



Imię i nazwisko studenta: Mykyta Brazhynskyy

Nr albumu: 172017

Studia drugiego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Automatyka i Robotyka

Profil: Systemy decyzyjne i robotyka

## **PRACA DYPLOMOWA MAGISTERSKA**

Tytuł pracy w języku polskim: Skaner 3d na bazie kamery RGBD

Tytuł pracy w języku angielskim: 3D Scanner based on RGBD camera

| Potwierdzenie przyjęcia pracy            |   |
|--|---|
| Opiekun pracy                            | Kierownik Katedry/Zakładu (pozostawić właściwe) |
| <i>podpis</i><br>dr inż. Michał Czubenko | <i>podpis</i>                                   |

Data oddania pracy do dziekanatu:



## **OŚWIADCZENIE dotyczące projektu dyplomowego zatytułowanego: Skaner 3d na bazie kamery RGBD**

Imię i nazwisko studenta: Mykyta Brazhynskyy

Data i miejsce urodzenia: 18.04.1998, Ługańsk

Nr albumu: 172017

Wydział: Wydział Elektroniki, Telekomunikacji i Informatyki

Kierunek: automatyka i robotyka

Poziom kształcenia: pierwszy

Forma studiów: stacjonarne

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. 2018 poz. 1191 z późn. zm.) i konsekwencji dyscyplinarnych określonych w ustawie z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. 2018 poz. 1668 z późn. zm.),<sup>1</sup> a także odpowiedzialności cywilnoprawnej oświadczam, że przedkładany projekt dyplomowy został opracowany przeze mnie samodzielnie.

Niniejszy projekt dyplomowy nie był wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem tytułu zawodowego.

Wszystkie informacje umieszczone w ww. projekcie dyplomowym, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami zgodnie z art. 34 ustawy o prawie autorskim i prawach pokrewnych.

Potwierdzam zgodność niniejszej wersji projektu dyplomowego z załączoną wersją elektroniczną

Gdańsk, dnia .....

.....

*podpis studenta*

<sup>1</sup> Ustawa z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce:

Art. 312. ust. 3. W przypadku podejrzenia popełnienia przez studenta czynu, o którym mowa w art. 287 ust. 2 pkt 1–5, rektor niezwłocznie poleca przeprowadzenie postępowania wyjaśniającego.

Art. 312. ust. 4. Jeżeli w wyniku postępowania wyjaśniającego zebrany materiał potwierdza popełnienie czynu, o którym mowa w ust. 5, rektor wstrzymuje postępowanie o nadanie tytułu zawodowego do czasu wydania orzeczenia przez komisję dyscyplinarną oraz składa zawiadomienie o podejrzeniu popełnienia przestępstwa.

## **STRESZCZENIE**

Celem niniejszej pracy dyplomowej było stworzenie skanera 3D oraz systemu wizualizacji utworzonych modeli rzeczywistych obiektów. Do budowy urządzenia wykorzystano kamerę głębi firmy Intel o nazwie RealSense D435i. W pracy został przedstawiony sposób budowy skanera 3D, jego kalibracji oraz algorytmy służące do przetwarzania otrzymanych danych pomiarowych w celu uzyskania wirtualnych modeli. W celu łatwiejszej obsługi programu został utworzony interfejs graficzny zawierający najważniejsze parametry wizualizacji i obróbki danych. Na koniec dane są eksportowane do modeli w formacie obsługiwany przez program Blender.

**Słowa kluczowe:** Skaner 3D ,Intel RealSense, Python, Kamera RGBD

**Dziedzina nauki i techniki, zgodnie z wymogami OECD:** Nauki inżynierijne i techniczne, Systemy automatyzacji i kontroli

## ***ABSTRACT***

The aim of this thesis was to create a 3D scanner and a system for visualization of created models based on real objects. In the work is presented how to build a 3D scanner, its calibration and algorithms used to process the obtained measurement data to obtain virtual models. In order to make the program easier to use, a graphic interface was created containing the most important parameters of visualization and data processing. Finally, the data are exported to the models in a format supported by the Blender program.

**Keywords:** 3D Scanner, Intel RealSense, Python, RGBD Camera

# **Spis treści**

|   |           |
|---|-----------|
| <b>1 Wprowadzenie</b>   | <b>8</b>  |
| 1.1 Cele i założenia . . . . .  | 8         |
| 1.2 Zawartość pracy . . . . .   | 8         |
| <b>2 Technologia skanerów 3D</b>  | <b>9</b>  |
| 2.1 Metoda triangulacji laserowej . . . . .                             | 9         |
| 2.2 Metoda emitowania światła strukturalnego . . . . .                  | 11        |
| 2.3 Fotogrametria . . . . .   | 13        |
| 2.4 Skanery impulsowe LIDAR . . . . .                                   | 13        |
| 2.4.1 Time of flight LIDAR . . . . .                                    | 13        |
| 2.4.2 Phase-Shift LIDAR . . . . .                                       | 13        |
| 2.5 Ogólne charakterystyki laserowych metod pomiarowych . . . . .       | 14        |
| 2.6 Zastosowania skanerów 3D . . . . .                                  | 14        |
| <b>3 Model skanera oraz koncepcja działania</b>                         | <b>16</b> |
| 3.1 Model i konstrukcja skanera 3D . . . . .                            | 16        |
| 3.2 Przejście do chmury punktów . . . . .                               | 18        |
| 3.2.1 Wstępna obróbka danych. . . . .                                   | 18        |
| 3.2.2 Przekształcenie danych z kamery RGBD do współrzędnych 3D. . . . . | 18        |
| 3.2.3 Normalizacja otrzymanych punktów. . . . .                         | 18        |
| 3.2.4 Interpolacja punktów . . . . .                                    | 19        |
| 3.2.5 Metoda światła strukturalnego . . . . .                           | 20        |
| 3.3 Rekonstrukcja powierzchni . . . . .                                 | 22        |
| 3.3.1 Trójwymiarowa triangulacja Delaunaya . . . . .                    | 22        |
| 3.3.2 Algorytm maszerujących sześciąnów . . . . .                       | 22        |
| 3.3.3 Ball pivoting algorithm . . . . .                                 | 23        |
| 3.3.4 Podsumowanie . . . . .  | 24        |
| <b>4 Implementacja oraz porównanie metod</b>                            | <b>25</b> |
| 4.1 Ogólny zarys funkcjonowania programu . . . . .                      | 25        |
| 4.1.1 Odczytanie danych z kamery RGBD . . . . .                         | 25        |
| 4.1.2 Przejście do chmury punktów . . . . .                             | 26        |
| 4.1.3 Oznaczanie błędnych próbek . . . . .                              | 27        |
| 4.1.4 Interpolacja punktów . . . . .                                    | 27        |
| 4.2 Porównanie metod rekonstrukcji powierzchni . . . . .                | 27        |
| 4.2.1 Algorytm BPA . . . . .  | 28        |
| 4.2.2 Triangulacja Delaunay'a . . . . .                                 | 29        |
| 4.3 Wyniki . . . . .  | 34        |
| 4.3.1 Wyniki metody światła strukturalnego . . . . .                    | 34        |
| 4.3.2 Wyniki metody skanera liniowego . . . . .                         | 36        |
| 4.3.3 Algorytm Delaunay . . . . .                                       | 38        |
| <b>5 Zakończenie</b>  | <b>41</b> |
| 5.1 Porównanie wyników przeprowadzonych badań . . . . .                 | 41        |
| 5.2 Błędy oraz możliwości poprawy . . . . .                             | 43        |
| <b>Spis tabel</b>   | <b>45</b> |

|                      |           |
|----------------------|-----------|
| <b>Spis rysunków</b> | <b>45</b> |
| <b>Bibliografia</b>  | <b>46</b> |

## **WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW**

### *RGB*

Paleta barw tworząca kolor piksela. Oznaczenie pochodzi od kolorów Red, Green, Blue czyli czerwony, zielony, niebieski.

### *Kamera RGBD*

Kamera głębi, oprócz wykonywania zdjęć RGB potrafi ona również dokonać pomiaru odległości od obiektów i nanieść te informację na powierzchnię poszczególnych pikseli obrazu.

### *LIDAR*

Ang.(Light Detection And Ranging) urządzenie służące do dokładnego pomiaru odległości. Działaniem przypomina funkcjonowanie radaru, lecz korzysta z odliczania czasu przelotu światła lasera, a nie mikrofal.

### *Blender*

Oprogramowanie służące do modelowania trójwymiarowego. Posiada szereg funkcji do animacji obiektów, generacji tekstur oraz importowania i eksportowania gotowych modeli.

### *Maya*

Program komputerowy, umożliwiający generację zaawansowanych modeli 3D przeznaczony do zastosowań przemysłowych. W tym programie zostały stworzone filmy takie jak Spiderman, Avatar oraz Up.

### *LIDAR*

Skaner impulsowy LIDAR jest urządzeniem mierzącym odległość za pomocą wiązki światła. Jego zasada działania jest zbliżona do funkcjonowania radaru. MS,AMS

# **1. WPROWADZENIE**

Kluczowym czynnikiem który wpłynął na rozwój technologii skanerów trójwymiarowych jest druk 3D. Jego historia rozpoczyna się w 1984 kiedy to Charles Hull zaprojektował oraz stworzył urządzenie korzystające z procesu zwany stereolitografią [1]. Jest to proces polegający na utwardzaniu materiału za pomocą lasera. Światło padając na powierzchnię płynnego fotopolimeru tworzy jego utwardzoną warstwę. Następnie platforma jest obniżana, a proces jest powtarzany aż do momentu utworzenia całego obiektu. Korzystając z druku 3D jest możliwe szybkie utworzenie prototypu z plastiku. Pozwala to na sprawdzenie na przykład właściwości aerodynamicznych modelu 3D bez konieczności wytwarzania go z docelowych materiałów, co znacząco zmniejsza koszty projektowania takich obiektów [2]. Do utworzenia komputerowych modeli 3D korzysta się z różnych metod projektowania. Jedną z nich jest trójwymiarowy skan rzeczywistego obiektu, a następnie przeniesienie go do wirtualnego systemu komputerowego. Istnieje wiele rozwiązań na rynku pozwalających na utworzenie precyzyjnych trójwymiarowych modeli. Jednym z takich programów jest Blender, otwarte oprogramowanie dzięki któremu użytkownik jest w stanie utworzyć elementy 3D jak również je animować. Poprzez wykorzystanie tego narzędzia użytkownicy są w stanie nawet wykonywać filmy animowane [3].

Drugim programem jest Maya. Jest to o wiele bardziej zaawansowane oprogramowanie do modelowania 3D. Jest ono wykorzystywane w branży filmowej. Niestety większość ogólnodostępnych programów jest płatna oraz ma mało możliwości konfiguracji poszczególnych parametrów związanych z rekonstrukcją kształtu obiektów. W niniejszej pracy zostanie zaprezentowany sposób na wierne utworzenie obiektu 3D ze skanów zrobionych za pomocą kamery Intel RealSense D435i.

## **1.1. Cele i założenia**

Celem niniejszej pracy jest zaprojektowanie skanera 3D korzystającego z metody triangulacji laserowej oraz wyeksportowanie modeli do programu Blender przy zastosowaniu kamery RGBD. Projekt składa się z dwóch części, budowy skanera trójwymiarowego oraz stworzenia programu do obróbki otrzymanych danych. Odległość z jakiej będzie wykonywany skan wynosi do 1 m. Powyżej tej wartości gęstość punktów będzie zbyt niska do wiernego odtworzenia modelu. Czas trwania obliczeń w programie wynosi poniżej 15 minut. Utworzony model powinien jak najwierniej oddawać wygląd rzeczywistego obiektu, luki w tekstuze powinny nieznacznie wpływać na ostateczny wygląd.

## **1.2. Zawartość pracy**

W pracy wykonano zestawienie istniejących metod mających na celu generację trójwymiarowych obiektów na podstawie danych z kamery głębi. Dokonane zostało porównanie pomiędzy dostępnymi na rynku skanerami 3D bazującymi na różnych technologiach pomiarowych na podstawie ich parametrów technicznych. Przeanalizowano w jakich warunkach dana metoda pomiarowa powinna zostać wykorzystana, a także dokonano przeglądu technologii, którymi posługiwano się w przeszłości do generacji trójwymiarowych modeli. Ostatecznie przedstawione zostały zastosowania współczesnych skanerów 3D.

W dalszej części pracy przedstawiony jest autorski model oraz konstrukcja skanera 3D. Omówione zostały metody służące do przetworzenia danych uzyskanych z kamery głębi w chmurę punktów. Następnie przedstawiono koncepcje istniejących rozwiązań służących do rekonstrukcji powierzchni oraz kształtu obiektów z chmury punktów.

Następny rozdział przedstawia metody analizy oraz obróbki danych, które mają posłużyć do cyfrowej implementacji rzeczywistych obiektów zarejestrowanych przez kamerę RGBD. Przedstawiono opisy zastosowanych algorytmów oraz kolejność ich wykonywania na podstawie autorskiego programu w języku Python. Skupiono się również na różnych metodach optymalizacji algorytmów. Poddano analizie pod względem dokładności rezultaty pomiarów w porównaniu do rzeczywistych wartości mierzonych.

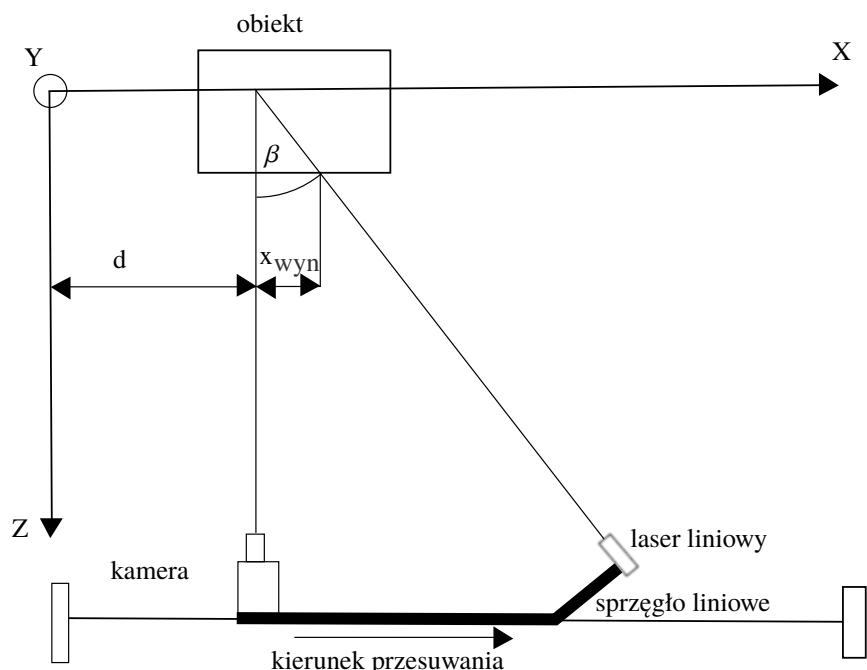
Ostatnia część zajmuje się podsumowaniem wykonanej pracy oraz otrzymanych wyników. Omówione są w nim możliwości ewentualnego udoskonalenia urządzenia. Przytoczone zostały również wady istniejącej metody obróbki danych oraz generacji modeli. Całość zostaje zamknięta porównaniem autorskiej metody tworzenia obiektów trójwymiarowych oraz programów dostępnych na rynku.

## 2. TECHNOLOGIA SKANERÓW 3D

Skanner jest urządzeniem służącym do rejestracji oraz konwersji do postaci cyfrowej płaskich obrazów na przykład dokumentów lub zdjęć. Jednakże zwykły skaner nie umożliwia rejestracji trójwymiarowych obiektów. Do tego zadania używany jest skaner 3D. Początki skanerów 3D datuje się na lata 60 XX wieku. Do uzyskania modeli używano wówczas oświetlenia oraz kamery, które rejestrowały odkształcenia cienia mierzzonego obiektu [4]. Niestety, wymagania dotyczące nakładu pracy nie były proporcjonalne do otrzymanych wyników. W połowie lat 80-tych komputery zyskały na popularności, a narzędzia pomiarowe stały się dokładniejsze w efekcie czego postanowiono użyć sondy stykowej. Mierzono odkształcenie sondy po zetknięciu się z obiektem wskutek czego można było wyznaczyć położenie punktów na płaszczyźnie obiektu w innym układzie współrzędnych [5]. Jej użycie pozwoliło na znaczne zwiększenie dokładności pomiarów, jednak prędkość ich wykonywania była powolna. Wobec tego zaistniała zauważalna potrzeba opracowania metody optycznej, która umożliwiła mierzenie obiektów z większą prędkością. Miałoby to na celu również pomiar elastycznych przedmiotów, które dotychczas nie były mierzalne ze względu na użyte technologie. Istnieje wiele różnych podejść do trójwymiarowych skanerów, każde z nich ma zastosowanie w określonej dziedzinie. Poniżej przedstawiono najważniejsze z nich.

### 2.1. Metoda triangulacji laserowej

Wśród urządzeń korzystających z metody triangulacji laserowej można wyróżnić dwa podstawowe typy. W pierwszym typie takiego skanera układ pomiarowy składa się z nadajnika laserowego, obiektu pomiarowego oraz kamery. Poprzez przesuwanie wiązki po nieruchomym obiekcie, znając położenie kamery oraz korzystając z odpowiednich wzorów można wyznaczyć położenie zmierzonych punktów w docelowym układzie współrzędnych [6]. Zaletami tej konstrukcji są dokładność i wysoka rozdzielcość [7]. Skanery te jednak nie sprawdzają się przy rekonstrukcji błyszczących oraz przezroczystych powierzchni. Zasada działania skanera została przedstawiona na rysunku 2.1.



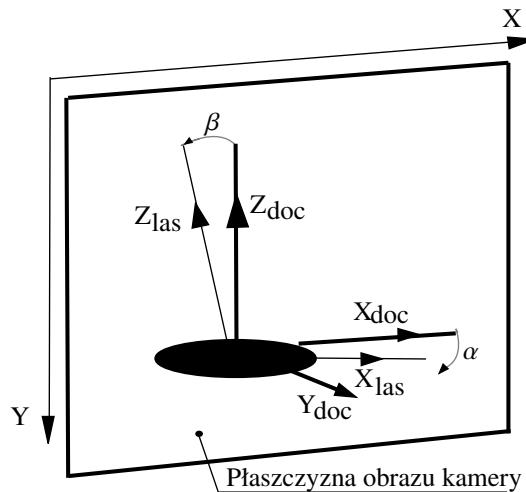
Rysunek 2.1: Schemat skanera wykorzystującego triangulację laserową [6].

Posługując się poniższymi wzorami można uzyskać współrzędne przestrzenne mierzonego obiektu, co w rezultacie pozwala na jego odwzorowanie w komputerowej symulacji:

$$\left\{ \begin{array}{l} X = d + x_{wyn} \\ Y = k_y \cdot y_{pic} \\ Z = ctg(\beta) \cdot x_{wyn} \\ x_{wyn} = k_x \cdot x_{pic} \end{array} \right. \quad (2.1)$$

Zmierzony obiekt posiada współrzędne X, Y, Z. Kamera jest oddalona od podstawy ramienia uchwytu o odległość  $d$ . Wynik pomiaru w kierunku osi X w docelowych jednostkach miary został oznaczony poprzez  $x_{wyn}$ . Linia lasera tworzy z płaszczyzną obrazu kąt  $\beta$ . Współrzędne linii lasera odczytane przez kamerę oznaczone są jako  $x_{pic}, y_{pic}$ . W celu przejścia na docelowe jednostki miary należy użyć odpowiednich skali przejścia dla obu współrzędnych X oraz Y, odpowiednio  $k_x$  i  $k_y$ .

Drugi typ skanerów opartych o metodę triangulacji laserowej steruje położeniem obiektu względem lasera. Korzystając z tego rozwiązania, możliwe jest odwzorowanie bardziej skomplikowanych kształtów. W skład elementów konstrukcyjnych takiego układu wchodzi laser, kamera oraz tacka obrotowa z umieszczonym na niej obiektem pomiarowym. W przeciwieństwie do skanera pierwszego typu, w tym układzie kamera oraz laser są nieruchome. Porusza się jedynie tacka obrotowa z obiektem. Laser wyświetla pionową linię na obiekt, a kamera rejestruje ten obraz. Tacka obracana jest o stały kąt  $\alpha$ . Po zmierzeniu kąta nachylenia pomiędzy kamerą, a laserem jak również kąta nachylenia kamery względem taki, można dokonać transformacji ze współrzędnych obiektu do współrzędnych obrazu przechodzących przez osią obrotu podstawy. Schemat działania takiego urządzenia został przedstawiony na rysunku 2.2.



Rysunek 2.2: Układ współrzędnych nieruchomego skanera [6].

Współrzędne zmierzonych obiektu w układzie kamery można opisać poniższymi równaniami. Przedstawiają one sposób na przejście ze współrzędnych obiektu do współrzędnych kamery.

$$\left\{ \begin{array}{l} Z_{doc} = y_0 - (y - (x_{las} - x_0)) \cdot \tan(\alpha) \\ X_{doc} = x - x_0 + (y_{las} - y_0) \cdot \tan(\beta) \\ Y_{doc} = 0 \end{array} \right. \quad (2.2)$$

Wartości X oraz Y są współrzędnymi obrazu kamery w pikselach.  $X_{las}, Z_{las}$  są współrzędnymi układu linii emitowanej przez laser. Kąt obrotu taki symbolizuje wartość  $\alpha$ . Układ docelowy nachylony jest do płaszczyzny obrazu pod kątem  $\beta$ . Środek taki obrotowej przedstawionej na obrazie kamery ma współrzędne  $x_0, y_0$  w pikselach. Docelowy układ współrzędnych określają współrzędne  $X_{doc}, Y_{doc}, Z_{doc}$ .

Ostatecznie korzystając z macierzy obrotu względem osi  $Z_{doc}$  można uzyskać współrzędne chmury punktów.

$$\begin{cases} X_{pkt} = X_{doc} \cdot \cos(\alpha) - Y_{doc} \cdot \sin(\alpha) \\ Y_{pkt} = X_{doc} \cdot \sin(\alpha) + Y_{doc} \cdot \cos(\alpha) \\ Z_{pkt} = Z_{doc} \end{cases} \quad (2.3)$$

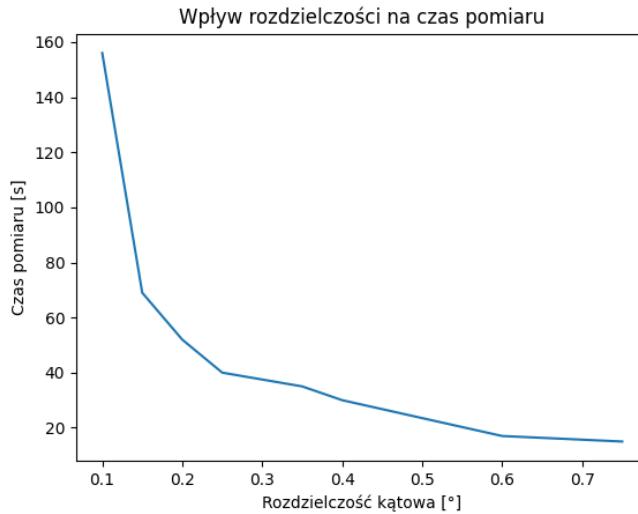
gdzie:  $\alpha$  przedstawia kąt obrotu tacki,  $X_{doc}, Y_{doc}, Z_{doc}$  są współrzędnymi układu kamery, natomiast  $X_{pkt}, Y_{pkt}, Z_{pkt}$  określają współrzędne w układzie chmury punktów względem środka tacki.

Na rynku istnieje wiele skanerów trójwymiarowych wykorzystujących metodę triangulacji laserowej. Poniżej zostaną omówione charakterystyki jednego z nich. Charakterystyki jakości pomiaru skanera Faro Focus zostały przedstawione w tabeli 2.1.

**Tabela 2.1:** Charakterystyki skanera Faro Focus S350 [8].

|                         |                           |
|-------------------------|---------------------------|
| Zasięg                  | 0.6 m - 350 m             |
| Błąd pomiaru odległości | $\pm 1$ mm                |
| Prędkość pomiaru        | do 976000 $\frac{pkt}{s}$ |
| Dokładność kątowa       | do 19 sekund kątowych     |
| Dokładność pozycji 3D   | 3.5 mm                    |
| Długość fali lasera     | 1550 nm                   |

Metoda triangulacji laserowej umożliwia mierzenie dużych obiektów z dalekiej odległości. Dlatego też znajduje ona zastosowanie na przykład w budownictwie. Dzięki niej inżynierzy pracujący przy konstrukcjach są w stanie skanować pełnowymiarowe budynki, a następnie dokonywać ich oględzin w postaci cyfrowej. Niestety jednak pomiar o wysokiej rozdzielczości zajmuje znacznie więcej czasu niż przy wykorzystaniu innych metod pomiarowych. Jest to spowodowane zależnością, iż czas pomiaru obiektu jest proporcjonalny do rozdzielczości kątowej skanera. To znaczy, że wraz ze wzrostem rozdzielczości kątowej rośnie również czas pomiaru, co zostało zobrazowane na rysunku 2.3.

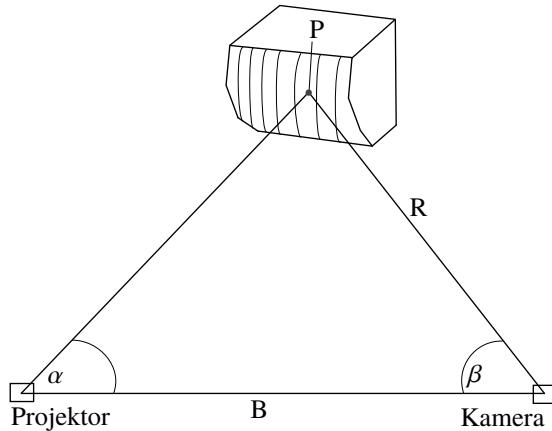


**Rysunek 2.3:** Czas pomiaru w zależności od dokładności pomiaru [9].

## 2.2. Metoda emitowania światła strukturalnego

Skanery oparte o metodę emitowania światła strukturalnego wyświetlają siatkę świetlną na mierzony obiekt. Następnie wzór na obiekcie jest mierzony przez jedną lub dwie kamery. Za sprawą badania mocy światła odbitego oraz kształtu siatki na przedmiocie, korzystając z metod triangulacji, można wyznaczyć położenie obiektu w innym układzie współrzędnych. Jednocześnie wskutek wykorzystania obrazu z dwóch kamer jest możliwe wykonanie pełnego zdjęcia 3D.

Główym walorem tej metody jest możliwość wykonywania zdjęć z dużą częstotliwością. Ponadto światło emitowane przez ten skaner nie jest światłem laserowym, dlatego też wszelkie aspekty zagrażające wzroku są wyeliminowane [7]. Słabym punktem tej metody jest ograniczona gęstość rzutowanej siatki. Co w przypadku obiektu o skomplikowanym kształcie sprawia, iż pomiar będzie musiał być wykonany parokrotnie pod różnymi kątami w celu uzyskania dokładnych wyników [7]. Zasada działania skanera bazującego na metodzie emitowania światła strukturalnego została przedstawiona na rysunku 2.4.



**Rysunek 2.4:** Metoda wyznaczania współrzędnych w technice światła strukturalnego [10].

Korzystając z podstawowych zależności trygonometrycznych możliwe jest wyrowadzenie równań pozwalających na wyznaczenie odległości poszczególnych punktów mierzonego obiektu od kamery.

$$\begin{cases} \frac{R}{\sin(\alpha)} = \frac{B}{\sin(180-\alpha-\theta)} \\ R = B \cdot \frac{\sin(\theta)}{\sin(\alpha+\theta)} \end{cases} \quad (2.4)$$

gdzie:  $\alpha$  to kąt pomiędzy wiązką światła emitowanego przez projektor, a odcinkiem łączącym projektor oraz kamerę. Odcinek łączący projektor oraz kamerę tworzy z wiązką światła docierającą do kamery kąt  $\theta$ . Odległość pomiędzy kamerą, a projektorem jest równa  $B$ , zaś  $R$  jest odlegością kamery od mierzonego przedmiotu.

W tabeli 2.2 zostały przedstawione charakterystyki urządzenia Faro Scan in a box.

**Tabela 2.2:** Charakterystyki skanera Faro Scan in a box [11].

|                  |                            |
|------------------|----------------------------|
| Zasięg           | 0.2 m - 1.12 m             |
| Gęstość siatki   | 0.078 mm-0.39 mm           |
| Prędkość pomiaru | < 4 s                      |
| Gęstość meshu    | Do 10 mln wierzchołków     |
| Dokładność       | Do 0.1 % wielkości obiektu |

Ze względu na krótki zasięg pomiaru skanery korzystające z metody emitowania światła strukturalnego nadają się jedynie do pomiarów przy bliskiej odległości. Ten sposób pomiarowy jest wykorzystywany w urządzeniach takich jak Xbox Kinect oraz kamerach RGBD firmy Intel. Gęstość siatki generowanej na obiekcie mierzonym jest bardzo duża, przez co dokładność odwzorowania takich modeli znacznie przewyższa inne metody pomiarów 3D. Ponadto przy wykonywaniu pomiaru, dokonywany jest skan całej powierzchni obiektu na którą trafiło emitowane światło. Wpływa to znacznie na prędkość pomiaru, ponieważ przy jednej obserwacji jest możliwe uchwycenie nawet połowy obiektu, o ile cały został umieszczony w kadrze kamery.

### 2.3. Fotogrametria

Metoda odtwarzania trójwymiarowego kształtu obiektów z płaskich dwuwymiarowych zdjęć nazywana jest fotogrametrią. Polega ona na mierzeniu korelacji między sobą poszczególnych obrazów, które wykonywane są w odstępie od 5 ° do 15 ° od siebie. W celu zwiększenia dokładności używa się również technologii SFM (ang. *Structure From Motion*). Opiera się ona na identyfikacji homologicznych punktów na różnych obrazach w celu uzyskania perspektywy między nimi. Poprzez wykorzystanie efektu paralaksy istnieje możliwość późniejszego określenia w jakiej odległości od kamery znajdują się poszczególne punkty na obrazie. Umożliwia to utworzenie funkcji przejścia między nimi oraz otrzymanie na końcowym etapie pełnego modelu 3D [12]. Pozytywnym aspektem tego rozwiązania jest niski koszt, gdyż do wykonania zdjęć wystarczy jedynie aparat w urządzeniu mobilnym. Największą wadą jest wysoka złożoność obliczeniowa. Często takie obliczenia wykonywane są w chmurze co zwiększa koszty eksploatacji takiej metody oraz dokładność pomiarów jest względnie niska w porównaniu ze skanerami RGBD.

### 2.4. Skanery impulsowe LIDAR

Zasada działania takiego lasera jest zbliżona do funkcjonowania radaru. Skanery impulsowe mierzą czas potrzebny wiązce lasera do przebycia drogi do przedmiotu i na tej podstawie określają odległość zmierzzonego punktu od źródła światła. Skanery te pozwalają na wykonywanie pomiarów na odległości nawet do 600 km w przypadku skanerów korzystających z pomiaru czasu przelotu oraz do 500 m w przypadku wykorzystania metody pomiaru przesunięcia fazowego, dlatego głównie wykorzystuje się je do mierzenia dużych odległości [13]. Z wielu typów skanerów impulsowych można wyróżnić dwa podstawowe, zostały one opisane poniżej.

#### 2.4.1. Time of flight LIDAR

Skaner impulsowy mierzący czas od momentu wysłania wiązki światła, aż do jej odbicia od obiektu nazywany jest (ang. *TOF-time of flight*). Zasada działania takiego skanera polega na wysłaniu wiązki światła w kierunku obiektu, a następnie zmierzenie w jakim czasie wiązka wróci do odbiornika. Takie urządzenia, ze względu na swoją konstrukcję i bardzo dobrą dokładność na dużych odległościach są stosowane najczęściej [14]. Charakterystyki przykładowego urządzenia Benewake TF03-180 zostały przedstawione w tabeli 2.3.

**Tabela 2.3:** Charakterystyki skanera Benewake TF03-180 [15].

|                             |   |
|-----------------------------|---|
| Zasięg                      | 0.1 m - 180 m                                   |
| Dokładność                  | do 1 cm   |
| Prędkość pomiaru            | do 1000 Hz                                      |
| Długość fali lasera         | 905 nm  |
| Kąt wykrycia wiązki światła | 0.5 °   |
| Błąd pomiaru                | ±10 cm przy odległości do 10 m, 1% powyżej 10 m |

#### 2.4.2. Phase-Shift LIDAR

Kolejnym typem skanera impulsowego jest (ang. *Phase-Shift LIDAR*). Mierzy on przesunięcie fazowe wiązki odbitej od obiektu, uzyskując na tej podstawie odległość od przedmiotu. Główną zasadą wykorzystywaną w tym skanerze jest fakt, że przy odbiciu od powierzchni faza światła zostaje przesunięta [16]. Mając również na uwadze, iż różnica przesunięcia fazowego pomiędzy wiązką odbitą, a wiązką wysłaną jest proporcjonalna do odległości przebytej przez światło można otrzymać następujące wzory [17]:

$$d = \frac{c}{2f} \cdot \frac{\phi}{2\pi} \quad (2.5)$$

Korzystając z powyższego wzoru można otrzymać odległość obiektu od odbiornika, oznaczoną jako  $d$ . Prędkość światła została oznaczona we wzorze symbolem  $c$ . Częstotliwość sygnału referencyjnego, na którego podstawie wyznaczane jest przesunięcie fazowe wynosi  $f$ . Zmierzone przesunięcie fazowe pomiędzy sygnałem referencyjnym, a odebranym przez odbiornik wynosi  $\phi$ .

Z powyższego wzoru wynika fakt, że w celu uzyskania dokładniejszych pomiarów odległości należy zwiększyć częstotliwość światła służącego do próbkowania. Skaner Benewake TF02 jest urządzeniem wykorzystującym pomiar przesunięcia fazowego do obliczania odległości (ang. *indirect time-of-flight*). Jego charakterystyki oraz wygląd zostały przedstawione w tabeli 2.4.

**Tabela 2.4:** Charakterystyki skanera Benewake TF02 [18].

|                             |  |
|-----------------------------|--|
| Zasięg                      | 0.4 m - 22 m                                 |
| Dokładność                  | do 1 cm                                      |
| Prędkość pomiaru            | do 100 Hz                                    |
| Długość fali lasera         | 850 nm                                       |
| Kąt wykrycia wiązki światła | 3 °  |
| Błąd pomiaru                | ±5 cm przy odległości do 5 m, 2% powyżej 5 m |

Obie te metody, zarówno pomiaru przesunięcia fazowego jak i mierząca czas przelotu światła charakteryzują się bardzo dobrą dokładnością. Jednak znajdują zastosowanie w różnych dziedzinach. Skanery LIDAR pierwszego typu stosuje się przy pomiarach daleko zasięgowych, głównie w terenie. Natomiast te drugiego typu wykorzystywane są przy pomiarach krótko-zasięgowych wymagających wysokiej dokładności oraz szybkości działania. Dlatego stosowane są głównie w pomiarach różnego typu odkształceń [19]. Porównanie charakterystyk obu metod znajduje się w tabeli 2.5.

**Tabela 2.5:** Porównanie charakterystyk skanera Benewake TF02 oraz TF03-180.

| Skaner              | Benewake TF02                     | Benewake TF03-180                    |
|---------------------|-----------------------------------|--------------------------------------|
| Zasięg              | 0.4 m - 22 m                      | 0.1 m - 180 m                        |
| Dokładność          | do 1 cm                           | do 1 cm                              |
| Prędkość pomiaru    | do 100 Hz                         | do 1000 Hz                           |
| Długość fali lasera | 850 nm                            | 905 nm                               |
| Kąt wykrycia        | 3 °                               | 0.5 °                                |
| Błąd pomiaru        | ±5 cm dla d < 5 m, 2% dla d > 5 m | ±10 cm dla d < 10 m, 1% dla d > 10 m |

## 2.5. Ogólne charakterystyki laserowych metod pomiarowych

Zestawienie uśrednionych parametrów poszczególnych metod pomiarowych znajduje się w tabeli 2.6. Na jej podstawie można dojść do wielu wniosków. Każda z metod triangulacji laserowej znajduje swoje zastosowanie w różnych dziedzinach nauki. W zależności od tego jaki obiekt ma zostać zmierzony oraz z jakiej odległości, wybór metody pomiarowej będzie inny.

**Tabela 2.6:** Charakterystyki metrologiczne laserowych metod pomiarowych [7].

| Metoda pomiarowa | Zakres pomiarowy [m] | Dokładność [mm] | Prędkość pomiaru $\frac{\text{punkty}}{\text{sekunda}}$ |
|------------------|----------------------|-----------------|---|
| ToF              | < 1500               | < 20            | do 12000  |
| P-S              | < 100                | < 10            | do 625000   |
| Triangulacja     | < 5                  | < 0.1           | do 10000  |

## 2.6. Zastosowania skanerów 3D

Rozwój technologii trójwymiarowych skanerów spowodował ich wdrożenie do znacznej liczby dziedzin. Wraz z rozwojem tej dyscypliny nauki, skanery 3D stały się dostępniejsze dla użytkowników. By wykonać wirtualny skan obiektu nie potrzeba już wielkiej mocy obliczeniowej komputera oraz dużych nakładów finansowych. Aktualnie w miarę dokładny skaner 3D jest dostępny nawet dla nieprofesjonalnego użytkownika. Współcześnie skanery 3D znajdują zastosowanie w większości gałęzi życia oraz przemysłu.

Przemysł budowlany wykorzystuje je w celu mierzenia odkształceń belek z dużą dokładnością [20]. W tej branży pomiary trójwymiarowe wykorzystywane są również do bardzo dokładnych pomiarów odległości oraz umiejscowienia różnych elementów w przestrzeni. Poprzez połączenie fotogrametrii oraz skanerów typu LIDAR uzyskuje się lepszą dokładność oraz szybszy czas pomiaru. Z pomiarów dokonanych w [9] wynika, że połączenie obu tych metod zmniejszyło czas badania o 75% w stosunku do samego wykorzystania skanera typu LIDAR.

Fotogrametria jest używana przy wielkoformatowych skanach 3D. Jednym z takich skanów jest Google Earth. Jest to wirtualny model kuli ziemskiej na którym została dokładnie odzwierciedlona znaczna część globu. Korzystając ze zdjęć satelitarnych tworzone są zdjęcia bardzo wysokiej rozdzielczości. Są one podstawą do ukazania kuli ziemskiej z dalszej perspektywy. Następnie wykorzystując zdjęcia zrobione z pokładów samolotów tworzone są modele w dokładniejszej rozdzielczości. Istotne jest zastosowanie fotogrametrii by z płaskich zdjęć samolotowych uzyskać pełne modele 3D skanowanych obiektów. Ta technologia została przez Google dopracowana do perfekcji co widać po niesamowitych obiektach, które można zaobserwować na portalu Google Earth. Dane z tej platformy mogą również posłużyć innym badaczom do przeprowadzania własnych eksperymentów. W pracy amerykańskich naukowców zostały przedstawione wyniki skanów fotogrametrycznych wykonanych na podstawie zdjęć z Google Earth [21]. Płaskie obrazy 2D zostały przez nich poddane obróbce oraz analizie, dzięki czemu w rezultacie otrzymali oni bardzo dobre wyniki. Błąd pomiarowy ich metody w porównaniu do rzeczywistych odległości zmierzonych na Google Earth wynosił zwykle mniej niż 1% na płaszczyźnie XY oraz do 5% na płaszczyźnie Z. Zarówno pomiary Google jak i naukowców miały trudności z poprawnym zobrazowaniem górzystych terenów. Wynikało to z faktu niskiej rozdzielczości zdjęć, przez co odkształcenia terenu było ciężko zmierzyć.

Skanery trójwymiarowe wykorzystywane są w medycynie do badania części ciała oraz modelowania komputerowego kończyn [22]. Wykorzystywane są również przy oględzinach pacjentów. Wysoka dokładność przy mierzeniu odkształceń na powierzchni ciała sprawia, że dzięki skanerom 3D można dostrzec złamania kości oraz inne ubytki [23]. Kolejną ważną rolą, jaką spełniają skanery 3D w służbie zdrowia jest druk 3D. Poprzez analizę danych otrzymanych z trójwymiarowych pomiarów oraz wykorzystując druk 3D można tworzyć modele kończyn. Takie elementy są pomocne przy nauce studentów oraz znajdują swoje zastosowanie w protetyce [24].

W przemyśle spożywczym skanery 3D używane są do kontroli jakości poszczególnych wyrobów [25]. Stosując daną metodę pomiarową, można bardzo szybko uzyskać informację na temat wymiarów danego produktu. Poprzez znajomość rozmiaru obiektu można obliczyć jego objętość, chropowatość oraz wypełnienie powierzchni. Wszystkie te informacje posługują się do weryfikacji parametrów produktu. Dzięki temu, będzie można wyznaczyć czy nadaje się on do sprzedaży.

Wykorzystuje się je także w branży ubraniowej do dokładnych pomiarów człowieka. Dzięki zastosowaniu technologii modelowania 3D możliwe jest zeskanowanie w całości człowieka, a następnie stworzenie idealnie pasujących ubrań [26]. Poprzez zastosowanie wyżej wymienionych środków można projektować ubrania, a następnie sprawdzać je na modelach bez konieczności bez konieczności wykorzystywania zasobów ludzkich i materialnych. Znacząco wpływa to na produktywność projektantów mody, którzy dotychczas musieli poświęcać długie godziny na wcielenie swoich pomysłów w życie. Taką technologię zaimplementowała Izraelska firma Optitex. Oferuje ona wirtualny wybieg, na którym chodzą modele ubrane w stroje dodane przez użytkownika [27]. Ta technologia może być również korzystna z perspektywy kupującego. Jeśli posiadałby on swoją wirtualną kopię to mógłby przymierzać ubrania bez konieczności wychodzenia z domu. Firma Visionix oferuje usługę skanu twarzy, dzięki której można przymierzać okulary na wirtualnym modelu [28]. Dzięki dokładnej znajomości wymiarów twarzy, możliwe jest dobranie pasujących oprawek w krótkim czasie. Została ona pierwszy raz opublikowana w roku 2001, na tamten moment była ona na tyle rewolucyjna, że została wspomniana w wielu innych pracach naukowych [26] [29].

### 3. MODEL SKANERA ORAZ KONCEPCJA DZIAŁANIA

Skanery 3D swoim wyglądem i zasadą działania przystosowane są do różnych zastosowań. W celu skanowania dużych obiektów takich jak budynki korzysta się z drogich skanerów wielkoformatowych. Za to jeśli użytkownik chce zeskanować mały obiekt, możliwe jest dokonanie tego urządzeniem ręcznym. W niniejszym rozdziale zostanie przedstawiony autorski projekt skanera 3D razem z opisem jego zasady działania. Przedstawione zostaną podstawy teoretyczne przekształcenia danych z kamery głębi do chmury punktów. Ukażane zostały również algorytmy służące do utworzenia siatki punktów (tzw. *meshu*) na podstawie chmury punktów.

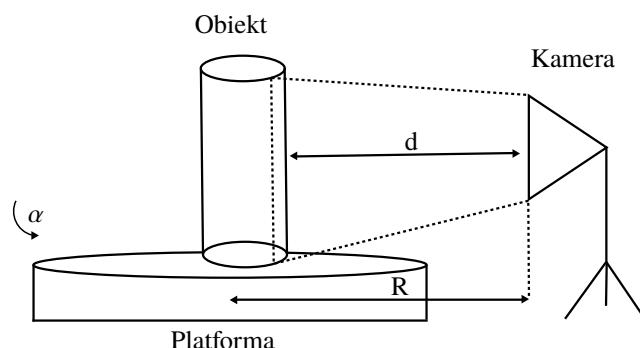
#### 3.1. Model i konstrukcja skanera 3D

W celu wykonania dokładnych modeli trójwymiarowych został utworzony skaner 3D na podstawie autorskiego projektu. W skład zestawu wchodzi kamera Intel RealSense D435i oraz platforma obrotowa. Wybór sensora od firmy Intel nie był przypadkowy. Posiada on szereg wbudowanych funkcji, takich jak łatwa możliwość kalibracji oraz nastaw odpowiednich parametrów wykrywania głębi. Jego rozdzielcość oraz dostępna liczba klatek na sekundę sprawia, że akwizycja danych jest o wiele dokładniejsza. Podczas budowy skanera dokonano porównania możliwości dwóch kamer trójwymiarowych: Intel RealSense D435i oraz Orbbec Astra Mini MX6000. W tabeli 3.1 dokonano przeglądu najważniejszych charakterystyk obu tych skanerów, zestawiono między innymi rozdzielcości głębi oraz kąt przechwytywania obrazu.

**Tabela 3.1:** Porównanie charakterystyk kamer Intel RealSense D435i oraz Orbbec Astra Mini MX6000 [30] [31].

| Kamera              | Astra Mini                                     | RealSense D435i                              |
|---------------------|--|--|
| Dokładność          | $\pm 1 - 3 \text{ mm na } 1 \text{ m}$         | < 2% na 2 m                                  |
| FOV                 | $60^\circ \text{H} \times 49.5^\circ \text{V}$ | $87^\circ \text{H} \times 58^\circ \text{V}$ |
| Rozdzielcość RGB    | 640 px x 480 px                                | 1920 px x 1080 px                            |
| Rozdzielcość głębi  | 640 px x 480 px                                | 1280 px x 720 px                             |
| FPS                 | 30   | 90   |
| Długość fali lasera | 830 nm   | 850 nm                                       |

Z powyższych charakterystyk wynika, że kamera firmy Intel jest dokładniejsza oraz lepiej spełni zadanie wiernego odwzorowania modelu 3D. Ponadto oprogramowanie dostarczane przez firmę Intel o nazwie RealSense Viewer pozwala na łatwą obsługę tego urządzenia. Umożliwia ono podgląd obrazu z kamery zarówno w 2D jak i w 3D. W programie dostępne są ustawienia, umożliwiające poprawną regulację parametrów rejestracji obrazu, takie jak moc lasera, wartość graniczną wykrywanej głębi oraz ekspozycję. Wszystkie te aspekty znacząco usprawniają proces kalibracji, produktywność oraz wpływają na poprawę dokładności generowanych obrazów. Konstrukcja zbudowanego skanera została zaprezentowana na rysunku 3.1.



**Rysunek 3.1:** Schemat budowy autorskiego skanera 3D.

Na powyższym rysunku można dostrzec dwa kluczowe elementy wchodzące w skład skanera. Platforma ruchoma napędzana silnikiem elektrycznym zapewnia stałą prędkość kątową obrotu tacki. Dzięki temu wyznaczanie położenia obiektu w przestrzeni jest dokładne. Wykonane zostały testy platformy napędzanej silnikiem elektrycznym oraz poruszanej ręcznie. Z wytworzonych w ten sposób modeli wynika jasno, iż stała prędkość kątowa obiektu jest kluczowa do poprawnego przekształcenia modelu. Kolejnym elementem wykorzystanym przy budowie skanera jest kamera RGBD. Wykonuje ona zdjęcia kolorowe oraz głębi z określoną częstotliwością oraz zapisuje je do pliku, w celu późniejszej ich obróbki. Ze względu na wykorzystanie platformy o stałej prędkości kątowej, dokonane zostało porównanie wpływu FPS na wygląd ostatecznego modelu. Zmiana liczby klatek na sekundę wpływa bezpośrednio na rozdzielcość kątową wykonanych zdjęć, gdy ta liczba jest większa, gęstość chmury punktów również się zwiększa.

Zasada funkcjonowania skanera została przedstawiona poniżej:

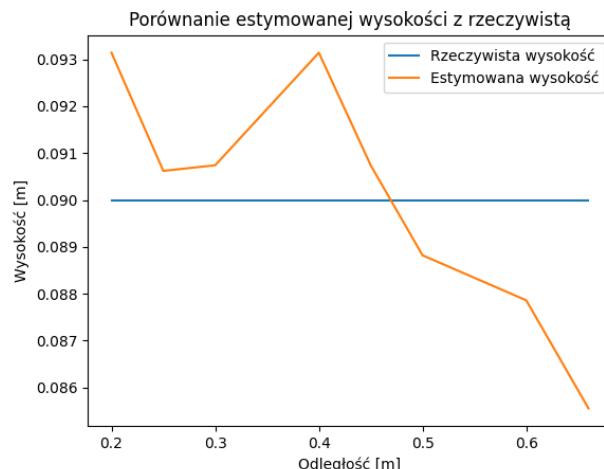
1. Mierzona jest dokładna odległość obiektywu kamery od środka tacki.
2. Obiekt umieszczany jest na obrotowej tacce.
3. Dokonuje się kalibracji tak ustawionego elementu, tak by stopień wypełnienia punktów był jak najdokładniejszy.
4. Tacka zostaje uruchomiona z prędkością  $0.1 \frac{rad}{s}$ .
5. Uruchomiony zostaje zapis obrazu głębi oraz RGB z kamery.
6. Gdy tacka wykona pełen obrót, nagrywanie oraz tacka zostają zatrzymane.

Wysokość obiektu jest mierzona na podstawie danych z kamery RGBD. Znając odległość kamery od obiektu, można wyznaczyć jego wysokość korzystając ze wzoru:

$$H_m = D \cdot \frac{h_{pix}}{h_{sens}} \cdot V_{FOV} \quad (3.1)$$

W powyższym wzorze  $H_m$  jest rzeczywistą wysokość obiektu,  $D$  określa odległość kamery od obiektu w metrach, zaś  $V_{FOV}$  oznacza pionowy kąt widzenia kamery. Wysokość soczewki kamery została oznaczona we wzorze jako  $h_{sens}$ , a  $h_{pix}$  jest wysokością obiektu w pikselach.

W celu wyznaczenia rzeczywistej wysokości obiektu w zależności od odległości i jego wysokości w pikselach, należy znać pionowy kąt widzenia kamery oraz wysokość soczewki. Informacje o kącie widzenia skanera znajdują się w dokumentacji urządzenia Intel RealSense D435i, natomiast wysokość soczewki należy wyznaczyć empirycznie [31]. W tym celu, dokonano pomiaru wysokości obiektu w pikselach na obrazie z kamery oraz jego odległość od obiektywu. Badanie powtórzono 8 razy w celu uzyskania dokładnej aproksymacji. Znając odległość oraz rzeczywistą wysokość obiektu, po przekształceniu wzoru można uzyskać wartość  $h_{sens}$ . Po uśrednieniu wyników ze wszystkich pomiarów przeprowadzono badanie jakości estymacji wysokości. Wykres rzeczywistej wysokości obiektu oraz jej przybliżenia znajduje się na rysunku 3.2.



**Rysunek 3.2:** Porównanie rzeczywistej wysokości z jej estymacją.

Na powyższym rysunku można zauważyc, że aproksymacja wysokości daje dobre rezultaty. W zakresie od 20 cm do 66 cm, maksymalny błąd względny wyniósł 4.2%. Otrzymane rezultaty są zadowalające do poprawnej wizualizacji rzeczywistego obiektu.

### 3.2. Przejście do chmury punktów

Po rejestracji pełnego obrotu obiektu na platformie, uzyskany plik należy odpowiednio przetworzyć. Na podstawie danych w nim zapisanych należy przejść z płaszczyzny dwuwymiarowej do chmury punktów. Poniżej opisany został szereg czynności, które należy wykonać w celu otrzymania docelowego rezultatu. Przedstawiono procedury, prowadzące do poprawnej reprezentacji punktów w przestrzeni trójwymiarowej. Na potrzeby projektu dokonano porównania dwóch metod generacji chmury punktów. W pierwszej metodzie wykorzystywana jest pojedyncza kolumna z każdej klatki nagrania, tym samym trójwymiarowy skaner wykorzystywany jest jako skaner liniowy. Kolejną użyta metodą jest wykorzystanie całego obrazu z kamery RGBD. Umożliwi to utworzenie chmury punktów z wielu kolumn obrazu.

#### 3.2.1. Wstępna obróbka danych.

Pierwszym krokiem użytej metody jest wstępna obróbka danych. Odczytywane są dane zapisane w pliku .bag z kamery RGBD. Poprzez zapis całego nagrania z kamery można dokonywać pomiaru wykorzystując narzędzie RealSense Viewer. Następnie wybierana jest kolumna obrazu która posłuży do ekstrakcji danych z kamery głębi. Filtracja w ten sposób danych ma na celu zwiększenie wydajności kodu. Dzieje się tak, ponieważ do późniejszych algorytmów będzie wykorzystywana tylko jedna kolumna z klatki obrazu, na podstawie której dokonana zostanie ekstrakcja informacji o głębi.

#### 3.2.2. Przekształcenie danych z kamery RGBD do współrzędnych 3D.

Następnym etapem schematu jest przejście z dwuwymiarowego układu współrzędnych kamery do trójwymiarowego układu obiektu. W tym celu zostanie użyte poniższe przekształcenie, wynikiem którego będzie macierz czterowymiarowych punktów. W skład punktu wchodzą trzy współrzędne odpowiadające pozycji w przestrzeni oraz wartość RGB danego punktu symbolizująca jego kolor.

$$\left\{ \begin{array}{l} D_\alpha = \begin{bmatrix} d_0 & \dots & d_{max} \end{bmatrix} \\ RGB_\alpha = \begin{bmatrix} rgb_0 & \dots & rgb_{max} \end{bmatrix} \\ X_\alpha = \cos(\alpha) \cdot (R - D_\alpha) \\ Y_\alpha = \sin(\alpha) \cdot (R - D_\alpha) \\ Z = \begin{bmatrix} 0 & \dots & h_{max} \end{bmatrix} \\ w_\alpha^n = \begin{bmatrix} X_\alpha[n] & Y_\alpha[n] & Z[n] & RGB_\alpha[n] \end{bmatrix} \end{array} \right. \quad (3.2)$$

Skaner został ustawiony w odległości R od osi obrotu tacki. Dla każdego kąta obrotu tacki  $\alpha$  mierzona jest odległość punktów od kamery i zostaje umieszczona w macierzy  $D_\alpha$ , zaś ich kolor umieszczony jest w macierzy  $RGB_\alpha$ . Macierze  $X_\alpha$ ,  $Y_\alpha$  zawierają współrzędne X oraz Y zmierzonych punktów dla danego kąta obrotu tacki  $\alpha$ . Macierz Z posiada współrzędne wzdłuż osi Z dla każdego z punktów, jej wartości zawierają się w przedziale od 0 do  $h_{max}$ , gdzie  $h_{max}$  jest wysokością obiektu w docelowych jednostkach miary. Wszystkie współrzędne n-tego punktu wraz z jego kolorem zawarte są w macierzy  $w_\alpha^n$ .

#### 3.2.3. Normalizacja otrzymanych punktów.

Kamera RGBD jest zaawansowanym technicznie urządzeniem. Pomimo kalibracji kamery, w przechwyconych obrazach mogą występować niedoskonałości. Przejawiają się one w złych pozycjach punktów względem pozostałych sąsiadów. Jest to spowodowane przekłamianymi odczytami głębi przez sensor kamery i jest to nieuniknione. Wynika to z konstrukcji skanera oraz odbić światła lasera od powierzchni obiektu. W celu zmniejszenia wpływu błędnych pomiarów

na ostateczny wygląd obiektu należy dokonać normalizacji punktów. Normalizacja sprowadza się do zbadania, czy owy punkt leży w odległości od środka podobnej do jego sąsiadów. W tym celu została wyznaczona średnia odległość punktów od osi przechodzącej przez środek obrotu tacki, która ma współrzędne (0,0,H). Warto zauważyć, że wartość współrzędnej Z nie płyną na wyznaczanie odległości, ponieważ utworzona ona została na podstawie równomiernego rozkładu wartości wysokości danego obiektu. Następnie, znając wartość średniej odległości punktów od środka układu współrzędnych, można empirycznie dobrać wartość graniczną odległości powyżej której punkty będą poddane interpolacji.

### 3.2.4. Interpolacja punktów

Interpolacja jest procesem aproksymacji współrzędnych punktów w miejscach, w których wystąpiły przeklamanie. Jest wiele różnych metod interpolacji, które mają za zadanie jak najwierniej oddać wartość punktu w nieznanym miejscu. Głównymi metodami są metoda najbliższych sąsiadów oraz interpolacja wielomianowa.

#### Metoda najbliższych sąsiadów

Interpolacja korzystająca z metody najbliższych sąsiadów jest często stosowanym algorytmem do rekonstrukcji danych w nieznanym miejscu. Wykorzystywana jest na przykład przy powiększaniu zdjęć do uzyskania lepszej rozdzielczości [32]. Zasada działania tej metody została przedstawiona poniżej.

1. Wybierany jest punkt P, którego wartość ma zostać wyznaczona.
2. Punkty w chmurze nie są rozłożone równomiernie, dlatego też wyznaczenie sąsiadów sprowadza się do wyznaczenia odległości wszystkich pozostałych punktów od wybranego.
3. Dla każdego punktu w chmurze mierzona jest jego odległość od wybranego punktu. Współrzędne punktu są trójwymiarowe, więc odległość od sąsiada  $D_n$  może zostać obliczona za pomocą wzoru:

$$D_n = \sqrt{(x_p - x_n)^2 + (y_p - y_n)^2 + (z_p - z_n)^2} \quad (3.3)$$

$x_p, y_p, z_p$  są współrzędnymi punktu, a  $x_n, y_n, z_n$  są współrzędnymi n-tego sąsiada.

4. Ze zbioru wyznaczonych w ten sposób odległości wyznaczana jest najmniejsza. Punkowi zostaje przypisana wartość najbliższego do niego punktu.

#### Interpolacja wielomianowa

Interpolacja wielomianowa jest szeroko stosowanym zagadnieniem w matematyce oraz fizyce. Jej szczególnymi odmianami są interpolacja liniowa oraz kwadratowa. Polega ona na dopasowaniu funkcji na przykład liniowej, kwadratowej lub dowolnego innego stopnia wielomianu do zbioru punktów. Znając funkcję przechodzącą przez wszystkie punkty ze zbioru, można określić jaka będzie jej wartość w punkcie którego wartość była dotychczas nieznana. Równania przedstawiające tę metodę zostały przedstawione poniżej.

$$\left\{ \begin{array}{l} X = \begin{bmatrix} x_0^0 & \dots & x_0^n \\ \vdots & \ddots & \vdots \\ x_n^0 & \dots & x_n^n \end{bmatrix} \\ A = \begin{bmatrix} a_0 \dots a_n \end{bmatrix}^T \\ Y = \begin{bmatrix} y_0 \dots y_n \end{bmatrix}^T \\ Y = X \cdot A \\ A = X^{-1} \cdot Y \\ W(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \dots + a_n \cdot x^n \end{array} \right. \quad (3.4)$$

Macierze X oraz Y przedstawia współrzędne punktów ze zbioru. Macierz A zawiera współczynniki funkcji W(x), zaś W(x) jest wielomianem przechodzącym przez wszystkie punkty ze zbioru. Znając wielomian interpolacyjny

można określić współrzędne poszukiwanego punktu, który znajdował się poza zbiorem dostępnych punktów, jego współrzędne to  $P(x_p, W(x_p))$ .

Na potrzeby implementacji powyższej metody, dokonano interpolacji punktów wzduż każdej z trzech osi układu współrzędnych. W tym wypadku, osią odciętych były indeksy poszczególnych punktów, zaś osią rzędnych były ich współrzędne wzduż wybranej osi.

### 3.2.5. Metoda światła strukturalnego

Drugą metodą akwizycji danych głębi jest użycie światła strukturalnego. Poprzez zastosowanie kamery RGBD możliwe jest uzyskanie informacji o trójwymiarowym rozłożeniu wszystkich punktów na obrazie. Przejście z nagrania do gotowego modelu składa się z kilku kroków.

## Utworzenie chmury punktów

W celu przejścia do chmury punktów wykorzystano narzędzie deweloperskie firmy Intel o nazwie rs-convert. Pozwala ono na przekształcenie każdej klatki nagrania, w pojedyncze ujęcie chmury punktów w układzie współrzędnych kamery. W ten sposób można wygenerować ilość chmur punktów bliską liczbie klatek w nagraniu. Niekiedy podczas wykorzystywania wspomnianego narzędzia występują błędy, sprawiające, iż pewne klatki nie zostaną przetworzone. Może to wynikać z błędnych odczytów sensora kamery podczas wykonywania nagrania. Kolejną przyczyną może być błąd podczas transmisji danych z urządzenia do komputera. Należy wtedy wziąć następną dostępną klatkę w celu spójności nagrania. Każda chmura przedstawia widok na obiekt obrócony względem klatki początkowej pod kątem  $\frac{\text{Indeks klatki} \cdot 360^\circ}{\text{Liczba klatek}}$ .

## Filtracja punktów

Na początek punkty należy odpowiednio odfiltrować. Ujęcia przedstawiają wszystkie punkty zarejestrowane przez kamerę takie jak tło oraz obracana taca. W celu wyizolowania skanowanego obiektu należy zmierzyć odległość każdego punktu od kamery. Zakres odległości mieści się w przedziale od początku tacki, do jej środka. Jest to punkt wyjściowy do dalszego dobierania odległości, które należy wykonać ręcznie w celu otrzymania najlepszych rezultatów. Punkty można również odfiltrować wzduż osi Y, by usunąć pozostałości punktów przynależących do wierzchu obracanej tacki.

## Translacja oraz obrót chmury punktów

Odfiltrowane chmury punktów należy odpowiednio obrócić by odwzorowywały one ostateczny wygląd modelu. W tym celu należy dokonać translacji punktów, by wszystkie były obracane względem środka układu współrzędnych. Macierz translacji ma następującą postać:

$$T = \begin{bmatrix} 1 & 0 & 0 & W_{mean} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & D \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Dla każdej chmury punktów zostały zmierzone wartości bocznych punktów skrajnych. Uśrednioną wartość obu tych punktów wzduż osi X stanowi  $W_{mean}$ . Odległość kamery od środka tacki została zmierzona na początku wykonywania pomiaru i wynosi D.

Poprzez zastosowanie powyższego wzoru możliwe jest przesunięcie chmury tak, że jej środek leży wzduż osi X, a odległość wzduż osi Z symbolizuje rzeczywistą odległość punktów na powierzchni obiektu od środka obrotu tacki. Wzdłuż osi Y nie dokonywano przekształceń, ponieważ dzięki zastosowaniu dedykowanego narzędzia do generacji chmur punktów firmy Intel odległości pionowe między poszczególnymi punktami są zachowane.

Następnym krokiem jest obrócenie chmury punktów wzduż osi Y. Zamiarem takiego przekształcenia jest

utworzenie z trójwymiarowych ujęć obiektu z różnych perspektyw jednej spójnej całości. Obrót wykonano za pomocą macierzy rotacji, której wzór został podany poniżej.

$$T = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \quad (3.6)$$

W powyższym wzorze kąt  $\alpha$  oznacza kąt obrotu chmury względem początkowego zbioru.

Poprzez zastosowanie wszystkich powyższych zabiegów otrzymano nakładające się chmury punktów w równych odstępach kątowych. Do otrzymania dokładniejszych rezultatów punkty należy odpowiednio do siebie dopasować. W tym celu posłużą funkcje z biblioteki Open3D. Użyte algorytmy oraz sposoby przeprowadzenia globalnej rejestracji pochodzą z zasobów dydaktycznych biblioteki [33].

### Decymacja punktów

Pierwszym krokiem metody dopasowania chmur punktów do siebie jest proces decymacji punktów. Algorytm ma na celu zmniejszenie ilości punktów w każdej z chmur, co znaczowo wpływa na wydajność pozostałych algorytmów. Ponadto zmniejszając zagęszczenie punktów nakładające się na siebie warstwy stają się mniej widoczne. W celu wykonania decymacji użyto funkcji z biblioteki Open3D o nazwie voxel\_down\_sample.

### Histogram cech punktów

W celu odpowiedniego nałożenia chmur punktów na siebie, dla każdej z chmur obliczono histogramu cech punktów. Jest to zabieg mający na celu wyznaczenie pewnych cech szczególnych w zbiorze punktów, takich jak na przykład zmiany geometrii na powierzchni chmury [34]. W celu wyznaczenia histogramu punktów należy również obliczyć wektory normalne do powierzchni chmury za pomocą funkcji estimate\_normals. Biblioteka Open3D umożliwia wyznaczenie szybkiego histogramu punktów implementując podstawy teoretyczne dostępne w pracy niemieckich naukowców [35]. Poprzez zastosowanie takiego kroku, możliwe jest następnie porównanie dwóch chmur punktów ze sobą oraz dopasowanie ich względem podobnych cech.

### Algorytm RANSAC

By odpowiednio nałożyć na siebie chmury punktów należy wyznaczyć macierz transformacji. Do tego celu posłuży algorytm RANSAC. Wymieniona metoda pozwala na iteracyjne wyznaczenie powierzchni generowanej przez obie chmury punktów, nawet wtedy gdy duża część z nich do niej nie należy. Poprzez wykorzystanie cech wspólnych na w obu chmurach możliwe jest wyznaczenie macierzy przejścia pomiędzy nimi, a co za tym idzie, zespolić ze sobą dane zbiory. Kroki funkcjonowania algorytmu prezentują się następująco:

1. Losowo wybierane jest N punktów ze zbioru źródłowego.
2. Dla każdego z punktów znajdywany jest jego najbliższy sąsiad w docelowej chmurze punktów na podstawie danych pochodzących z histogramu cech. Jeśli są one podobne, to punkty powinny zostać połączone.
3. Iteracyjnie znajdująca jest transformacja minimalizująca odległość podobnych do siebie punktów.
4. Algorytm jest powtarzany wiele razy do momentu osiągnięcia zbieżności lub przekroczenia maksymalnej ilości iteracji.

Funkcja pochodząca z biblioteki Open3D umożliwia przeprowadzenie operacji RANSAC na dostępnych parach chmur punktów. W celu uzyskania lepszej dokładności należy odpowiednio dostosować jej parametry. Głównym parametrem jest maksymalna ilość iteracji, jeśli będzie ona zbyt mała to metoda może zatrzymać się przed znalezieniem optymalnego rozwiązania. Kolejnym istotnym argumentem funkcji jest liczba wybieranych punktów testowych. W zależności od ich liczby chmury mogą zostać lepiej dopasowane, ponieważ więcej punktów będzie użyte do wyznaczenia macierzy transformacji. Kryterium zbieżności algorytmu RANSAC oprócz wyżej wspomnianych, stanowi również porównywanie odległości dwóch chmur od siebie. Jeżeli będą one dostatecznie blisko siebie, to funkcja zakończy swoje działanie, dając w rezultacie wynikową macierz transformacji.

### **3.3. Rekonstrukcja powierzchni**

Wszystkie powyższe metody pozwalały na określenie chmury punktów danego obiektu. W celu wyeksportowania gotowego modelu, potrzebna jest również znajomość całej powierzchni danego przedmiotu. W tym pomocne okazują się różne metody rekonstrukcji powierzchni. Jest to kluczowy element działania algorytmu wirtualizacji rzeczywistych obiektów do postaci modeli 3D. Powstało wiele prac naukowych na temat różnych metod rekonstrukcji powierzchni, poniżej zostały omówione najważniejsze z nich.

#### *3.3.1. Trójwymiarowa triangulacja Delaunaya*

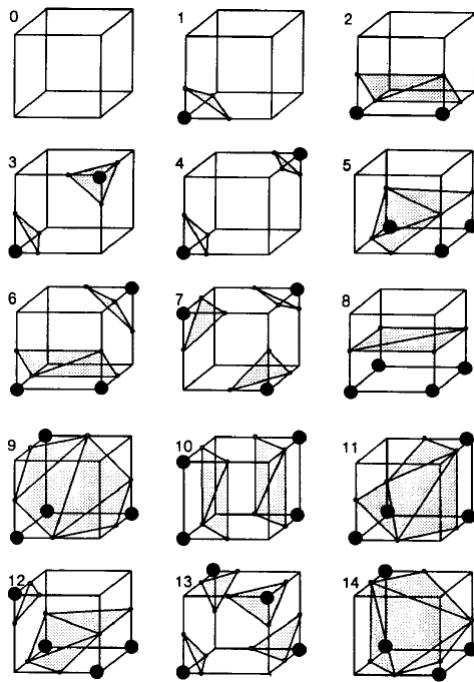
Triangulacja powierzchni metodą Delaunaya jest jedną z popularniejszych metod nakładania siatki na nieregularnie rozprowadzone punkty. Polega ona na połączeniu punktów w nieprzecinające się trójkąty, dzięki czemu możliwe jest nałożenie na nie koloru oraz utworzenie meshu na danym obiekcie. Triangulację Delaunaya można wykonać dla dowolnej N-wymiarowej płaszczyzny, w danym przykładzie zostanie opisana metoda dla trójwymiarowej chmury punktów [36]. Schemat działania tej metody został opisany poniżej.

1. Na początku tworzony jest super-ostrosłup zawierający wszystkie punkty ze zbioru. Będzie on stanowił wstęp do triangulacji.
2. Wybierany jest punkt ze zbioru, dla którego będzie dokonywana triangulacja.
3. Dla każdego z dostępnych ostrosłupów sprawdzane jest, czy punkt leży wewnątrz sfery opisanej na nim. Jeśli tak, to czworościan dodawany jest do zbioru ostrosłupów do usunięcia.
4. Dla każdego ostrosłupa ze zbioru do usunięcia, tworzone są cztery nowe, zawierające trzy wcześniejsze wierzchołki oraz punkt dla którego dokonywana jest triangulacja. W ten sposób utworzone czworościany dodawane są do zbioru dostępnych ostrosłupów.
5. Proces jest powtarzany dla każdego z punktów. Na końcu początkowy super-ostrosłup oraz wszystkie te które mają z nim wspólny wierzchołek zostają usunięte.

Zaletą tej metody jest to, że jest ona inkrementacyjna. Punkty do triangulacji są dodawane po kolej, a co za tym idzie, nie ma potrzeby tworzenia nowej siatki po dodaniu pojedynczego punktu. Wystarczy jedynie wykonać jedną fazę algorytmu i dopisać utworzone wierzchołki czworościanów do listy. Modyfikacją tej metody jest triangulacja DeWall zaproponowana przez włoskich naukowców w 1997 roku [36]. Jej nazwa pochodzi od Delaunay i ściany (ang. *Wall*). Pojęcie ściany oznacza w tym przypadku rekurencyjne dzielenie zbioru punktów na dwa podobne do siebie obszary, następnie równoległe utworzenie dwóch niezależnych od siebie triangulacji Delaunaya. Dzięki podzieleniu zbioru punktów na niezależne od siebie obszary, możliwe staje się wykorzystanie wielowątkowości procesora do obliczeń równoległych. To znaczco przyspiesza pracę algorytmu.

#### *3.3.2. Algorytm maszerujących sześciianów*

Algorytm maszerujących sześciianów jest metodą typu dziel i podbijaj zaproponowaną w 1987 roku przez amerykańskich naukowców w Nowym Yorku [37]. W przeciwieństwie do triangulacji Delaunaya, punkty na które ma zostać nałożona siatka meshu muszą być w regularnych odstępach od siebie. W związku z tym, chcąc wykorzystać tę metodę trzeba dokonać interpolacji punktów, w celu przejścia z nieregularnych przestrzeni pomiędzy elementami do równomiernie leżących punktów. Ze względu na schemat działania algorytmu, jest on niezwykle szybki, ponieważ pozwala on na uniknięcie części obliczeń. Algorytm dzieli przestrzeń na regularne sześciiany, a następnie sprawdza przecięcia punktów ze ścianami sześciianu. Przy regularnej siatce punktów istnieje  $2^8$  takich przecięć, ponieważ sześciian ma 8 krawędzi, na każdej z nich może leżeć punkt lub nie. Po uwzględnieniu symetrii obrotowej sześciianu liczbę kombinacji można zredukować do zaledwie 15. Zostały one przedstawione na rysunku 3.3.



Rysunek 3.3: Możliwe przecięcia sześcianu przez punkty [37]

Przecięcia punktów z sześcianami tworzą trójkąty, które później zostaną dodane do triangulacji. Należy zwrócić uwagę na to, że gęstość siatki na obiekcie poddanym triangulacji bezpośrednio zależy od długości boku sześcianu. Im jest on mniejszy, tym więcej będzie przecięć punktów z krawędziami, a co za tym idzie, więcej utworzonych trójkątów triangulacyjnych. Występują odmiany tej metody wprowadzające zmienną długość boku danego sześcianu [38]. Adaptacyjna metoda maszerujących sześcianów bada krzywiznę powierzchni wewnętrz sześcianów za pomocą wyznaczania wektorów normalnych utworzonych dla danych trójkątów. Jeśli wektory odchycone są w podobną stronę, oznacza to, że powierzchnia jest dostatecznie gładka. W przeciwnym wypadku sześciyan dzielony jest na kolejne cztery sześciany i algorytm jest powtarzany do momentu uzyskania odpowiedniej krzywizny powierzchni. Dzięki wykorzystaniu tej metody tworzona jest dokładniejsza triangulacja powierzchni względem początkowego algorytmu. Ponadto dzięki wykorzystaniu adaptacyjnej metody algorytm jest szybszy niż ten który by miał ustaloną długość boku. Porównanie algorytmu maszerujących sześcianów (MS) oraz jej adaptacyjnej odmiany (AMS) dla różnych ilości punktów znajduje się w tabeli 3.2.

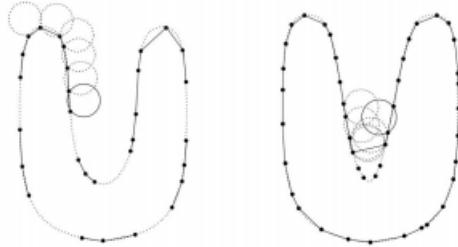
Tabela 3.2: Porównanie algorytmu MS oraz adaptacyjnych MS dla  $7.4 \cdot 10^6$  oraz  $2.8 \cdot 10^6$  punktów [38].

| Metoda                    | MS               | AMS              | MS               | AMS              |
|---------------------------|------------------|------------------|------------------|------------------|
| Liczba punktów            | $7.4 \cdot 10^6$ | $7.4 \cdot 10^6$ | $2.8 \cdot 10^6$ | $2.8 \cdot 10^6$ |
| Czas wykonywania obliczeń | 331 s            | 230 s            | 164 s            | 81 s             |
| Ilość trójkątów           | 718964           | 299292           | 393606           | 102868           |

### 3.3.3. Ball pivoting algorithm

Algorytm toczącej się kuli jest metodą służącą do rekonstrukcji powierzchni na podstawie chmury punktów. Został on zaproponowany przez Fausto Bernardini i pozostałych w 1999 roku [39]. BPA został stworzony by szybko i dokładnie odwzorowywać kształt powierzchni. Schemat postępowania programu, opiera się o kulę, która toczy się po powierzchni punktów. Z początku, wybierany jest pierwszy trójkąt triangulacyjny. Trzy punkty utworzą pierwszy trójkąt triangulacyjny, jeśli kula styka się tylko i wyłącznie z nimi. Gdy tocąc się, kula natrafi na dodatkowy punkt, to zostaje on połączony z bokiem wcześniejszego trójkąta. Cały proces powtarza się do momentu, gdy kula pokona całą powierzchnię. Pomimo przejścia kuli przez całą powierzchnię, nie oznacza to, że ze wszystkich punktów zostanie utworzony mesh.

Warto nadmienić, że głównym parametrem wpływającym na czas trwania algorytmu jest promień kuli. W większości przypadków, ustala się go na podstawie średniej odległości punktów od siebie. Jeśli promień będzie niedostatecznie duży, to kula może "wypaść" przez punkty, więc nie zostaną one połączone. Powstają wtedy dziury, które negatywnie wpływają na ostateczny wygląd meshu. Można je później załatać wykorzystując algorytmy interpolacji liniowej lub ponownie uruchomić algorytm BPA, tym razem z większym promieniem i powtórzyć cały proces jeszcze raz. W efekcie czego większość dziur zostanie załatwiona. Wpłynie to jednak na znaczne zwiększenie czasu trwania algorytmu. Wizualizacja zasady działania algorytmu została przedstawiona na rysunku 3.4



**Rysunek 3.4:** Zasada działania algorytmu BPA [39].

Główną zaletą tego algorytmu jest fakt, że jego złożoność obliczeniowa zależy liniowo od liczby punktów w zbiorze. Jeśli dodany zostanie dodatkowy punkt, to kula musi się przez niego przetoczyć kilka razy. Jednak cała struktura pozostałych punktów pozostaje niezmienna. Ten aspekt algorytmu stanowi największą odmianę względem triangulacją Delaunay'a. W triangulacji Delaunay'a, po dodaniu dodatkowego punktu cała struktura triangulacji może się zmienić. Ważną zaletą algorytmu jest możliwość regulacji promienia kuli  $R$ , im promień jest mniejszy, tym algorytm zajmie więcej czasu, ponieważ zetknie się z mniejszą ilością punktów podczas toczenia. Zatem eksperymentując z długością promienia, można otrzymać zadowalająco dobre rezultaty w krótszym czasie.

### 3.3.4. Podsumowanie

Wszystkie wyżej opisane metody generacji siatki z chmury punktów dają bardzo dobre rezultaty. Jednak do dalszych badań wybrano metodę BPA oraz triangulację Delaunay'a. W przypadku algorytmu toczącej się kuli, przedstawione zostaną kroki potrzebne do jego implementacji z biblioteki Open3D. Przy triangulacji Delaunay'a omówiona zostanie szczegółowa implementacja oraz optymalizacja autorskiego programu. Dla obu powyższych metod przedstawiono porównanie wyników oraz jakości wygenerowanej siatki.

## 4. IMPLEMENTACJA ORAZ PORÓWNANIE METOD

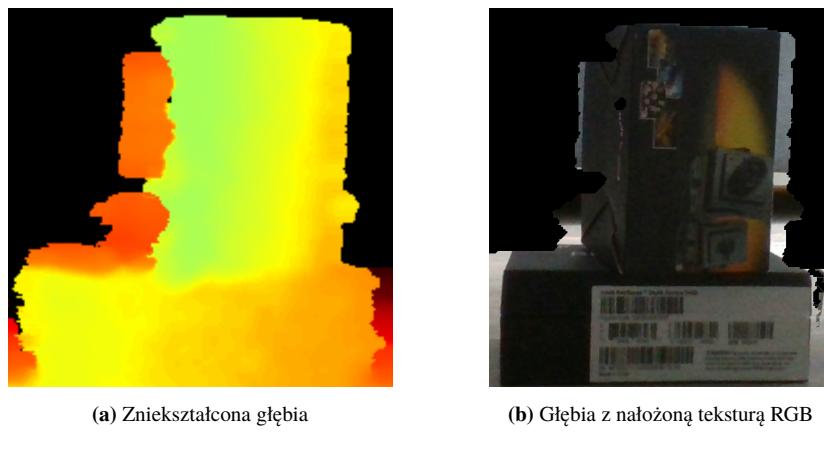
W bieżącym rozdziale przedstawione zostaną sposoby użycia dwóch metod rekonstrukcji powierzchni z chmury punktów. Omówione zostaną charakterystyki poszczególnych algorytmów oraz rezultaty dzięki nim uzyskane. Ukazana zostanie implementacja wybranego z nich. W celu uzyskania trójwymiarowych modeli na podstawie skanów rzeczywistych obiektów, przetestowano dwie metody generacji meshu. Pierwszą z nich jest algorytm tocżącej się kuli. Kolejnym algorymem jest trójwymiarowa triangulacja Delaunay'a wraz ze sposobami jej optymalizacji.

### 4.1. Ogólny zarys funkcjonowania programu

Autorski program, mający na celu przetworzenie danych ze skanera 3D w celu uzyskania modeli trójwymiarowych składa się z wielu istotnych kroków. Począwszy od wczytania danych z kamery trójwymiarowej, a skończywszy na eksportowaniu modeli do programu Blender. Istnieje wiele różnych podejść do implementacji poniższych algorytmów. Poniżej zostaną omówione autorskie metody na ich implementację oraz optymalizację. W rozdziale zostaną opisane najważniejsze algorytmy oraz ich szczegółowe charakterystyki. Program zawiera graficzny interfejs, który ma umożliwić użytkownikom łatwą obróbkę danych.

#### 4.1.1. Odczytanie danych z kamery RGBD

Pierwszym krokiem pracy programu jest odczytanie danych z kamery trójwymiarowej. Wczytane dane są poddawane różnym obróbkom w celu usunięcia przekłamań spowodowanych złymi odczytami z czujnika głębi. Głównym tego powodem jest rozpraszanie wiązki światła na obiekcie, co uniemożliwia poprawny odczyt głębi przez skaner. Przykład takiej sytuacji został przedstawiony na rysunku 4.1.



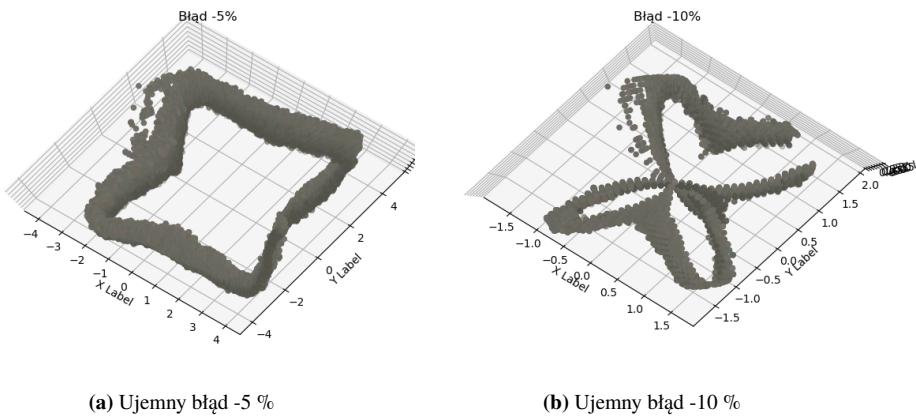
Rysunek 4.1: Porównanie widoku kamery głębi oraz RGB

Biblioteka pyrealsense dostarczana przez firmę Intel do kamer trójwymiarowych zapewnia szereg funkcji pozwalających na oczyszczenie wynikowych danych z dziur. W pakiecie SDK zawarte są filtry chwilowe oraz przestrzenne. Filtr chwilowy umożliwia wypełnienie dziur wartościami z poprzednich klatek. Jest to funkcja, która pozwala na uzyskanie bardziej czytelnych danych. Są one również wiarygodniejsze niż użycie interpolacji do uzyskania danych. Filtr przestrzenny umożliwia wygładzenie danych zachowując jednocześnie dane brzegowe. W przypadku wcześniejszego filtru te dane są tracone w skutek mieszania danych z aktualnych oraz wcześniejszych klatek. Przydatna okazuje się również być funkcją do nakładania na siebie obrazu głębi oraz koloru. Czujnik natężenia siatki światła oraz kamera RGB znajdują się obok siebie, dlatego też piksel o współrzędnych (x,y) będzie różnił się w przypadku koloru oraz głębi [31]. By uzyskać poprawne dane trzeba nałożyć odpowiednio klatki pochodzące z kamery RGB oraz czujnika głębi. Zostało to dokonane poprzez wykorzystanie odpowiedniej funkcji z biblioteki pyrealsense. Możliwy jest wybór wzgldem, którego źródła mają być nakładane na siebie klatki. Na potrzeby programu klatki nakładane są na siebie wzgldem ujęcia RGB.

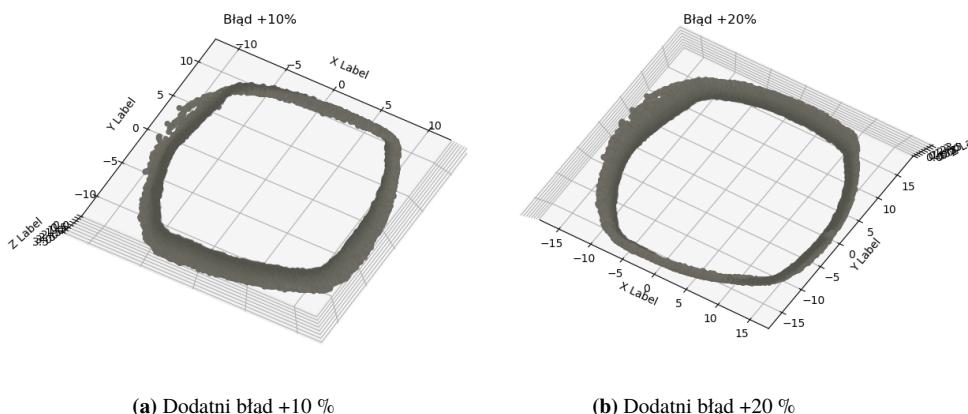
Następnym krokiem jest ekstrakcja danych głębi oraz koloru ze wszystkich klatek obrazu. Porównano dwie metody akwizycji danych. W metodzie skanera liniowego brana pod uwagę jest pojedyncza kolumna z całego obrazu, zaś w metodzie światła strukturalnego przetwarzane są dane głębi pochodzące z całego obrazu. Po przeprowadzeniu należytych badań oraz analizy wyników, powstały w skutek implementacji obu metod generacji chmur punktów, do dalszego ciągu pracy wybrano metodę skanera liniowego. Zbiór kolumn powstały dzięki zastosowaniu tej metody posłuży w późniejszym etapie do przejścia w chmurę punktów. Wszystkie te informacje zapisywane są w pliku z rozszerzeniem npy. Ten format pliku używany jest przez bibliotekę Numpy do wykonywania szybkich zapisów binarnych danych. Dzięki temu zabiegowi, uzyskane dane z nagrania zajmują jedynie kilkanaście kB. Ten zabieg ma na celu późniejszy odczyt danych bezpośrednio z zapisanego pliku. Dzięki temu nie będzie konieczności ładowania ponownie pełnego filmu z kamery RGBD. Przekształcenie otrzymanych danych zostanie przedstawione w dalszym ciągu pracy.

#### 4.1.2. Przejście do chmury punktów

Kolejnym istotnym krokiem algorytmu jest przejście do chmury punktów. Polega on na wyznaczeniu dla każdego dwuwymiarowego punktu, odpowiadających mu współrzędnych w przestrzeni trójwymiarowej. Przed przystąpieniem do dalszej pracy należy jednak zmierzyć odległość środka tacki, na której stał obiekt, od kamery. Jeśli ta odległość zostanie źle zmierzona, uzyskany model będzie błędny. Zostało sporządzone porównanie wpływu błędnego pomiaru na ostateczny wygląd modelu. Wszystkie modele zostały zrzutowane na płaszczyznę dwuwymiarową, ponieważ dokonywanie przekształcenie nie wpływa na współrzędną Z danych punktów. Początkowa odległość kamery od środka tacki wynosiła 50 cm.



**Rysunek 4.2:** Porównanie wpływu ujemnego błędu pomiarowego na wygląd chmury punktów dla -5 %, -10 %.



**Rysunek 4.3:** Porównanie wpływu dodatniego błędu pomiarowego na wygląd chmury punktów dla +10 %, +20 %.

Na rysunkach 4.2 oraz 4.3 można zaobserwować jak duży wpływ na ostateczny wygląd modelu ma błędne zmierzenie odległości. Jeśli stosunek błędu do rzeczywistej odległości jest większy od 1, wtedy kształt modelu przybiera

formę okręgu. Jeśli zaś jest mniejszy od 1, obwód modelu zaczyna być wklesty. Przyczyną takiego zachowania algorytmu jest wzór wykorzystywany do przejścia ze współrzędnych obiektu do współrzędnych kamery (3.2). Ze wzoru można wywnioskować, że gdy stosunek błędnie zmierzonej odległości od środka jest o wiele większy od rzeczywistej odległości równania zaczynają opisywać współrzędne okręgu. Z kolei, przy ujemnym błędzie pomiarowym, dla  $R_b << R$ , promień  $R - D_\beta$  może przybierać ujemne wartości, tworząc tym samym wstęgi.

Kolejnym krokiem jest wyznaczenie rzeczywistej wysokości obiektu na podstawie wzoru (3.1). Równanie pozwala na obliczenie wysokości obiektu, znając odległość od niego oraz jego wymiary w pikselach. Znając wysokość obiektu jest możliwe ustalenie jaka powinna być gęstość punktów wzdłuż osi Z. Dzięki takiemu podejściu wymiary obiektu zostaną wiernie odwzorowane, a co za tym idzie, będzie można sprawdzić prawdziwe charakterystyki modelu. Mając informacje o rzeczywistych wymiarach, można ustalić objętość danego obiektu oraz jego powierzchnię w  $m^2$ .

Mając poprawnie wyznaczoną odległość od obiektu można przejść do wyznaczenia chmury punktów korzystając z równania 3.2. Do otrzymanej chmury punktów należy również dodać kolory poszczególnych pikseli uzyskane na podstawie obrazu RGB. Na potrzeby użytej metody zapisu trójwymiarowych modeli, piksele muszą zostać przeskalowane z wartości <0,255> do wartości z zakresu <0,1> dla każdego z trzech kolorów, czerwonego, zielonego oraz niebieskiego.

#### *4.1.3. Oznaczanie błędnych próbek*

Istotnym aspektem działania algorytmu jest eliminacja błędnych próbek. Jest to proces mający na celu znalezienie oraz oznaczenie przekłamanych punktów, które mogą negatywnie wpływać na ostateczny wygląd modelu. Składa się ona z kilku faz.

Pierwszym krokiem tej metody jest wyznaczenie średniej odległości wszystkich punktów od środka obiektu. Jest ona wyznaczana na podstawie poniższego wzoru.

$$D_{mean} = \frac{\sum_{n=0}^N \sqrt{p_{nx}^2 + p_{ny}^2}}{N} \quad (4.1)$$

Punkt P o indeksie n posiada współrzędne  $p_{nx}, p_{ny}$ . Zbiór na którym dokonywania jest eliminacja posiada N punktów. Średnia odległość punktu od środka układu wynosi D.

Warto zauważyć, że we wzorze nie występuje współrzędna Z n-tego punktu. Powodem tego jest fakt, iż punkty są równomiernie rozłożone wzdłuż osi Z. Liczenie odległości wzdłuż tej osi nie wpłynęło by na poprawienie dokładności wyznaczania odpowiednich punktów. Można więc ten etap pominąć i wykorzystać jedynie dwie pierwsze współrzędne punktu do wyznaczenia jego pozycji względem środka.

Dla każdego punktu ze zbioru sprawdzana jest odległość od środka. Jeśli jest ona większa, niż  $p\% \cdot D_{mean}$ , gdzie  $p\%$  oznacza graniczny współczynnik odległości od środka, to punkt zostaje oznaczony odpowiednią flagą. Flaga umożliwia w późniejszym etapie rozpoznanie przeklamanego punktu oraz dodanie go do zbioru interpolacyjnego.

#### *4.1.4. Interpolacja punktów*

Po tym jak wszystkie przeklamane punkty zostały odpowiednio oznaczone można przejść do ich interpolacji. Wykorzystując technikę interpolacji można wyznaczyć współrzędne tych punktów, których wartości początkowo były błędne. Podczas badań otrzymanych rezultatów interpolacji, został wybrany najlepszy algorytm. Spośród interpolacji liniowej, sześciennej oraz najbliższych sąsiadów wybrana została metoda sześcienna. Daje ona najlepsze rezultaty spośród wszystkich wypróbowanych metod. Punkty zostały przekształcone za pomocą (3.6). Wynikiem takiej operacji jest nowa chmura punktów, która nie zawiera już przekłamanych wartości. Dzięki temu możliwe będzie dokładne utworzenie siatki triangulacyjnej na modelu. Korzystając z danej metody interpolacji uzyskane wartości mają bardziej naturalny rozkład. W przypadku jej liniowej odmiany, współrzędne punktów rosły liniowo. Sprawiało to wrażenie, że punkty na obiekcie były ułożone błędnie. Po zmianie na interpolację wielomianem trzeciego stopnia, ten problem jest niezauważalny.

### **4.2. Porównanie metod rekonstrukcji powierzchni**

W poniższej sekcji przedstawione zostały wyniki implementacji dwóch metod rekonstrukcji powierzchni. Dokonano porównania dokładności odwzorowania powierzchni jak również ich szczegółowości. W algorytmie BPA

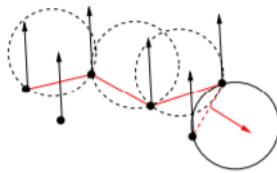
porównano wpływ promienia kuli na ostateczny wygląd modelu oraz czas trwania algorytmu. Dla triangulacji Delaunay'a zostanie przedstawiona szczegółowa implementacja oraz użyte metody optymalizacji. Ukażany zostanie również wpływ ilości punktów na czas trwania triangulacji.

#### 4.2.1. Algorytm BPA

Dla porównania został zaimplementowany algorytm BPA. Jego użycie jest możliwe dzięki wykorzystaniu biblioteki Open3D. Pakiet umożliwia dokonywanie wielu operacji na chmurach punktów. Dostępne są w nim rekonstrukcje Poissona, wyznaczanie wektorów normalnych oraz dane testowe do przeprowadzania badań dokładności implementowanych algorytmów. Na potrzeby wdrożenia danej metody wykorzystana zostanie funkcja umożliwiająca stworzenie meshu przy wykorzystaniu algorytmu BPA. Implementacja algorytmu składa się z kilku kroków które zostały omówione poniżej.

#### Estymacja normalnych

Główną ideą algorytmu jest toczenie kuli po chmurze punktów. W przypadku kiedy gęstość 3 dotkniętych punktów jest większa od średniej gęstości to utworzony w ten sposób trójkąt będzie mniejszy. W przypadku, gdy lokalna gęstość jest jednak mniejsza niż średnia, kula może dotknąć punktu, który jest ukryty. Znajduje się on wtedy pod górną warstwą i w celu dokładnego odwzorowania powierzchni, tak utworzony trójkąt powinien zostać odrzucony [40]. Dana sytuacja została opisana na rysunku 4.4.



**Rysunek 4.4:** Dotknięcie przez kule punktu pod powierzchnią [40].

W celu wykrycia poprawnie skonstruowanego trójkąta używane są normalne. W tym celu należy obliczyć wektory normalne do powierzchni danego trójkąta oraz do średnie wektory normalne do powierzchni blisko danego trójkąta [41].

$$\begin{cases} N_{tri} = \vec{AB} \times \vec{AC} \\ N_{mean} = \frac{\sum_{i=1}^K \vec{A_iB_i} \times \vec{A_iC_i}}{K} \\ \theta = \arccos\left(\frac{N_{tri} \cdot N_{mean}}{\|N_{tri}\| \|N_{mean}\|}\right) \end{cases} \quad (4.2)$$

Wektorem normalnym do powierzchni trójkąta jest  $N_{tri}$ . Wektory  $\vec{AB}$  oraz  $\vec{AC}$  są skonstruowane z wierzchołków A, B, C trójkąta triangulacyjnego. Wokół trójkąta triangulacyjnego znajdują się sąsiednie trójkąty w liczbie K, tworzące uśredniony wektor normalny  $N_{mean}$ . Kąt pomiędzy wektorem normalnym do trójkąta, a do powierzchni wokół niego wynosi  $\theta$ .

Trójkąt interpolacyjny należy odrzucić, gdy iloczyn skalarny jego wektora normalnego oraz normalnej do powierzchni jest ujemny. Oznacza to, że kąt  $\theta$  pomiędzy nimi będzie większy od  $\frac{\pi}{2}$ . Dla wartości powyżej 90 stopni trójkąt będzie skierowany przeciwnie do powierzchni i należy go odrzucić. Do wyznaczenia wektorów normalnych do chmury punktów użyta została funkcja dostępna w bibliotece open3D. Znajduje ona punkty w okolicy danego punktu i przecinając je prostą tworzy wektor normalny. Następnie wektory normalne do punktu są uśredniane i wyznaczane za pomocą analizy kowariancji.

#### Wyznaczenie średniej odległości punktów

Kolejnym krokiem niezbędnym do poprawnego funkcjonowania algorytmu BPA jest odpowiedni dobór promienia kuli R. Jest to bardzo istotny parametr algorytmu. Odpowiedni jego dobór sprawi, że wyjściowy model

trójwymiarowy obiektu będzie szczególny i będzie zawierał mało luk. Jednakże wraz ze wzrostem długości promienia rośnie również czas obliczeniowy programu. Promień powinien być więc dobierany z uwzględnieniem obu tych aspektów. W celu jego najlepszego doboru należy wyznaczyć średnią odległość punktów od siebie. Znając tę odległość, będzie można odpowiednio dobrą długość  $R$ , by kula nie wpadała w przestrzeń pomiędzy punktami. Za obliczenie średnich odległości odpowiada funkcja z biblioteki open3D. Dla każdego punktu w chmurze wyznacza jest odległość od najbliższego sąsiada. Z tak powstałego wektora odległości można wyznaczyć średnią korzystając z biblioteki Numpy w celu optymalizacji czasu obliczeniowego. Następnie, traktując średnią odległość pomiędzy obiektami jako punkt wyjściowy, można empirycznie dobrą odpowiednią wartość promienia kuli  $R$ . W tabeli 4.1 został przedstawiony wpływ wielkości promienia  $R$  na czas trwania obliczeń oraz ilość trójkątów.

**Tabela 4.1:** Czas trwania BPA w zależności od promienia  $R$ . Średnia odległość między punktami  $D_{mean} = 0.047$ .

| Długość promienia | $\frac{R}{D_{mean}}$ | Czas trwania | Liczba trójkątów |
|-------------------|----------------------|--------------|------------------|
| 0.047             | 1                    | 0.51 s       | 42819            |
| 0.09              | 2                    | 1.44 s       | 44025            |
| 0.14              | 3                    | 7.5 s        | 32474            |
| 0.19              | 4                    | 29.8 s       | 24936            |
| 0.23              | 5                    | 93 s         | 21496            |
| 0.28              | 6                    | 215 s        | 18641            |
| 0.33              | 7                    | 430 s        | 16348            |
| 0.38              | 8                    | 737s         | 14457            |

W powyższej tabeli przedstawiony został wpływ wielkości promienia na czas obliczeń oraz liczbę trójkątów. Z danych zebranych podczas testów można wysnuć wniosek, iż wraz ze wzrostem długości promienia kuli, rośnie czas trwania algorytmu. Maleje przy tym ilość utworzonych trójkątów. Podstawa takiego zachowania jest fakt, iż przy większym promieniu kula dotyka mniejszej ilości punktów, tym samym tworząc mniej trójkątów. Jednocześnie czas trwania algorytmu rośnie, ponieważ badanie sąsiedztwa punktów zajmuje więcej czasu. Przy większym promieniu kuli, poszukiwana jest liczba sąsiadów danego punktu w większym otoczeniu. To znaczco wpływa na złożoność obliczeniową.

Po wyznaczeniu normalnych oraz dobraniu promienia kuli ostatnim krokiem jest utworzenie triangulacji BPA. Wykorzystując funkcję z biblioteki open3D można przeprowadzić algorytm toczący się kuli i uzyskać ściany trójkątów triangulacyjnych. Funkcja dostępna w bibliotece jest udoskonaleniem początkowej metody zaproponowanej przez Bernardini w 1999 roku. Wprowadziła ona możliwość równoległego przeprowadzania obliczeń w celu ich szybszego wykonywania [42]. By z większą prędkością odnajdywać sąsiadujące punkty użyto również drzewa ósemkowego. Jest to struktura danych umożliwiająca podzielenie przestrzeni na mniejsze części, a następnie utworzenie drzewa sąsiedztw. Znając strukturę drzewa, można je przeszukiwać w bardzo łatwy sposób, by odnaleźć odpowiednich sąsiadów punktu. Tworzenie tej struktury ma charakter rekurencyjny. Początkowo generowany jest sześcian otaczający wszystkie punkty, następnie jest on dzielony na 4 sześciany. Każdy kolejny na kolejne 4 rekurencyjnie. Podział trwa do momentu, kiedy wewnętrz sześcienu znajduje się nie więcej niż 8 punktów. Każdy korzeń w drzewie ma 8 gałęzi. Współrzędne każdego punktu w drzewie można zapisać binarnie. Znając pozycję danego punktu w drzewie, korzystając ze wzoru oraz przesunięć binarnych można odnaleźć jego sąsiadów. Dzięki tym operacjom przeprowadzony zostanie algorytm BPA znacznie szybciej, niż jego początkowa implementacja z 1999 roku. W ten sposób powstały model może wciąż zawierać luki, wynikające z nierównomiernej gęstości punktów. Na koniec otrzymane trójkąty triangulacyjne można wyeksportować do pliku obsługiwanej przez program Blender. W tym programie można dodać ręcznie płaszczyznę by zakryć luki powstałe w skutek przeprowadzenia algorytmu BPA.

#### 4.2.2. Triangulacja Delaunay'a

W celu implementacji trójwymiarowej metody triangulacji Delaunay'a zastosowano algorytm Bowyer-Watson [43]. Przedstawia on sposób na dołączanie dodatkowych punktów do triangulacji oraz sposób na jej utworzenie. Przy początkowej implementacji algorytmu, czas obliczeń był zbyt długi. Ilość punktów występująca przy triangulacji rzeczywistych danych pomiarowych sięga około 20000. Potrzebna była metoda szybszego obliczania triangulacji oraz

generacji meshu. Dokonano wielu kluczowych optymalizacji mających na celu minimalizację czasu trwania programu. Początkowe czasy trwania obliczeń zostały przedstawione w tabeli 4.2.

**Tabela 4.2:** Czas trwania triangulacji przed optymalizacją.

| Liczba punktów | Czas    |
|----------------|---------|
| 1000 pkt       | 10.85 s |
| 5000 pkt       | 246 s   |
| 10000 pkt      | 915 s   |

W powyższej tabeli można dostrzec, iż czas triangulacji przekracza początkowe założenia. Maksymalna zbadana ilość punktów testowych wynosi 10000. Czas triangulacji takiej ilości punktów powinien być znacznie niższy. Poprawiono, więc metody użyte w algorytmie, by zwiększyć jego wydajność. Głównym aspektem algorytmu, który potrzebował najwięcej czasu obliczeniowego jest wyznaczenie przynależności punktu do sfery opisanej na czworościanie. Dokonano porównania dwóch metod wyznaczania środka sfery opisanej na czworościanie.

### Wykorzystanie iloczynu skalarnego oraz wektorowego

Pierwsza metoda oparta jest o kombinację iloczynu wektorowego oraz skalarnego w celu uzyskania współrzędnych środka okręgu [44]. Wykorzystując podstawowe przekształcenia geometryczne, na podstawie współrzędnych wierzchołków możliwe jest określenie współrzędnych barycentrum. Równanie przedstawiające wzór na środek sfery opisanej na ostrosłupie zostało przedstawione poniżej.

$$\left\{ \begin{array}{l} \vec{u}_i = v_i - v_0, i = 1, 2, 3 \\ \vec{p} = O - v_0 \\ R^2 = \|p^2\| = \|u_1 - p\|^2 = \|u_2 - p\|^2 = \|u_3 - p\|^2 \\ O = v_0 + \frac{l_{01}^2(u_2 \times u_3) + l_{02}^2(u_3 \times u_1) + l_{03}^2(u_1 \times u_2)}{2u_1 \cdot (u_2 \times u_3)} \end{array} \right. \quad (4.3)$$

Czworościan dla którego ma zostać wyznaczona sfera na nim opisana posiada wierzchołki  $v_0 \dots v_3$ . Wektor przejścia z wierzchołka  $v_0$  do pozostałych wierzchołków  $v_i$  został oznaczony poprzez  $\vec{u}_i$ . Sfera opisana na ostrosłupie posiada środek w punkcie  $O$  oraz promień  $R$ . Wektor przejścia z wierzchołka  $v_0$  do środka sfery oznaczony jest symbolem  $\vec{p}$ . Długość krawędzi łączącej wierzchołek  $v_0$  oraz  $v_i$  wynosi  $l_{0i}$ .

W powyższych wzorach znajdują się 4 iloczyny wektorowe oraz jeden iloczyn skalarny. Wszystkie te operacje są bardzo kosztowne obliczeniowo. W tabeli 4.3 przedstawione zostały czasy triangulacji w zależności od ilości punktów. Pomiary są znacznie lepsze niż na początku, lecz wymagana jest dalsza ich optymalizacja.

**Tabela 4.3:** Czasy triangulacji dla pierwszej metody wyznaczania przynależności punktu do sfery.

| Liczba punktów | Czas   |
|----------------|--------|
| 1000 pkt       | 4.55 s |
| 5000 pkt       | 67 s   |
| 10000 pkt      | 194 s  |
| 15000 pkt      | 455 s  |

### Metoda wyznaczników macierzy

Druga metoda określania współrzędnych środka sfery opisanej na czworościanie polega na obliczeniu wyznaczników [45]. Z punktu widzenia złożoności obliczeniowej jest ona bardziej optymalna w porównaniu do pierwszej metody. Równania przedstawiające odpowiednie wyznaczniki macierzy oraz wzory pozwalające wyznaczyć współrzędne

środka sfery zostały przedstawione poniżej.

$$\begin{aligned}
 A &= \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix} \\
 D_x &= \begin{vmatrix} x_1^2 + y_1^2 + z_1^2 & y_1 & z_1 & 1 \\ x_2^2 + y_2^2 + z_2^2 & y_2 & z_2 & 1 \\ x_3^2 + y_3^2 + z_3^2 & y_3 & z_3 & 1 \\ x_4^2 + y_4^2 + z_4^2 & y_4 & z_4 & 1 \end{vmatrix} \\
 D_y &= - \begin{vmatrix} x_1^2 + y_1^2 + z_1^2 & x_1 & z_1 & 1 \\ x_2^2 + y_2^2 + z_2^2 & x_2 & z_2 & 1 \\ x_3^2 + y_3^2 + z_3^2 & x_3 & z_3 & 1 \\ x_4^2 + y_4^2 + z_4^2 & x_4 & z_4 & 1 \end{vmatrix} \\
 D_z &= \begin{vmatrix} x_1^2 + y_1^2 + z_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 + z_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 + z_3^2 & x_3 & y_3 & 1 \\ x_4^2 + y_4^2 + z_4^2 & x_4 & y_4 & 1 \end{vmatrix} \\
 C &= \begin{vmatrix} x_1^2 + y_1^2 + z_1^2 & x_1 & y_1 & z_1 \\ x_2^2 + y_2^2 + z_2^2 & x_2 & y_2 & z_2 \\ x_3^2 + y_3^2 + z_3^2 & x_3 & y_3 & z_3 \\ x_4^2 + y_4^2 + z_4^2 & x_4 & y_4 & z_4 \end{vmatrix} \\
 x_0 &= \frac{D_x}{2a} \\
 y_0 &= \frac{D_y}{2a} \\
 z_0 &= \frac{D_z}{2a} \\
 R &= \frac{\sqrt{D_x^2 + D_y^2 + D_z^2 - 4ac}}{2|a|}
 \end{aligned} \tag{4.4}$$

Wierzchołki ostrosłupa mają współrzędne  $x_i, y_i, z_i$  dla  $i=1, 2, 3, 4$ . Środek sfery o promieniu  $R$  opisanej na czworościanie opisują współrzędne  $x_0, y_0, z_0$ . Wyznaczniki czwartego stopnia  $D_x, D_y, D_z, A, C$  powstały w skutek przekształcenia ogólnego wzoru na środek sfery opisanej na ostrosłupie za pomocą metody Laplace'a.

Z powyższych równań wynika, że do obliczenia współrzędnych środka okręgu wystarczy określić 4 wyznaczniki macierzy. Jest to operacja o wiele szybsza niż liczenie iloczynów wektorowych i skalarnych. W celu większej optymalizacji promień sfery został obliczony jako odległość wierzchołka od środka okręgu. W porównaniu do obliczania kolejnych dwóch wyznaczników macierzy  $A$  oraz  $C$ , jest to szybsza operacja. Zestawienie czasu trwania algorytmów dla obu wymienionych metod zostało przedstawione w tabeli 4.4.

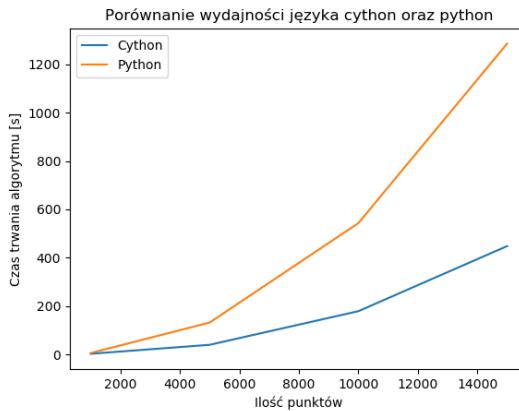
**Tabela 4.4:** Czasy triangulacji dla pierwszej oraz drugiej metody wyznaczania przynależności punktu do sfery.

| Liczba punktów | Czas metody iloczynowej | Czas metody wyznaczników |
|----------------|-------------------------|--------------------------|
| 1000 pkt       | 4.55 s                  | 2.97 s                   |
| 5000 pkt       | 67 s                    | 40 s                     |
| 10000 pkt      | 194 s                   | 179 s                    |
| 15000 pkt      | 455 s                   | 448 s                    |

Dane przedstawione w powyższej tabeli świadczą o wyższości drugiej metody nad pierwszą. Oba algorytmy zawierają przekształcenia macierzowe. W metodzie iloczynowej zawarte są 4 iloczyny wektorowe oraz jeden iloczyn

skalarnej. W metodzie wyznacznikowej obliczane są 4 wyznaczniki macierzy. Widać, iż w drugim przypadku obliczenia okazały się być mniej złożone. Średni spadek czasu trwania programu wynosił 30% względem pierwszej metody. Z powyższej tabeli wynika również, że wraz ze wzrostem ilości punktów maleje różnica pomiędzy tymi dwoma metodami. Wynika to ze stałej różnicy kosztu obliczeniowego dla pierwszej oraz drugiej metody. Wiele czynników wpływa na wydłużenie czasu trwania algorytmu. Podstawowym parametrem jest ilość punktów. Jako, że różnica między dwoma metodami jest stała, a cały algorytm jest zależny od objętości chmury punktów, to różnica pomiędzy nimi spada wraz ze wzrostem ilości punktów.

Kolejną znaczącą optymalizacją dokonaną w programie jest przejście z języka python do cython. Oba te języki programowania posiadają bardzo podobną strukturę, jednakże znacząco różnią się one sposobem wykonywania programu. Python jest językiem interpretowanym. Program punkt po punkcie uruchamia każdą instrukcję i wykonuje ją w locie. Dzięki tej możliwości, nawet jeśli w późniejszym etapie programu jest błąd, zostanie on wykryty dopiero w momencie wykonania instrukcji. Z kolei cython jest językiem komplikowanym. Każdy plik wchodzący w skład programu jest komplikowany do języka C. Szereg instrukcji dla programu jest odgórnie wyznaczony, a wszystkie operacje mogą zostać wcześniej przekształcone do języka maszynowego. Dzięki takiej operacji wydajność programu znacznie się zwiększyła. W celu sprawdzenia wzrostu wydajności dzięki zastosowaniu języka cython dokonano porównania. Zaimplementowano dwa jednakowe algorytmy, jeden napisany w języku python, drugi zaś w cython. Wyniki pomiarów znajdują się w tabeli 4.5, zaś ich graficzna reprezentacja na rysunku 4.15.



**Rysunek 4.5:** Porównanie czasu algorytmu dla cython oraz python.

**Tabela 4.5:** Porównanie czasu trwania algorytmu dla cython oraz python.

| Liczba punktów | Czas cython | Czas python |
|----------------|-------------|-------------|
| 1000 pkt       | 2.97 s      | 6 s         |
| 5000 pkt       | 40 s        | 132 s       |
| 10000 pkt      | 179 s       | 543 s       |
| 15000 pkt      | 448 s       | 1285 s      |

Na wykresie 4.15 oraz w tabeli 4.5 można zauważyć, że wstępna komplikacja kodu nie tylko wpłynęła procentowo na wydajność algorytmu. Sprawiła ona, iż tempo wzrostu złożoności obliczeniowej w zależności od ilości punktów również się zmniejszyło. Dzięki temu możliwe jest wykonanie triangulacji jeszcze większej ilości punktów. Zmniejszenie tempa wzrostu ma związek z optymalizacją pętli iteracyjnych w języku cython. Wraz ze wzrostem ilości punktów poddanych triangulacji, rośnie również ilość iteracji pętli. W języku python pętle wykonywane są wolniej niż w cython, dlatego też przyrost czasu obliczeniowego jest znacznie bardziej dostrzegalny.

Następnym elementem który wpływał pozytywnie na optymalizację czasu działania programu jest zmiana biblioteki. Dotychczas do obliczania wyznacznika macierzy używana była funkcja z biblioteki NumPy. Jest to biblioteka typu open-source. Dostęp do niej jest darmowy, a aktualizować ją może każdy kto zostanie zaakceptowany przez

społeczności. Dzięki pracy dużej liczby ludzi kod jest zoptymalizowany oraz uniwersalny. Funkcje dostępne w NumPy mają bardzo dużą wydajność pod względem pamięci i zasobów obliczeniowych. Biblioteka zawiera wiele różnych modułów począwszy od algebry liniowej, po równania różniczkowe oraz obliczenia na liczbach zespolonych. Z testów przeprowadzonych na programie do triangulacji wynika, że do obliczeń macierzowych najlepiej jednak się biblioteka SciPy. Ona też jest zbiorem programów typu open-source. Korzysta ona po części z NumPy jako struktury do przechowywania danych obliczeniowych. Obie biblioteki zawierają podobne funkcje, lecz do zastosowań projektowych wybrana została ta druga. Porównanie pracy programu, przy zastosowaniu funkcji z biblioteki NumPy oraz SciPy do obliczeń macierzowych zostało przedstawione w tabeli 4.6.

**Tabela 4.6:** Porównanie czasu trwania algorytmu przy obliczaniu macierzy przez NumPy oraz SciPy.

| Liczba punktów | Czas SciPy | Czas NumPy |
|----------------|------------|------------|
| 1000 pkt       | 2.97 s     | 2.48 s     |
| 5000 pkt       | 40 s       | 36 s       |
| 10000 pkt      | 179 s      | 231 s      |
| 15000 pkt      | 448 s      | 486 s      |

W powyższej tabeli został przedstawiony wpływ użycia różnych bibliotek na ostateczny czas trwania algorytmu triangulacji Delaunay'a. Dla mniejszej ilości punktów, biblioteka NumPy jest wydajniejsza. Jednak przy wzroście punktów, biblioteka SciPy osiągnęła lepsze wyniki, dlatego została wybrana jako docelowa metoda przeprowadzania obliczeń macierzowych.

Kolejnym aspektem algorytmu wymagającym znacznej mocy obliczeniowej jest badanie styczności czworościanów. W algorytmie, z każdej ściany ostrosłupa niestykającej się z inną ścianą jest budowany jest dodatkowy czworościan. W jego skład wchodzą trzy wierzchołki początkowej ściany oraz dodatkowy punkt triangulacyjny. Należy, więc znaleźć te ściany algorytmu, które nie stykają się z pozostałymi. Jest to zadanie wymagające dwóch pętli iteracyjnych i wykonywane jest dla każdego dodatkowego punktu. W przypadku autorskiego programu, zostało zastosowane kryterium przynależności zbiorów. Tablica ścian została utworzona poprzez kombinację 4 różnych wierzchołków. Dla każdego z pozostałych czworościanów sprawdzane jest czy którykolwiek z trzech wierzchołków znajduje się w zbiorze ścian początkowego ostrosłupa. Jeśli tak, to ściana ta jest dodawana do zbioru zajętych ścian i proces jest kontynuowany do momentu wyczerpania pozostałych ostrosłupów lub ścian. Zadanie to nie jest trywialne z punktu widzenia struktury danych. Matematyczne podstawy jego funkcjonowania nie są złożone, jednakże opis programowy tak. Wymagana jest struktura, która umożliwia przechowywanie nieuporządkowanej sieci wierzchołków. Dzieje się tak ze względu na to, że punkty mogą tworzyć ścianę w różnej kolejności, więc porządek ich występowania w danej ścianie może być różny dla dwóch osobnych ostrosłupów. Zostały sprawdzone dwie metody odnajdywania sąsiadujących ścian.

Pierwsza metoda zakładała iterację przez wszystkie wierzchołki początkowego oraz aktualnego czworościanu. Dzięki temu można było sprawdzić ile z nich występuje w obu tych zbiorach. To zadanie wymagało dużo czasu obliczeniowego, ponieważ korzystało z trzech pętli iteracyjnych. Pierwsza odpowiadała za wybór sąsiadniego ostrosłupa, druga za wybór ściany do sprawdzenia w aktualnym czworościanie, a trzecia za wybór ściany w drugim ostrosłupie.

Drugą metodą, która była możliwa dzięki zastosowaniu języka python były operacje na zbiorach. Struktura danych set() umożliwia przechowywanie nieuporządkowanych zbiorów danych i dokonywanie na nich operacji. Na potrzeby algorytmu triangulacyjnego zastosowano iloczyn dwóch zbiorów.

$$\left\{ \begin{array}{l} V_a = v_{a1}, v_{a2}, v_{a3}, v_{a4} \\ V_b = v_{b1}, v_{b2}, v_{b3}, v_{b4} \\ S_a = \binom{V_a}{3} \\ S_b = \binom{V_b}{3} \\ W = V_a \cap V_b \\ S_{bi} \subset S_a \Leftrightarrow |W| = 3 \\ S_f = S_a / S_{bi} \end{array} \right. \quad (4.5)$$

Ostrosłupy A oraz B są opisane wierzchołkami znajdującymi się odpowiednio w zbiorach  $V_a$  i  $V_b$ . Ściany obydwoch ostrosłupów znajdują się w zbiorach  $S_a$  oraz  $S_b$ . Wspólne wierzchołki ostrosłupów stanowią zbiór  $W$ , zaś zbiór  $S_f$  zawiera ściany wolne niestykające się z pozostałymi ostrosłupami.

Z powyższych równań widać, że jeśli moc zbioru będącego iloczynem zbiorów wierzchołków ostrosłupów będzie równa 3, to obie te figury będą posiadały wspólną ścianę. Należy ją wtedy dodać do zbioru współdzielonych ścian. Operacja zostanie powtórzona dla wszystkich pozostałych ostrosłupów. Ze ścian wolnych które pozostaną, zostaną zbudowane nowe ostrosłupy.

Na sam koniec, ze zbioru ostrosłupów triangulacyjnych należy usunąć te, które współdzielą krawędzi z początkowym super-ostrosłupem. Operację można szybko przeprowadzić, ponieważ znane są współrzędne wierzchołków początkowego czworościanu.

Powyższe operacje przedstawiają algorytm triangulacji Delaunay'a, dzięki któremu uzyskiwane są ostrosłupy. Do generacji meshu potrzebna jest jeszcze dodatkowa rzecz. Mianowicie, muszą zostać utworzone poszczególne ściany tych ostrosłupów. Są one generowane poprzez kombinację wierzchołków ostrosłupa. Następnie, współrzędne wierzchołków muszą zostać przekształcone na odpowiadające im indeksy w zbiorze punktów. W celu optymalizacji algorytmu każdy punkt posiada czwartą współrzędną. Jest nią indeks na którym znajduje się punkt w ogólnym zbiorze punktów. Umożliwia to w końcowym etapie szybkie przekształcenie ścian bryły do postaci indeksów.

Ostatnim etapem programu jest zapis utworzonego w ten sposób meshu do pliku. Wykonywane jest to za pomocą biblioteki Trimesh. Jako parametr podawany jest tam zbiór ścian ostrosłupa. Kolejnymi parametrami są kolory wierzchołków oraz ich współrzędne. Tak wygenerowany plik może zostać wyeksportowany i otwarty w programie Blender. Poprzez wykorzystanie programu Blender, możliwe jest dokonywanie różnych operacji na wygenerowanym modelu. Aplikacja zawiera szereg funkcji, umożliwiających wykonywanie różnych przekształceń na obiekcie. Dodatkowo można w sposób ręczny usunąć błędnie wygenerowane trójkąty triangulacyjne, udoskonalaając w ten sposób przygotowany model trójwymiarowy.

### 4.3. Wyniki

Powyższe algorytmy miały na celu jak najdokładniej odwzorować obiekt na podstawie danych z kamery głębi. W poniższej sekcji omówione zostaną wyniki użytych metod akwizycji danych. Ukażane zostaną modele utworzone poprzez wykorzystanie algorytmu BPA oraz triangulacji Delaunay'a. Przedstawiona zostanie dokładność odwzorowania powierzchni przez obie te metody. W celu testów wykonano dwa skany przedmiotów. Dokonano przekształcenia na model 3D jabłka oraz pudełka. Oba te przedmioty mają swoje barwy, które również zostały oddane w modelach.

#### 4.3.1. Wyniki metody światła strukturalnego

Po przeprowadzeniu wszystkich operacji niezbędnych do zespolenia ze sobą chmur punktów powstały w skutek użycia metody światła strukturalnego należy nałożyć na nie siatkę. W tym celu zostanie użyta triangulacja Delaunay'a oraz algorytm BPA. Przeprowadzono porównanie wpływu promienia toczącej się kuli na ostateczny wygląd wygenerowanego meshu. Dla triangulacji sprawdzono dokładność odwzorowania ścian oraz kolorów badanych modeli. Dokonano porównania wpływu częstotliwości próbkowania klatek na ostateczny wygląd zbioru punktów dla kąta równego  $11.25^\circ$ ,  $45^\circ$ . Początkowe chmury punktów, które zostały poddane obróbce zostały przedstawione na rysunkach 4.6 oraz 4.7.



(a) Próbkowanie  $45^\circ$

(b) Próbkowanie  $11.25^\circ$

**Rysunek 4.6:** Początkowe chmury punktów jabłka, próbkowane co  $45^\circ$  i  $11.25^\circ$ .



(a) Próbkowanie  $45^\circ$

(b) Próbkowanie  $11.25^\circ$

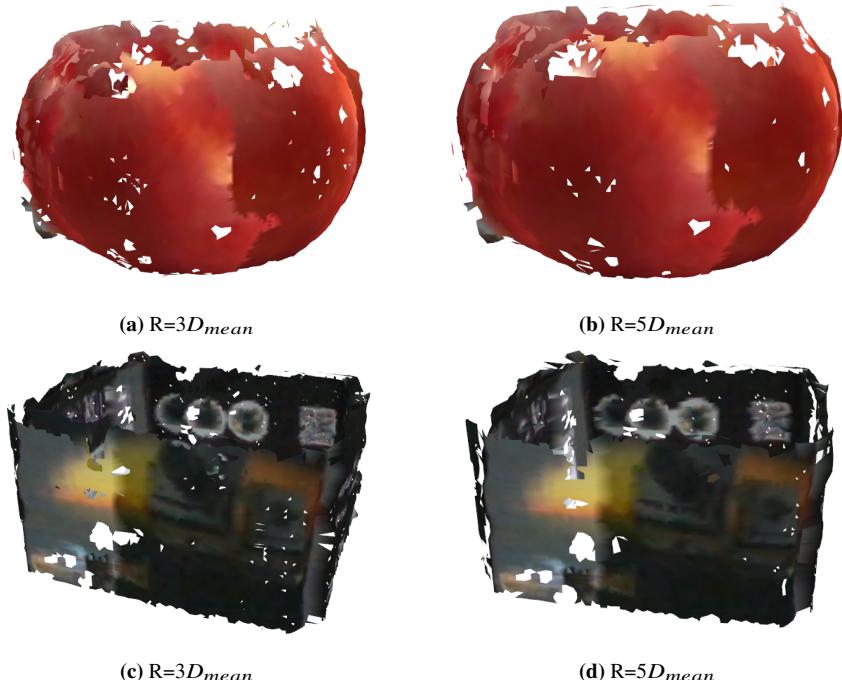
**Rysunek 4.7:** Początkowe chmury punktów pudełka, próbowane co  $45^\circ$  i  $11.25^\circ$ .

Na powyższych rysunkach można zaobserwować, że występują drobne niedoskonałości wynikające z błędного odczytu danych głębi oraz koloru. Widać to zwłaszcza przy chmurze punktów jabłka. Po boku chmury znajduje się wycinek tła, który powinien mieć większą odległość i zostać usunięty. By tego uniknąć można odfiltrować punkty pod względem koloru. Taki zabieg jednak zmniejszyłby uniwersalność danego rozwiązania.

Ponadto z powyższych chmur wynika obserwacja, że wraz ze wzrostem częstotliwości próbkowania powierzchnia chmury zaczyna być rozmażana. Powodem takiego zachowania są nakładające się na siebie punkty, które są nieznacznie przesunięte względem siebie. Problem ten występuje dla obu chmur i pogłębia się wraz ze wzrostem liczby punktów poddanych obróbce. Dla małej liczby punktów, problem jest niemalże niedostrzegalny. Jednakże, przy próbkowaniu co  $11.25^\circ$  ostateczna chmura zawiera ponad 40000 punktów i ten mankament staje się bardziej widoczny.

Zwiększenie ilości punktów estymacyjnych w metodzie RANSAC wpływa na poprawę wyglądu chmury, jednak rośnie przy tym złożoność obliczeniowa. By osiągnąć najlepsze efekty każdy z parametrów należy dobierać ręcznie w zależności od częstotliwości próbkowania i typu obiektu. Do dalszych analiz wybrano chmurę punktów, dla której próbki wykonywano co  $45^\circ$ .

W celu otrzymania siatki triangulacyjnej użycie algorytmu BPA z biblioteki Open3D. Liczba punktów w chmurze wynosiła 27000 oraz 40000, odpowiednio dla jabłka oraz pudełka. Wyniki algorytmu dla długości promienia równych  $R=3D_{mean}$  oraz  $R=5D_{mean}$ , gdzie  $D_{mean}$  jest średnią odlegością punktów od siebie, zostały przedstawione na rysunku 4.8.



(a)  $R=3D_{mean}$

(b)  $R=5D_{mean}$

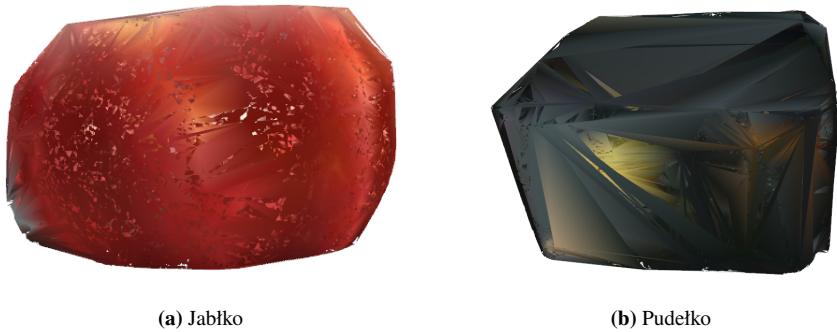
(c)  $R=3D_{mean}$

(d)  $R=5D_{mean}$

**Rysunek 4.8:** Siatka utworzona za pomocą algorytmu BPA jabłka oraz pudełka dla promieni  $R=3D_{mean}$  oraz  $R=5D_{mean}$

Na powyższych rysunkach przedstawiono wpływ długości promienia na ostateczny wygląd siatki wygenerowanej przez algorytm tocżącej się kuli. Otrzymane rezultaty nie są jednak zadowalające. Pomimo znacznego zwiększenia liczby oraz gęstości punktów, wyniki znacznie odbiegają od tych otrzymanych przy metodzie skanera liniowego. Ponadto zwiększenie promienia kuli z  $3D_{mean}$  do  $5D_{mean}$  nie zmniejszyło liczby dziur w modelu. Wygląd obiektu zależy w znacznym stopniu od ułożenia punktów. W miejscach, w których się ze sobą pokrywają powstaje gładka powierzchnia. Jednak na łączeniach oraz górze oraz na dole punkty są gorzej dopasowane. Podstawą tego jest działanie algorytmu RANSAC, który ma na celu minimalizację błędu pomiędzy większością punktów. Efekt algorytmu został uzyskany, jednak przez to, iż skrajne punkty nie stanowią większości całego zbioru, zostały one niedokładnie zespolone. W rezultacie powoduje to, że średnia gęstość punktów w chmurze jest znacznie mniejsza niż gęstość punktów na skrajnych powierzchniach. Do wyznaczenia długości promienia kuli można posłużyć się średnią odlegością pomiędzy punktami. W przypadku powyższych badań wybrano promień stanowiący trzykrotność oraz pięciokrotność średniej odlegości między punktami. Niestety, przez znacznie mniejszą gęstość na krańcach chmury, algorytm nie był w stanie połączyć ze sobą sąsiednich punktów. Skutkowało to występowaniem dziur w obiekcie.

Na rysunku 4.9 zestawiono wyniki algorytmu triangulacji Delaunay'a utworzone na podstawie jabłka oraz pudełka.



**Rysunek 4.9:** Wyniki triangulacji Delaunay'a

Na powyższych rysunkach zaprezentowano rezultaty przeprowadzenia triangulacji na dwóch chmurach punktów. Można na nich zaobserwować, że dużo szczegółów powierzchni jest przykrytych. Wynika to z nakładających się ścian utworzonych przez trójkąty triangulacyjne. W dalszej części pracy przedstawiono dokładne omówienie zagadnienia wraz z opracowaniem wyników. Jednakże niewątpliwą zaletą takiego rozwiązania jest brak dziur w powierzchni obiektu. Poprzez zastosowanie takiej metody generacji siatki, można również utworzyć dotychczas nie istniejące ściany. Na rysunkach przedstawione są obiekty, dla których górne ściany nie zostały zeskanowane. Jednak w wyniku przeprowadzenia triangulacji zostały one utworzone. Wykorzystując narzędzia do modelowania oraz obróbki obiektów trójwymiarowych, na przykład Blender, można usunąć część źle rozmieszczonej ściany.

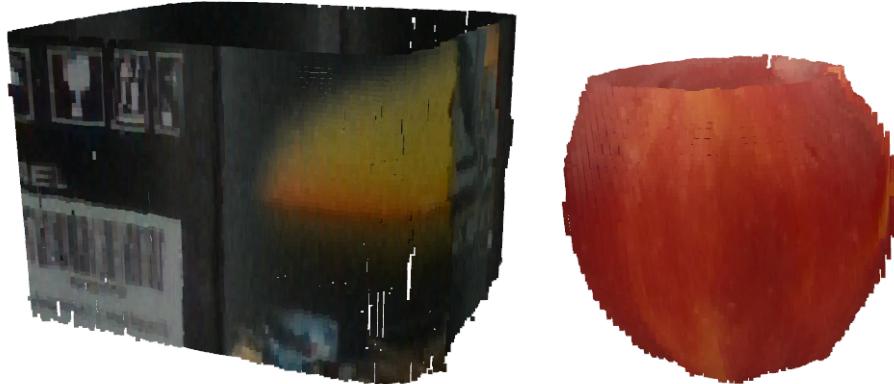
Po przeprowadzeniu analizy wyników można wysnuć wiele wniosków. Na powstałych w rezultacie chmurach punktów dokonano wielu przekształceń, by je ze sobą scalić. Jednakże rezultaty tych działań są niezadowalające. Punkty nie pokrywają się z należytą dokładnością. Ponadto znacznie zwiększyła się liczba punktów w chmurze, nie wpływając przy tym na poprawę wyników. Z kolei znacznie wpłynęło to na zwiększenie złożoności obliczeniowej. Zbiory punktów powstałe w skutek użycia metody skanera liniowego zawierają mniej punktów, generując przy tym lepszy efekt końcowy. Zarówno algorytm BPA jak i triangulacja Delaunay'a dla zastosowanej metody wypadają znacznie gorzej na tle metody skanera liniowego. Po przeprowadzeniu należytych analiz uznano, że najlepiej wypada metoda skanera liniowego ze względu na jej możliwość dokładniejszego odwzorowania powierzchni obiektu.

#### 4.3.2. Wyniki metody skanera liniowego

Kolejną metodą użytą do akwizycji danych o głębi obrazu jest metoda skanera liniowego. Poprzez jej zastosowanie możliwe jest otrzymanie znacznie lepszych rezultatów niżeli w przypadku metody światła strukturalnego. Poniżej przedstawiono wyniki generacji chmur punktów oraz siatek triangulacyjnych na jej podstawie.

## Chmura punktów

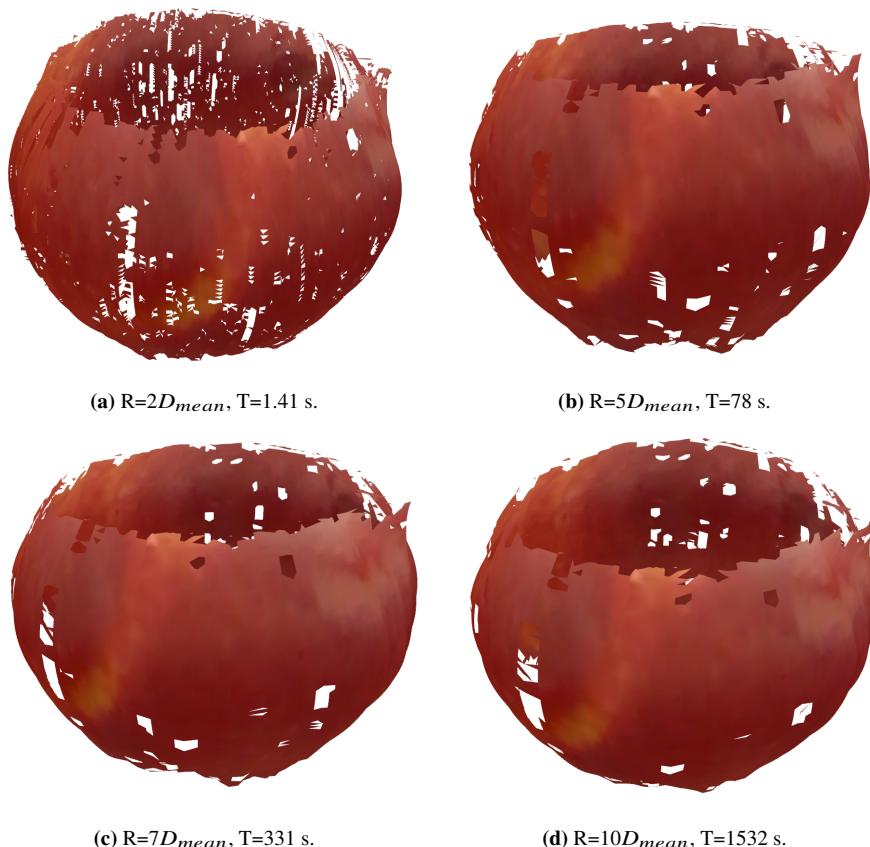
Dla metody skanera liniowego wygenerowane chmury punktów są bardzo dokładne. Poprzez zastosowanie odpowiednich przekształceń luki pomiędzy punktami są niemalże niezauważalne. Na rysunku 4.10 znajdują się punkty z nałożoną tekstonurą dla modeli jabłka oraz pudełka.



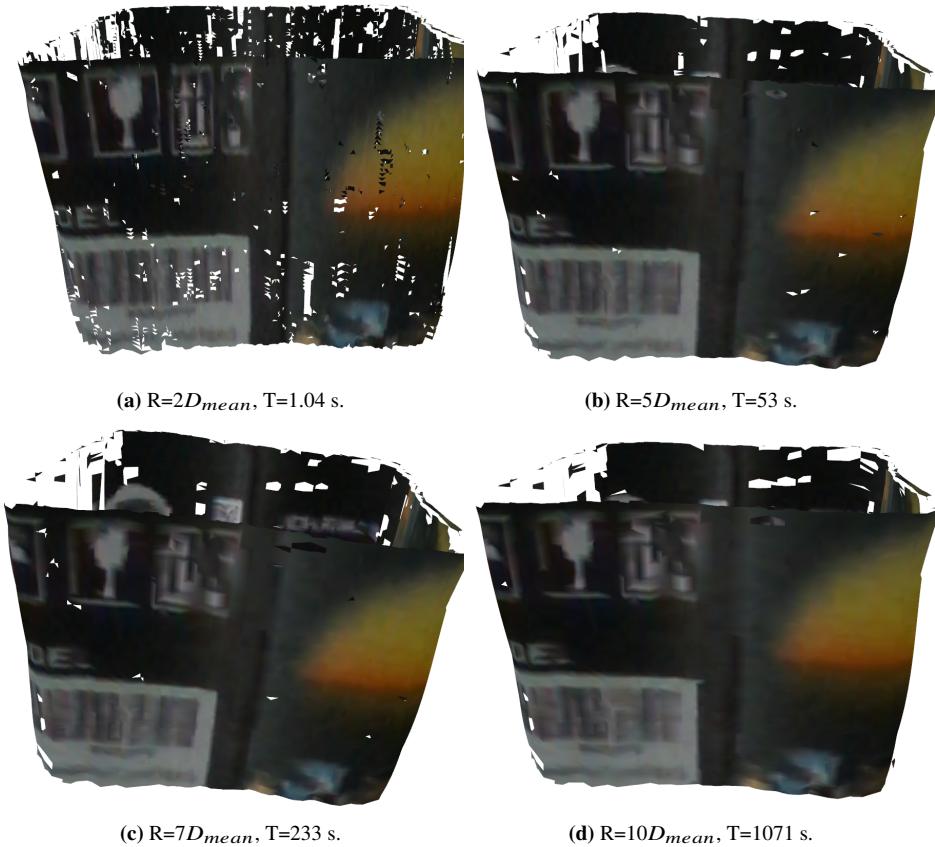
Rysunek 4.10: Chmury punktów pudełka oraz jabłka.

## Algorytm BPA

Dla algorytmu toczącej się kuli dokonano pomiaru wpływu promienia na ostateczny wygląd modelu. Na rysunkach 4.11 i 4.12 znajdują się modele jabłka oraz pudełka na które został nałożony mesh korzystając z metody BPA. Dla każdego z nich sprawdzono wartość promienia  $R$  równą odpowiednio 2,5,7,10 razy średnia odległość pomiędzy punktami. Dla lepszego porównania zdjęcia modeli robiono pod tym samym kątem.



Rysunek 4.11: Czas trwania algorytmu BPA dla jabłka w zależności od długości promienia  $R$ .



**Rysunek 4.12:** Czas trwania algorytmu BPA dla pudełka w zależności od długości promienia R.

Na powyższym zestawieniu widać, że wygląd chmury punktów znacznie różni się od gotowego meshu. W modelu jest wiele luk, które należy zakryć. Dokonać tego można przez odpowiednie dobranie promienia kuli. Znaczący wpływ na luki ma nierównomierna gęstość rozłożenia punktów w chmurze. Dla długości promienia równej dwukrotnej średniej odległości punktów, na rysunku 4.11 (a) oraz 4.12 (a) widać znaczny ich odsetek. Zwiększenie promienia do 5 znacznie poprawia wypełnienie. Po tym etapie, dalsze powiększanie promienia wpływa nieznacznie na ostateczny wygląd.

Przy zwiększaniu promienia rośnie również czas obliczeniowy. Powodem tego jest większa ilość punktów w sąsiedztwie kuli. Dla każdego jej przesunięcia wymagane jest wyznaczenie sąsiadów z odległości  $2R$ , co przy większym promieniu oznacza znaczną ilość punktów. Dla małej długości promienia czas obliczeń jest stosunkowo dobry. Algorytm zajmuje około 50 sekund. Można jednak zauważyć gwałtowną tendencję wzrostową. Zwiększenie o mniej niż 50 % długości promienia, sprawia, że długość obliczeń wzrasta pięciokrotnie. W celu najlepszych wyników zarówno czasowych jak i wyglądu modeli należy empirycznie dobrać promień. Z obserwacji wynika, że promień równy trzykrotności średniej odległości punktów daje dobre rezultaty. Został to też potwierdzone w pracach naukowych [40]. Z badań innych naukowców wynika, że średnio w sąsiedztwie kierunku toczenia się kuli powinno znajdować się 20 punktów. Przy  $R=3D_{mean}$  wzduż oraz wokół promienia znajduje się 17 punktów. Oznacza to, że empirycznie dobrany promień jest dobrym punktem startowym do poszukiwań.

#### 4.3.3. Algorytm Delaunay

W autorskim programie została w całości zaimplementowana metoda trójwymiarowej triangulacji Delaunay'a. W poniższej sekcji opisane zostaną wyniki pomiarów, wygenerowane modele oraz ich udoskonalenia w programie Blender. Na rysunku 4.13 ukazane zostały triangulacje Delaunay'a dla chmur punktów jabłka oraz pudełka.



**Rysunek 4.13:** Mesh Delaunay dla jabłka oraz pudełka.

Na rysunku powyżej można zauważyc najważniejszą różnicę algorytmu BPA od triangulacji Delaunay'a. W przypadku pierwszego, wygenerowane modele zawsze zawierały pewien odsetek dziur. Jednakże, ze względu na istotę działania, modele uzyskane dzięki wykorzystaniu algorytmu Bowyer-Watsona ich nie zawierają. Wynika to z faktu, iż do poprawnego funkcjonowania tej metody, z każdej trójki punktów musi zostać utworzony trójkąt. Dzięki temu z każdego punktu w chmurze tworzona jest ściana ostrosłupa. Ponadto korzystając z danej metody, zostają również uzupełnione luki wynikające ze sposobu pomiaru. Akwizycja danych następuje z boku obiektu, dlatego nie istnieją dane o wierzchniej ścianie modeli. Triangulacja Delaunay'a rozwiązuje ten problem, ponieważ graniczne punkty są łączone z pozostałymi na drugiej ścianie, tworząc w ten sposób dotychczas nieistniejącą ścianę. Kolory ścian wyznaczane są na podstawie wierzchołków trójkąta.

Mankamentem tej metody jest fakt, że często ściany występujące w modelu są duże. Oznacza to, że niekiedy część krawędzi obiektu może zostać przysłonięta przez zbudowaną ścianę. Taka sytuacja została zaprezentowana na rysunku 4.13 w przypadku modelu pudełka. Kolory zostały wiernie oddane, jednak część z nich na siebie nachodzi, przysłaniając w ten sposób pozostałe. W przypadku jabłka ten problem nie występuje.

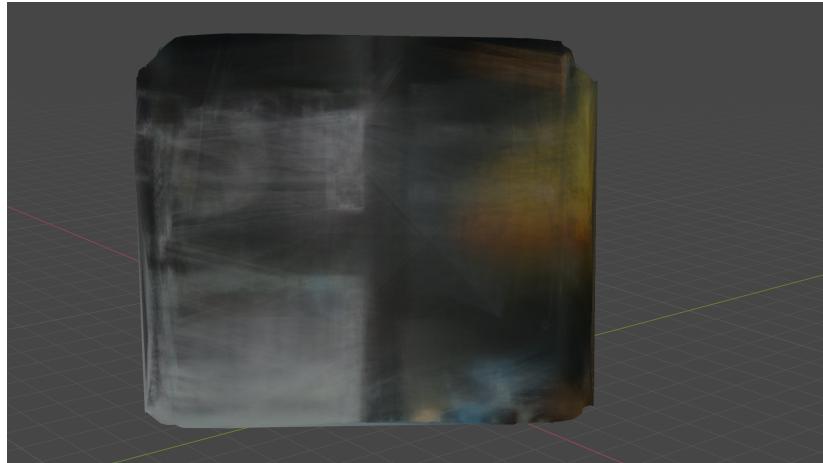
Wykorzystując program do obróbki modeli Blender, można dodatkowo wygładzić ściany z ostrych krawędzi powstających w skutek triangulacji. Program zawiera szereg funkcji wpływających na wygląd gotowego meshu, zmieniając sposób w jaki światło się od niego odbija. Na rysunku 4.14 przedstawione zostały modele poddane obróbkom.



**Rysunek 4.14:** Mesh Delaunay dla pudełka przed i po wygładzaniu w programie Blender.

Na powyższym rysunku przedstawiono modele przekształcone za pomocą programu Blender. Zastosowano funkcję flat do wyrównania poziomu cieni wokół wystających krawędzi, dzięki czemu stają się one mniej widoczne. Po lewej stronie wprowadzono funkcję, zakrywającą tylne krawędzie. Wykorzystując te metodę można uzyskać gładkie boki sześcianu, co wpływa pozytywnie na wygląd modelu. Część krawędzi przekształcono za pomocą edycji wierzchołków. Jest to funkcja pozwalająca na wygładzanie wystających wierzchołków trójkątów triangulacyjnych. Możliwe jest też podnoszenie tych, które leżą zbyt nisko, dzięki czemu ściany mają naturalny kształt oraz bardziej oddają wygląd

rzeczywistego obiektu. We wspomnianym programie edycji wierzchołków można dokonywać pojedynczo lub zbiorowo, wykorzystując narzędzie do zaznaczania punktów.



**Rysunek 4.15:** Ukryte ściany pudełka widoczne w programie Blender.

Na rysunku 4.15 zastosowano funkcję Xray. Pozwala ona na wyświetlenie dolnych warstw obiektu. Wykorzystując to przekształcenie, można zauważać, że triangulacja Delaunay'a działa poprawnie. Ściany które dotychczas były ukryte, ukazują faktyczną teksturę obiektu, która pokrywa się z tą wygenerowaną przez chmurę punktów.

Ostatecznie wybrano triangulację Delaunay'a do przeprowadzania generacji meshu. Pomimo wad takich jak przykrywanie wierzchnich warstw powierzchni przez trójkąty daje ona dobre rezultaty. Ściany modelu nie zawierają dziur. Poprzez wykorzystanie tego algorytmu generowane zostają też ściany, które w przypadku BPA nie mogły zostać utworzone. Taka sytuacja jest dla dolnej oraz górnej ściany prostopadłościennu oraz jabłka. Delaunay generując trójkąty łączy również przeciwległe ściany zakrywając tym samym obszary nieobjęte przez chmurę punktów. Poprzez zastosowanie danej metody uzyskuje się zawsze taki sam rezultat. Wygląd modelu nie zależy od nastaw parametru, tak jak miało to miejsce w przypadku BPA. Usprawnia to proces konwersji fizycznych obiektów do postaci 3D, a dzięki programowi Blender można dodatkowo usunąć niepotrzebne trójkąty zakrywające pozostałe ściany.

## 5. ZAKOŃCZENIE

Tematem pracy jest zbudowanie skanera 3D oraz wyeksportowanie gotowych modeli do programu Blender. Dane zadanie zostało wykonane pomyślnie. Przeprowadzono analizę dostępnych metod oraz teoretyczną analizę algorytmów. Dokonano szeregu optymalizacji, wpływających na zwiększenie wydajności autorskiego programu. Postawiony problem był złożony, a na ostateczny wygląd modelu wpływały szereg czynników. W poniższej sekcji przytoczone zostaną najważniejsze z nich.

### 5.1. Porównanie wyników przeprowadzonych badań

Poniżej omówione zostaną wyniki przeprowadzonych badań. Przytoczone zostaną modele wygenerowane za pomocą metody triangulacji Delaunay'a oraz algorytmu BPA. Zostaną one zestawione z gotowymi modelami dostępnymi w innych pracach naukowych. Omówiona zostanie ich dokładność względem autorskich metod.

Implementacja triangulacji Delaunay'a oraz użycie BPA miały na celu jak najwierniejsze oddanie powierzchni zeskanowanych modeli. Obie te metody nadają się do różnych celów. Triangulacja może być stosowana, gdy nie wszystkie boki obiektu zostały zeskanowane. Kiedy górna warstwa obiektu nie jest widoczna przez kamerę, to powstanie w tym miejscu luka. Takimi obiektemi są wszystkie prostopadłościany oraz walce. Natomiast modele dla których uda się utworzyć górną ścianę to na przykład piramida lub stożek.

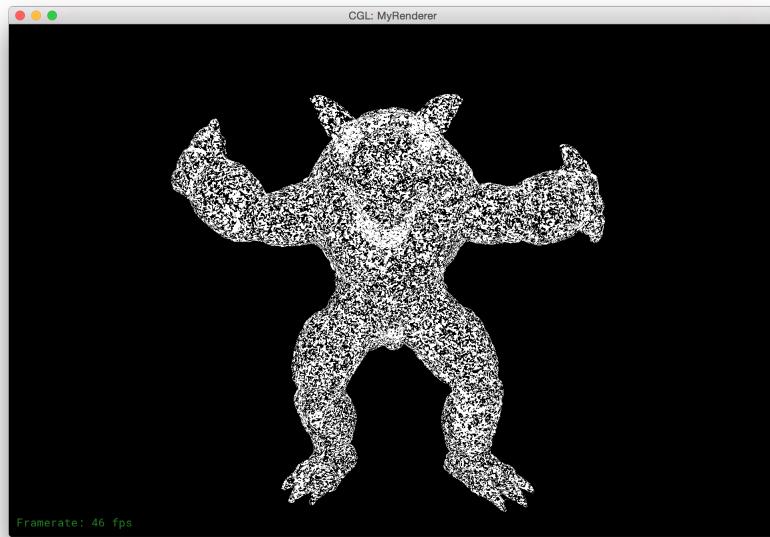
Kolejnym istotnym aspektem metody Delaunay'a jest brak dziur w wygenerowanych ścianach. Poprzez połączenie wszystkich punktów trójkątami, nie będzie między nimi prześwitu. Jej ograniczeniem jest długi czas obliczeniowy algorytmu. Nawet po dokonaniu szeregu optymalizacji czas trwania programu jest wciąż wysoki.

Algorytm BPA również może znaleźć swoje zastosowanie przy generacji tekstur obiektów. Jego zdolność do odwzorowywania kolorów płaszczyzny jest o wiele lepsza niż w przypadku triangulacji. Jego ograniczeniem natomiast są luki w powierzchni. Wynikają one z nierównomiernego rozkładu punktów w chmurze. Część z dziur można poprawić dobierając odpowiednio promień kuli, jednak inne pozostaną. Z przeprowadzonych badań wynika, że w wypadku danego projektu najlepiej wypadł algorytm triangulacji Delaunay'a. Brak luk w gotowym modelu jest ważniejszy, niż dokładna jakość tekstuury.

Do analizy wyników badań przeprowadzonych w projekcie, porównano dokładność odwzorowania powierzchni z gotowymi zdjęciami dostępnymi w innych pracach naukowych. Na rysunku 5.1 zostało przedstawione porównanie autorskich modeli jabłka oraz pudełka. Zestawiono je z człowiekiem Armadillo na rysunku 5.2. Jest to popularny zbiór danych, służący do testowania algorytmów generacji meshu. Pochodzi on z uczelni Stanford i przez wiele lat służył naukowcom do testowania oraz implementacji nowych metod tworzenia siatki na podstawie chmury punktów. Występuje w publikacjach zarówno o triangulacji Delaunay'a jak i BPA.



Rysunek 5.1: Mesh utworzony za pomocą BPA dla jabłka oraz pudełka.



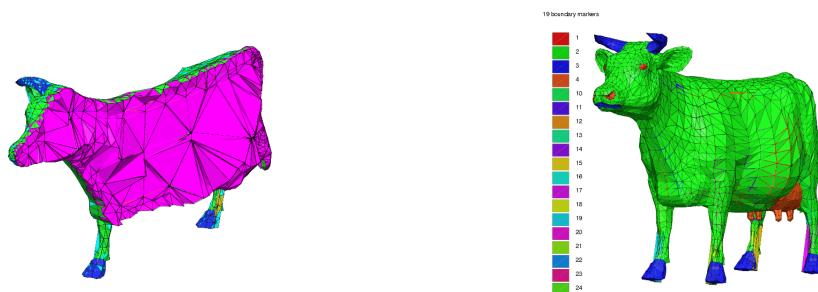
**Rysunek 5.2:** Mesh człowieka armadillo wygenerowany za pomocą BPA [46].

W meshu człowieka Armadillo można zauważać wiele luk. Autorzy stwierdzili, że ciężko im było dobrać odpowiedni promień kuli, dla którego powierzchnia obiektu stałaby się gładka. Na danym przykładzie widać, że algorytm BPA ma swoje wady które czasami utrudniają generację poprawnego modelu. Dla autorskich modeli również występują dziury, lecz jest ich o wiele mniej.

Na rysunku 5.3 oraz 5.4 zestawione zostały wyniki triangulacji Delaunay'a wykonane za pomocą autorskiego programu oraz przez program TetGen firmy Wias-Berlin.



**Rysunek 5.3:** Mesh utworzony za pomocą triangulacji Delaunay'a dla jabłka oraz pudełka.



**Rysunek 5.4:** Mesh triangulacji Delaunay'a oraz chmura punktów z przykładowych zdjęć programu TetGen [47].

Powyżej można zauważać, że nawet obiekt wygenerowany przez profesjonalne oprogramowanie TetGen zawiera wiele niedoskonałości. Na początkowej chmurze punktów krowy, powierzchnia jest gładka. Jednak po poddaniu obróbce przez triangulację powierzchnia staje się kanciasta i zawiera wiele odstających ścian. Z kolei istotnym elementem jest

brak dziur na powierzchni. Poddając modele z programu obróbce w programie Blender, można by się było pozbyć części niedoskonałości powierzchni. Porównując autorskie obiekty oraz te pochodzące z biblioteki programu firmy Wias-Berlin można dostrzec wiele podobieństw. Wynika to z użycia tego samego algorytmu, a różnice wynikają jedynie z innego podejścia do jego implementacji.

## 5.2. *Błędy oraz możliwości poprawy*

Podczas wykonywania testów zaimplementowanych metod napotkano szereg niedoskonałości. Istnieją również możliwości poprawy istniejących rozwiązań. Część z nich wynika z użytych algorytmów i metod, natomiast inne wynikają z ograniczeń sprzętowych. Poniżej przedstawione zostaną rozwiązania, których zaimplementowanie pozwoli na usprawnienie programu. Umożliwią one również zwiększenie jakości otrzymanych modeli obiektów.

Pierwszym mankamentem istniejącej metody skanowania obiektów jest brak możliwości rejestracji przezroczystych oraz błyszczących obiektów. Wynika to z zasady działania kamery 3D. Rzutuje ona siatkę interpolacyjną na mierzony obiekt. Jednak, gdy jest on przezroczysty lub błyszczący to odbija on promień lasera. Przez dyfrakcję światła, tylko część zostaje odbita do kamery. By temu zapobiec należy użyć sprawić by obiekt odbijał mniej światła w przypadku obiektów błyszczących oraz sprawić by je odbijał w przypadku przezroczystych. By tego dokonać należy na przykład pomalować powierzchnię na matowy kolor lub przykleić taśmę. Jest to skuteczna metoda pozwalająca na akwizycję danych z tego typu obiektów.

Kolejnym problemem który został napotkany podczas testów jest brak możliwości skanowania skomplikowanych kształtów. Jeżeli obiekt posiada wystające elementy to mogą one nie zostać zeskanowane poprawnie. W przypadku kubka, jeśli wiązka światła trafi na jego ucho, to nie zeskanuje ona tego co jest pod spodem. W efekcie na uzyskanym obiekcie będzie dostrzegalny odstający element, lecz nie to co jest pod nim. By temu zapobiec można dokonać kilku skanów, przemieszczając przy tym obiekt wzdłuż tacki. Dzięki temu jego oś obrotu zostanie przesunięta i zostanie on zeskanowany z innej strony. Dzięki temu będzie można zeskanować powierzchnię pod wystającym elementem.

Wykorzystanie metody triangulacji Delaunay'a jest bardzo kosztowne obliczeniowo. Z przeprowadzonych testów wynika, że czas trwania programu wynosi około 10 minut. Pomimo wykonania szeregu optymalizacji algorytm wciąż trwa znacznie dłużej, niż BPA dla małej długości promienia. W celu dalszej optymalizacji należy przekształcić algorytm, by wykonywał obliczenia równolegle na karcie graficznej. Kolejną możliwością jest utworzenie programu w innym języku. Utworzenie metody wykorzystującej dany algorytm w c++ pozwoli na dokonywanie szybszych obliczeń. Ten język charakteryzuje się lepszym wykorzystaniem mocy obliczeniowej procesora oraz wydajniejszym zarządzaniem pamięcią. Aktualnie język cython, pomimo tego, że jest częściowo kompilowany wciąż zawiera szereg ograniczeń wywodzących się z pythona. Poprzez zastosowanie list o nieokreślonej objętości, zarządzanie pamięcią jest mniej efektywne, a czas operacji jest dłuższy.

Dokładność skanów jest czymś, na czym należałoby się skupić najbardziej. Istnieje wiele możliwości jej udoskonalenia. W celu zwiększenia precyzji można podnieść liczbę klatek na sekundę nagrania. Dzięki czemu zwiększone zostanie zagęszczenie punktów chmurze. Zwiększoną gęstość punktów wpłynie pozytywnie na luki w algorytmie BPA. Jednocześnie przy wykorzystaniu triangulacji Delaunay'a gęstsze rozmieszczenie punktów w chmurze wpływa na zmniejszenie wielkości ścian generowanych trójkątów. Gdy pola trójkątów interpolacyjnych będą mniejsze, to będą one siebie nawzajem mniej zakrywać. Poprzez taki zabieg, tekstura obiektu zostanie wierniej oddana. Zmniejszając prędkość obrotową tacki można uzyskać ten sam rezultat, co w przypadku zwiększania liczby klatek. Przy wolniejszym ruchu tacki, wykonane zostanie więcej zdjęć. Wydaje się to być lepszym rozwiązaniem, ponieważ prędkość transmisji danych przez protokół USB nie będzie już ograniczeniem, wzrośnie jedynie czas pomiaru. Znaczną poprawę może również wnieść zmiana sposobu akwizycji danych o głębi. Z przeprowadzonych analiz wynika, że skaner liniowy lepiej oddaje powierzchnię modelu niż metoda światła strukturalnego. Jednakże wprowadzenie dodatkowych przekształceń do tej metody, mogłyby skutkować zwiększeniem jakości generowanych chmur punktów. Kolejnym sposobem na zwiększenie dokładności modeli jest ekstrapolacja kolumn. W tym celu należy w pustej przestrzeni pomiędzy kolejnymi kolumnami dodać nowe. Dokonanie tego będzie możliwe dzięki wykorzystaniu pozycji wcześniejszych punktów. Uśrednione pozycje dwóch sąsiednich punktów będą stanowić pozycję środkowego. Dana operacja może zostać wykonana wielokrotnie. Zwiększy to zagęszczenie punktów, jednak mogą zaistnieć przekłamania. Pomiędzy kolumnami występują efekty nieuchwytnie, które są niemożliwe do wyznaczenia. W miejscu gdzie nie znajduje się żaden punkt mogło na przykład być

wgłębienie. Przy ekstrapolacji zostanie ono jednak zastąpione uśrednioną wartością z okolicznych punktów. Najlepszym rozwiążaniem wydaje się być wykonanie dodatkowych skanów. Należy zarejestrować w jakich miejscach nastąpiły przekłamania punktów, a następnie uruchomić ponownie skan. Można również zarejestrować kilka pełnych obrotów tacki. Posiadając kilka ujęć tego samego kąta obrotu, będzie można wybrać tę wartość która nie jest przekłamana. Poprzez taki zabieg będzie można uniknąć aktualnie dokonywanej interpolacji punktów. Znane wówczas będą rzeczywiste pozycje punktów, a modele w ten sposób wygenerowane będą wiernie odwzorowywały rzeczywistość.

Ostatnim widocznym w testach ograniczeniem aktualnie wykorzystywanej metody skanowania trójwymiarowego jest brak górnej oraz dolnej ściany. Wykonując skan od boku, nie ma możliwości wyznaczenia kształtu górnej oraz dolnej ściany. Pierwszym sposobem uniknięcia tego jest wykonanie skanu z różnych perspektyw, a następnie nałożenie różnych części modelu na siebie w programie Blender. Będzie to jednak bardzo skomplikowane, ponieważ należy dokonać ekstrakcji jedynie potrzebnych płaszczyzn. Innym sposobem jest wykonanie skanu pod kątem. Dzięki takiemu zabiegowi, oprócz ścian bocznych zarejestrowana zostanie też ściana górnna. Jednakże taki zabieg znaczaco utrudnia przekształcenia geometryczne. Oprócz już istniejących będzie należało dodać dodatkowe, uwzględniające przesunięcie kamery oraz jej kąt względem obiektu.

## **Spis rysunków**

|      |  |    |
|------|--|----|
| 2.1  | Schemat skanera wykorzystującego triangulację laserową [6]. . . . .  | 9  |
| 2.2  | Układ współrzędnych nieruchomego skanera [6]. . . . .  | 10 |
| 2.3  | Czas pomiaru w zależności od dokładności pomiaru [9]. . . . .  | 11 |
| 2.4  | Metoda wyznaczania współrzędnych w technice światła strukturalnego [10]. . . . .                           | 12 |
| 3.1  | Schemat budowy autorskiego skanera 3D. . . . .   | 16 |
| 3.2  | Porównanie rzeczywistej wysokości z jej estymacją. . . . .   | 17 |
| 3.3  | Możliwe przecięcia sześciangu przez punkty [37] . . . . .  | 23 |
| 3.4  | Zasada działania algorytmu BPA [39]. . . . .   | 24 |
| 4.1  | Porównanie widoku kamery głębi oraz RGB . . . . .  | 25 |
| 4.2  | Porównanie wpływu ujemnego błędu pomiarowego na wygląd chmury punktów dla -5 %, -10 %. . . . .             | 26 |
| 4.3  | Porównanie wpływu dodatniego błędu pomiarowego na wygląd chmury punktów dla +10 %, +20 %. . .              | 26 |
| 4.4  | Dotknięcie przez kule punktu pod powierzchnią [40]. . . . .  | 28 |
| 4.5  | Porównanie czasu algorytmu dla cython oraz python. . . . .   | 32 |
| 4.6  | Początkowe chmury punktów jabłka , próbkowane co 45 ° i 11.25 °. . . . .                                   | 34 |
| 4.7  | Początkowe chmury punktów pudełka, próbkowane co 45 ° i 11.25 °. . . . .                                   | 35 |
| 4.8  | Siatka utworzona za pomocą algorytmu BPA jabłka oraz pudełka dla promieni $R=3D_{mean}$ oraz $R=5D_{mean}$ | 35 |
| 4.9  | Wyniki triangulacji Delaunay'a . . . . .   | 36 |
| 4.10 | Chmury punktów pudełka oraz jabłka. . . . .  | 37 |
| 4.11 | Czas trwania algorytmu BPA dla jabłka w zależności od długości promienia R. . . . .                        | 37 |
| 4.12 | Czas trwania algorytmu BPA dla pudełka w zależności od długości promienia R. . . . .                       | 38 |
| 4.13 | Mesh Delaunay dla jabłka oraz pudełka. . . . .   | 39 |
| 4.14 | Mesh Delaunay dla pudełka przed i po wygładzaniu w programie Blender. . . . .                              | 39 |
| 4.15 | Ukryte ściany pudełka widoczne w programie Blender. . . . .  | 40 |
| 5.1  | Mesh utworzony za pomocą BPA dla jabłka oraz pudełka. . . . .  | 41 |
| 5.2  | Mesh człowieka armadillo wygenerowany za pomocą BPA [46]. . . . .  | 42 |
| 5.3  | Mesh utworzony za pomocą triangulacji Delaunay'a dla jabłka oraz pudełka. . . . .                          | 42 |
| 5.4  | Mesh triangulacji Delaunay'a oraz chmura punktów z przykładowych zdjęć programu TetGen [47]. . .           | 42 |

## **Spis tabel**

|     |  |    |
|-----|--|----|
| 2.1 | Charakterystyki skanera Faro Focus S350 [8]. . . . .   | 11 |
| 2.2 | Charakterystyki skanera Faro Scan in a box [11]. . . . .   | 12 |
| 2.3 | Charakterystyki skanera Benewake TF03-180 [15]. . . . .  | 13 |
| 2.4 | Charakterystyki skanera Benewake TF02 [18]. . . . .  | 14 |
| 2.5 | Porównanie charakterystyk skanera Benewake TF02 oraz TF03-180. . . . .   | 14 |
| 2.6 | Charakterystyki metrologiczne laserowych metod pomiarowych [7]. . . . .  | 14 |
| 3.1 | Porównanie charakterystyk kamer Intel RealSense D435i oraz Orbbec Astra Mini MX6000 [30] [31]. . . . .         | 16 |
| 3.2 | Porównanie algorytmu MS oraz adaptacyjnych MS dla $7.4 \cdot 10^6$ oraz $2.8 \cdot 10^6$ punktów [38]. . . . . | 23 |
| 4.1 | Czas trwania BPA w zależności od promienia R. Średnia odległość między punktami $D_{mean} = 0.047$ . . . . .   | 29 |
| 4.2 | Czas trwania triangulacji przed optymalizacją. . . . .   | 30 |
| 4.3 | Czasy triangulacji dla pierwszej metody wyznaczania przynależności punktu do sfery. . . . .                    | 30 |
| 4.4 | Czasy triangulacji dla pierwszej oraz drugiej metody wyznaczania przynależności punktu do sfery. . . . .       | 31 |
| 4.5 | Porównanie czasu trwania algorytmu dla cython oraz python. . . . .   | 32 |
| 4.6 | Porównanie czasu trwania algorytmu przy obliczaniu macierzy przez NumPy oraz SciPy. . . . .                    | 33 |

## Bibliografia

- [1] V. G. Gokhare, D. Raut, and D. Shinde, “A review paper on 3d-printing aspects and various processes used in the 3d-printing,” *International Journal of Engineering Research & Technology*, vol. 6, no. 6, pp. 953–958, 2017.
- [2] K. Bassett, R. Carriveau, and D.-K. Ting, “3d printed wind turbines part 1: Design considerations and rapid manufacture potential,” *Sustainable Energy Technologies and Assessments*, vol. 11, pp. 186–193, 2015.
- [3] B. A. Studio, “Spring - Blender Open Movie,” 2019.
- [4] M. A.-B. Ebrahim, “3d laser scanners’ techniques overview,” *Int J Sci Res*, vol. 4, no. 10, pp. 323–331, 2015.
- [5] M. Abdel-Bary Ebrahim, “3d laser scanners: History, applications, and future,” *τχ. October*, vol. 2014, p. 5, 2011.
- [6] S. Mikulski, “Metody triangulacji laserowej w skanerach trójwymiarowych,” *Poznan University of Technology Academic Journals. Electrical Engineering*, pp. 240–245, 2013.
- [7] J. Nowacki, N. Sieczkiewicz, and M. Nocoń, “Pomiar odkształceń spawalniczych metodami skanowania 3d,” *Badania Nieniszczące i Diagnostyka*, pp. 4–5, 2018.
- [8] FARO, *FARO Focus specyfikacja techniczna*, 2017.
- [9] S. El-Omari and O. Moselhi, “Integrating 3d laser scanning and photogrammetry for progress measurement of construction work,” *Automation in construction*, vol. 18, no. 1, pp. 1–9, 2008.
- [10] M. Wrona and W. Piotrowska, “Nowe metody skanowania 3d na potrzeby inwentaryzacji geometrycznej pojazdów,” *TTS Technika Transportu Szynowego*, vol. R. 22, nr 12, pp. 1692–1695, CD, 2015.
- [11] S. in a box, *SIAB STRUCTURED LIGHT 3D SCANNER*, 2015.
- [12] E. Głowienka, B. Jankowicz, B. Kwoczyńska, P. Kuras, K. Michałowska, S. Mikrut, A. Moskal, I. Piech, M. Strach, and J. Sroka, “Fotogrametria i skaning laserowy w modelowaniu 3d,” *WSIE, Rzeszów*, pp. 118–119, 2015.
- [13] J. U. Eitel, B. Höfle, L. A. Vierling, A. Abellán, G. P. Asner, J. S. Deems, C. L. Glennie, P. C. Joerg, A. L. LeWinter, T. S. Magney, *et al.*, “Beyond 3-d: The new spectrum of lidar applications for earth and ecological sciences,” *Remote Sensing of Environment*, vol. 186, pp. 372–392, 2016.
- [14] H. B. Andrew Bell, Brad Chambers *et al.*, *Introduction to LiDAR*, 2010.
- [15] Benewake, *TF03-180 Product Specification*, 2018.
- [16] A. Wehr and U. Lohr, “Airborne laser scanning—an introduction and overview,” *ISPRS Journal of photogrammetry and remote sensing*, vol. 54, no. 2-3, pp. 68–82, 1999.
- [17] H. Yoon, H. Song, and K. Park, “A phase-shift laser scanner based on a time-counting method for high linearity performance,” *The Review of scientific instruments*, vol. 82, p. 075108, 07 2011.
- [18] Benewake, *TF02 Product Specification V3.3*, 2017.
- [19] S. 3D, *Time-of-Flight vs. Phase-Based Laser Scanners: Right Tool for the Job*, 2004.
- [20] B. Goszczyńska, W. Trąmpczyński, K. Bacharz, M. Bacharz, J. Tworzecka, and P. Tworzewski, “Doświadczalna analiza odkształceń przestrzennych belek żelbetowych z zastosowaniem skanera optycznego 3d,” *Inżynieria i Budownictwo*, vol. 70, no. 3, pp. 156–159, 2014.
- [21] J. Chen and K. Clarke, “Rapid 3d modeling using photogrammetry applied to google earth,” 09 2016.

- [22] A. Tomaka, L. Luchowski, K. Skabek, and M. Tarnowski, “3d head surface scanning techniques for orthodontics,” *Journal of Medical Informatics & Technologies*, vol. 9, pp. 124–130, 2005.
- [23] M. J. Thali, M. Braun, and R. Dirnhofer, “Optical 3d surface digitizing in forensic medicine: 3d documentation of skin and bone injuries,” *Forensic science international*, vol. 137, no. 2-3, pp. 203–208, 2003.
- [24] P. G. McMenamin, M. R. Quayle, C. R. McHenry, and J. W. Adams, “The production of anatomical teaching resources using three-dimensional (3d) printing technology,” *Anatomical sciences education*, vol. 7, no. 6, pp. 479–486, 2014.
- [25] A. Anders, Z. Kaliniewicz, and P. Markowski, “Zastosowanie skanera 3d do pomiarów cech geometrycznych produktów spożywczych na przykładzie pieczywa typu "kajzerka" i "minikajzerka",” *Postępy Techniki Przetwórstwa Spożywczego*, pp. 22–26, 2012.
- [26] N. D’Apuzzo, “3d body scanning technology for fashion and apparel industry,” in *Videometrics IX*, vol. 6491, p. 64910O, International Society for Optics and Photonics, 2007.
- [27] N. Blackburn, *Setting clothes in motion with Israel’s OptiTex*, 2006.
- [28] Admin, *3D Face Capture and Virtual Try-On System for the Eyewear Market*, 2001.
- [29] N. D’Apuzzo, “State of the art of the methods for static 3d scanning of partial or full human body,” in *Proceedings of Conference on 3D Modeling, Paris, France*, Citeseer, 2006.
- [30] Orbbec, “Astra Mini specification,” 2018.
- [31] Intel, “RealSense D435i specification,” 2020.
- [32] D. Han, “Comparison of commonly used image interpolation methods,” in *Proceedings of the 2nd international conference on computer science and electronics engineering*, Atlantis Press, 2013.
- [33] Q.-Y. Zhou, J. Park, and V. Koltun, “Open3D Global registration,” 2018.
- [34] D. Warchał, “Porównanie i łączenie cech deskryptorów chmur punktów w rozpoznawaniu statycznych układów dloni,” *Zeszyty Naukowe Politechniki Rzeszowskiej. Elektrotechnika*, no. 35 [294], nr 1, pp. 51–67, 2016.
- [35] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (fpfh) for 3d registration,” in *2009 IEEE international conference on robotics and automation*, pp. 3212–3217, IEEE, 2009.
- [36] P. Cignoni, C. Montani, and R. Scopigno, “Dewall: A fast divide and conquer delaunay triangulation algorithm in ed,” *Computer-Aided Design*, vol. 30, no. 5, pp. 333–341, 1998.
- [37] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.
- [38] R. Shu, C. Zhou, and M. S. Kankanhalli, “Adaptive marching cubes,” *The Visual Computer*, vol. 11, no. 4, pp. 202–217, 1995.
- [39] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, “The ball-pivoting algorithm for surface reconstruction,” *IEEE transactions on visualization and computer graphics*, vol. 5, no. 4, pp. 349–359, 1999.
- [40] F. B. J. Mittleman, H. R. C. Silva, and G. Taubin, “The ball-pivoting algorithm for surface reconstruction,”
- [41] OpenGL, “Calculating a Surface Normal,” 2013.
- [42] J. Digne, “An analysis and implementation of a parallel ball pivoting algorithm, image processing on line, 4 (2014),” 2014.
- [43] S. Rebay, “Efficient unstructured mesh generation by means of delaunay triangulation and bowyer-watson algorithm,” *Journal of computational physics*, vol. 106, no. 1, pp. 125–138, 1993.

- [44] A. Hui, “Circumsphere of tetrahedron,” 2017.
- [45] E. Weisstein, “Circumsphere,” 2017.
- [46] M. S. Brett Rapponotti and A. Zeng, “Point Cloud to Mesh, Ball-Pivoting Algorithm,” 2016.
- [47] H. Si, “TetGen example: Cow,” 2015.