

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-xxxx-xxxx

**POROVNANIE SPRACOVANIA 3D DÁT ANALYTICKÝM
METÓDAMI A NEURÓNOVÝMI SIEŤAMI**

BAKALÁRSKA PRÁCA

2022

Maroš Kocúr

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-xxxx-xxxx

**POROVNANIE SPRACOVANIA 3D DÁT ANALYTICKÝM
METÓDAMI A NEURÓNOVÝMI SIEŤAMI**

BAKALÁRSKA PRÁCA

Študijný program:	Robotika a kybernetika
Názov študijného odboru:	kybernetika
Školiace pracovisko:	Ústav robotiky a kybernetiky
Vedúci záverečnej práce:	prof. Ing. Jarmila Pavlovičová, PhD.
Konzultant:	Ing. Miroslav Kohút

Bratislava 2022

Maroš Kocúr

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Robotika a kybernetika
Autor:	Maroš Kocúr
Bakalárska práca:	Porovnanie spracovania 3D dát analytickým metódami a neurónovými sieťami
Vedúci záverečnej práce:	prof. Ing. Jarmila Pavlovičová, PhD.
Konzultant:	Ing. Miroslav Kohút
Miesto a rok predloženia práce:	Bratislava 2022

Na základe rastúcej potreby práce s 3D dátami dochádza k neustálemu testovaniu a výskumu nových možností algoritmov. Súčasná práca sa venuje algoritmu RANSAC pomocou, ktorého je možné matematicky opísať rôzne zoskupenia bodov. Modely je následne možné použiť pri ďalšom spracovaní 3D dát. Cieľom práce je porovnať analytickú metódu implementácie RANSAC modelu s jeho rozšírením, ktoré implementuje aj modely neurónových sietí na dosiahnutie lepších výsledkov.

Kľúčové slová: RANSAC, PCL, neurónová sieť

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Robotics and cybernetics
Author:	Maroš Kocúr
Bachelor's thesis:	Comparison of 3D Data Processing by Analytical Methods and Neural Networks
Supervisor:	prof. Ing. Jarmila Pavlovičová, PhD.
Consultant:	Ing. Miroslav Kohút
Place and year of submission:	Bratislava 2022

Based on the growing need to work with 3D data, there is constant testing and research of new algorithm possibilities. The current work is devoted to the RANSAC algorithm, with the help of which it is possible to mathematically describe various groupings of points. The models can then be used for further processing of 3D data. The aim of the work is to compare the implementation of the analytical method of the RANSAC model with its extension, which also implements neural network models to achieve better results.

Keywords: RANSAC, PCL, neural network

Pod'akovanie

I would like to express a gratitude to my thesis supervisor.

Obsah

Úvod	1
1 PCL	2
1.1 Inštalácia PCL	2
1.2 Používanie PCL	2
1.3 Kompilácia projektu	2
2 Point Cloud	3
2.1 General	3
2.2 Použitie	3
2.3 Konverzia do 3D povrchu	3
3 RANSAC	4
3.1 Overview	4
3.2 Algoritmus	4
3.3 RANSAC 3D	5
3.4 Neural guided RANSAC	5
4 KINECT	6
4.1 Ako funguje	6
4.2 Výstup z KINECTU	6
4.2.1 List potrebných lib pre Ubuntu 22.04 LTS	6
5 Implementovanie RANSAC algoritmu	8
6 Vystupy	9
7 Implementovanie RANSAC algoritmu s použitím NN	10
8 Zaver	11
Záver	12
Zoznam použitej literatúry	13
Prílohy	13
A Štruktúra elektronického nosiča	14

B	Algoritmus	15
C	Výpis sublime	16

Zoznam obrázkov a tabuliek

Obrázok 3.1	Priklad algoritmu v 2D rovine	4
Obrázok 4.1	Výstupný Point Cloud z Kinectu	7

Zoznam skratiek

NN	Neural Network
PCL	Point Cloud Library
RANSAC	Random sample consensus

Zoznam algoritmov

B.1	Vypočítaj $y = x^n$	15
-----	-------------------------------	----

Zoznam výpisov

1	Vytvorenie PCD pomocou Kinectu	11
C.1	Ukážka sublime-project	16

Úvod

Ukazka upraveného šablony FEIStyle.cls (<https://github.com/Kyslik/FEIStyle>) s použitím Times New Roman fontu. Nastavenie fontov som prebral z oficiálneho IEEE šablony, vložil som ho do FEIStyle.cls na riadky 228-230.

Tu bude krásny úvod s diakritikou atď.

A možno aj viac riadkový úvod.

1 PCL

PCL je samostný, vysoko škalovaliteľný, otvorený projekt pre 2D a 3D obrázky a spracovávanie point cloud. Knižnica je cross platform, napísaná v jazyku C++ a Python. Najčastejšie sa používa na operačnom systéme Linux. Existujú package aj pre macOS a Windows vytvorené tretími stranami. My v tomto projekte budeme používať Ubuntu 22.04. LTS a verzia PCL je 1.12.1. Knižnica je vydaná pod BSD licenciami čo znamená, že je voľne použiteľná na úroveň komerčné účely a za účelom výskumu. Random sample consensus (RANSAC) je dlhá skratka naopak NN je skratka v krátkej forme. PCL je skratka v krátkej forme.

1.1 Inštalácia PCL

PCL je dostupný na mnoho distribúcií Linuxu ako Ubuntu, Debian, Fedora, Gentoo a Arch Linux. PCL na distribúciách Ubuntu a Debian môžeme nainštalovať pomocou.

```
sudo apt install libpcl-dev
```

Na Windowse sa PCL inštaluje pomocou wvpack package manažéra vytvoreného Microsoftom.

```
PS> .install pcl
```

MacOS má Homebrew package manažéra ktorý podporuje inštaláciu package, ktoré Apple alebo Linux nedokáže nainštalovať. `brew install pcl` Toto sú odporúčane inštalácie pre PCL na daných operačných systémoch.

1.2 Používanie PCL

Na používanie PCL si potrebujeme v našom kóde vložiť potrebné knižnice. Po nainštalovaní sa v našom systéme nastavujú premenné, takže stačí nám použiť príkaz v C++, `include <pcl/“názov_knižnice1.h” > .`

1.3 Kompilácia projektu

Vytvoríme si súbor CMakeList.txt v ktorom zadefinujeme potrebné premenné aby make vedel najst' cestu ku knižnici a vedel aké súbory má skompilovať. Ďalej vytvoríme priečinok s názvom build, v termináli vojdeme do neho a pomocou príkazu `cmake` “cesta k CMakeList.txt“, si vytvoríme makefile. Ďalej používame len príkaz `make` ktorý nám vytvorí súbor pomocov ktorého môžeme spustiť projekt.

2 Point Cloud

2.1 General

Je to súbor bodov v priestore. Body môžu reprezentovať 3D tvary alebo objekty. Každý bod má karteziánske súradnice (X,Y,Z). Point cloud je generovaný pomocou 3D skenera alebo pomocou softwaru na fotogrametriu, ktorý meria veľa bodov na externom povrchu objektov okolo. My v tomto projekte použijeme Kinect 2 (odsek 6). Point cloud sa používa v 3D modelovaní, metrológii, meranie kvality výrobkov a rôzne vizualizácie. Point cloud sa často zarovnáva s 3D modelmi alebo inými point cloudmi ako registrácia množín bodov.

2.2 Použitie

Pre priemyselnú metrológiu alebo inšpekciu pomocou priemyselnej počítačovej tomografie možno mračno bodov vyrobeného dielu zosúladiť s existujúcim modelom a porovnať, aby sa skontrolovali rozdiely. Geometrické rozmery a tolerancie možno získať aj priamo z point cloudu.

2.3 Konverzia do 3D povrchu

V geografických informačných systémoch sú point cloudi jedným zo zdrojov využívaných na tvorbu digitálneho výškového medelu terénu. Používajú sa aj na generovanie 3D modelov mestského prostredia. Drony sa často využívajú na nazbieranie RGB obrázkov ktoré sa neskôr pomocou algoritmu strojového videnia ako je AgiSoft Photoscan, PixčD alebo DroneDeploy používajú na vytvorenie RGB point cloudu, kde sa môže požiť vzdialenostná a objemová aproximácia.

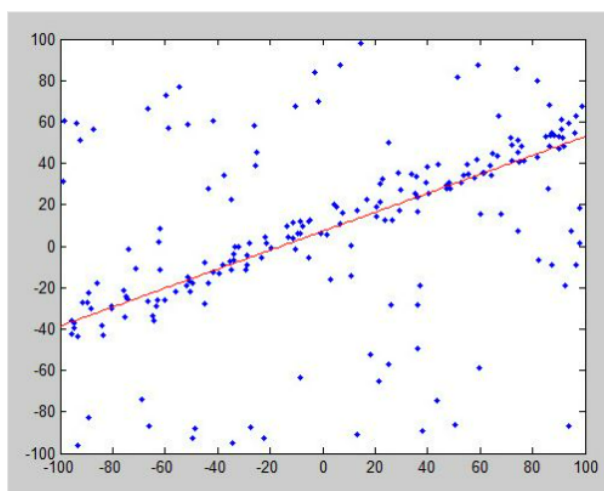
3 RANSAC

3.1 Overview

RANSAC je algoritmus vytvorený Fischlerom a Bollesom, určuje všeobecný prístup k odhadu parametrov, s veľkým podielom outliers v stupnom datasete. Na rozdiel od iných výkonných algoritmov odhadu, ako napríklad M-estimators a meródov najmenších štvorcov s prepojením na strojové učenie. RANSAC bol vytváraný komunitou ľudí používajúcich strojové učenie. RANSAC je vzorkovacia technika ktorá generuje kandidáta na minimálny počet pozorovaní potrebných na zistenie odhadu parametrov ležiacich pod modelom. Na rozdiel od ostatných vzorkovacích algoritmov, ktoré používajú čo najviac bodov ako môžu, RANSAC používa najmenší počet bodov ako môže.

3.2 Algoritmus

1. Vybranie minimum náhodných bodov potrebných na určenie parametrov modelu
2. Vyhľadanie parametrov pre model
3. Určenie koľko bodov z množiny všetkých bodov leží s preddefinovanou .
4. Ak zlomok bodov ležiacich v preddefinovanej , presahuje preddefinovaný prah , prehodnotí parametre modelu použitím všetkých identifikovaných inliers a ukončí.
5. Inak, opakuj kroky 1 až 4, s maximálnym N opakovaniami.



Obr. 3.1: Příklad algoritmu v 2D rovine

Modré sú body z datasetu, pomocou RANSAC algoritmu sa určili 2 body, ktoré majú vo svojom subsele najmenej bodov ležiacich mimo alfy.

3.3 RANSAC 3D

Podobne ako pri opise vyššie RANSAC 3D funguje s výberom náhodných troch bodov na ktorých zostaví rovinu a spočíta body ležiace v rovine a body ležiace mimo roviny. Algoritmus sa opakuje n -opakovaní a výstupom je najlepší model.

3.4 Neural guided RANSAC

4 KINECT

Kinect je vstupné zariadenie snímajúce pohyb, vyrobené Microsoftom. Zariadenie obsahuje RGB kamery a infračervený projektor a detektor ktorý monitoruje hĺbku priestoru na základe štrukturovateľného svetla alebo na základe času trvajúceho svetlu dopadnúť na objekt, vďaka ktorému vie kinect poskytnúť rekognizáciu miest v reálnom čase. Kinect sa používa hlavne v hernom priemysle, ale používa sa taktiež na komerčné a akademické účely pretože poskytuje mapovanie priestoru a je lacnejší než profesionálne zariadenia.

4.1 Ako funguje

Infračervený projektor na kinecte posiela modulované infračervené svetlo ktoré je zachytené sensormi. Infračervené svetlo ktoré sa odrazí od bližších objektov má kratší čas letu ako svetlo ktoré sa odrazí od vzdialenejších objektov, takže sensor sníma ako vymodulovaný vzor bol deformovaný z času letu svetla, pixel po pixeli. Čas priletu meranej hĺbky touto metódou môže byť presnejšie vypočítaný v kratšom čase, čo zabezpečí viac snímkov za sekundu. Hneď ako kinect naskenuje hĺbkovú fotografiu, použije metódu zisťovania hrán k vytýčeniu bližších objektov z pozadia fotky.

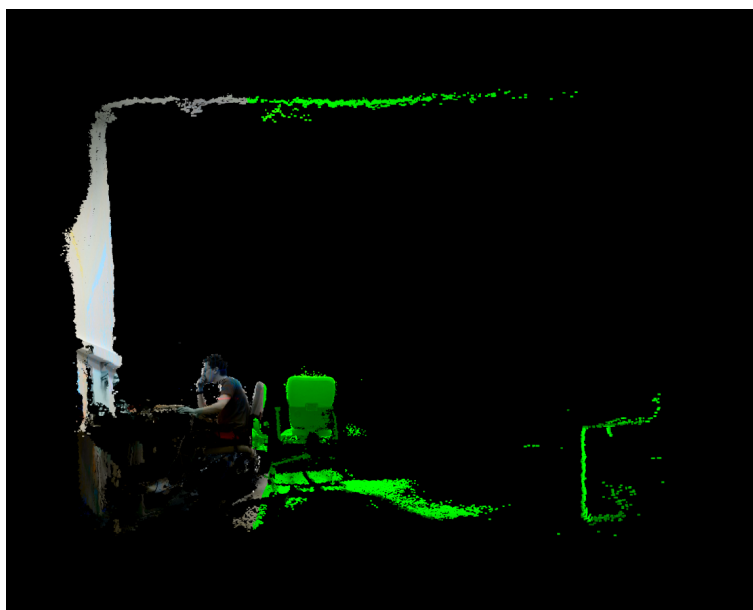
4.2 Výstup z KINECTU

Na získanie výstupu z Kinectu je potrebné si nainštalovať Open-source knižnicu LibFreenect2, ktorá je určená na získavanie videí z Kinectu verzie 2. Knižnica disponuje vzorovým kódom, ktorý po spustení zapne Viewer a okno rozdelí na štyri časti a v nich uvidíme výstupy Infra Red kamery, farebnej kamery a depth kamery. Nato aby sme knižnicu mohli používať potrebovali sme doinštalovať potrebné knižnice a súčasti. Na získanie snímky z Kinectu sme použili Python script, ktorý nám vytvorí point cloud súbor, v ktorom budeme hľadať požadované tvary.

4.2.1 List potrebných lib pre Ubuntu 22.04 LTS

1. libusb
2. TurboJPEG
3. OpenGL
4. OpenCL (optional)
5. CUDA (optional, NVIDIA only)
6. VAAPI (optional)

7. OpenNI2



Obr. 4.1: Výstupný Point Cloud z Kinectu

5 Implementavanie RANSAC algoritmu

6 Vystupy

7 Implementovanie RANSAC algoritmu s použitím NN

8 Zaver

```
from freenect2 import Device, FrameType
import numpy as np

# Open the default device and capture a color and depth frame.
device = Device()
frames = {}
with device.running():
    for type_, frame in device:
        frames[type_] = frame
        if FrameType.Color in frames and FrameType.Depth in frames:
            break

# Use the factory calibration to undistort the depth frame and register the RGB
# frame onto it.
rgb, depth = frames[FrameType.Color], frames[FrameType.Depth]
undistorted, registered, big_depth = device.registration.apply(
    rgb, depth, with_big_depth=True)

# Combine the depth and RGB data together into a single point cloud.
with open('output.pcd', 'wb') as fobj:
    device.registration.write_pcd(fobj, undistorted, registered)

with open('output_big.pcd', 'wb') as fobj:
    device.registration.write_big_pcd(fobj, big_depth, rgb)
```

Výpis 1: Vytvorenie PCD pomocou Kinectu

Záver

Conclusion is going to be where?

Here.

Prílohy

A	Štruktúra elektronického nosiča	14
B	Algoritmus	15
C	Výpis subline	16

A Štruktúra elektronického nosiča

/CHANGELOG.md

- file describing changes made to FEIstyle

/example.tex

- main example *.tex* file for diploma thesis

/example_paper.tex

- example *.tex* file for seminar paper

/Makefile

- simply Makefile – build system

/fei.sublime-project

- is project file with build in Build System for Sublime Text 3

/img

- folder with images

/includes

- files with content

/bibliography.bib

- bibliography file

/attachmentA.tex

- this very file

B Algoritmus

Algoritmus B.1 Vypočítaj $y = x^n$

Require: $n \geq 0 \vee x \neq 0$

Ensure: $y = x^n$

$y \leftarrow 1$

if $n < 0$ **then**

$X \leftarrow 1/x$

$N \leftarrow -n$

else

$X \leftarrow x$

$N \leftarrow n$

end if

while $N \neq 0$ **do**

if N is even **then**

$X \leftarrow X \times X$

$N \leftarrow N/2$

else { N is odd }

$y \leftarrow y \times X$

$N \leftarrow N - 1$

end if

end while

C Výpis sublime

```
../.. ./ fei .sublime-project
```

Výpis C.1: Ukážka sublime-project