

## Spis treści

1 Wstęp.....	2
2 Trójwymiarowa transformata Hougha.....	3
2.1 Historia.....	3
2.2 Teoria.....	4
3 Implementacja.....	8
4 Działanie transformaty na danych testowych.....	9
4.1 Poprawność.....	9
4.2 Wydajność.....	9
5 Działanie transformaty na danych geologicznych.....	10
5.1 Pochodzenie danych.....	10
5.2 Detekcja krawędzi.....	10
5.3 Poprawność.....	10
5.4 Wydajność.....	10
6 Podsumowanie i wnioski.....	11
7 Bibliografia.....	12

1 Wstep

## 2 Trójwymiarowa transformata Hougha

### 2.1 Historia

Transformata Hougha jest metodą wykorzystywaną do detekcji linii prostych, okręgów lub innych parametryzowalnych figur geometrycznych w obrazie binarnym. Została opracowana przez Paula V.C. Hougha w 1959 roku jako narzędzie do analizy obrazów z komory pęcherzykowej (Hough, 1959). Urządzenie to pozwalało na wykrycie cząstek elementarnych poprzez obserwacje pozostawianych przez nie śladów w postaci pęcherzyków pary w przegrzanej cieczy. W roku 1962 transformata została opatentowana przez Komisję Energii Atomowej Stanów Zjednoczonych, a w późniejszych latach była nadal rozwijana. W 1972 Duda i Hart w pracy „*Use of the Hough Transformation to Detect Lines and Curves in Pictures*”, przedstawiono obecnie stosowaną wersję transformaty. Użyto w niej parametryzacji  $(\theta, \rho)$ , która była stosowana dużo wcześniej w ogólniejszej metodzie jaką jest transformata Radona. Parametryzacja ta gwarantuje skończoność przestrzeni Hougha (Duda i Hart, 1972). Oprócz zaproponowania innej parametryzacji poszukiwanych prostych, autorzy pracy pokazali też jak użyć transformaty do detekcji innych figur dających się opisać analitycznie, jak np. okręgi i elipsy.

Istnieje także możliwość uogólnienia transformaty Hougha tak, aby pozwalała na detekcję dowolnych kształtów (Ballard, 1981; Chmielewski, 2021). Uogólnienie jest realizowane poprzez zmianę sposobu opisu szukanego wzorca. Zamiast równania analitycznego stosuje się tabelę, która za pomocą kierunku gradientu i wektora, opisuje położenie punktów brzegowych względem uprzednio wybranego punktu referencyjnego.

Na początku tego wieku pojawiły się publikacje traktujące o trójwymiarowej transformacie Hougha jako narzędziu do wykrywania płaszczyzn. Przedstawiono w nich rozmaite metody parametryzacji płaszczyzny (Vosselman i Dijkman, 2001; Rabbani, 2006). Równanie kierunkowe  $z(x,y) = a_x x + a_y y + d$  nie nadaje się do opisu pionowych płaszczyzn, ale świetnie sprawdza się tam, gdzie poszukiwane płaszczyzny są bliskie poziomemu, jak np. na laserowych skanach lotniczych (Vosselman i Dijkman, 2001). Aby uniknąć problemu parametrów rosnących do nieskończoności parametryzuje się płaszczyznę przez kierunek jej wektora normalnego i jej odległość od początku układu odniesienia (Rabbani, 2006). Istnieje także możliwość bardziej ogólnego podejścia, w którym używa się multiwymiarowych przestrzeni (trzy i więcej wymiarowych). Warunkiem koniecznym jest, aby heurystyki wyszukiwania pozostały wykonalne (Achtert i in., 2008).

## 2.2 Teoria

Opis teoretyczny transformaty Hougha postanowiono rozpocząć od przedstawienia schematu działania, wspólnego dla wszystkich typów szukanych wzorców. Transformata polega na takim przekształceniu ośrodka poszukiwań, aby w tworzonej przez nas przestrzeni Hougha szukany wzorec był punktem. Następnie wszystkie piksele „głosują” na wzorce, których częścią mogą być. Tym samym problem wykrycia złożonej figury sprowadza się do wykrycia lokalnych maksimów w przestrzeni Hougha.

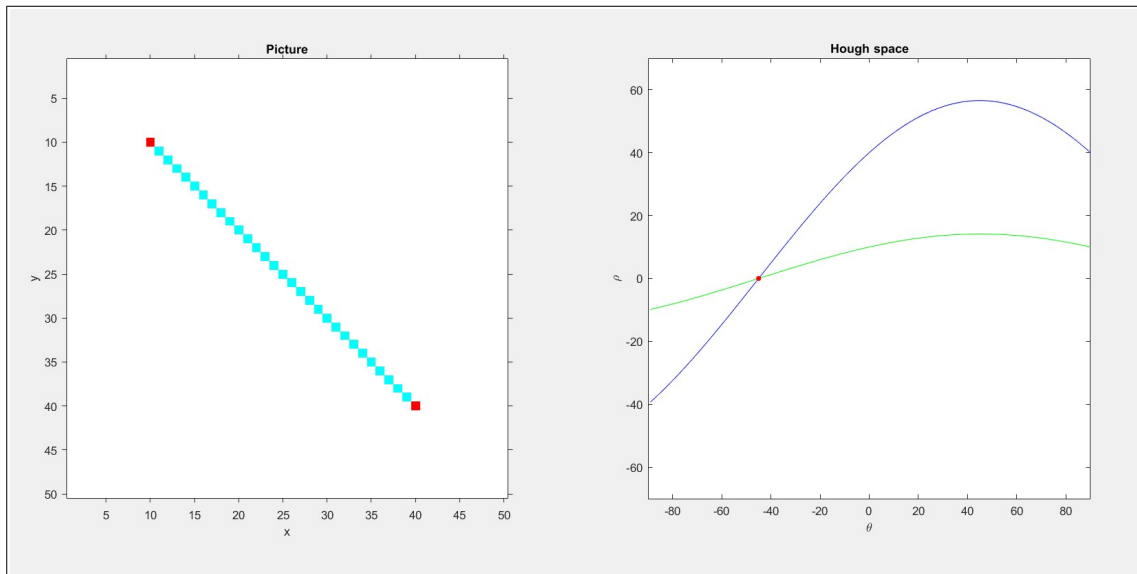
Najprostszymi przypadkami transformaty są detekcja prostych na obrazie i płaszczyzn w przestrzeni. Jak napisano w rozdziale o historii opisanie ich za pomocą równań ze współczynnikami kierunkowymi, stwarza problemy z wykrywaniem wystąpień w kierunkach zbliżonych do pionowych, dlatego rozpatrzmy parametryzację za pomocą kąta i odległości. W przypadku prostej używa się do tego równania:

$$x \cos(\theta) + y \sin(\theta) = \rho, \quad (2.1)$$

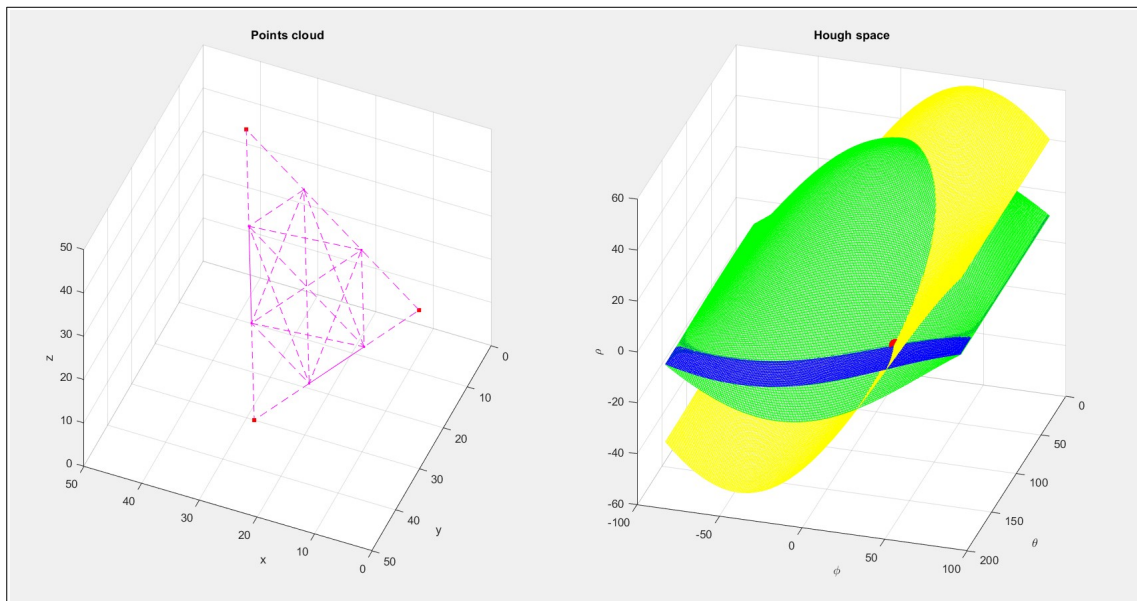
gdzie  $\rho$  jest odległością prostej od początku układu współrzędnych, a  $\theta$  kątem pomiędzy osią  $x$  i  $\rho$  (Duda i Hart, 1972). W ten sposób każdy punkt  $(x, y)$  przechodzi w sinusoidę w przestrzeni Hougha, których przecięcie wyznacza prostą (Rys. 2.1). W przypadku płaszczyzny opisuje się ją za pomocą równania:

$$x \cos(\theta) \cos(\varphi) + y \sin(\theta) \cos(\varphi) + z \sin(\varphi) = \rho, \quad (2.2)$$

gdzie  $\rho$  jest odległością płaszczyzny od początku układu odniesienia, kąt  $\varphi$  zawiera się pomiędzy płaszczyzną  $OXY$ , a wektorem normalnym płaszczyzny szukanej. Z kolei kąt  $\theta$  zawiera się pomiędzy rzutem wektora normalnego na płaszczyznę  $OXY$ , a osią  $OX$  (Rabbani, 2006). Parametryzacja taka powoduje przejście każdego punktu z wejściowej objętości danych w płaszczyznę sinusoidalną w przestrzeni Hougha, a ich przecięcie wskazuje płaszczyznę (Rys. 2.2). Pokazuje to dobrze podobieństwa między tymi przypadkami.

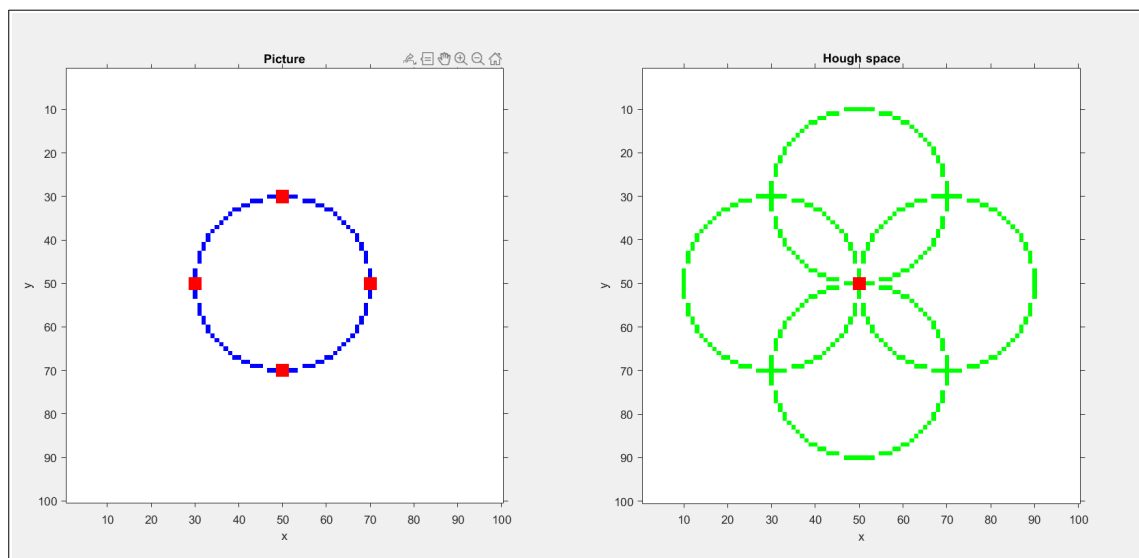


*Rys. 2.1 Dwa czerwone punkty na płaszczyźnie i ich reprezentacja w przestrzeni Hougha w postaci dwóch sinusoid z zaznaczonym przecięciem wskazującym prostą.*

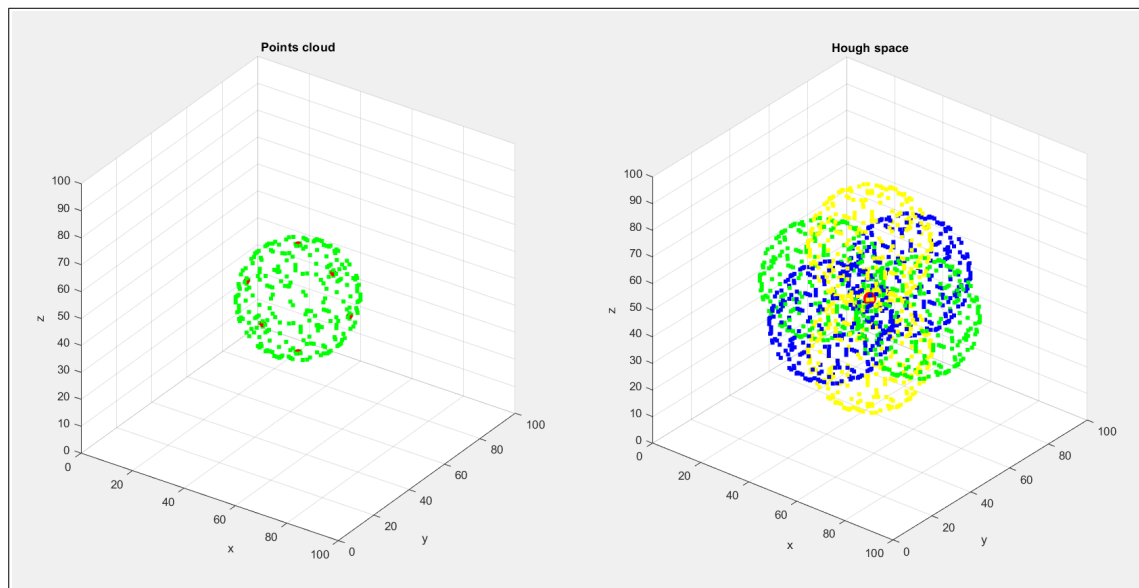


*Rys. 2.2 Trzy czerwone punkty na płaszczyźnie i ich reprezentacja w przestrzeni Hougha w postaci płaszczyzn sinusoidalnych z zaznaczonym miejscem przecięcia wskazującym płaszczyznę, na której znajdują się rozpatrywane punkty.*

Podobieństwo widać także pomiędzy sposobami detekcji okręgów i sfer. W tych przypadkach zdefiniowanie przejścia do przestrzeni Hougha nie wymaga dodatkowych operacji. Figury te z definicji stanowią zbiory punktów równo odległych od środka figury, dlatego też zbiorem potencjalnych figur, których częścią może być dany punkt jest taka sama figura. Każdy punkt z danych wejściowych przechodzi w okrąg (Rys. 2.3) lub sferę (Rys. 2.4) w przestrzeni Hougha, a ich przecięcie wskaże wykrytą figurę. Metoda ta wymaga podania promienia szukanego wzorca jeszcze przed detekcją, jednak nie wyklucza to poszukiwań figur różnej wielkości. Operację taką przeprowadza się poprzez sekwencyjne wykonywanie transformaty z kolejnymi wartościami promienia z rozpatrywanego przedziału. Tak zdefiniowaną transformatę można uogólnić na elipsy i elipsoidy. Uwzględnienie różnych wartości odległości w kierunkach prostopadłych wymaga od nas jednoczesnego uwzględnienia rotacji wzorca. Oznacza to zwiększenie złożoności problemu i liczby wymiarów przestrzeni Hougha, która w przypadku trójwymiarowej przestrzeni danych i elipsoidy nieobrotowej osiąga 6 wymiarów przy 3 stałych promieniach.



*Rys. 2.3 Cztery czerwone punkty na okręgu i ich reprezentacja w przestrzeni Hougha w postaci okręgów z zaznaczonym miejscem przecięcia wskazującym środek szukanej figury.*



*Rys. 2.4 Sześć czerwonych punkty na sferze i ich reprezentacja w przestrzeni Hougha w postaci sfer z zaznaczonym miejscem przecięcia wskazującym środek szukanej figury.*

Istnieje także możliwość użycia transformaty Hougha jako jednego z etapów bardziej złożonej detekcji. Przykład takiego zastosowania mogą stanowić algorytmy wyszukiwania prostokąta w 2D i prostopadłościanu w 3D. Wykrycie takich kształtów sprowadza się do znalezienia ich składowych. Po wykryciu odpowiednio prostych lub płaszczyzn, następuje analiza wyniku w poszukiwaniu par wzajemnie prostopadłych i równoległych, które mogłyby utworzyć figurę będącą początkowym celem detekcji.

### 3 Implementacja

W celu zbadania możliwości trójwymiarowej transformaty Hougha stworzono w środowisku Mathworks MatLAB R2021b szereg skryptów implementujących transformatę dla różnych kształtów. Wszystkie funkcje szukające przyjmują binarną (logiczną) objętość danych jako argument. Punkty w niej zawarte stanowią krawędzie brył przestrzennych. Wynikiem działania funkcji jest wyznaczenie wartości przestrzeni Hougha dla danego kształtu. Oprócz przestrzeni transformacji funkcje te zwracają też wektory będące wartościami wymiarów przestrzeni.

Ogólne zasady działania algorytmów trójwymiarowej transformaty Hougha zostały przedstawione w rozdziale 2.2. W tym zostaną rozpatrzone praktyczne aspekty implementacji algorytmów. Każdy z algorytmów zaczyna się od stworzenia przestrzeni Hougha odpowiednich wymiarów, a następnie wyznaczenia wszystkich punktów o wartości TRUE punktów z wejściowej objętości danych i iteracyjnego wykonywania czynności dla każdego w nich. Podczas nich zwiększane są odpowiednie komórki przestrzeni Hougha, które odpowiadają figuram, w które został przekształcony punkt.

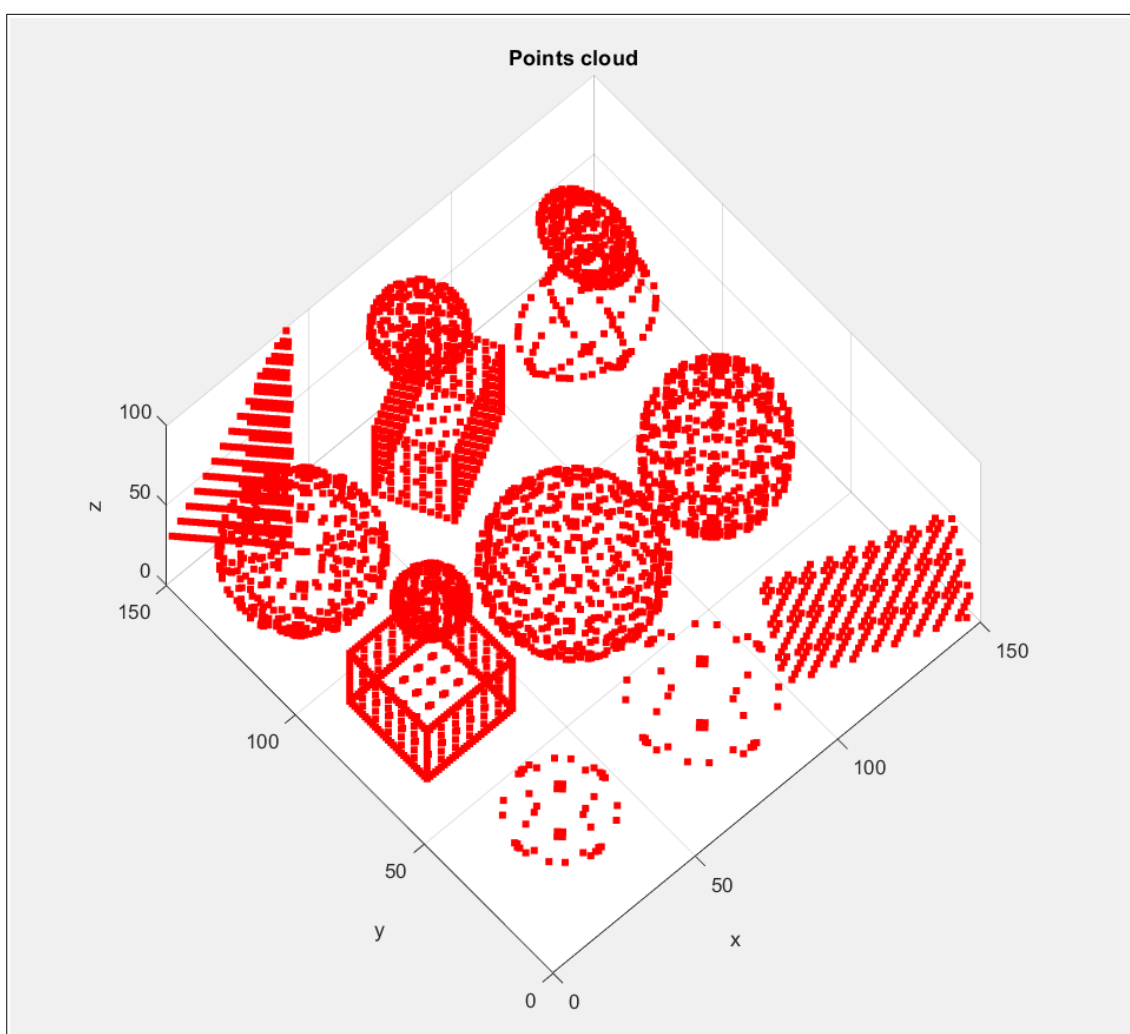
Przestrzeń transformaty nie jest jednakowa dla każdego skryptu. Dla płaszczyzn przestrzeń Hougha jest trójwymiarowa i ma 2 stałe wymiary będące zakresami kątów, dla  $\theta$  jest to zakres od  $0^\circ$  do  $179^\circ$ , a dla  $\phi$  od  $-89^\circ$  do  $90^\circ$ , w obu przypadkach z krokiem 1 stopień. Trzeci wymiar dla tego przypadku zależny od danych wejściowych i długości  $d$  przekątnej objętości danych. Zakres odległości rozciąga się od  $-d$  do  $d$  z krokiem 0.1. Dla przypadku detekcji sfer wymiary przestrzeni wyjściowej są takie same jak wymiary przestrzeni wejściowej. W takim przypadku nie są też zwracane żadne inne dane oprócz wartości przestrzeni transformaty. W przypadku bardziej ogólnym niż okrąg, jakim jest elipsoida, przestrzeń Hougha znacząco się komplikuje, o czym wspomniano wcześniej. Podczas implementowania tego algorytmu barierą stały się ograniczenia środowiska MatLAB, które nawet przy małych testowych danych wejściowych, nie pozwoliły na próbowanie kątów odchylenia od osi co 1 stopień z powodu ograniczenia na wielkość pojedynczej zmiennej. Wymusiło to zwiększenie kroku próbkowania kąta odchylenia do 45 stopni co w efekcie dało 6 wymiarową przestrzeń transformaty. Zmienna ta dla danych wejściowych o wymiarach  $(a, b, c)$  ma wymiar  $(a, b, c, 4, 4, 4)$ .



## 4 Działanie transformaty na danych testowych

### 4.1 Poprawność

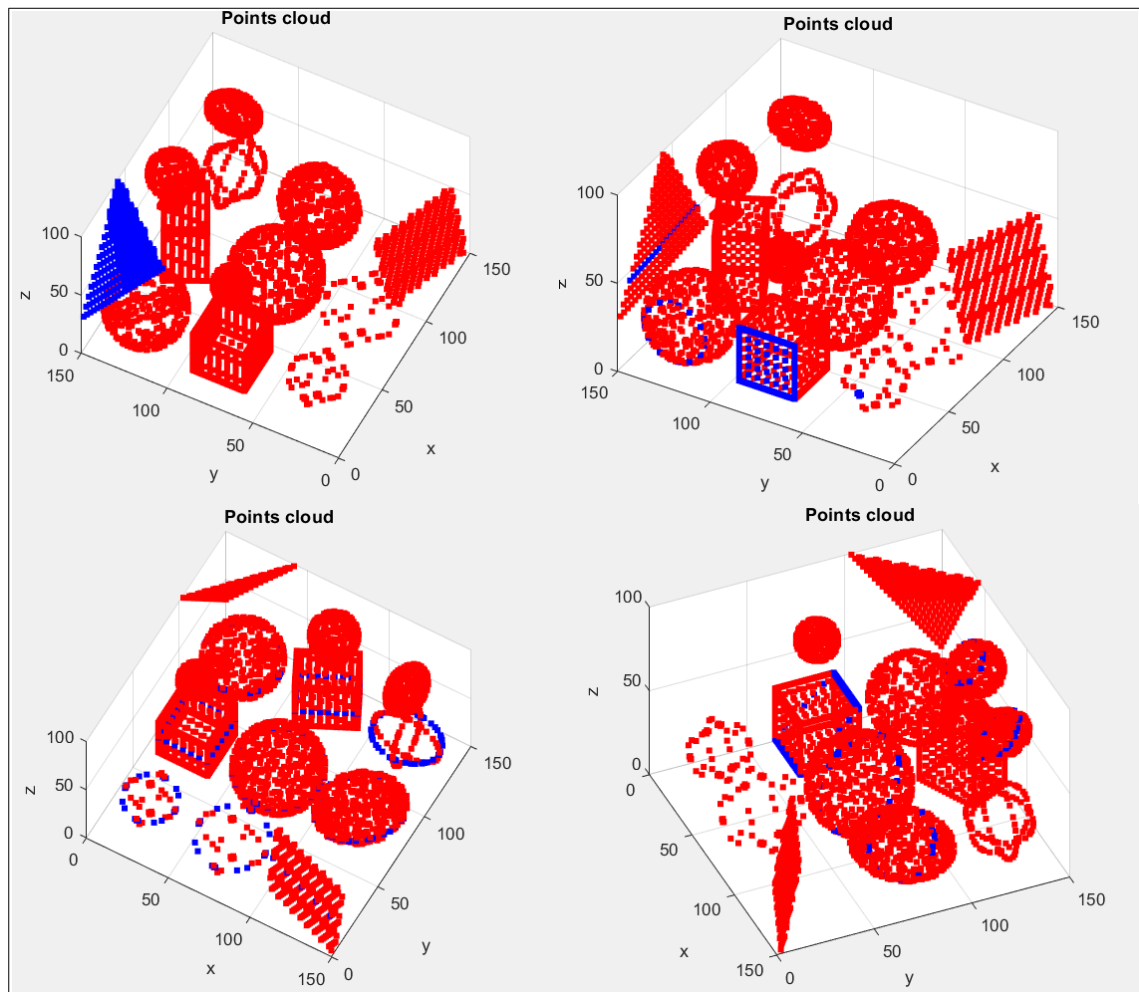
Aby sprawdzić działanie trójwymiarowej transformaty Hougha stworzono dane testowe. Objętość danych ma wymiary 150x150x100 px i zawiera wzorce do detekcji zaimplementowanymi skryptami. Znajdują się tam 2 płaszczyzny złożone z pojedynczej warstwy pikseli, 4 sfery o różnych promieniach, 2 elipsoidy i 2 prostopadłościany reprezentowane przez swoje krawędzie. Dodatkowo znajdują się tam figury będące szukanymi wzorcami, ale mające mniej punktów. Są to 2 sfery i jedna elipsoida reprezentowane przez 3 wzajemnie prostopadłe figury płaskie (Rys 4.1).



*Rys. 4.1 Testowa objętość danych zawierająca wzorce do detekcji.*

Badanie poprawności rozpoczęto od sprawdzenia wyników detekcji płaszczyzn. W tym celu sprawdzono 16 płaszczyzn z największym wynikiem i porównano z pożądanym efektem, jakim jest wykrycie dwóch płaszczyzn w rogach stworzonej objętości danych i boków dwóch prostopadłościanów, sześcianu o boku 30 oraz prostopadłościanu o przybliżonych wymiarach (33.5, 22.4, 50). Wyniki nie były w pełni satysfakcjonujące. Dwa z trzech najwyższych wyników, z powodu dużej liczby punktów wzorca, odpowiadały płaszczyznom z rogów objętości danych. W rezultacie działania funkcji szukającej znajdują się także płaszczyzna przechodząca przez podstawy obu prostopadłościanów oraz płaszczyzny, w których zawierają się pozostałe boki sześcianu występujące na przemian z płaszczyznami przechodzącymi przez różne figury (Rys 4.2). Takiego wyniku nie można uznać za niepoprawny, ale uniemożliwia on późniejszą detekcję prostopadłościanu, dlatego zwiększono liczbę wyników do 30, co pozwoliło wykryć górny bok prostopadłościanu. Dalsze zwiększanie liczby wyniku uznano za nieoptymalne, ze względu na zwracanie płaszczyzn przechodzących przez różne obiekty. W celu wykrycia płaszczyzn zawierających w sobie pionowe boki prostopadłościanu podjęto próby zwiększenia dokładności poprzez próbkowanie kąta  $\theta$  co  $0.1^\circ$  oraz detekcji na danych pozbawionych innych figur. Udało się osiągnąć wykrycie wszystkich boków prostopadłościanów, wśród 20 najwyższych wyników, ale wymagało to wspomnianej zwiększonej dokładności, usunięcia innych figur, a także zwiększenia liczby punktów zawartych w jednym z boków. Problemy z detekcją pionowych boków prostopadłościanu wynikały z 2 aspektów. Po pierwsze boki te zawierały mniej punktów niż boki sześcianu, co przekładało się na niższy wynik. Drugim powodem było ich położenie, z wyliczeń ręcznych wynika, że kąty  $\theta$  dla dwóch par płaszczyzn są nie wymierne i wynoszą około  $26,57^\circ$  i  $116,57^\circ$ .

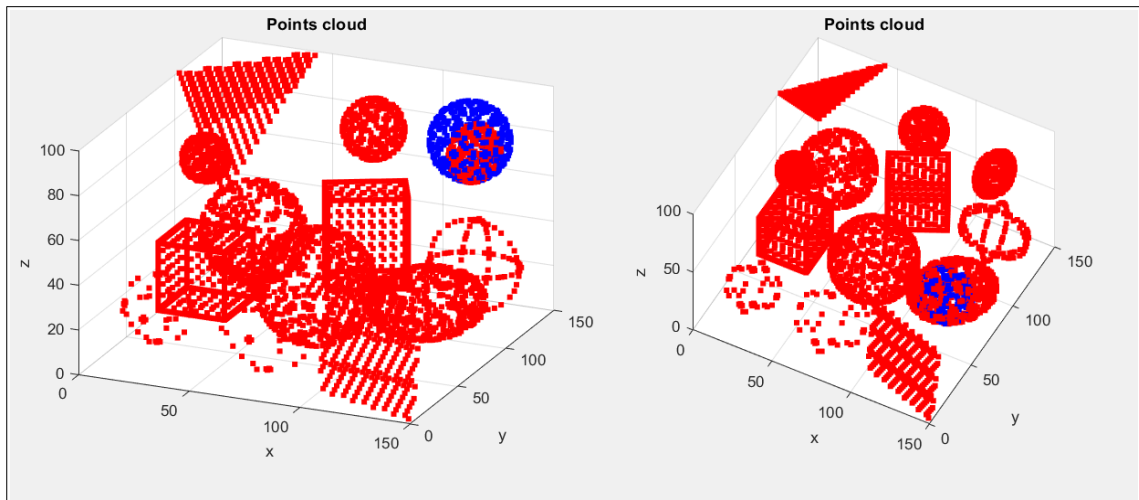
Detekcja wszystkich płaszczyzn, które w sobie zawierają boki prostopadłościanów pozwoliła na skorzystanie ze skryptu do detekcji tych figur. Algorytm wśród zwróconych 20 płaszczyzn wyszukał pary wzajemnie prostopadłe i równoległe, a następnie skazał środki ciężkości obiektów i równania płaszczyzn, w których zawierają się ich boki.



*Rys. 4.2 Cztery przykładowe wyniki detekcji płaszczyzn na danych testowych.  
W górnym rzędzie płaszczyzny, które chciano znaleźć, w dolnym rzędzie  
płaszczyzny przechodzące przez wiele figur.*

W następnym kroku sprawdzono poprawność działania algorytmów wyszukiwania sfer i elipsoid. Próby wyszukania w danych istniejących sfer zakończyły się powodzeniem. Dla każdej z czterech figur najwyższy wynik stanowiło jej dokładne położenie, a kolejne wyniki oscylowały wokół niego. Podjęto także próbę wykrycia sfer reprezentowanych jedynie przez 3 okręgi wzajemnie prostopadłe, ale nie udało się tego osiągnąć. Sfera o promieniu 15 została wskazana w elipsoidzie, której jeden z promieni był równy szukanemu, a ta o promieniu 20 została wykryta w elipsoidzie i nieco większej sferze o promieniu 22. Rezultaty działania skryptów wykrywających elipsoidy były takie same, udało się wyznaczyć położenie figur złożonych z dużej liczby punktów, ale tej z ich ograniczoną ilością już nie. Detekcja ta zwróciła fałszywe wystąpienia obok innych elipsoid.

W kolejnym kroku sprawdzono jakie efekty przyniesie detekcja jeśli promień będzie się minimalnie różnił. W tym celu wykonano próby wyszukania sfer o promieniach od 14 do 21. Dla każdej z tych wartości wyniki były zbliżone. Szukane wzorce były wskazywane w pobliżu sfer i elipsoid o podobnych promieniach (Rys 4.3), a zatem przekazanie do funkcji szukającej przybliżonego argumentu dało przybliżone rezultaty.



*Rys. 4.3 Sfery wykryte w pobliżu figur o podobnej krzywiźnie.*

Kolejnym krokiem badań nad poprawnością transformaty było sprawdzenie jej wrażliwości na szum punktowy znajdujący się w danych wejściowych. W tym celu dodawano do testowej objętości danych losowe punkty i sprawdzano jaka ich ilość jest w stanie zakłócić i zmienić wyniki działania skryptów.

Szanowny Panie Doktorze, opiszę to jutro. Dodam też ładne wykresiki jak to wszystko zwalnia w zależności od wielkości danych ilości punktów w nich i wielkości szukanej figury.

## 4.2 Wydajność

## 5 Działanie transformaty na danych geologicznych

### 5.1 Pochodzenie danych

### 5.2 Detekcja krawędzi

### 5.3 Poprawność

### 5.4 Wydajność

## 6 Podsumowanie i wnioski

## 7 Bibliografia

1. D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, Vol.13, No.2, p.111-122, 1981
2. E. Achtert, Ch. Böhm, J. David, P. Kröger, A. Zimek, "Global Correlation Clustering Based on the Hough Transform", 2008
3. G. Vosselman, S. Dijkman, "3D Building Model Reconstruction from Point Clouds and Ground Plans", International Archives of Photogrammetry and Remote Sensing, Volume XXXIV-3/W4 Annapolis, MD, 22-24, październik, 2001
4. P. Chmielewski, "Analiza uogólnionej Transformaty Hougha do wyszukiwania ziaren minerałów o zadanym kształcie", luty, 2021
5. P.V.C. Hough, "Machine Analysis of Bubble Chamber Pictures", Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959
6. R.O. Duda and P.E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures", Comm. ACM, Vol. 15, s. 11–15, styczeń, 1972
7. T. Rabbani, "Automatic reconstruction of industrial installations – Using point clouds and images", pages 43–44, Publications on Geodesy 62, Delft, 2006.