

POLITECHNIKA WARSZAWSKA  
WYDZIAŁ MECHATRONIKI

Optyczne techniki skanowania 3D

Projekt nr 2:  
Segmentacja ścian budynków  
(Point cloud nr 4)

Wykonanie:  
Konrad Bedełek

20 maja 2022

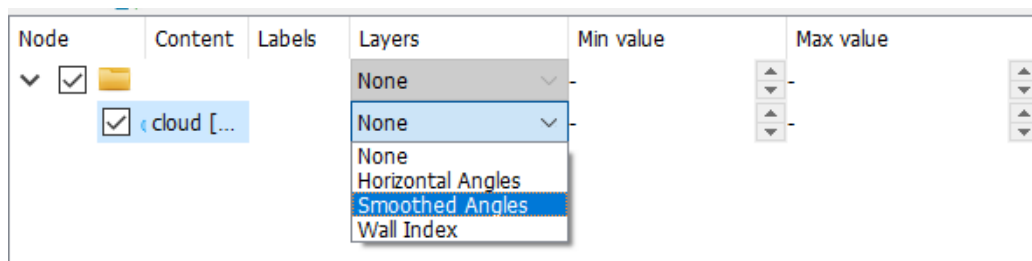
# Spis treści

1. Ogólny opis projektu . . . . .	2
2. Opis algorytmu . . . . .	4
3. Porównanie algorytmów . . . . .	6
4. Działanie programu . . . . .	7
5. Fajna tabela . . . . .	10
6. Podsumowanie . . . . .	11
Bibliografia . . . . .	12

# 1. Ogólny opis projektu

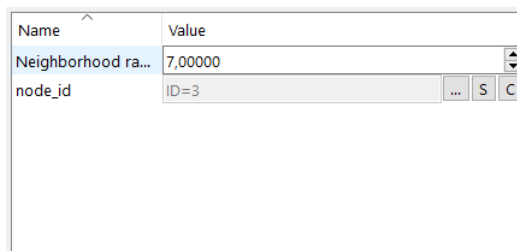
Projekt powstał przy użyciu platformy FRAMES, w całości w języku programowania C++.

Projekt dotyczy segmentacji ścian budynków w chmurze punktów 3D. Obliczone informacje zapisywane są do trzech warstw na wybranych etapach. Warstwa Horizontal Angles zawiera wartości kątów między wektorem normalnym najlepiej dopasowanej płaszczyzny a osią Z. Warstwa Smoothed Angles zawiera wygładzone wartości tych kątów. Ostatnia warstwa - Wall Index, zawiera odpowiednie indeksy wszystkich punktów. Indeksy te zostały podzielone na spełniające i niespełniające kryterium. Kryterium dotyczy wartości kątów - punkty o wartościach kątów powyżej ustalonego progu są kolejno numerowane, a te niespełniające kryterium otrzymują indeks 0. Dzięki temu, ustawiając wyświetlanie od wartości minimalnej 1, można dokładnie zobaczyć oddzielone obszary.[2]



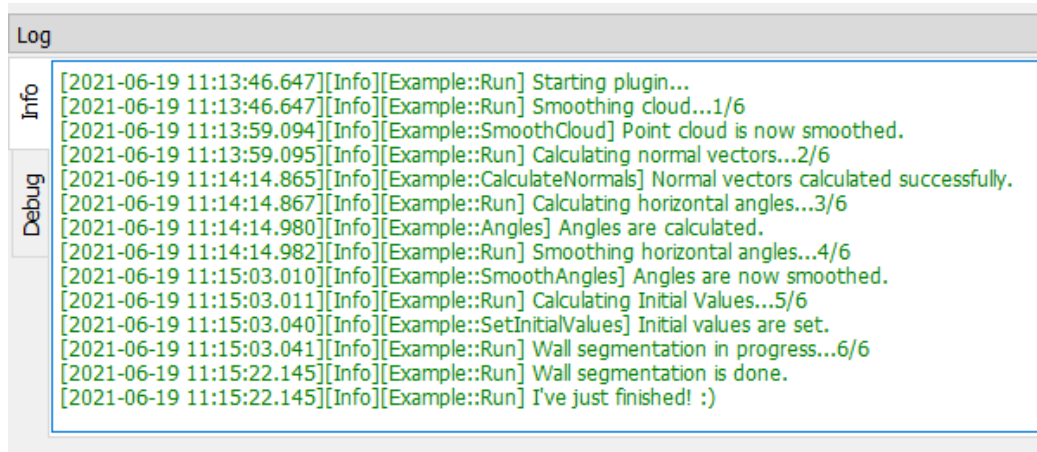
Rysunek 1.1. Warstwy

Jedyną wartością wprowadzaną do programu jest Neighbourhood radius, czyli promień sąsiedztwa wykorzystywany podczas segmentacji. Można jednak go nie wprowadzać, ponieważ domyślnie ustawiona jest wartość 7 i jest ona optymalna.



Rysunek 1.2. Dane wejściowe

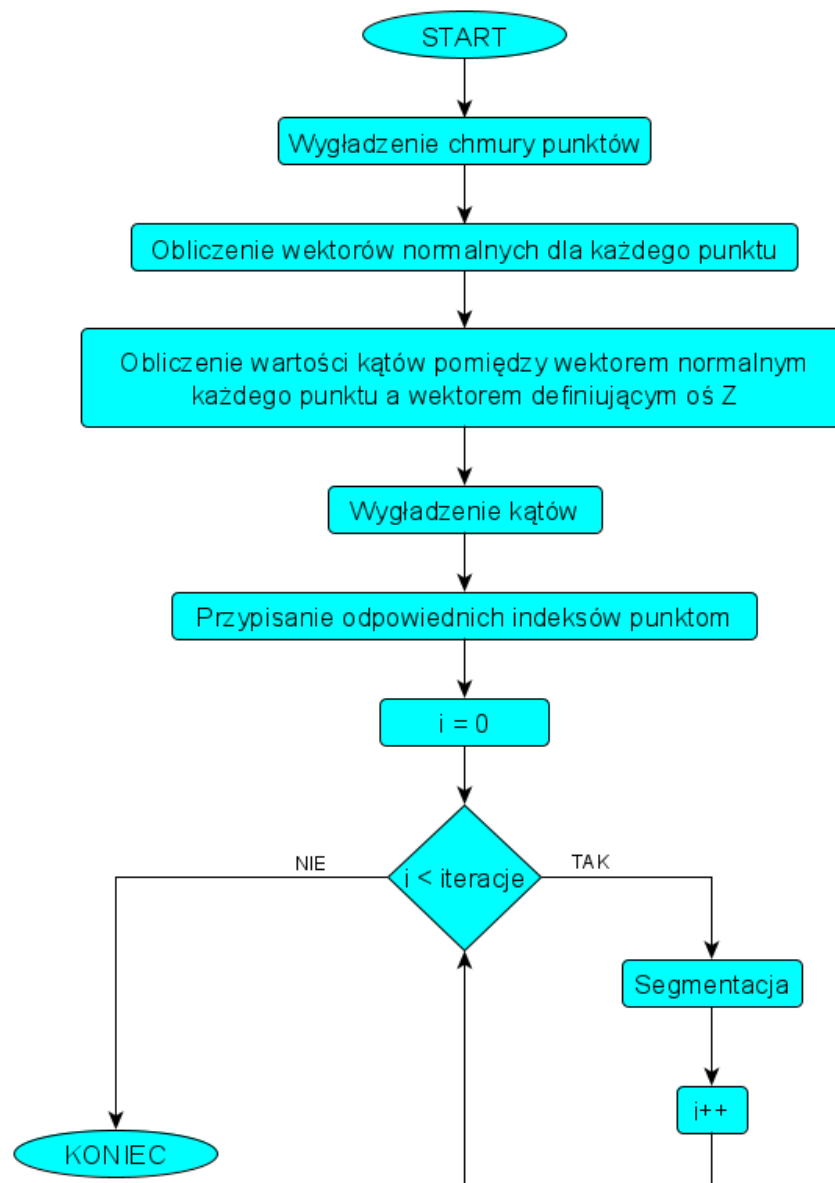
Program podczas wykonywania się, na bieżąco informuje użytkownika o postępie. Drukuje informację który algorytm z ilu jest aktualnie wykonywany, a gdy zostanie on wykonany, użytkownik zostaje o tym poinformowany. Dodatkowo dla każdego algorytmu powstał pasek postępu (progress bar).



Rysunek 1.3. Wyświetlanie informacji o postępie

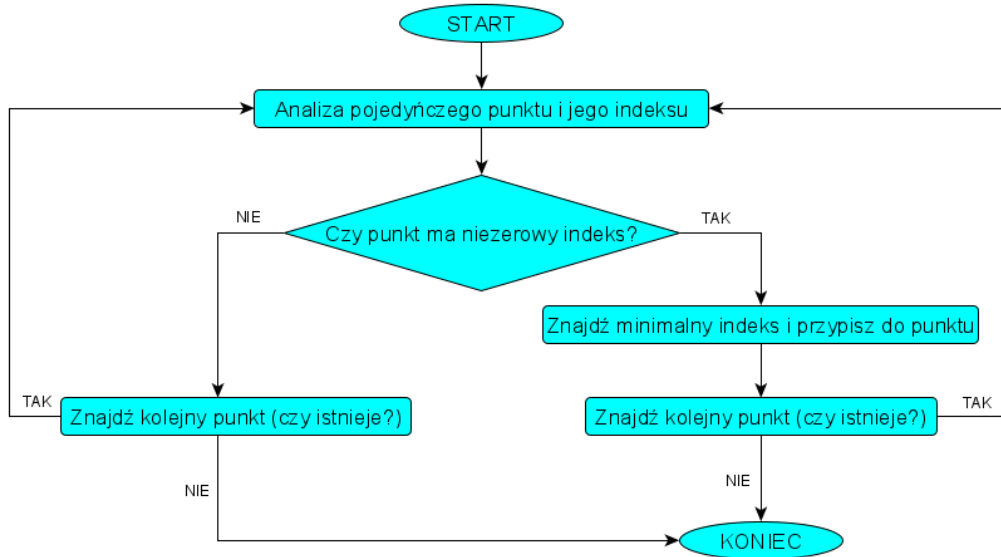
## 2. Opis algorytmu

Algorytm działania programu zakłada stworzenie funkcji odpowiedzialnych za różne etapy. Kolejno wykonywane są wygładzanie chmury, obliczanie wektorów normalnych, obliczanie i wygładzanie kątów, przypisywanie indeksów oraz segmentacja.



Rysunek 2.1. Algorytm działania programu

Algorytm segmentacji działa w oparciu o odległości Hausdorfa. Odległością jest tu promień sąsiedztwa, który definiuje, ile punktów wraz z ich indeksami ma być wziętych pod uwagę podczas ustalania indeksu pojedynczego punktu. Nie jest to jednak wierne odwzorowanie algorytmu odległości Hausdorfa[2].

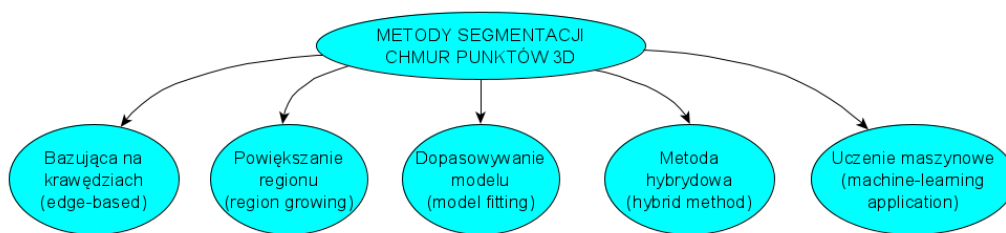


Rysunek 2.2. Algorytm segmentacji

### 3. Porównanie algorytmów

Jednym z algorytmów segmentacji chmur punktów 3D jest algorytm odległości Hausdorfa. Niniejszy projekt bazuje na nim, jednak zawiera pewne modyfikacje. Algorytm Hausdorfa grupuje punkty ze względu na ich odległości, oznaczając punkty jako obliczone i nieobliczone a także przypisując im odpowiednie numery grup.[3]

Warto jednak przedstawić ogólny podział metod segmentacji chmur punktów 3D. Przedstawiony został on na rysunku 3.1.



Rysunek 3.1. Metody segmentacji

**Metoda bazująca na krawędziach** - opiera się na wyodrębnieniu obszarów przy użyciu krawędzi i pogrupowaniu ich.

**Powiększanie regionu** - metoda ta może zostać zaimplementowana w wersji z góry na dół lub z dołu w górę. Może też być połączeniem obu tych podejść. Metoda ta opiera się na wyodrębnieniu punktów o określonych cechach i pogrupowaniu ich.

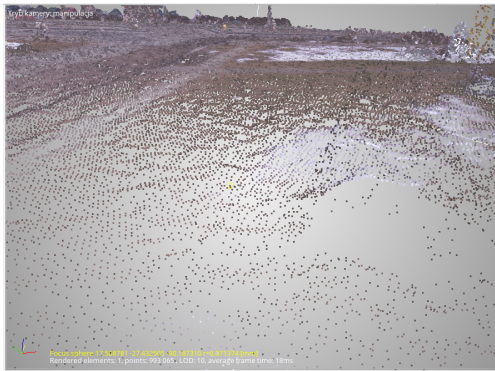
**Dopasowywanie modelu** - metoda opiera się na dopasowywaniu prymitywów.

**Metoda hybrydowa** - kombinacja dwóch lub więcej metod segmentacji.

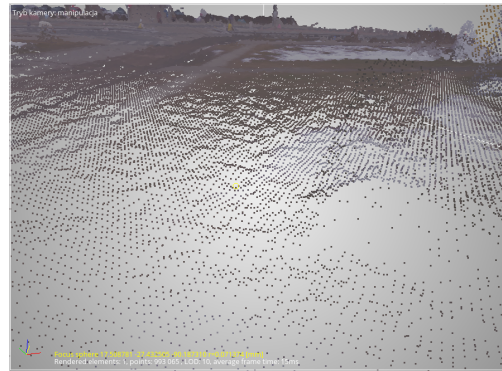
**Uczenie maszynowe** - wykorzystanie algorytmów sztucznej inteligencji, tak by komputer mógł nauczyć się odpowiednio segmentować punkty.

## 4. Działanie programu

Program rozpoczyna swoje działanie wygładzaniem chmury punktów. Jest to krok który ułatwia wykonanie dalszych algorytmów. Na rysunku 4.1 przedstawiono przybliżony widok chmury punktów przed wygładzaniem. Rysunek 4.2 przedstawia przybliżenie na część punktów po wygładzeniu całej chmury.



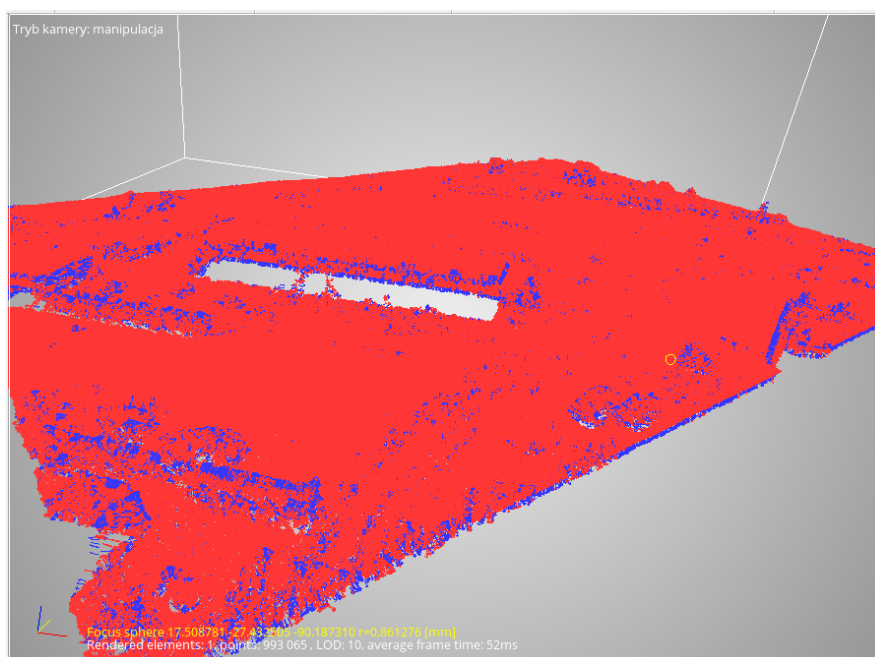
Rysunek 4.1. Chmura punktów przed wygładzeniem



Rysunek 4.2. Chmura punktów po wygładzeniu

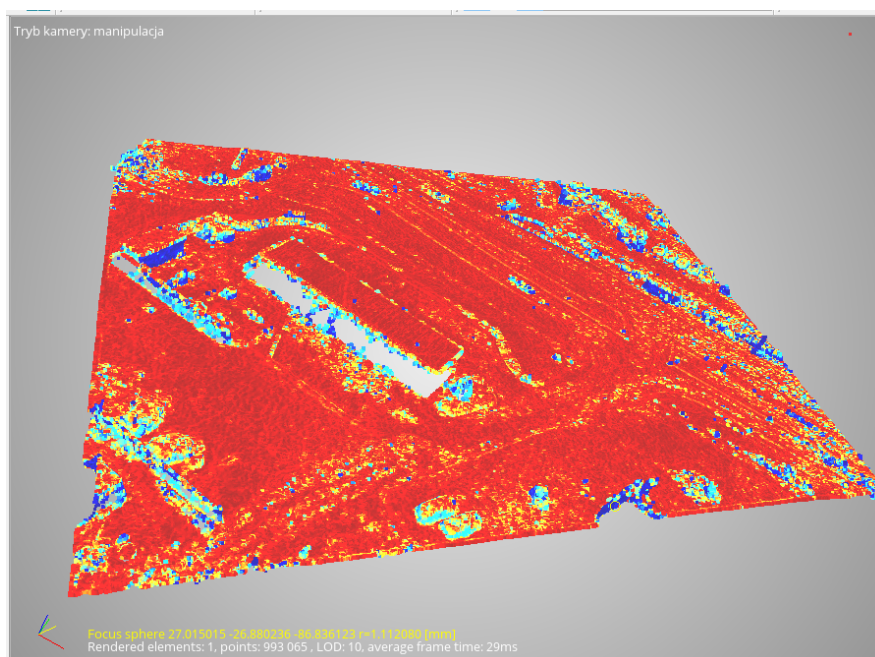
Kolejnym krokiem koniecznym do segmentacji jest obliczenie wektorów normalnych dla każdego punktu. Dzieje się to przy użyciu najlepiej dopasowanej płaszczyzny do rozważanego punktu oraz jego sąsiedztwa wyrażonego przez ilość sąsiadów.





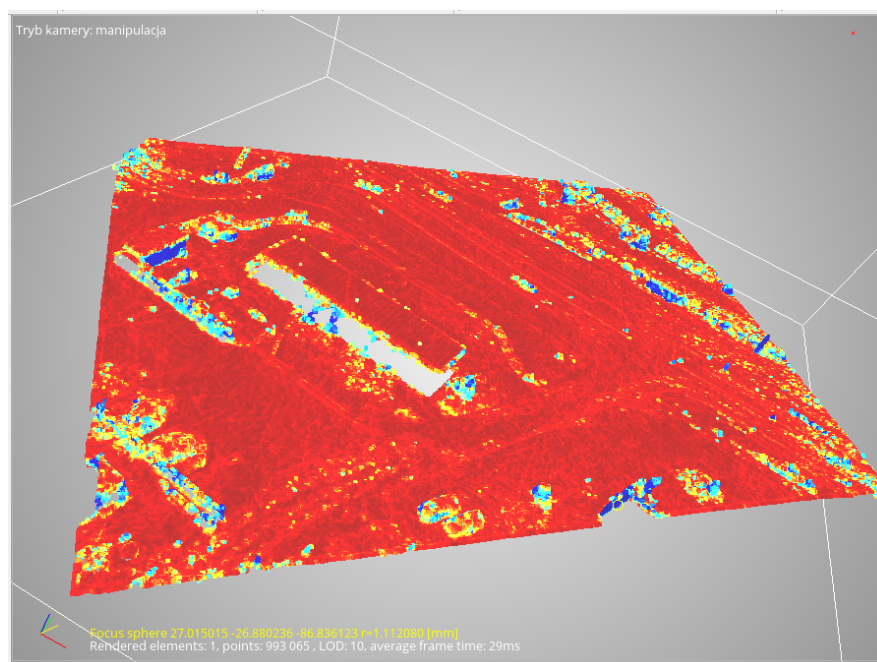
Rysunek 4.3. Wektory normalne

Na podstawie obliczonych wektorów normalnych, liczone są kąty między nimi a osią Z. Zapisywane są one do warstwy Horizontal Angles.



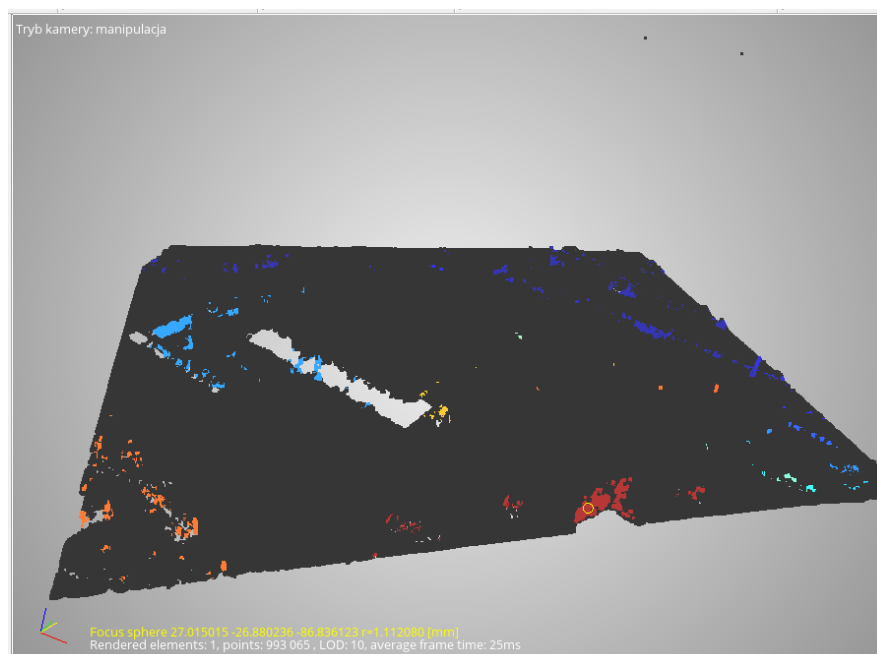
Rysunek 4.4. Kąty między wektorami normalnymi punktów a osią Z

Obliczone kąty wygładza się w celu ułatwienia segmentacji, analogicznie jak wcześniej była wygładzana chmura. Wygładzone wartości zapisywane są do warstwy Smoothed Angles.



Rysunek 4.5. Kąty między wektorami normalnymi punktów a osią Z - wygładzone

Ostatnim krokiem jest segmentacja, która wykonywana jest w pętli for określoną ilość razy. Eksperymentalnie dobrana została wartość 10 iteracji. Daje ona zadowalające rezultaty i pozwala oszczędzić czas.



Rysunek 4.6. Ostateczny rezultat po segmentacji

## 5. Fajna tabela

Na poniższej tabeli przedstawiono skład niebezpiecznych składników domestosa, ponieważ tabela musi być. Bibliografia w sumie też[1].

Nazwa składnika	Zawartość	Nr CAS
Podchloryn sodu	1 - 5 %	7681-52-9
Alkilodimetyloamina	1 - 5 %	68955-55-5
Wodorotlenek sodu	<1 %	1310-73-2

Tabela 5.1. Skład domestosa

## 6. Podsumowanie

Zaproponowane rozwiązanie miało na celu segmentację ścian budynków. Posiada ono zarówno zalety jak i wady.

### **Zalety rozwiązania:**

- Poza całymi ścianami wykrywa także fragmenty ścian. W rozważanej chmurze punktów wystąpił problem z zeskanowaniem ścian. Znaczna ich część była zeskanowana fragmentarycznie, jednak nawet te niewielkie fragmenty są uwzględniane.
- Rozwiązanie jest proste i przejrzyste.

### **Wady rozwiązania:**

- Poza ścianami budynków wykrywa część podobnych obiektów. Wykrywane zostają niektóre ściany samochodów lub obiekty podobne do ścian. Dzieje się tak przez progi ustawione w kodzie. Zakres kątów jest duży, jednak było to bardzo pomocne przy segmentacji źle zeskanowanych ścian.
- Rozwiązanie jest stosunkowo wolne. Wynikać to może częściowo z samego faktu, iż chmura punktów 3D jest wymagająca obliczeniowo dużo bardziej niż np. obraz 2D.

# Bibliografia

- [1] Xian-Feng Han, Jesse S. Jin, Ming-Jie Wang, Wei Jiang, Lei Gao, and Liping Xiao. A review of algorithms for filtering the 3d point cloud. *Signal Processing: Image Communication*, 57:103–112, 2017.
- [2] François Pomerleau, Francis Colas, and Roland Siegwart. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015.
- [3] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2021.