaniach sme využili bodové aj plošné osvetlenie, každé na iný účel.

2.3 Snímanie tieňov

Funkcia akejkoľvek kamery či fotoaparátu je premietnuť (zobraziť) 3D scénu na obraz v menšej dimenzii. Táto operácia využíva geometrickú projekciu.

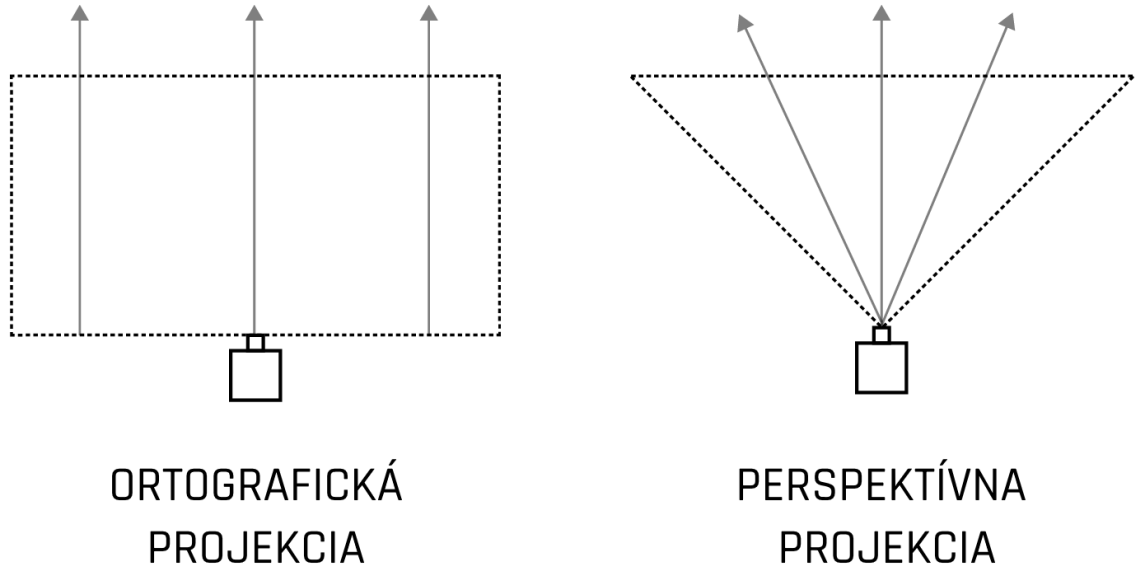
Projekcia je definovaná ako zobrazenie bodov z 3-rozmerného sveta do 2-rozmerného obrazu nasledovne:

$$I(i, j) = P(x, y, z)$$

Projekcia P zobrazuje trojrozmerný bod (x, y, z) na bod (i, j) v dvojrozmernom obraze I . Vidíme, že kvôli strate dimenzie, projekcia P nikdy nebude zobrazením jedna k jednej. Viacero bodov z trojrozmerného sveta sa totiž zobrazí na ten istý bod v dvojrozmernom obraze.

Smer lúčov z kamery do priemetne určuje typ projekcie. Dnešné snímacie zariadenia bežne používajú dva typy projekcií (obrázok 2.3):

- ortografickú projekciu - smer premietania je pre každý bod obrazu rovnobežný
- perspektívnu projekciu - smery premietania sa zbiehajú do jedného bodu



Obr. 2.3: Typy projekcií

Tieto projekcie môžeme presnejšie zadefinovať nasledovne:

Definícia 3. *Ortografická projekcia je zobrazenie $R^3 \rightarrow R^2$:*

$$(x, y, z) \rightarrow (x, y)$$

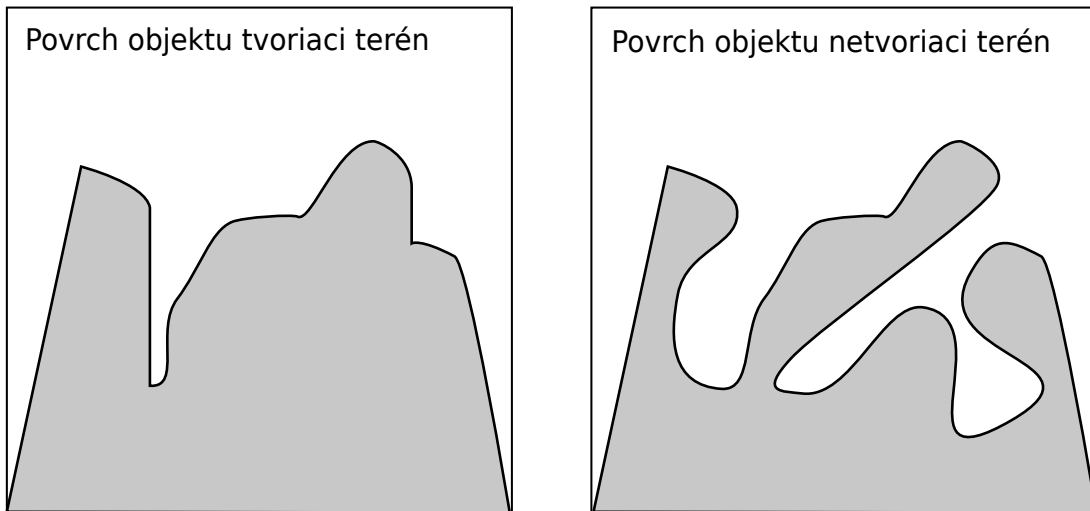
Definícia 4. *Perspektívna projekcia je zobrazenie $R^3 \rightarrow R^2$:*

$$(x, y, z) \rightarrow \left(\frac{f \cdot x}{z}, \frac{f \cdot y}{z} \right)$$

kde f je vzdialenosť priemetne od počiatku súradnicovej sústavy.

Azda najzásadnejším rozdielom medzi ortografickou a perspektívnou projekciou je vplyv na zobrazovaný objekt. V obraze vzniknutom perspektívnou projekciou sa objekt zdá byť menší, ak je ďalej od kamery. Tento efekt je ľuďom blízky, lebo aj oko zobrazuje svet pomocou perspektívnej projekcie. Perspektíva sa často využíva v umení, vo filmovom priemysle alebo v počítačovej grafike. Pri rovnobežnom premietaní sa objekt zobrazí bez skreslenia. Jeho veľkosť sa nemení, ak sa približuje alebo vzdaluje od kamery, navyše ortografická projekcia má tú vlastnosť, že zachováva rovnobežnosť. Teda ak boli dve línie pôvodne rovnobežné, tak vo vzniknutom obraze budú stále rovnobežné.

Príkladom perspektívnej projekcie je fotoaparát a kamera. Všetky snímajú realitu z jedného ohniskového bodu. Klasické stolové skenery a kopírky zobrazujú objekty pomocou ortografickej projekcie. V našej práci používame skener CRUSE, ktorý využíva kombináciu oboch typov projekcií. V osi y (kratšia strana stola) využíva perspektívnu projekciu a v osi x (dlhšia strana stola) využíva ortografickú projekciu. Je to zapríčinené tým, že v osi y sú všetky body snímané pásom pixlov, ktorý je v jednom momente zachytený objektívom, zatiaľ čo v osi x sú jednotlivé pixle zložené kvôli pohybu stola.



Obr. 2.4: Terén a oblasť netvoriaca terén

2.4 3D model a terén

Scéna v trojrozmernom modelovaní je definovaná [3] ako oblasť v priestore, v ktorej je časť vyplnená a zvyšok je prázdny. Oblasť, ktorá je vyplnená sa považuje za pevnú, a svetlo cez ňu nemôže prechádzať. Súvislá oblasť vyplneného priestoru sa nazýva objekt. Scéna sa nazýva terén, ak sa dá hrana objektu h zapísať ako funkcia v tvare:

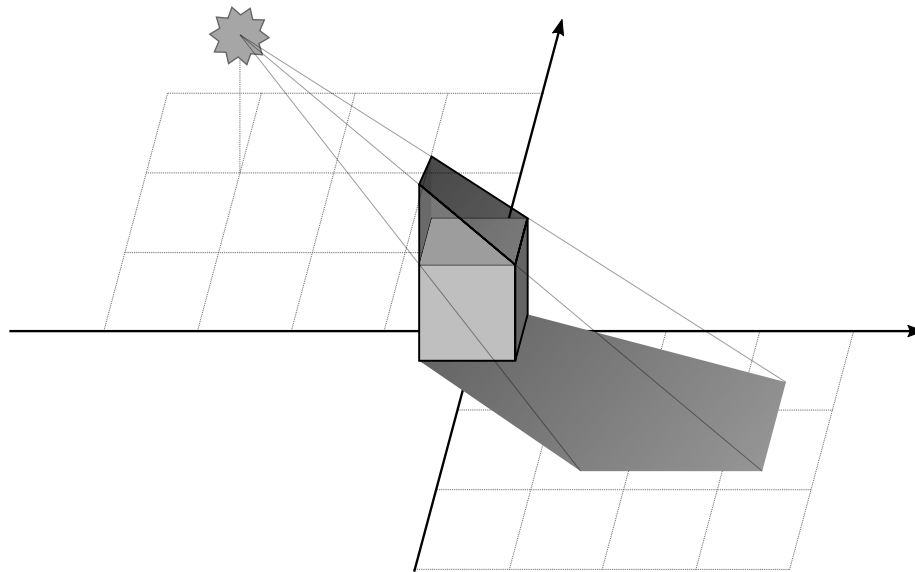
$$h : R^2 \rightarrow R = h(x, y)$$

Teda každý bod v rovine sa zobrazí na jednu výškovú hodnotu. Ako už samotné slovo hovorí, scény v tvare terénu sú podobné scénam na Zemi (ako napríklad pohoria). Na druhej strane scény, ktoré sa nedajú zapísať v tvare terénu sú pravé také, ktoré obsahujú viacero komplikovaných štruktúr, ako napríklad vznášajúce sa objekty alebo objekty s dierami. Oba typy scén sú znázornené na obrázku 2.4.

Existujú aj spôsoby ako analyzovať skrytý priestor pod terénom [9], avšak v našej práci je terén objektu dostatočným vyjadrením vlastností objektu. Tento prístup je odôvodnený hlavne tým, že samotný sken objektu je zobrazenie priestoru do roviny. Inými slovami, ak by sme sa snažili vytvoriť nielen terén, ale celý objekt, tak by to utrpelo na presnosti, lebo dodatočné dáta by boli získané z menej presných zdrojov ako je samotný skener CRUSE.

2.5 Rekonštrukcia objektu z tieňov

V tejto časti si opíšeme algoritmus pre rekonštrukciu objektov zo sekvencie snímok. V databáze snímok je vždy naskenovaný daný objekt, ktorý sa snažíme zrekonštruovať, spolu s tieňom, ktorý vrhá na podložku. Na každom snímku sa objekt nachádza na tom istom mieste, ale mení sa poloha tieňa. Alternatívou je určenie pevného bodu v priestore



Obr. 2.5: Konštrukcia objektu z jedného tieňa kocky

a na základe jeho umiestnenia v konkrétnom obrázku určiť posunutie objektu a tieňu oproti pôvodnému umiestneniu. Tento pevný bod sa zvyčajne realizuje nakreslením krížika na podložku vedľa objektu. My ale budeme pracovať s tým, že objekt v rámci viacerých skenov zostáva na tom istom mieste.

Algoritmus na rekonštrukciu objektov:

V nasledujúcom texte budeme predpokladať, že vzniknuté tieňe sa premietajú na rovinu. Musíme urobiť sériu snímok (skenov) pri rôznom umiestnení zdroja svetla a na základe toho, ako vzniknú tieňe, budeme schopní vytvoriť 3D reliéf modelu. Celý algoritmus možno rozdeliť na 3 kroky [4]:

Krok 1: Vyhladať tieňe v sekvencii snímok a určiť ich ohraničenie. Tento krok predstavuje samotné spracovanie obrázku. Vo výsledku bude každý obrázok rozdelený na tri časti a to skenovaný objekt, tieň a podložka bez tieňu. Pre ďalšiu postupnosť nám bude stačiť uchovanie jednej z týchto troch informácií pre každý pixel vo vhodnej dátovej štruktúre, inými slovami, nie je potrebné naďalej pracovať s obrázkom, ale postačujúce sú iba tieto tri hodnoty.

Krok 2: Pre každý obrázok vytvor objekt generovaný lúčmi, ktoré spájajú body tieňa so zdrojom svetla. Pri realizácii prirodzene stačí, aby bol objekt generovaný v miestach, kde sa objekt nachádza v zmysle osi x a y . Na obrázku 2.5 je znázornená konštrukcia objektu, ktorý vznikne z jedného tieňa kocky. Tento objekt je terén a na každej súradnici má výškovú hodnotu väčšiu alebo rovnú výške pôvodného skenovaného objektu na danej súradnici.

Krok 3: Vytvor prienik všetkých vzniknutých objektov, ktoré vznikli v kroku 2.

Výsledkom tohoto algoritmu je oblak bodov, určujúci povrch objektu. Inými slovami, každému pixelu na obrázku priradíme jeho výšku.

Kapitola 3

Pozorovania a merania

V tejto kapitole opíšeme, aké merania sme prostredníctvom skenera CRUSE uskutočnili, aké objekty sme skenovali a aké výstupy sme dostali.

3.1 Priebek skenovania

Samotné skenovanie objektu je možné pri rôznych nastaveniach skenera opísaných v kapitole 1.2. Skenovanie každého objektu sme rozdelili na dve časti:

Prvou je naskenovanie objektu pri plnom osvetlení s cieľom dosiahnuť obrázok objektu bez akýchkoľvek tieňov. Na túto časť sme využívali zabudované žiarivkové osvetlenie v samotnom skeneri pričom skenovanie prebiehalo v režime LRFB (všetky svetlá zapnuté). V tomto kroku sú svetlá umiestnené tak, aby sa čo najviac zabránilo vzniku tieňov. Získaný obrázok slúži na rozoznanie pixlov, ktoré patria objektu od ostatných pixlov. Zároveň slúži ako podklad pre export 3D modelu daného objektu.

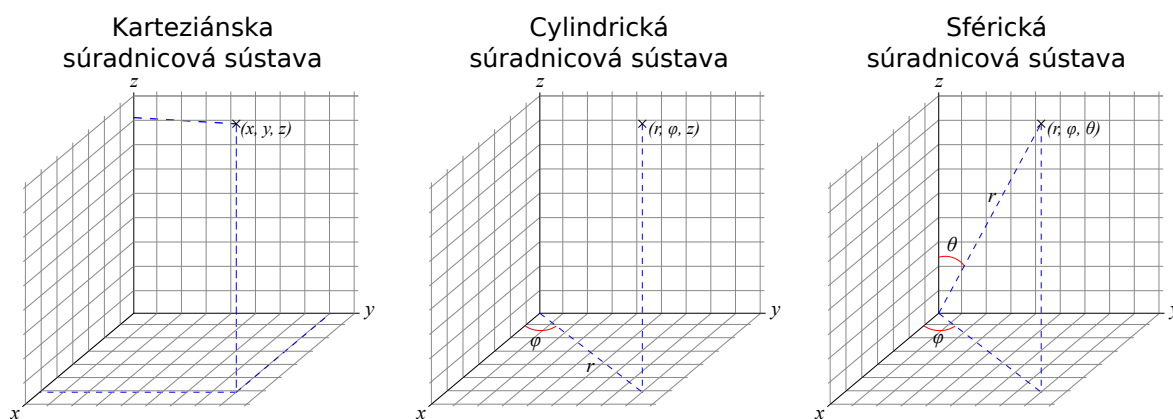
V druhom kroku skenujeme objekt viackrát pri osvetlení bodovým zdrojom svetla. V tomto kroku je zabudované osvetlenie vypnuté a jediným zdrojom svetla je LED dióda, ktorá je pri každom skenovaní umiestnená na inej pozícii.

Pri všetkých meraniach je dôležité, aby kontrast medzi objektom a podložkou bol čo najväčší. Toto môžeme dosiahnuť použitím bielej podložky a tmavého objektu. Nakoľko sa objekt vyhľadáva na obrázku s plným osvetlením (t.j. nie sú prítomné žiadne tieňe), môže byť farba objektu aj úplne čierna. Algoritmus totiž najprv nájde objekt pri plnom osvetlení a pri bodovom osvetlení hľadá iba tieňe.

3.2 Umiestnenie zdroja svetla

Umiestnenie zdroja svetla môžeme charakterizovať tromi premennými. Najpoužívanejšie spôsoby (pozri obrázok 3.1), ako určiť pozíciu bodu v priestore sú:

- Karteziánska súradnicová sústava – zdroj svetla bude mať súradnice x, y, z , pričom začiatok súradníc je v „červenom bode“ (viac o červenom bode je v kapitole 3.5) a jednotkou je 1mm.
- Cylindrická súradnicová sústava – zdroj svetla bude mať súradnice r, φ, z , kde r je vzdialenosť bodu od osi z ; φ je uhol odklonu sprievodcu bodu od osi x a z je vzdialenosť bodu od roviny xy , pričom začiatok súradníc je v „červenom bode“ a jednotkou je 1mm.
- Sférická súradnicová sústava – zdroj svetla bude mať súradnice r, φ, θ , kde r je vzdialenosť bodu od počiatku súradnicovej sústavy; φ je uhol odklonu sprievodcu bodu od osi x a θ je uhol medzi sprievodcom a osou z , pričom začiatok súradníc je v „červenom bode“ a jednotkou je 1mm.

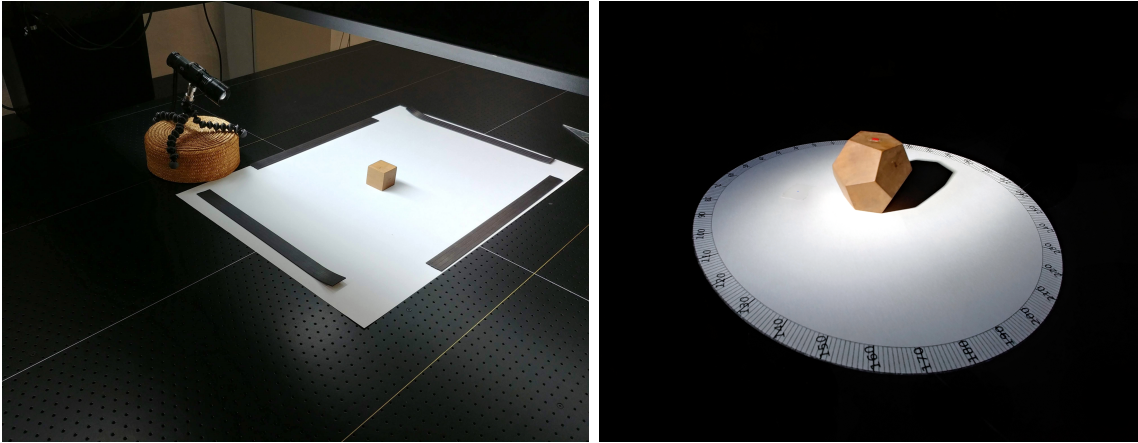


Obr. 3.1: Zobrazenie rôznych súradnicových osí [1]

Všetky tri súradnicové sústavy sú rovnocenné a existujú konverzie medzi každými dvomi, takže v algoritme môžeme použiť hociktorú z nich a dostaneme výsledky s rovnakou presnosťou. Avšak pri meraní hodnôt môžu nastať rozdiely v presnosti, keďže pri rôznych súradnicových sústavách sa použijú odlišné nástroje.

My sme použili karteziánsku pravouhlú súradnicovú sústavu a cylindrickú súradnicovú sústavu. V oboch prípadoch sme na meranie vzdialenosti použili pravítko. Na meranie uhla φ v cylindrickej súradnicovej sústave sme použili uhlomer.

Obe sústavy (karteziánska aj cylindrická) majú svoje výhody, aj nevýhody. Azda najzásadnejším faktorom, podľa ktorého je dobré zvoliť jednu alebo druhú sústavu, je presnosť meradiel. Ak vieme odmerať vzdialenosť s presnosťou jedného milimetra, ale uhol môže mať odchýlku až niekoľko stupňov, tak je vhodnejšie použiť karteziánsku sústavu a tým sa úplne vyhnúť meraniu uhlov. Ale ak vieme uhol merať naozaj s presnosťou jedného stupňa a plánujeme robiť veľa skenovaní, tak je lepšie uprednostniť



Obr. 3.2: Nastavenie scény počas skenovania

cylindrickú sústavu. Vtedy odmeriame na začiatku vzdialenosť zdroja svetla od červeného bodu a pred každým ďalším skenovaním už budeme merať iba uhol pootočenia.

V našich pozorovaniach mali merania vzdialeností aj uhlov približne rovnakú presnosť, a preto sme využili obe sústavy. Karteziánsku sme používali pri menšom počte skenovaní a cylindrickú v prípade, že sme objekt skenovali veľa krát. Dôvodom je snaha o zníženie celkového času skenovania a nastavovania scény na minimum.

Sférickú súradnicovú sústavu sme pri našich meraniach nevyužili, a to najmä z dôvodu komplikovaného merania uhla θ . Nastavenie scény a zdroja svetla počas našich pozorovaní je vidieť na obrázku 3.2.

3.3 Použité nástroje

Pri všetkých algoritmoch, ktorých cieľom je vytvorenie 3D modelu objektu z obrázku na základe tieňov, je kľúčová analýza obrazu a rozlíšenie pixlov obsahujúcich tieň od ostatných pixlov. Pre dosiahnutie čo najväčšej presnosti je pri tejto analýze potrebný vysoký kontrast osvetlených miest oproti zatieneným častiam. Inými slovami, našim cieľom je vytvorenie ostrých tieňov, aby bolo možné určiť ohraničenie zatienenej oblasti s vysokou presnosťou. Toto nám zabezpečí malý zdroj svetla s vysokou intenzitou. Ideálnym zdrojom svetla by bol nularozmerný bod v priestore s nekonečnou svetelnou intenzitou, avšak v reálnom svete je postačujúcou alternatívou aj silnejšia LED dióda. My sme pri všetkých skenovacích procesoch využívali LED diódu so svetelným tokom približne 200 lúmenov.

Okrem vhodného zdroja svetla je dôležitým aspektom presnosť pri meraní vzdialenosti zdroja svetla od objektu. Na tieto účely sme používali pravítko a všetky merania sme robili s presnosťou $\pm 0,5\text{mm}$.

3.4 Typy skenovaných objektov

Na testovanie funkčnosti programu sme využili objekt s podstavou 6-uholníka (pozri obrázok 3.5). Tento objekt je použitý ako ilustračný vo všetkých častiach tejto práce. Pre určenie presnosti algoritmu sme odmerali rozmery objektu posuvným pravítkom s presnosťou $\pm 0,05\text{mm}$.

Po otestovaní programu na spomínanom objekte sme urobili ešte niekoľko meraní. Naším cieľom bolo otestovať algoritmus a samotnú implementáciu na telesách s rôznymi tvarmi. Funkčnosť algoritmu sme sa rozhodli otestovať na telesách týchto tvarov:

- objekt s podstavou nekonvexného mnohouholníka
- ihlan
- rotačné teleso

Tieto objekty sme zvolili preto, aby sa otestovala schopnosť programu vypočítat povrch telies, ktoré sú v niektorom aspekte výnimočné. Za konkrétne telesá na skenovanie sme vybrali ozubené koleso, štvorsten a polguľu (pozri obrázok 3.3). Porovnanie skenovaných objektov s objektami skonštruovanými pomocou tieňov je bližšie popísané v kapitole 5.3.



Obr. 3.3: Telesá so špecifickými vlastnosťami

3.5 Červený bod

Pri zaznamenávaní polohy zdroja svetla vzhľadom na objekt je nutné vytvorenie nejakej súradnicovej sústavy a určenie počiatku danej sústavy. Následne odmeriame súradnice zdroja svetla a súradnice objektu. Potom môžeme narábať s danými hodnotami, ako so vstupnými údajmi pre trojrozmerný priestor, a tak skonštruovať v programe scénu obsahujúcu skenovaný objekt a zdroj svetla.

Pri bližšom preskúmaní vzťahov medzi bodmi v ktorejkoľvek súradnicovej sústave zistíme, že počiatok sústavy sa môže nachádzať kdekoľvek, bez ujmy na presnosti. Podstatné je iba to, aby sa pri každom skenovaní objektu nachádzal presne na tom istom mieste.

Našou prvou ideou bolo umiestniť počiatok sústavy do jedného z rohov v zoskenovanom obrázku. Ak by pri každom skenovaní zostala oblasť skenovacej plochy rovnaká, tak by aj počiatok sústavy bol vždy na tom istom mieste. Takto by bol na každom obrázku objekt umiestnený presne na tých istých pixloch a menila by sa iba pozícia zdroja svetla. Avšak stretli sme sa s chybou softvéru skenera CRUSE, ktorý nezaručoval presnosť vybratej oblasti. Inými slovami, ak sa v rozhraní softvéru CSx nastavila rovnaká oblasť skenovania, tak po skenovaní bola oblasť niekedy posunutá. Odchýlka bola až 60 pixlov, čo je príliš veľa vzhľadom na očakávanú presnosť algoritmu.

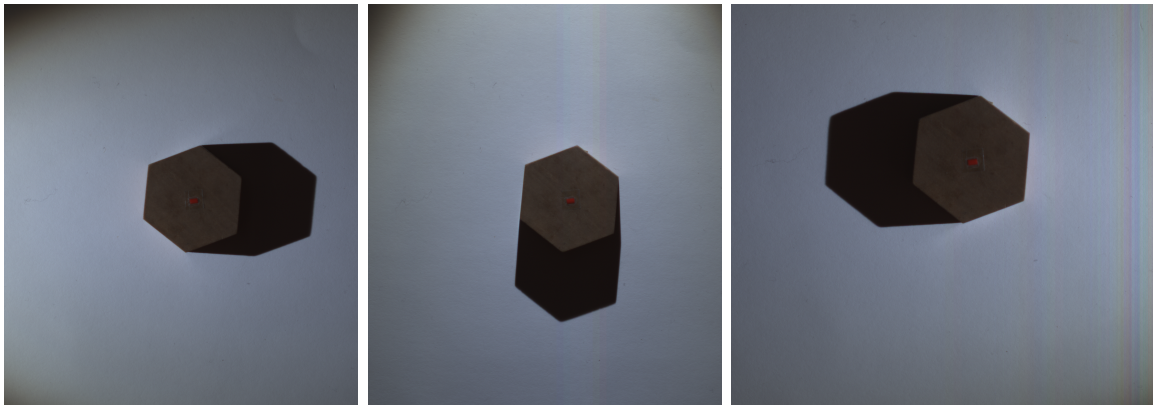
Druhou možnosťou je označiť počiatočný bod sústavy v samotnom skenovanom obrázku. Algoritmus by tento označený bod našiel a mohol by ho pri výpočte použiť. Tento spôsob sme aj otestovali a ukázal sa ako vhodný. Červený bod sme vytvorili pomocou bodky na podložke s výraznou farbou, ktorá sa na skenovanom objekte nevyskytuje. V programe následne nájdeme bod s najväčšou intenzitou danej farby a určíme ho za počiatok súradníc. Od tohoto bodu je potom aj merané umiestnenie zdroja svetla. Odchýlka v meraní je v tomto prípade priemer nakreslenej bodky, čo predstavuje hodnotu menšiu ako 20 pixlov.

Do samotnej implementácie sme zakomponovali výber farby červeného bodu. Používateľ tak do súboru so vstupnými údajmi pridá aj informáciu o tom, aká farba bola použitá na nakreslenie červeného bodu. Na výber sú možnosti červená, zelená a modrá farba. Napriek tomu, že tento orientačný bod nemusí byť červenej farby, budeme kvôli jednoduchosti aj naďalej používať označenie tohoto bodu ako „červený bod“. Využitie dvoch červených a jedného modrého bodu je vidieť na obrázku 3.3.

Pri tomto spôsobe je ešte veľmi dôležité, aby pozícia objektu vzhľadom na červený bod zostala pri všetkých skenoch rovnaká, čo znamená, že je treba zabezpečiť, aby sa nejakým nedopatrením objekt neposunul, lebo to by viedlo k signifikantým nepresnostiam. Pomôcť môže napríklad upevnenie podložky ku stolu magnetickými páskami alebo menším závažím.

3.6 Výstupy zo skenera

Po dokončení skenovacieho procesu je do počítača uložený obrázok vo formáte TIF. Pre ďalšiu prácu je vhodné konvertovať tento súbor do niektorého z formátov s bezstratovou kompresiou, ako sú *png* a *bmp*. Tieto formáty sú podporované širšou skupinou programov a sú užívateľsky častejšie používané ako formát TIF. Tým sa ušetrí miesto



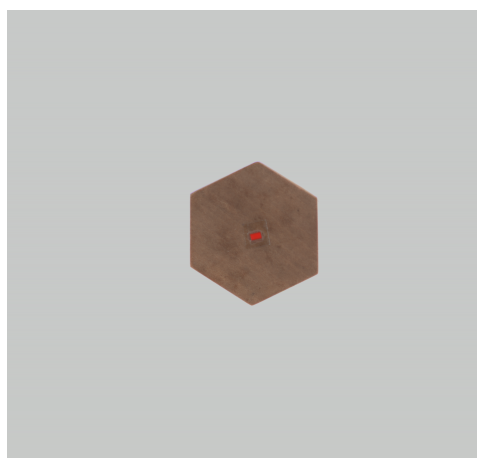
Obr. 3.4: Výstupné súbory zo skenovania pri bodovom osvetlení

a zrýchli čas samotného výpočtu programu bez straty kvality.

Niekoľko výstupných súborov môžeme vidieť na obrázku 3.4. Na všetkých obrázkoch je jasne rozoznateľný tieň, objekt a pozadie (podložka). Azda jedinou mylnou informáciou je tieň vzniknutý v rohoch obrázkov. Tento tieň je zapríčinený nedostatočným rozptylom svetla z bodového zdroja (LED diódy). Avšak napriek tomu, že tento jav nastáva, nijak to neovplyvní náš algoritmus, pretože tento tieň sa v blízkosti objektu nenachádza.

Jeden zo súborov musí byť nasnímaný pri plnom osvetlení. Tu treba dať pozor a nastaviť gain skenera na nižšiu hodnotu ako pri predošlých snímaníach tak, aby bol na obrázku jasne rozoznateľný skenovaný objekt. Príklad takého výstupu môžeme vidieť na obrázku 3.5. Na tomto obrázku je možné jasne rozoznať, ktoré pixle zobrazujú objekt a na ktorých pixloch je iba biela podložka.

Na obrázkoch 3.4 a 3.5 je slabo vidieť aj červenú bodku, ktorá označuje náš vyššie spomínaný červený bod. Táto bodka môže byť umiestnená na objekte alebo mimo neho. V tomto prípade je umiestnená na objekte.



Obr. 3.5: Výstupný súbor zo skenovania pri plošnom osvetlení

Kapitola 4

Implementácia

V tejto kapitole opíšeme ako sme implementovali algoritmus na analýzu tieňov. Aplikáciu sme písali v jazyku C#. Celá aplikácia je rozdelená do nasledujúcich súborov:

- MainClass.cs
- Nacitaj.cs
- Krok1.cs
- Krok2.cs
- Krok3.cs
- Vykresli.cs

Názvy súborov zodpovedajú názvom tried, ktoré sú v nich uložené. Okrem týchto súborov je pre spustenie programu potrebný aj:

- priečinok s naskenovanými súbormi
- priečinok so súbormi typu .txt, v ktorých sú popísané parametre jednotlivých meraní

4.1 Implementácia algoritmu

Celý algoritmus popísaný v kapitole 2.5 sme rozdelili do troch tried. Každý krok algoritmu predstavuje jednu triedu, t.j. jeden súbor. Okrem týchto troch tried sme vytvorili ešte dve triedy - *Nacitaj* a *Vykresli*.

Nacitaj

Na základe vstupných údajov prečítaných z textového súboru, načítame jednotlivé obrázky a aktualizujeme hodnoty skenera, ktoré boli použité. Okrem toho medzi

vstupné informácie ešte patrí farba červeného bodu, ktorý sa používa pri rozpoznávaní telesa a umiestnenie zdroja svetla vzhľadom na červený bod pre každý obrázok. Zároveň skonvertujeme obrázky z formátu bitmapy na pole celých čísel, ktoré určujú postupne červenú, zelenú a modrú zložku každého pixlu. Vďaka tomu je následne čítanie a zapisovanie do obrázkov omnoho rýchlejšie, nakoľko predvolené metódy `SetPixel` a `GetPixel` sú veľmi pomalé.

Krok 1

V tomto kroku postupne vykonáme tri metódy, ktorých cieľom je analyzovať jednotlivé obrázky a každý z nich rozdeliť na objekt, tieň objektu a prázdne pixle. Obrázky sme dostali z predošlej triedy (*Nacítaj*) vo formáte poľa celých čísel.

1. NajdiCervenyBod()

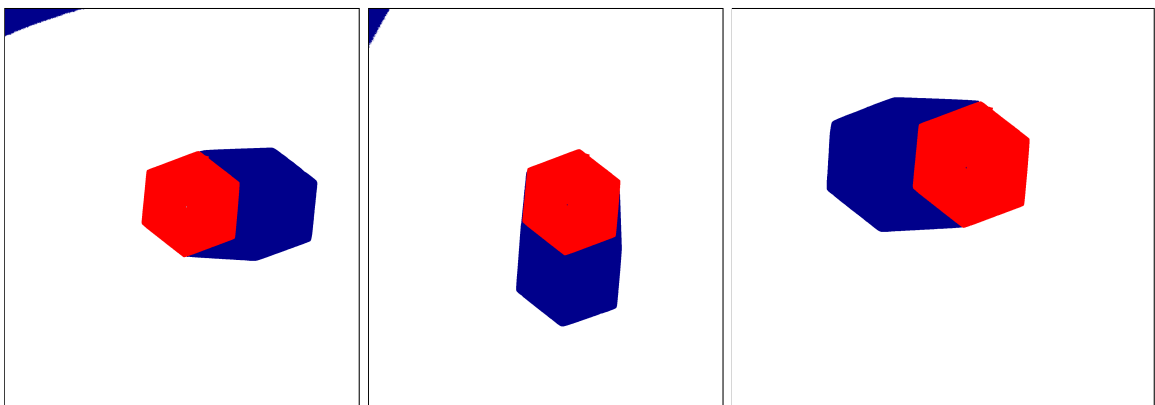
Postupne v každom obrázku nájdeme body červenej farby a priemer ich súradníc určíme za červený bod. Bude to začiatok karteziánskych súradníc na danom obrázku.

2. NajdiObjekty()

V obrázku s plným osvetlením priradíme každému pixlu informáciu či je to objekt alebo nie. Ako porovnávací argument používame jas pixlu. Následne každému inému obrázku priradíme to isté ohodnotenie posunuté o rozdiel červených bodov medzi obrázkom s plným osvetlením a aktuálnym obrázkom.

3. NajdiTiene()

Vo všetkých obrázkoch (okrem obrázku s plným osvetlením) určíme každému pixlu, ktorý nie je objekt či je to tieň alebo nie. Ako porovnávací argument používame jas pixlu.



Obr. 4.1: Rozdelenie na objekt, tieň a pozadie (podložku)

Po vykonaní týchto troch metód máme pre každý obrázok pole s informáciou či je daný pixel objekt alebo tieň, alebo ani jedno. Analýzu obrazu a následne rozoznanie objektu a tieňa môžeme vidieť na obrázku 4.1. Červenou farbou je znázorňený objekt, modrou tieň a bielou podložka (pozadie).

Krok 2

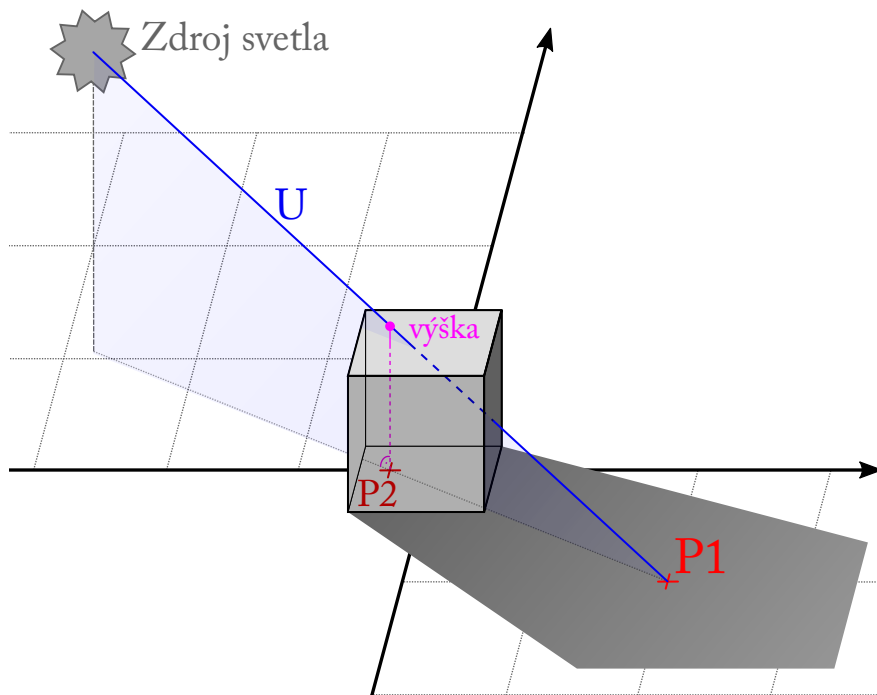
V kroku 2 je naším cieľom určiť pre každý obrázok maximálne výškové hodnoty jednotlivých pixlov. Presnejšie povedané, pre každý obrázok (okrem obrázku s plným osvetlením) vykonávame:

```

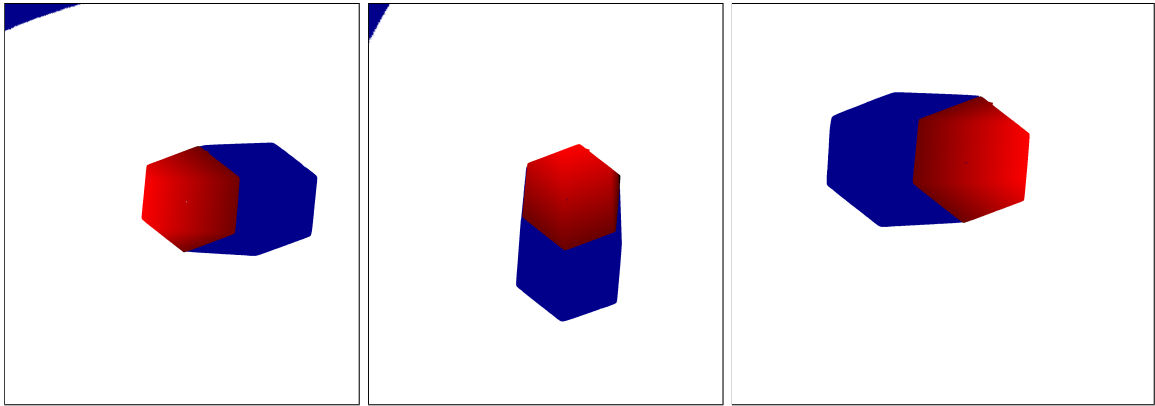
for (pixel  $P_1$ ;  $P_1$  je tieň) do
  úsečka  $U \leftarrow (\text{ZdrojSvetla} - P_1)$ 
  for (pixel  $P_2$ ;  $P_2$  leží pod  $U$ ) do
     $P_2.\text{vyska} = \max(P_2.\text{vyska}, \text{výška } U \text{ v bode } P_2)$ 
  end for
end for

```

Na obrázku 4.2 je znázornený výpočet algoritmu. V každom obrázku tak dostaneme výšku každého pixla objektu a z fyzikálnych vlastností svetla a tvorby tieňov vieme, že reálna výška konkrétneho pixlu určite nie je väčšia ako výška, ktorá nám v tomto kroku vyšla. Na obrázku 4.3 môžeme vidieť zobrazenie výškového poľa jednotlivých obrázkov (čím je červená farba sýtejšia, tým je výška konkrétneho pixlu väčšia).



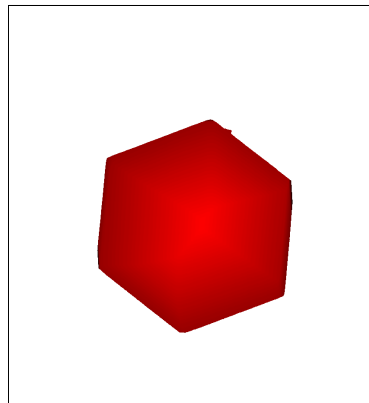
Obr. 4.2: Náčrt algoritmu z Kroku 2



Obr. 4.3: Výškové pole jednotlivých obrázkov

Krok 3

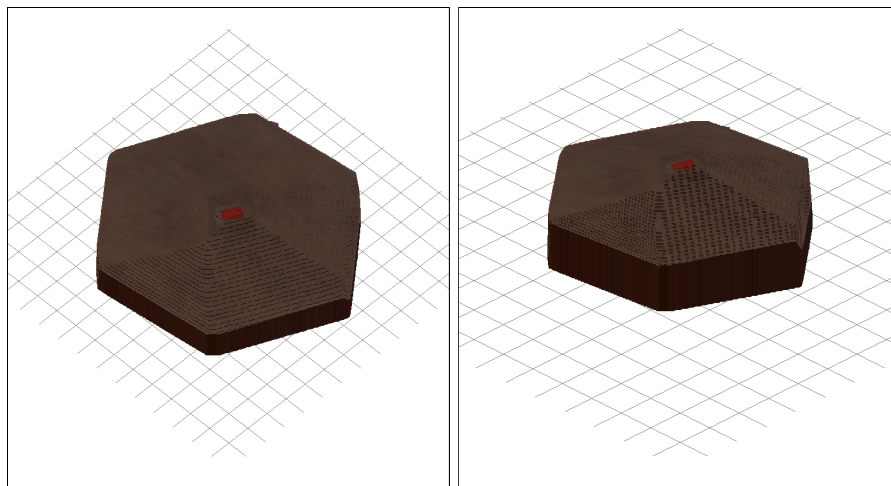
V poslednom kroku algoritmu vytvoríme prienik všetkých objektov, ktoré nám vznikli v *Kroku 2*. Výsledný objekt bude mať rozmery zhodné s rozmermi objektu na obrázku s plným osvetlením. Na obrázku 4.4 môžeme vidieť zobrazenie výškového poľa vzniknutého objektu (čím je červená farba sýtejšia, tým je výška konkrétneho pixlu väčšia).



Obr. 4.4: Prienik výšok

Vykresli

V triede určenej na export vypočítaných dát ukladáme nami získané hodnoty do súborov v počítači. Táto trieda vytvorí textový súbor obsahujúci kľúčové údaje a tvar vymodelovaného objektu. Medzi ne patrí napríklad aj maximálna, minimálna a priemerná výška vytvoreného objektu. Textový súbor sa uloží do rovnakého adresára, ako sa nachádza vstupný textový súbor. Trieda vytvorí súbor typu *.xyz*, v ktorom je uložený oblak bodov objektu a ktorý sa dá následne zobrazovať a upravovať v bežných modelovacích programoch. Okrem toho vytvorí obrázky typu *png*, v ktorých zobrazuje vytvorený 3D model vždy s iným sklonom pohľadu kamery. Na obrázku 4.5 môžeme vidieť vzniknutý 3D model naskenovaného objektu.



Obr. 4.5: 3D model

MainClass

V samotnej spustiteľnej aplikácii sú postupne volané vyššie spomínané triedy a do konzoly je priebežne vypisovaný stav algoritmu a uplynutý čas.

4.2 Ďalšie vlastnosti kódu

Paralelizácia

Celý program je ľahko paralelizovateľný, a tak je možné využiť potenciál dnešných počítačov. Tým, že v triedach *Nacitaj*, *Krok1* a *Krok2* sa pracuje s každým z naskenovaných obrázkov nezávisle, je možné tieto spracovania obrázkov robiť na rôznych jadrách procesoru. V triede *Vykresli* je naším cieľom exportovať rôzne obrázky do súborov, takže opäť môžeme využiť paralelizáciu a každý obrázok môže byť renderovaný iným jadrom procesoru. Síce viaceré procesy pristupujú k tej istej dátovej štruktúre súčasne, ale vďaka tomu, že všetky z nej iba čítajú, je táto paralelizácia korektná.

Modularita kódu

Vďaka rozdeleniu jednotlivých krokov algoritmu do rôznych tried má náš kód vysokú modularitu. Každá trieda (a aj každá metóda v nej) má jeden špecifický význam. Triedy medzi sebou navzájom nekomunikujú, ale všetky sú volané iba z hlavnej triedy (*MainClass*). Rovnako žiadna trieda nepristupuje k premenným z iných tried, ale z hlavnej triedy dostane do konštruktora potrebné dáta a metódami *Get()* vráti novovytvorené dátové štruktúry. Je teda jednoduché zmeniť implementáciu niektorého konkrétneho kroku, zrýchliť alebo zefektívniť načítavanie a vypisovanie, alebo pracovať s priebežne vypočítanými hodnotami v jednotlivých krokoch algoritmu.

Vstupné údaje

Vstupné údaje sa delia na dve časti: textový súbor, v ktorom sú popísané parametre jednotlivých meraní a naskenované obrázky formátu *png*.

Textový súbor sa musí nachádzať v rovnakom priečinku ako spustiteľná aplikácia (*exe*) a jeho názov musí byť *vstup.txt*. Štruktúra súboru vyzerá nasledovne:

1. riadok = hodnota *dpi*, pri ktorej boli skenované všetky obrázky.

2. riadok = písmeno *r*, *g* alebo *b*, určujúce farbu červeného bodu.

3. riadok = *cesta* k priečinku, v ktorom sú uložené naskenované obrázky.

4. riadok = číslo *k*, určujúce počet naskenovaných obrázkov s tieňmi.

nasleduje *k* riadkov = na každom riadku sú tri čísla - súradnice *x*, *y*, *z* zdroja svetla s počiatkom v červenom bode pre konkrétny naskenovaný obrázok.

V priečinku s naskenovanými súbormi sa musí nachádzať obrázok s názvom *fl.png*. Je to obrázok, na ktorom je objekt zoskenovaný pri plnom osvetlení (t.j. zo všetkých strán) a na ktorom nie sú žiadne tieň. Okrem neho sú tu uložené aj obrázky s osvetlením vždy z niektorej strany. Názvy týchto obrázkov sú prirodzené čísla od jednotky (t.j. *1.png*, *2.png*, *3.png*...).

Kapitola 5

Výsledky

5.1 Presnosť programu

Presnosť programu na jednoduchých objektoch (kocka, kváder) je veľmi vysoká už pri nízkom počte skenovaní. Kváder s podstavou šesťuholníka mal výšku 1,45 cm a náš program vypočítal priemernú výšku 1,6cm. To je síce odchýlka 1,5mm, ale dôvodom tejto odchýlky nie je chyba v programe. Pri všetkých objektoch platí, že ak teleso obsahuje plochy rovnobežné s podložkou, tak tieto plochy nie je možné detegovať. Tento jav nie je zapríčinený zlým algoritmom alebo implementáciou, ale vyplýva čisto z geometrických vlastností svetelných lúčov. Preto sa na vrchných rovných plochách vytvára vizuálny oblúčik alebo strieška. Napriek tomu, že neexistuje algoritmus, ktorý by sa dokázal s týmto problémom vysporiadať, je možnosť vyrovnáť tieto plochy v sekundárnom spracovaní objektu. Ak by sme napríklad vedeli o skenovanom objekte, že je to rovnobežnosť, tak je možné upraviť tento model do požadovaného tvaru. Avšak toto sekundárne spracovanie sme zatiaľ neimplementovali, a tak to zostáva ako jedna z možností budúceho vylepšenia programu.

Najväčšia odchýlka v meraniach bola spôsobená najmä nepresnosťami v meraní fyzikálnych veličín. Merali sme vzdialenosť zdroja svetla, pričom odchýlka v meraniach bola niekedy až 1 centimeter. Táto veličina by sa dala zmerať presnejšie použitím lepších pracovných pomôcok a nástrojov. Nepresnosti niekedy vznikali aj kvôli nedostatočnému výkonu svetelného zdroja. Ten by sa dal nahradiť silnejším zdrojom s väčším rozptylom svetla a tým by sa zamädzilo vzniku neželaných tieňov v rohoch obrázka.

Samotná aplikácia nemala odchýlku v tvorbe reliéfu viac ako 30 pixlov, čo je menej ako 3mm. Táto nepresnosť je spôsobená nesprávnym rozlíšením tieňa od podložky, prípadne nepresným určením červeného bodu. Na okraji tieňa vzniká polotieň, a preto sa musí veľmi presne určiť hranica jasů, od ktorej sa považuje konkrétny pixel za tieň. My sme túto hranicu získali ako polovicu priemerného jasů na obrázku. Táto metóda síce funguje, ale pre vyššiu presnosť by sa dala táto hranica určiť z histogramu obrázka.

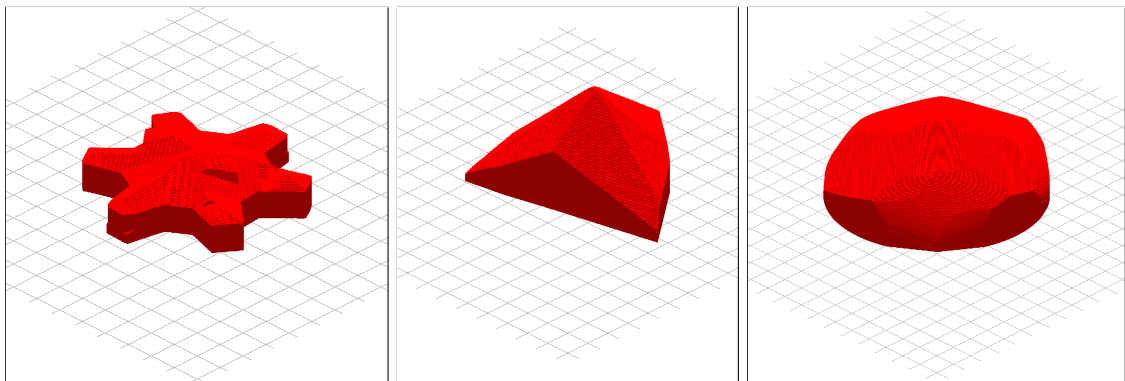
5.2 Zložitosť programu

Časová zložitosť samotného programu je najvyššia v kroku 2. Tu je pre každý obrázok časová zložitosť až $O(n \cdot m \cdot (n + m))$ kde n a m sú dĺžky strán obrázka v pixloch. Pamäťová zložitosť programu je rovná vstupu, teda súčtu veľkostí všetkých obrázkov, ktoré sa načítavajú.

Na počítanie a vykresľovanie sme používali zostavu s Windows 10 Home Edition, procesorom Intel Core i7-7700HQ 2.4GHz, 8GB pamäťou DDR4. Spracovávali sme obrázky rôznych rozmerov až do veľkosti 4000x4000px. Dĺžka výpočtu závisela od počtu naskenovaných obrázkov a od veľkosti týchto obrázkov. Pri 5 obrázkoch bolo trvanie celého výpočtu cca 1 minúta. Pri počte 26 naskenovaných obrázkov trval výpočet až 5 minút. Tento čas síce vyzerá byť dlhý, ale vzhľadom na čas strávený pri skenovaní je to pre používateľa akceptovateľná dĺžka.

5.3 Netriviálne objekty

V kapitole 3.4 sme opísali, aké typy objektov sme skenovali a na ktorých sme skúšali funkčnosť algoritmu. Všetky 3 objekty (objekt s podstavou nekonvexného mnohoúhelníka, ihlan a rotačné teleso) sme skenovali pri 4 rôznych pozíciách bodového svetla. Skonstruované 3D modely sú zobrazené na obrázku 5.1.



Obr. 5.1: 3D modely naskenovaných objektov

Na základe vzniknutých obrázkov a presných rozmerov vzniknutých 3D modelov vieme povedať nasledujúce zhodnotenia:

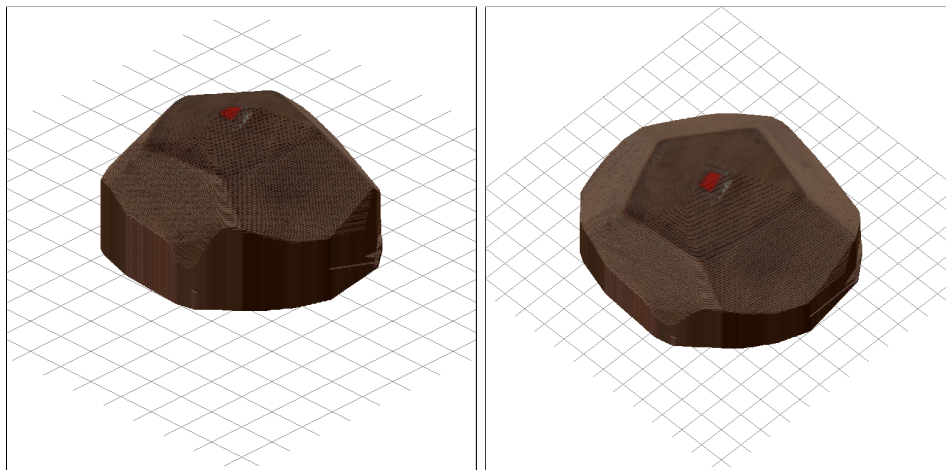
1. **Nekonvexné telesá:** Program dokáže úspešne skonstruovať reliéf nekonvexných telies s presnosťou rovnakou, akú by mal pri vizuálne podobnom konvexnom objekte.
2. **Ihlany:** Program dokáže skonstruovať reliéf ihlanu, ale vizuálna aj rozmerová odchýlka je veľmi vysoká. Táto odchýlka je spôsobená tým, že existujú plochy

objektu, ktoré vrhajú tieň na samotný objekt. Preto ich nie je možné detegovať a nedá sa určiť či v danom bode v priestore sa objekt nachádza alebo nie.

3. **Rotačné telesá:** Program dokáže veľmi úspešne skonštruovať reliéf rotačných telies a pri dostatočne vysokom počte skenovaní (10 a viac) majú rotačné telesá presnejšie rozmery ako napríklad kvádre alebo ihlany.

5.4 Súradnicové sústavy

Pri niektorých skenovaniach sme využili cylindrickú súradnicovú sústavu a pri niektorých karteziánsku. Cylindrickú sústavu sme využili na meranie súradníc pri pravidelnom 12-stene. Vzhľadom na komplikovanosť tohoto objektu, bolo potrebné urobiť vysoký počet skenovaní, aby sme dostali model s dostatočnou presnosťou. Uskotočnili sme preto 25 skenovaní pri bodovom osvetlení a vždy sme otočili 12-sten o 14.4° . Vďaka tomu, že sme nemerili umiestnenie zdroja svetla, dochádzalo k menšej odchýlke v meraniach. Výsledný 3D model mal výšku 3,6cm, zatiaľ čo skutočný 12-sten meral 3,75cm. To predstavuje je odchýlku iba 1,5mm. Vytvorený 3D model je vidieť na obrázku 5.2.



Obr. 5.2: 3D model 12-stenu

V porovnaní s karteziánskou sústavou, v ktorej sme uskutočňovali všetky ostatné merania, je cylindrická približne rovnako presná. Výhodou preto zostáva iba rýchlejšie presunutie zdroja svetla na novú pozíciu, čo sa využije najmä pri vysokom počte skenovaní, ale na presnosť programu to nemá vplyv.

Záver

V tejto práci sme najprv opísali fungovanie a prácu so skenerom CRUSE. Taktiež sme vysvetlili a zhrnuli základné definície a poznatky z oblasti tieňov, aby boli sebestačným materiálom a aby boli vhodné pre úplného začiatočníka v tejto oblasti. Zachovali sme tiež konzistenciu a pokúsili sa o dostatočnú zrozumiteľnosť s minimálnymi predpokladanými znalosťami. V tretej kapitole sme opísali jednotlivé pozorovania a vysvetlili sme dôvody prečo sme zvolili dané osvetlenie, označenie červeného bodu či konkrétne typy objektov. V štvrtej kapitole sme uviedli niektoré detaily implementácie nášho programu, ako aj ilustračné výstupné obrázky priamo z aplikácie, ktorá je súčasťou tejto práce. Dôležitou časťou bola tiež kapitola venovaná porovnaniu vzniknutých 3D modelov s pôvodnými objektami a určeni presnosti programu.

Ako pokračovanie práce, za účelom vylepšenia presnosti programu, by bolo možné navrhnúť sekundárne spracovanie vzniknutých modelov. Používateľ by mohol vložiť na začiatku dodatočné vlastnosti o skenovanom objekte a na základe týchto informácií by program upravil výstupný oblak bodov, tak aby sa viac podobal originálu. Táto dodatočná vlastnosť môže byť napríklad počet stien objektu, výška objektu alebo informácia, že objekt je rovnobežnosten, prípadne rotačné teleso.

Inou možnosťou vylepšenia algoritmu by mohlo byť určovanie sklonu objektu v konkrétnom mieste na základe jas v danej oblasti. Jas objektu je v danom mieste najväčší ak je povrch objektu v danom mieste kolmý na prichádzajúce svetelné lúče. Naopak ak je povrch objektu v niektorom mieste rovnobežný s prichádzajúcimi svetelnými lúčami, tak tam je jas minimálny. Ako dodatočné vylepšenia samotného programu je možné ešte pridať exportovanie vzniknutých modelov aj v niektorom formáte používajúcom trianguláciu a vytvoriť užívateľské rozhranie aplikácie.

Pevne veríme, že táto práca pomôže ďalším ľuďom, ktorí by chceli svojim naskenovaným objektom pridať výškovú hodnotu na jednotlivých pixloch. Nakoľko je táto metóda jednoduchou a lacnou alternatívou oproti 3D skenerom a parametre skenera CRUSE sú momentálne špičkou vo svojej oblasti, je náš program naozaj využiteľný. Zároveň sa táto práca môže stať aj podkladovým materiálom pre narábanie so skenerom CRUSE, nakoľko je to prvá dokumentácia fungovania a obsluhy tohoto skenera v slovenskom jazyku.

Literatúra

- [1] Andeggs. Cartesian, spherical and cylindrical coordinate systems. https://commons.wikimedia.org/wiki/File:Cartesian_spherical_cylindrical.svg, 2008. [Online; accessed 6-May-2019].
- [2] S. Savarese; M. Andretto. 3D reconstruction by shadow carving: Theory and practical evaluation. *Computer Vision*, 2006.
- [3] M. Daum. Extending the shape-from-darkness algorithm to three dimensions. *Computer Vision*, 1999.
- [4] F. M. Dickey; A. W. Doerry. Recovering shape from shadows in synthetic aperture radar imagery. *Computer Vision*, 2008.
- [5] M. Daum; G. Dudek. On 3-D surface reconstruction using shape from shadows. *Computer Vision*, 1998.
- [6] J. F. Hughes. Computer graphics: Principles and practice (3rd edition). *Computer Graphics*, 2013.
- [7] P. Radzim. Poznámky k školeniu CRUSE. 2018. Bratislava, FMFI UK.
- [8] CRUSE Scanner. Cruse software csx 3.9. *Firemná dokumentácia*, 2018.
- [9] M. S. Langer; G. Dudek; S. W. Zucker. Space occupancy using multiple shadow images. *Computer Vision*, 1995.

Prílohy

K práci je priložené CD-médium, na ktorom sa nachádza zdrojový kód softvérového riešenia a ukážkové 3D modely naskenovaných objektov.