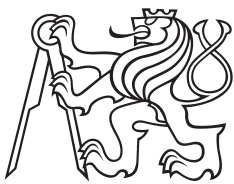


Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Detekce objektů z hloubkové kamery

Lukáš Kunt

Vedoucí: Dr. Petr Štěpán
Studijní program: Kybernetika a robotika
Květen 2020

Poděkování

Prohlášení

Abstrakt

Klíčová slova: slovo, klíč

Vedoucí: Dr. Petr Štěpán
Ústav X,
Uliční 5,
Praha 99

Abstract

Keywords: word, key

Title translation: Object Detection
from Depth camera — Journey to the
who-knows-what wondeland

Obsah

1 Matematický aparát	1
1.1 Homogenní souřadnice	1
1.2 Rotace a rotační matice	1
1.3 Sobelův operátor pro detekci hran	2
1.4 Hledání konvexního obalu množiny bodů	3
1.5 Vlastní čísla a vektory	3
1.6 Singulární rozklad	4
1.7 Analýza hlavních komponent	4
2 Kamera	5
2.1 Dírkový model kamery	5
2.2 Stereo kamera	6
2.2.1 Výpočet vzdálenosti	6
2.2.2 Chyba výpočtu vzdálenosti . . .	7
2.2.3 Epipolární geometrie	7
2.3 Hledání stereo páru	8
2.3.1 Plošné metody	8
2.3.2 Porovnávání rysů	9
2.4 Intel Realsense D435	9
2.5 Metody v detekci objektů z hloubkových dat	11
2.5.1 Klasické metody	11
2.5.2 Segmentace obrazu	12
2.5.3 Určování normálového vektoru	14
2.6 Výpočet normálového vektoru plochy neregulárního mráčka bodů	15
2.7 RANSAC	15
2.8 Zvolený postup	15
2.8.1 Segmentace obrazu	15
2.8.2 Detekce objektů ze segmentovaného obrazu	17
Literatura	21

Obrázky

Tabulky

2.1 Model dírkové kamery	6
2.2 [2]	8
2.3 Placeholder	10
2.4 Fotografie kamery Intel® Realsense™ D435 [1]	10
2.5 Postupně zleva: hodnota limitu, počet falešných shod pokud je scéna blíže, než je minimální vzdálenost detekce kamerou. Vzdálenost, kdy vyplněnost hloubkové mapy klesne pod 95% a vzdálenost při které je směrodatná odchylka ve směru osy z menší, než ve směru x a y [11]	11
2.6 Tabulka minimální detekovatelné hloubky [3]	11
2.7 Výstupní data kamery Realsense, hloubková data byla převedena do odstínů šedé	12
2.8 Červenou barvou jsou znázorněny body, ze kterých probýhal algoritmus 1, modré body byly podle výše uvedených kritérií vyřazeny a zeleně jsou znázorněny výsledné body reprezentující zem	17
2.9 Vzdálenost boudů od roviny zobrazena jako počet na sobě lžících cihel	18
2.10 Výstup kamery - mračno bodů zobrazující skupinu cihel. V pravé části je vidět zkreslení hrany způsobující problém při segmentaci	19
2.11 Výsledek segmentace obrazu podle velikosti derivace výšky.	19

Kapitola 1

Matematický aparát

1.1 Homogenní souřadnice

Během celé práce budu pracovat jak s kartézskými, tak i s homogenními souřadnicemi. Tyto jsou v Euklidovském prostoru dimenze n reprezentovány pomocí vektoru, který má $n + 1$ prvků. Je-li \mathbf{r} bod v trojdimenzionálním Euklidovském prostoru \mathbb{E}^3 reprezentován parametry x, y, z , pak pro převod mezi kartézskými a homogenními souřadnicemi platí následující vztahy.

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \mathbf{r}_{hom} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1.1)$$

$$\mathbf{r}_{hom} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \rightarrow \mathbf{r} = \begin{bmatrix} x/w \\ y/w \\ z/w \end{bmatrix} \quad (1.2)$$

Pomocí homogenních souřadnic můžeme sestavit transformační matici \mathbf{A} , takto se skládá z matice rotace \mathbf{R} a vektoru translace \mathbf{t} .

$$\mathbf{A} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (1.3)$$

1.2 Rotace a rotační matice

Natočení souřadného systému, popřípadě objektu s tímto svázaným vzhledem k jinému systému v prostoru dimenze n se nejjednodušeji popíše pomocí rotační matice $\mathbf{R} \in \mathbb{R}^n$. Mezi nejběžnější rotační matice patří rotace o úhel θ

kolem os kartézského souřadného systému v \mathbb{E}^3 . Tyto vypadají následovně.

$$\mathbf{R}_x = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$

$$\mathbf{R}_y = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

$$\mathbf{R}_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.6)$$

Obecné natočení pak můžeme dostat složením 3 a více těchto rotací¹ Pro vytvoření matice rotace \mathbf{R} v prostoru \mathbb{R}^3 kolem obecné osy dané normalizovaným vektorem \mathbf{u} o úhel θ slouží Rodriguesův vzorec [17].

$$\mathbf{R} = \mathbf{I} + \tilde{\mathbf{u}} \sin \theta + \tilde{\mathbf{u}}^2 (1 - \cos \theta); \quad \tilde{\mathbf{u}} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \quad (1.7)$$

Chceme-li natočit vektor \mathbf{a} tak, aby byl rovnoběžný s vektorem \mathbf{b} použijeme vzorec 1.7, úhel rotace θ a vektor \mathbf{v} kolem kterého rotace proběhne se určí následovně.

$$\mathbf{v} = \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a} \times \mathbf{b}\|} \quad (1.8)$$

$$\theta = \arccos \left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \right) \quad (1.9)$$

Podle [9] je tato metoda nevhodná pro výpočet na počítači. Zejména pak v případech, kdy bychom měli počítat arccos hodnot blízko ± 1 . Toto je způsobeno omezenou přesností reprázentace desetinných čísel v počítači (takzvaná floating point aritmetika). Přesnější je tedy použít tento vzorec.

$$\theta = \text{atan2}(\|\mathbf{a} \times \mathbf{b}\|, \mathbf{a} \cdot \mathbf{b}) \quad (1.10)$$

Výhoda funkce atan2 oproti arcus tangens je oštréné dělení nulou, ke kterému může dojít při zaokrouhlování desetinných čísel. Navíc funkce vrací úhel v rozmezí $[0, 2\pi]$

1.3 Sobelův operátor pro detekci hran

Hrana v obrazu může být také pozorována, jako prodká změna intenzity obrazu v daném místě. Rychlé změny se dají detekovat pomocí gradientu, který je pro spojitou funkci $f(x, y)$ v bodě (x, y) vyjádřen následovně.

$$\nabla f(x, y) = \begin{bmatrix} \mathbf{G}_x & \mathbf{G}_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix} \quad (1.11)$$

¹Na pořadí zde záleží, není možné dostat libovolnou rotační matici, pokud budou dvě po sobě doucí rotace probíhat kolem stejné osy

Velikost a směr gradientu se jsou určeny následovně.

$$\text{mag}(\nabla f) = \|\nabla f\|_2 = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \quad (1.12)$$

$$\theta = \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right) \quad (1.13)$$

Parciální derivace musí být spočítané pro každý pixel (x, y) obrazu \mathbf{A} . K tomuto se využívá Sobelových kernelů $\mathbf{K}_x, \mathbf{K}_y$. Tyto se konvolvuji s obrazem a v každém bodě aproximují hodnotu obou parciálních derivací $g_x(x, y), g_y(x, y)$. Ze kterých se pak podle vztahu 1.11 určí gradient.

$$\mathbf{K}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & -1 \end{bmatrix} \quad \mathbf{K}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (1.14)$$

$$g_x(x, y) = \sum_{k=-1}^1 \sum_{l=-1}^1 \mathbf{K}_x(k, l) \mathbf{A}(x+k, y+l) \quad (1.15)$$

$$g_y(x, y) = \sum_{k=-1}^1 \sum_{l=-1}^1 \mathbf{K}_y(k, l) \mathbf{A}(x+k, y+l) \quad (1.16)$$

$$(1.17)$$

Výpočet gradientu intenzity tímto způsobem je výpočetně relativně nenáročný. Pro další zrychlení výpočtu se používá aproximace magnitudy gradientu místo exaktního výpočtu 1.12. Nejjednodušší aproximací je

$$\text{mag}(\nabla f) \approx \|\nabla f\|_1 = |\mathbf{G}_x| + |\mathbf{G}_y| \quad (1.18)$$

tato dává sice velice nepřesný výsledek, ale pro aplikace, kde stačí hrubý odhad velikosti gradientu může značně snížit výpočetní dobu.

1.4 Hledání konvexního obalu množiny bodů

Je-li dána množina S obsahující n bodů v euklidovském prostoru \mathcal{E}^3 , popřípadě euklidovské ploše \mathcal{E}^2 , pak konvexní obal množiny S je nejmenší konvexní množina obsahující množinu S [6].

1.5 Vlastní čísla a vektory

Pro čtvercovou matici $\mathbf{A} \in \mathbb{R}^{n \times n}$ a nenulový vektor $\mathbf{v} \in \mathbb{R}^n$ a skálár $\lambda \in \mathbb{R}$ platí

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}. \quad (1.19)$$

Skalár λ se nazývá vlastní číslo matice \mathbf{A} a vektor \mathbf{v} vlastní vektor příslušný vlastnímu číslu λ . Spektrální věta (viz. [18]) říká: Je-li matice \mathbf{A} symetrická, pak má n vlastních čísel, kde $n = \text{rank}\mathbf{A}$ a všechna vlastní čísla jsou reálná.

1.6 Singulární rozklad

Singulární SVD²rozklad umožní každou matici $\mathbf{A} \in \mathbb{R}^{m \times n}$ rozložit jako

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (1.20)$$

kde matice \mathbf{S} je diagonální, matice \mathbf{U} je ortogonální a obsahuje vlastní vektory matice $\mathbf{A}\mathbf{A}^T$, obdobně je pak i matice \mathbf{V} ortogonální a je složena z vlastních vektorů matice $\mathbf{A}^T\mathbf{A}$

1.7 Analýza hlavních komponent

Analýza hlavních komponent (PCA³)

²Z anglického Singular Value Decomposition

³Z anglického Principal Component Analysis

Kapitola 2

Kamera

2.1 Dírkový model kamery

Kamera zobrazuje body $\mathbf{p} \in \mathbb{E}^3$ na $\mathbf{p}_o \in \mathbb{E}^2$. Je to tedy zobrazení mapující body ve světovém souřadném systému na body v souřadném systému obrazu kamery. Nejjednodušším modelem je dírkový. Jedná se o model kamery s konečným optickým centrem, neboli centrem projekce. Toto umístíme do počátku kartézského souřadného systému a budeme uvažovat plochu kolmou na osu Z našeho souřadného systému, kterou nazveme obrazovou rovinou tato je popsána jediným parametrem a to ohniskovou vzdáleností f . V dírkovém modelu kamery se bod v prostoru $\mathbf{p} = (x, y, z)^T$ zobrazí na bod obrazu pomocí následující zobrazení

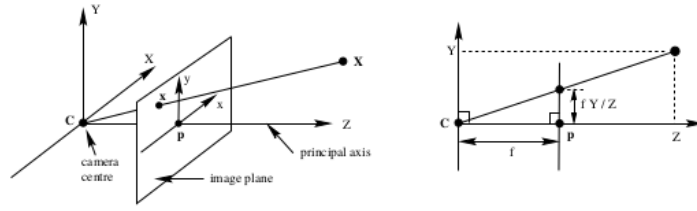
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \mapsto \begin{bmatrix} fx/z \\ fy/z \end{bmatrix} \quad (2.1)$$

což odpovídá bodu, kde přímka spojující centrum projekce a bod v prostoru, protne obrazovou rovinu (viz obrázek 2.1). Ve vztahu 2.1 jsme předpokládali, že počátek souřadného systému obrazu je v optickém středu. Konvencí však je za počátek souřadnic zvolit levý horní roh obrazu. K zápisu využijeme homogenních souřadnic a dostaneme následující rovnici pro zobrazení z souřadného systému kamery do systému obrazu s počátkem v levém horním rohu obrazu

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fx + zc_x \\ fy + zc_y \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.2)$$

kde jako c_x a c_y jsme označili souřadnice optického středu. Označíme-li obecný bod v prostoru, který je popsán v souřadném systému kamer \mathbf{x}_p a odpovídající bod v rovině obrazu \mathbf{x} , pak můžeme rovnici 2.2 přepsat jako

$$\mathbf{x} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}] \mathbf{x}_p \quad \mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$



Obrázek 2.1: Model dírkové kamery

Matice \mathbf{K} se nazývá matice vnitřních parametrů kamery.

Obecně je bod v prostoru \mathbf{x} popsán v jiném souřadném systému, než je ten se kterým je svázána kamera, tento se nejčastěji nazývá světový souřadný systém. Přechod do souřadného systému kamery ze světového je možno popsat následovně

$$\mathbf{x}_p = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{x} \quad (2.4)$$

Kde jsme jako $\tilde{\mathbf{C}}$ označili počátek souřadného systému kamery vyjádřený v světových souřadnicích a \mathbf{R} rotaci světového souřadného systému vzhledem k kameře. Dosadíme-li nyní rovnici 2.3 do 2.4 dostane vztah popisující transformaci bodu z světových souřadnic do souřadnic obrazu.

$$\mathbf{x} = \mathbf{KR}[\mathbf{I} \mid -\tilde{\mathbf{C}}]\mathbf{x} \quad (2.5)$$

Parametry $\tilde{\mathbf{C}}$ a \mathbf{R} se souhrnně nazývají vnější parametry kamery.

2.2 Stereo kamera

2.2.1 Výpočet vzdálenosti

Stereo kamera je tvořena dvěma nezávislými čočkami. Zpracováním obrazů jsme schopní získat informaci o hloubce každého bodu obrazu. Budeme uvažovat model stereo kamery, kde osy obou čoček jsou rovnoběžné (viz obrázek 2.2). Osy jsou od sebe ve vzdálenosti b , tomuto parametru se říká báze (baseline). Z podobnosti trojúhelníků plyne

$$\frac{z}{f} = \frac{x}{xl} \quad (2.6)$$

$$\frac{z}{f} = \frac{x - b}{xr} \quad (2.7)$$

pokud z rovnice 2.6 vyjádříme x a dosadíme do rovnice 2.7, pak po úpravě dostaneme výsledný vztah pro vzdálenost bodu od kamery. Hodnota d se pak nazývá rozdíl (disparity).

$$z = \frac{fb}{xl - xr} = \frac{fb}{d} \quad (2.8)$$

Kde x_l a x_r jsou souřadnice bodu \mathbf{p} v levém a pravém obraze stereokamery. Pro každý bod v levém (popřípadě pravém) obraze je tedy potřeba najít odpovídající bod v obraze pravém (tzv. stereo pár). Bez jakékoliv apriorní znalosti by to znamenalo pro každý pixel prohledat celý druhý obraz, tedy náročnost hledání $\mathcal{O}(wh)$, tj. plocha obrazu vyjádřená v pixelech. Pomocí epipolární geometrie můžeme hledání zúžit na přímku a snížit tak náročnost hledání stereo páru na $\mathcal{O}(w)$. [5]

2.2.2 Chyba výpočtu vzdálenosti

Chyba výpočtu vzdálenosti se podle [11] určí derivací rovnice 2.8 podle d .

$$\frac{\partial z}{\partial d} = \frac{z}{fb} \quad (2.9)$$

$$|\epsilon_z| = \frac{z^2}{fb} |\epsilon_d| \quad (2.10)$$

Kde ϵ_z je chyba určení vzdálenosti bodu a ϵ_d je chyba rozdílu, toto je vlastnost kamery, popřípadě algoritmu použitého ke hledání odpovídající párů (viz. TODO fill section) a nabývá téměř konstantních hodnot [11]. Z 2.10 tedy plyne, že chyba je úměrná kvadrátu vzdálenosti daného bodu a přesnost stereo kamer se vzdáleností rychle klesá.

2.2.3 Epipolární geometrie

Epipolární geometrie se zabývá průnikem obrazových rovin se svazkem ploch, jejichž společná osa je báze. Uvažujme dvě dírkové kamery v obecném natočení, tedy jejich obrazové roviny nemusí být rovnoběžné (viz 2.3). Z dírkové modelu kamery plyne, že bod \mathbf{X} a optické středy kamer C, C' jsou koplanární a tvoří rovinu p . V místě, kde tato rovina protíná rovinu obrazu se nachází epipolární přímka l . Hledáme-li stereo pár, pak pro známý bod \mathbf{x} (tj. projekce bodu \mathbf{X} do optické roviny levé kamery) musíme najít odpovídající bod \mathbf{x}' . Tento může ležet pouze na epipolární přímce l' . Přímka l' je určena epipólem e' a bodem x' . Epipól se nachází v místě, kde přímka spojující optická centra kamer protíná rovinu obrazu. Bod x' se určí jako

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \quad (2.11)$$

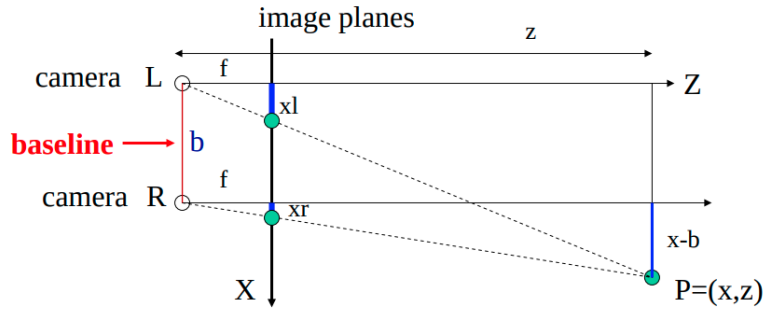
Kde \mathbf{H} je homografie mezi rovinami obrazu. Toto mapování je ovlivněno pouze vzájemnou polohou kamer a jejich vnitřními parametry a nezávisí tedy na scéně. Z bodu \mathbf{x}' se pak již určí epipolární přímka l' .

$$\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' \quad (2.12)$$

$$\mathbf{l}' = \mathbf{e}' \times \mathbf{H}\mathbf{x} \quad (2.13)$$

$$\mathbf{l}' = \mathbf{F}\mathbf{x} \quad (2.14)$$

Matice \mathbf{F} se nazývá fundamentální a $\text{rank } \mathbf{F} = 2$, tedy zobrazení reprezentované maticí \mathbf{F} zobrazí bod na přímku. Pokud jsou obě optické osy stereo kamery rovnoběžné, pak se výpočet epipolární přímky velice zjednoduší, jelikož homografie mezi rovinami obrazu je pouze posunem ve směru báze.



Obrázek 2.2: [2]

2.3 Hledání stereo páru

Metody užívané pro hledání stereo parů se dělí na metody plošné (area based) a ty založené na porovnávání rysů (features). Při hledání stereo páru se často využívá následujících omezení, která tento proces zjednoduší. Tato však v extrémních případech nemusí platit.

- Epipolární omezení: pixel v druhém obraze se nachází (pokud existuje) na epipolární přímce (vit. 2.2.3)
- Spojitost: Rozdílová, popř. hloubková mapa by měla být po částech spojitá.
- Jedinečnost: Každý pixel v levém obraze má k sobě právě jeden odpovídající v obraze pravém.
- Pořadí: Pořadí pixelů v levém obraze odpovídá pořadí v pravém

2.3.1 Plošné metody

Využívají informaci z okolí pixelu. Tyto jsou porovnány se všemi přípustnými pozicemi v druhém obraze a bod, jehož hodnota je nejbližší vzoru tvoří stereo pár. Mezi nejpoužívanější algoritmy patří.

Součet absolutních rozdílů

Je definována konstantní velikost okolí pixelu $\mathbf{x}(u, v)$, pro který se hledá stereo pár. Tato je následně porovnávána s body, kde se může korespondující pixel nacházet, tj. body které splňují námi používaná omezení. Pro odpovídající bod \mathbf{x}' tedy platí následující vztah.

$$\mathbf{x}' = \operatorname{argmin} \left(\sum_i \sum_j |\mathbf{I}_l(u + i, v + j) - \mathbf{I}_r(u + i, v + j + d)| \right) \quad (2.15)$$

kde $\mathbf{I}(u, v)$ je intenzita příslušného pixelu, d je posun okna ve druhém obraze a součet probíhá přes celé okno, tedy přes každé i, j . Vztah byl zjednodušen

uvažováním horizontálních epipolárních přímek. To odpovídá konfiguraci kamery s rovnoběžnými optickými osami.

Existuje spousta podobných algoritmů, které pouze používají jinou metrickou funkci. Místo absolutní hodnoty například kvadrát rozdílu, nebo normalizovaný kvadrát rozdílu. Jedná se o nejrychlejší algoritmy pro hledání stereo párů, přesto však dosahují vysokých přesností [12].

■ Cenzus

Jedná se o metodu patřící do rodiny plošných metod, které před porovnáním hodnot pixelů provedou určitou transformaci obou obrazů, sem patří například i transformace podle hodnoty (rank transform). Tato metoda opět pracuje se definovanou maskou kolem pixelu, pro který hledá stereo pár. Pixel, který se nachází v šabloně se přemění na vektor jedniček a nul, podle toho zda je hodnota intenzity větší, popřípadě menší než intenzita centrálního pixelu. Stejná transformace se provede i v pravém obrazu pro každý pixel, který může doplnit stereo pár. Vektory jsou následně porovnány a je vybrán ten, jehož vektor má nejmenší Hammingovu vzdálenost. [12, 5]

■ 2.3.2 Porovnávání rysů

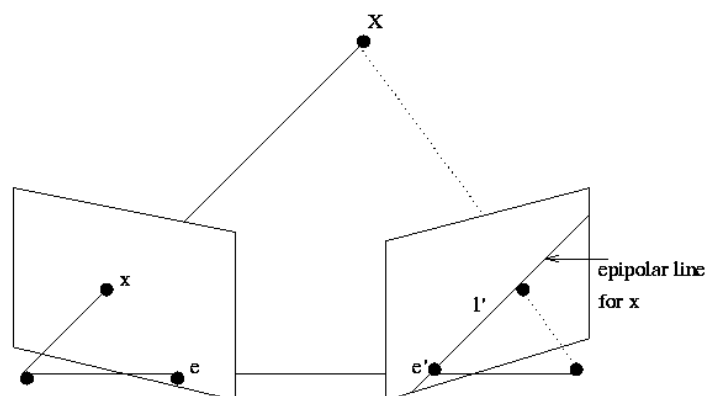
Pro každý obraz jsou vygenerovány důležité body. Tyto mohou být například hrany či rohy. Tyto jsou porovnány s analogicky vygenerovanými rysy v druhém obrazu. Tento postup se často opakuje, kdy v každém dalším kroku je rozložen každý rys na několik menších, o kterých již máme apriorní znalost jejich přibližné polohy. Aby bylo možné sestavit obraz pouze z několika rysů je nutné provést časově náročné předzpracování obrazu za účelem najetí vhodných rysů a po nalezení korespondujících rysů provést zpětnou rekonstrukci obrazu. [5]

Pro porovnání rysů mezi levým a pravým obrazem se využívá algoritmů jako je například SIFT (Scale Invariant Feature Transform), SURF (Seeded-Up Robust Features) nebo BRIEF (Binary Robust Independent Elementary Features).

■ 2.4 Intel Realsense D435

Intel® Realsense™ D435 je širokoúhlá stereo kamera, která se skládá z RGB kamery, infračerveného (infra red) projektoru, a dvou infračervených kamer jejichž data jsou zpracovává přímo v čipu kamery procesor. Výstupem z kamery je tedy barevný obraz (RGB) a vzdálenost jednotlivých pixelů neboli hloubková mapa.

Dvě infračervené kamery jsou v konfiguraci s rovnoběžnými optickými osami ve vzdálenosti 50 mm. Infračervený projektor promítá na scénu statický obrazec, který se nachází mimo viditelné spektrum a není tedy zachycen RGB kamerou. Tento obrazec slouží k najetí stereo párů, zejména pak u objektů s řídkou texturou.



Obrázek 2.3: Placeholder



Obrázek 2.4: Fotografie kamery Intel® Realsense™ D435 [1]

Kamera je vybavena specializovaným procesorem Vision Processor D4 pro výpočet hloubkové mapy. Tento je schopen při rozlišení hloubkové kamery 848x480 zpracovávat až 90 snímků za sekundu (FPS), toto je vykoupeno sníženou přesností [11] a pro maximální přesnost je vhodné snížit frekvenci na 30 snímků za sekundu. Procesor má stejně jako celá kamera velice nízký odběr. Celá kamera včetně IR projektoru má odběr menší než jeden watt a je tedy vhodná pro mobilní aplikace, kde je velikost baterie a spotřeba energie omezujícím faktorem. [3]

Pro hledání stereo párů je využit census algoritmus popsáný v 2.3.1 s maskou velikosti 7×7 . Výsledky jsou následně ověřovány sadou filtrů, které měří důvěryhodnost shody. Podle nastaveného limitu pak pro tento pixel buď vygenerují záznam v hloubkové mapě, nebo pixel zůstane nevyplněn. Výsledky kamery pro různé hodnoty limitu jsou vidět v tabulce.

Procesor kamery není schopen určit hloubku bodu, pokud se objekt nachází příliš blízko. Konkrétní hodnoty je možno vidět v tabulce 2.6. Maximální měřitelná vzdálenost je za ideálních podmínek až 40 metrů [11]. S rostoucí vzdáleností se kamera chová podle rovnice 2.10, toto chování není patrné, pokud kamera operuje s objekty, které se nachází na obou hranicích měřitelné vzdálenosti. Ilustrace výstupu kamery, pokud se objekt nachází ve vzdálenosti mezi, než specifikuje tabulka 2.5 je vidět na obrázku 2.7b.

Preset	FPR	$r_{p=95\%}$	$\max(\sigma_x, \sigma_y) > \sigma_z$
Off	91.3%	7.0 m	6.1 m
Low	19.8%	6.9 m	6.6 m
Medium	5.8%	5.8 m	6.7 m
High	0.5%	4.2 m	6.8 m

Obrázek 2.5: Postupně zleva: hodnota limitu, počet falešných shod pokud je scéna blíže, než je minimální vzdálenost detekce kamerou. Vzdálenost, kdy vyplněnost hloubkové mapy klesne pod 95% a vzdálenost při které je směrodatná odchylka ve směru osy z menší, než ve směru x a y [11]

Resolution	D400/D410/D415	D420/D430
	Min-Z (mm)	Min-Z (mm)
1280x720	450	280
848x480	310	195
640x480	310	175
640x360	240	150
480x270	180	120
424x240	160	105

Obrázek 2.6: Tabulka minimální detekovatelné hloubky [3]

2.5 Metody v detekci objektů z hloubkových dat

Existuje spousta různých metod pro detekci objektů z hloubkových dat. Tyto se liší podle různorodosti detekovaných objektů, požadavků na čas výpočtu, možnosti získání ténovacích dat nebo formátem vstupních dat. Obvyklým formátem je mračno bodů, neboli point cloud, což je nestrukturovaná množina bodů v trojrozměrném prostoru. Zejména v posledních letech s nástupem hloubkových kamer jako je například Microsoft Kinect, Asus Xtion nebo námi používaný Intel RealSense, se čím dál více používají data ve formátu RGB-D obrazu, tento se dá pomocí vztahu 2.2 převést na point cloud. Výstupní data z hloubkové kamery bývají většinou oproti LIDARu méně přesné. Výhodami naopak jsou: Vysoká vzorkovací frekvence, velké množství výstupních bodů a cena.

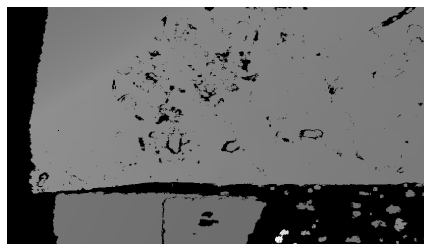
2.5.1 Klasické metody

Určení normálového vektoru plochy z nestrukturovaných dat

Téměř všechny metody detekce z objektů se skládá z více kroků a jedním z nich většinou bývá určení normálového vektoru plochy. Obvykle se pak jedná o plochu reprezentující zem, popřípadě jinou podložku, popřípadě plochy reprezentující stěny místnosti. Přístupy k hledání ploch v obrazu se výrazně liší, uvedeme příklady některých z nich.



(a) : Snímek bez výrazných výpadků dat hloubkové mapy



(b) : Snímek s chybějícími hloubkovými daty

Obrázek 2.7: Výstupní data kamery Realsense, hloubková data byla převedena do odstínů šedé

- Použití hrubé síly (brute force). Postupně jsou vyzkoušeny všechny možnosti, popřípadě část z nich a vybere se nejlépe vyhovující. Využívá se zde zejména algoritmů jako je RANSAC, popřípadě jeho modifikace. Tato metoda je vhodná zejména pokud jsou data zatížena silným šumem, jelikož při správném počtu iterací téměř vždy vrátí správný výsledek, toto je však vykoupeno vysokou časovou náročností.[7, 4] Tato se dá snížit vhodným předzpracováním dat. Například segmentací na menší celky a určením plochy pouze pro tyto segmenty [14]
- Výpočet normálového vektoru v každém bodě, kde je normálový vektor kolmý na plochu reprezentující okolí bodu, toto je tvořeno body nacházejícími se v okruhu o poloměru r [16] popřípadě k nejbližšími body[8, 15]. Poté je provedena segmentace pomocí metody rostoucích oblastí viz. sekce 2.5.2. Výstupem jsou tedy skupiny bodů, které reprezentují jednotlivé plochy. V některých případech je ještě použit RANSAC na již nalezené segmenty, aby byly vyloučeny body, které leží mimo plochu přesto, že jejich normálový vektor dané ploše odpovídá [13].
- Jsou nalezeny body, které patří do dané roviny, těmito je poté pomocí PCA proložena rovina [19]. Tato metoda je náchylná na body ležící mimo rovinu (outliers), jelikož rovina je proložena body ve smyslu nejmenších čtverců a jediný bod ležící ve velké vzdálenosti od roviny může způsobit velké nepřesnosti.

2.5.2 Segmentace obrazu

Důležitým krokem pro porozumění obrazu a pro detekci objektů je segmentace obrazů, tj. rozdělení vstupních dat na skupiny. Rozdělujeme dva typy segmentace a to sémantickou, která sjednocuje body, které reprezentují danou skupinu objektů (například kostky ležící na zemi). Druhým typem je segmentace jednotlivých instancí, tedy skupina objektů je rozdělena na jednotlivé instance dané skupiny.

Přístup k segmentaci se liší podle dané aplikace. Nejpoužívanějšími metodami jsou tyto:

- Metoda prahování. Body $\mathbf{p}(x, y)$ jsou rozděleny do m skupin podle toho, zda jejich vlastnost V , což je například vzdálenost bodu od kamery nebo od plochy, dosahuje stanovené hodnoty T_m , tato můžeme být pevně určená, nebo se může dynamicky měnit a to v závislosti na okolních bodech, nebo v závislosti na pozici v obrazu, tj. $T(x, y)$. Pro prahovací metodu tedy platí následující vztah

$$\left\{ \begin{array}{l} \mathbf{p} \in m \text{ iff } V(\mathbf{p}) > T_m \\ \mathbf{p} \in n \text{ iff } V(\mathbf{p}) > T_n \\ \mathbf{p} \in \emptyset \text{ iff } V(\mathbf{p}) \leq T_1 \end{array} \right\} \quad (2.16)$$

- Metoda detekce hran. V obrazu jsou detekovány prudké změny pozorované vlastnosti V bodů obrazu. Pokud první derivace bodu \mathbf{p} dosáhne dané hodnoty je \mathbf{p} registrován jako hrana. Povýpočtu bývá provedena úprava těchto hran, kterou je například filtrování, popřípadě spojení některých sousedních bodů. Nakonec jsou v obrazu identifikovány segmenty bodů, které jsou od zbytku odděleny hranou. K výpočtu se využívá například Sobelova operátoru, který byl popsán v sekci 1.3 nebo Cannyho operátor.

- Metoda rostoucí oblasti. Tato se dělí na dva typy: S počátečním bodem (seed) a bez počátečního bodu.

Je zvolen jeden, popřípadě několik počátečního bodů. K těmto se postupně přidávají sousední body, pokud splňují předem definované podmínky. Těmito je například maximální odchylka normálových vektorů bodů od normálového vektoru seedu, nebo rozdíl velikost některé souřadnice. Metoda bez počátečního bodu pak rozdělí všechny body do skupin, jejichž body spolu sousedí a zároveň mají podobné vlastnosti. Jedná se o algoritmus, který pracuje s konstantní časovou náročností, která odpovídá $\mathcal{O}(2n)$, kde n je počet pixelů obrazu.

- Metoda rozdělování a slučování. Obraz je postupně rozdělen na části, které mají podobnou charakteristiku. Následně jsou sousední regiony, které jsou si podobné sloučeny do jednoho. Pro reprezentaci regionů se obvykle využívá quadtree.

Pokud o je obraz a V daná vlastnost jednotlivého bodu p a skupina bodů má vlastnost V , pokud tuto má každý bod skupiny pak lze metodu rozdělování a slučování popsat následovně.[10]

- Region R_1 je roven o
- Pokud platí $V(R_i) = False$, pak je region rezdělen na několik menších
- Pokud platí $V(R_i) = True$, pak je R_i sloučen se všemi sousedními regiony R_j , přičemž musí platit $V(R_i \cup R_j)$. Tento krok se opakuje dokud je možné některý region sloučit.

- Metoda shlukování (clustering). Obraz je rozdělen do clustrů, kde body v každém clustru mají podobné vlastnosti, resp. jejich vlastnosti jsou si v rámci obrazu nejbližší. Vlastnost v tomto případě bývá nejčastěji euklidovská vzdálenost. Mezi nejpopulárnější algoritmy patří tyto ¹
 - K-means algoritmus. V tomto algoritmu je předem nutné znát počet clustrů m , do kterých budou body rozděleny. Následně se náhodně vybere m bodů, které představují střed clustru. Následně je pro každý bod spočítána vzdálenost od jednotlivých středů. Bod je přiřazen do clustru, jehož vzdálenost středu je nejmenší. Následuje přepočítání středů a postup se opakuje. Ve chvíli kdy již nedochází ke změnám mezi jednotlivými clustery je zaznamenán součet rozptylů bodů v rámci jednotlivých clusterů a celý proces počínaje náhodnou volbou středů se n -krát opakuje. Nakonec je vybrán výsledek, který má nejmenší roptyl.
 - Mean-shift (posun průměru) algoritmus. Je předem definován poloměr kruhového okna r . Následně je náhodně určeno (popřípadě podle předem definované masky rozmístěno) n bodů \mathbf{p} . V každé iteraci se spočítá střed bodů \mathbf{t} , pro které platí $\|\mathbf{p} - \mathbf{t}\| < r$ a tento poradí bod \mathbf{p} , tento postup se opakuje, dokud body konvergují. Výsledná poloha bodů \mathbf{p}_i pak představuje střed jednotlivých clustrů.
 - DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algoritmus. Všechny body obrazu jsou označeny jako nenavštívené. Náhodně je vybrán nenavštívený bod \mathbf{a} a pokud se v jeho okolí o velikosti ϵ nachází minimální předem zvolený počet bodů, pak je celé okolí bodu \mathbf{p} přidáno do clusteru a postup se opakuje pro přidání bodů. Pokud již nelze žádný další bod přidat zvolí se další nenavštívený bod a proces se opakuje pro další cluster. Toto probíhá dokud existují nenavštívené body.

■ Detekce objektů

■ 2.5.3 Určování normálového vektoru

Ve většině scén počítačového vidění lze pozvat velké plochy, jakými jsou například stěny, či zem. Tyto jsou důležitým blokem pro většinu postupů detekce objektů. Například přesné určení směru podložky je nezbytné pro detekci objektů ležících na této podložce. Rozlišujeme dva časté postupy: Již známe body, které jsou součástí dané plochy, tj. již byla provedena segmentace obrazu, nebo

¹Princip algoritmů bude vysvětlen při shlukování podle euklidovské vzdálenosti

2.6 Výpočet normálového vektoru plochy nerganizovného mračna bodů

Normálový vektor plochy je definován, jako vektor kolmý na každý vektor dané plochy.

2.7 RANSAC

Výpočet normálového vektoru v bodě proběhne selekcí n bodů jejichž vzdálenost od

2.8 Zvolený postup

Bylo navrženo a implementováno několik postupů. Tyto se dělí do dvou kategorií.

- Segmentace obrazu na skupiny bodů odpovídající jednotlivým cihlám
- Detekce samotné cihly a její vizualizace pomocí ohraničujícího boxu

2.8.1 Segmentace obrazu

Přístup založený na výšce

Tento postup vychází z předpokladu, že na zemi se krom cihel nanachází jiné předměty. Scénu tedy tvoří pouze zem a na ní umístěné cihly, které mají předem známé rozměry. Začneme určením normálového vektoru roviny reprezentující zem, jelikož z jeho znalosti můžeme najít body ležící nad zemí, tj. body představující cihly. Vytvoříme mřížku bodů, které jsou v místech, kde předpokládáme přesný výstup kamery. U těchto ověříme, zda pro ně byla kameoru vygenerována hloubka, dále ověříme, zda bod leží v maximální vzdálenosti $4m$ jelikož podle [11] chyba stereo kamery RealSense D435 nad tuto vzdálenost začíná být markantní a správné určení normálového vektoru je zásadní pro všechny ostatní kroky. Množinu bodů $M = \mathbf{a}_1 \dots \mathbf{a}_m$ které splňují výše uvedená kritéria proložíme rovinou pomocí PCA algoritmu a dostaneme tedy normálový vektor \mathbf{n} plochy p minimalizující kvadrát vzdáleností od bodů tato je posunuta mimo počátek o d .

$$\bar{\mathbf{a}} = \frac{1}{m} (\mathbf{a}_1 \dots \mathbf{a}_m) \quad (2.17)$$

$$d = \mathbf{n} \cdot \bar{\mathbf{a}} = \mathbf{n}^T \bar{\mathbf{a}} \quad (2.18)$$

$$p : n_1x + n_2y + n_3z - d = 0 \quad (2.19)$$

Každý bod \mathbf{a} je buď součástí země, nebo součástí cihly. Je-li v množině M k bodů, které jsou součástí země G , pak zbylých $m - k$ bodů musí být součástí

mračna bodů reprezentující cihlu C . Tyto body se vždy nachází ve výšce z ² nad zemí a tedy platí, pro každý bod \mathbf{a}

$$n_1a_1 + n_2a_2 + n_3a_3 + \epsilon > d \rightarrow \mathbf{a} \in C \quad (2.20)$$

$$n_1a_1 + n_2a_2 + n_3a_3 + \epsilon \leq d \rightarrow \mathbf{a} \in G \quad (2.21)$$

Kde ϵ je experimentálně určená hodnota, která ošetří případy, kde $k = m$, tj. všechny body reprezentují zem. V tomto případě část bodů bude ležet nad plochou p . To jest dáno šumem dat a nepřesností kamery. Přidáním parametru ϵ je zajištěno, že každý bod reprezentující zem bude registrován i při zašuměných datech.

Po výběru bodů z mřížky podle výše uvedených kritérií dostaneme k bodů, které reprezentují zem a nacházejí se v místech, kde je přesnost kamery v rámci jejich možností nejvyšší. Tyto body by se daly znovu proložit plochou a dostat tak aproximaci natočení země pomocí normálového vektoru \mathbf{n} . Vzhledem k důležitosti přesnosti určení \mathbf{n} získáme další body reprezentující zem pomocí algoritmu 1. Tyto body opět pomocí PCA proložíme přímkou a dostaneme dobrou aproximaci normálového vektoru země. Na obrázku je vidět výsledek algoritmu 1.

Algorithm 1 Epandování bodů

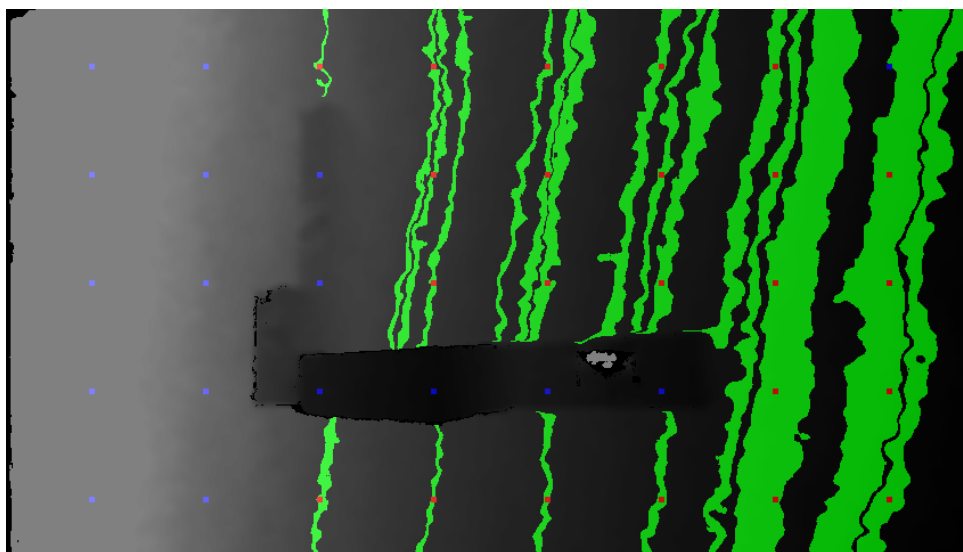
```

Stack ← ∅
G ← ∅
for  $\mathbf{a}$  in  $M$  do
    Stack.push( $(\mathbf{a})$ )
     $z \leftarrow$  height of  $\mathbf{a}$ 
    while not Stack.empty() do
         $\mathbf{t} \leftarrow$  Stack.pop()
         $G \leftarrow G \cup \mathbf{t}$ 
         $V \leftarrow V \cup p$ 
        for  $\mathbf{n}$  in neighbours of  $\mathbf{t}$  do
             $zn \leftarrow$  height of  $\mathbf{n}$ 
            if  $zn \in [z - \delta, z + \delta]$  and not  $\mathbf{n} \in G$  then
                stack.push( $\mathbf{n}$ )
            end if
        end for
    end while
end for
end for

```

Nyní se vypočítá výška každého bodu nad rovinou a podle této hodnoty se přidělí hodnota 0 až 3, kde číslo reprezentuje očekávaný počet na sobě naskládaných cihel. Vizualizace tohoto výsledku je vidět na obrázku 2.9

²Osa z směřuje směrem z kamery a tedy body nacházející se nad zemí mají menší hodnotu z



Obrázek 2.8: Červenou barvou jsou znázorněny body, ze kterých probíhal algoritmus 1, modré body byly podle výše uvedených kritérií vyřazeny a zeleně jsou znázorněny výsledné body reprezentující zem

■ Přístup založený na změně výšky

Velice populární postup při segmentaci obrazu je založena na normálovém vektoru, tento je vypočten pro každý bod obrazu a následně jsou body, jejichž normálový vektor má podobný směr, popřípadě i velikost, sloučeny do větších celků. V našem případě je vrchní stěna cihly paralelní se zemí a má tedy stejný normálový vektor. Nachází se však v jiné výšce. Pomočí výpočtu gradientu, který budeme realizovat pomocí Sobelova operátoru aplikovaného na výšku bodů, tedy spočítáme derivaci výšky, tato by měla být nejvyšší v oblasti přechodu země - cihla. Následně pak sloučíme body s hodnotou gradientu výšky, která si liší v rámci skupiny maximálně o δ . K tomu bylo využito upraveného CCL (Connected-component labeling) algoritmu.

Tato metoda při správném odladění parametru δ funguje velice přesně. Problémem je však robustnost. Hodnoty parametru δ fungující na cihly v blízkosti kamery, nedtekují cihly ve vzdálenosti řadově 2 m a více a naopak. Toto je způsobeno jednak šumem kamery, který podle rovnice 2.10 roste kvadraticky a zejména pak zkreslením vzdálenější hrany cihly kamerou, viz. obrázek 2.10. Kamera zde špatně nachází stereo páry a generuje mračna bodů, které ve skutečnosti neexistují. Tyto zmírňují přechod cihla-zem a snižují tak hodnotu gradientu. Toto chování je dobře vidět na obrázku 2.11.

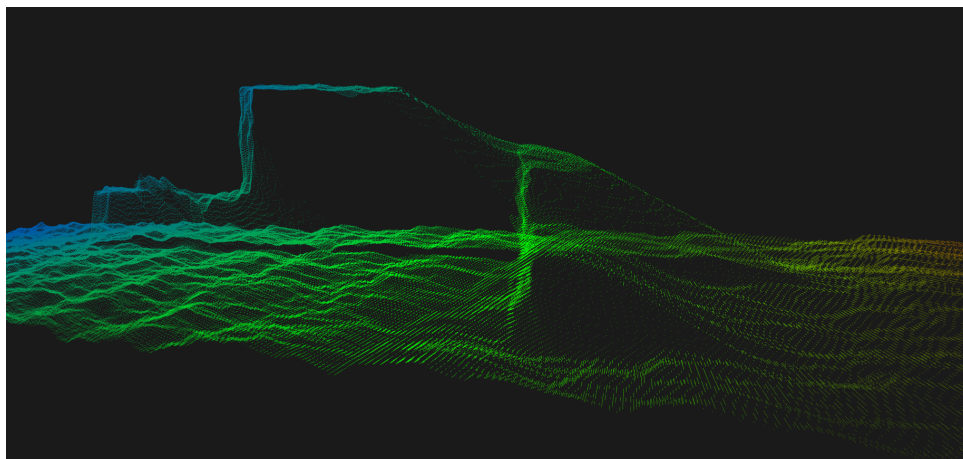
■ 2.8.2 Detekce objektů ze segmentovaného obrazu

Jako při segmentaci obrazu, tak i zde bylo vyzkoušeno několik algoritmů. Některé z nich byly nastaveny na detekci za zjednodušujících podmínek a nabízí tak nižší výpočetní náročnost při stejné přesnosti detekce. Vstupem všech prezentovaných algoritmů bude segmentovaný obraz jako je například

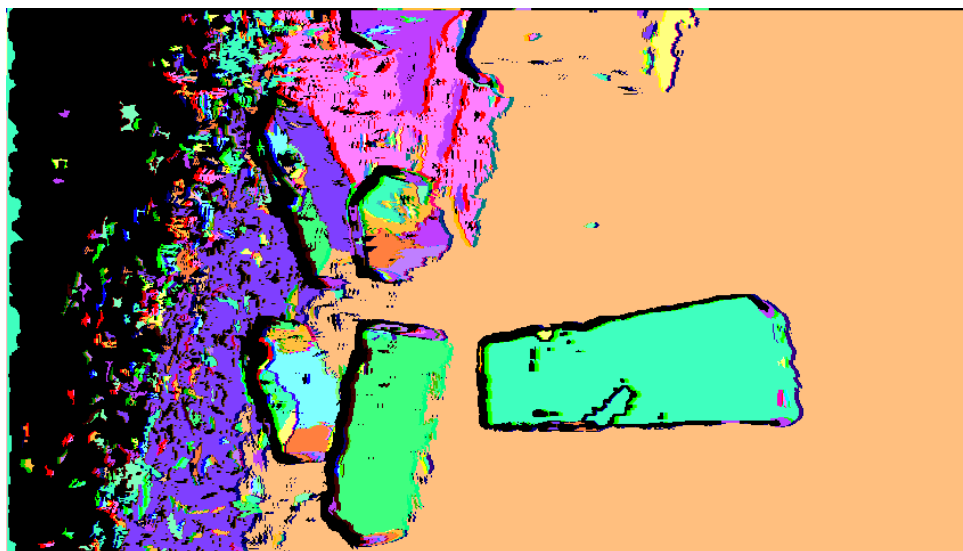


Obrázek 2.9: Vzdálenost boudů od roviny zobrazena jako počet na sobě lžících cihel

obrázek 2.9, pro tuto segmentaci jsme využili algoritmus 2.8.1.



Obrázek 2.10: Výstup kamery - mračno bodů zobrazující skupinu cihel. V pravé části je vidět zkreslení hrany způsobující problém při segmentaci



Obrázek 2.11: Výsledek segmentace obrazu podle velikosti derivace výšky.



Literatura

- [1] *The Intel® RealSense™ Depth Camera D435*.
- [2] Stereo and 3d vision.
- [3] *Intel® RealSense™ D400 Series Product Family*, 2019.
- [4] Panagiotis-Alexandros Bokaris, Damien Muselet, and Alain Trémeau. 3D reconstruction of indoor scenes using a single RGB-D image. In *12th International Conference on Computer Vision Theory and Applications (VISAPP 2017)*, Porto, Portugal, February 2017.
- [5] Myron Z Brown, Darius Burschka, and Gregory D Hager. Advances in computational stereo. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):993–1008, 2003.
- [6] Timothy M Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16(4):361–368, 1996.
- [7] Orazio Gallo, Roberto Manduchi, and Abbas Rafii. Cc-ransac: Fitting planes in the presence of multiple surfaces in range data. *Pattern Recognition Letters*, 32(3):403–410, 2011.
- [8] Dirk Holz, Stefan Holzer, Radu Bogdan Rusu, and Sven Behnke. Real-time plane segmentation using rgb-d cameras. In *Robot Soccer World Cup*, pages 306–317. Springer, 2011.
- [9] Walker James W. Angle Between Vectors, nov 2014.
- [10] Dilpreet Kaur and Yadwinder Kaur. Various image segmentation techniques: a review. *International Journal of Computer Science and Mobile Computing*, 3(5):809–814, 2014.
- [11] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–10, 2017.

- [12] Annika Kuhl. Comparison of stereo matching algorithms for mobile robots. *The University of Western Australia Faculty of Engineering, Computing and Mathematics*, 2005.
- [13] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE, 2011.
- [14] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–6. IEEE, 2009.
- [15] Alexander JB Trevor, Suat Gedikli, Radu B Rusu, and Henrik I Christensen. Efficient organized point cloud segmentation with connected components. *Semantic Perception Mapping and Exploration (SPME)*, 2013.
- [16] Jiefei Wang, Matthew Garratt, and Sreenatha Anavatti. Dominant plane detection using a rgb-d camera for autonomous navigation. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 456–460. IEEE, 2015.
- [17] Serge Weisstein Eric W., Belonie. Rodrigues’ rotation formula. Visited on 8/04/20.
- [18] Tomáš Werner. *Optimalizace*. České vysoké učení technické, 11.2.2020 edition, 2020.
- [19] Lizhi Zhang, Diansheng Chen, and Weihui Liu. Fast plane segmentation with line primitives for rgb-d sensor. *International Journal of Advanced Robotic Systems*, 13(6):1729881416665846, 2016.