

# Cyfrowe Przetwarzanie Obrazu

## Projekt

**Detekcja piłki siatkowej na obrazach pochodzących ze stereopary wraz z wyznaczaniem współrzędnych 3D trajektorii piłki**

**Wykonał:** Konrad Bedełek, 289809

**Data:** 15.05.2022

## 1. Wstęp

Celem projektu jest detekcja piłki na obrazach pochodzących ze stereopary oraz wyznaczenie współrzędnych 3D trajektorii poruszającej się piłki na podstawie obrazu z trzech ustawionych wokół sceny kamer. Na rysunku nr 1 przedstawiono jeden z obrazów przedstawiający scenę na której należy wykryć przemieszczającą się piłkę.



*Rysunek 1. Scena do detekcji piłki.*

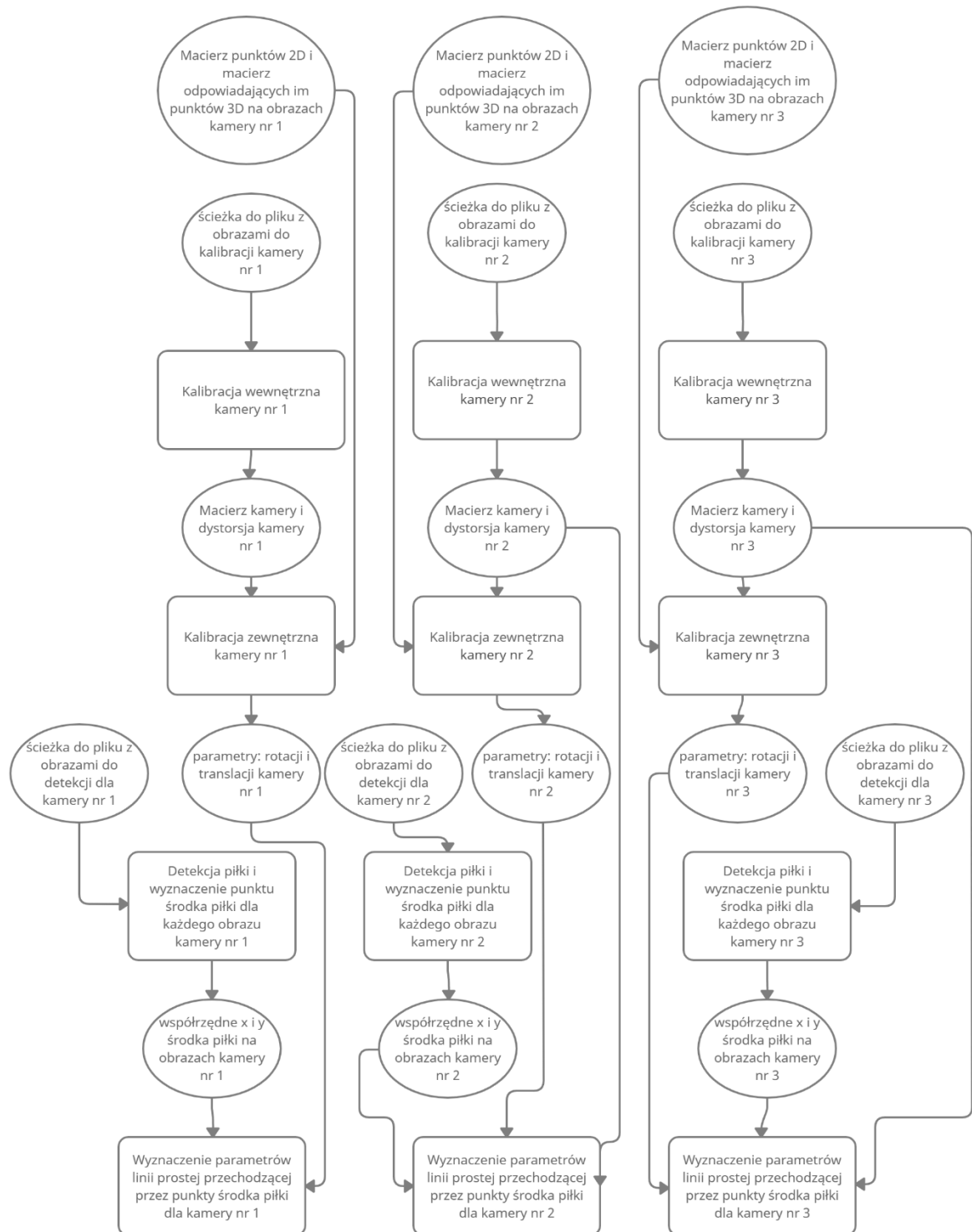
Spostrzeżenia dotyczące sposobu rozwiązania zadania:

- Obecność jednej poruszającej się piłki na obrazach upraszcza zadanie ponieważ nie ma konieczności śledzenia trajektorii kilku obiektów.
- Piłka nie zostaje przysłonięta przez inne obiekty (brak zawodników na boisku).
- Piłka nie jest jedynym poruszającym się obiektem na scenie co może utrudnić detekcję na podstawie estymacji tła. Jednakowoż jest ona najszybciej poruszającym się obiektem, a więc jej przemieszczenie wprowadzi znaczącą zmianę na obrazie względem tła.
- Detekcja piłki na podstawie koloru jest problematyczna ze względu na rozmycie piłki wynikające z jej ruchu oraz dużą ilość kolorów w tle. Dodatkowo wykryte zostaną elementy nieruchome. Ten sposób detekcji będzie ponadto ograniczony do jednego rodzaju (koloru) piłki.
- Na obrazie można zauważyć również inne okrągłe obiekty inne niż piłka co utrudni rozpoznawanie na podstawie kształtu obiektu.
- W danych można spotkać obrazy na których występuje jedna poruszająca się piłka i kilka piłek nieruchomych co może wprowadzić konieczność śledzenia trajektorii obiektów celem wyeliminowania obiektów stacjonarnych.
- Kamery w sekwencji 3 są inaczej ustawione niż w sekwencji 1 i 2.
- Oświetlenie na obrazach jest niezmiennie jednak na obrazach obecne są regiony o różnym natężeniu światła.
- Na piłkach obecny jest wzór który odróżnia się od reszty elementów na obrazie.

## 2. Schemat programu

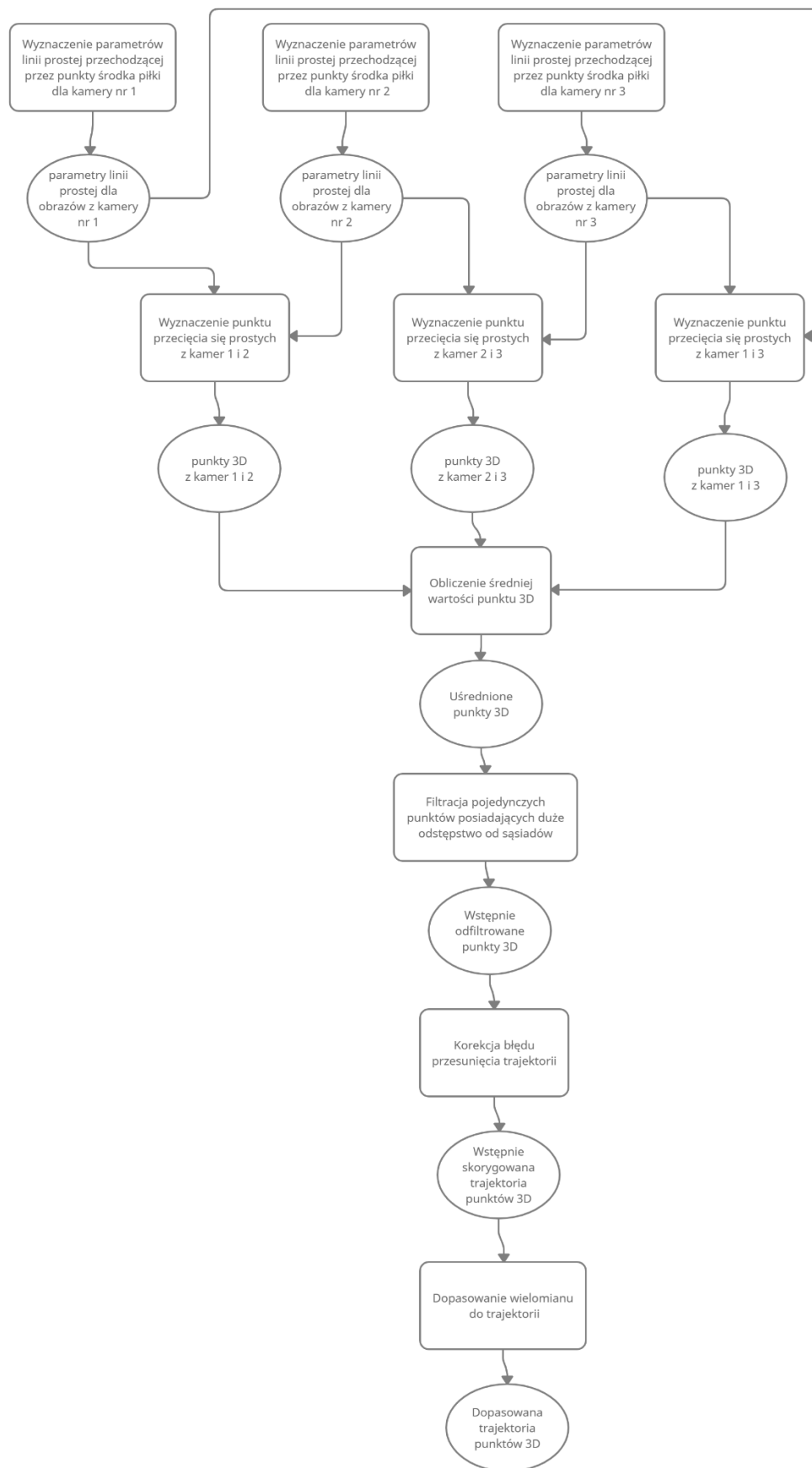
Rysunki nr 2 i 3 przedstawiają schemat blokowy opisujący w skrócie kolejne kroki podejmowane w celu wyznaczenia trajektorii 3D piłki.

W elipsach opisano dane przekazywane na wejściu poszczególnych funkcji oraz dane z nich wychodzące. Prostokąty odpowiadają poszczególnym funkcjom, a treść w nich zawarta opisuje po krótku zadanie jakie realizuje dana metoda.



Rysunek 2. Schemat blokowy programu cz.1

Ciąg dalszy schematu blokowego rozwiązania poczynwszy od końca poprzedniej części schematu:



Rysunek 3. Schemat blokowy programu cz.2.

### 3. Opis sposobu rozwiązania.

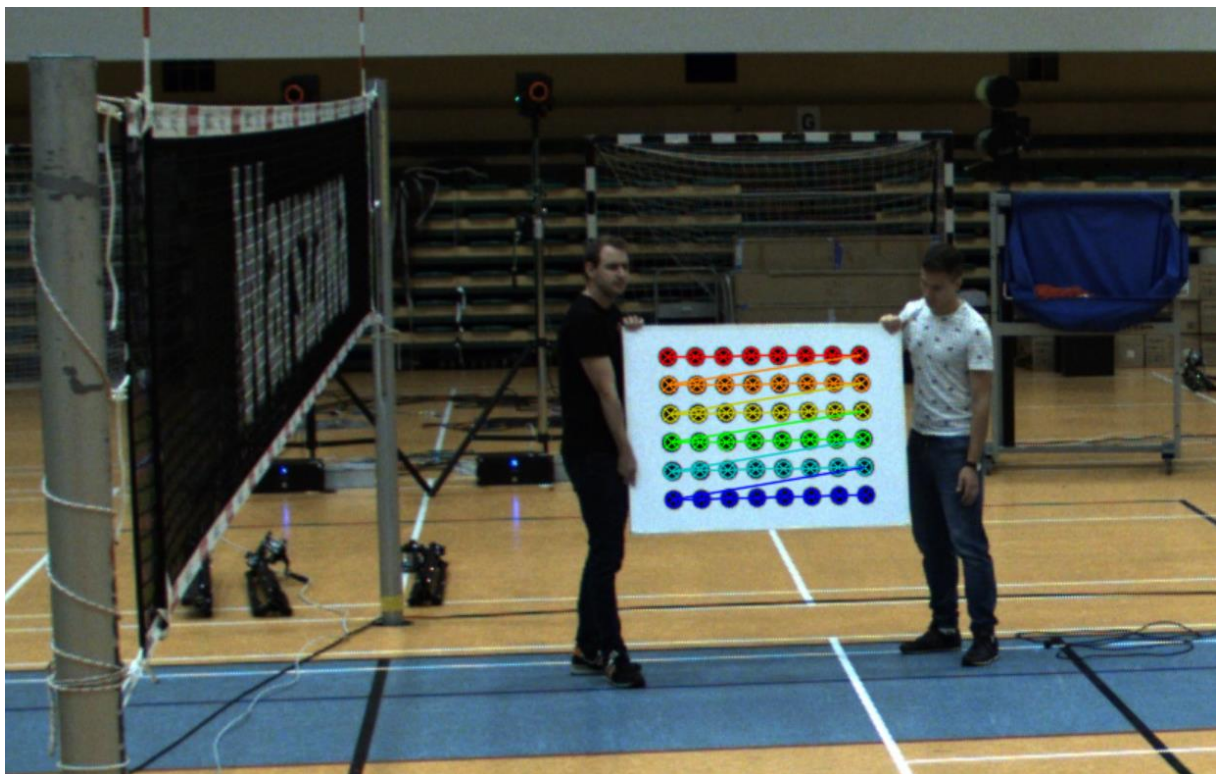
W tym rozdziale zostały opisane kolejno poszczególne kroki podjęte celem rozwiązania zagadnienia.

#### 3.1. Kalibracja wewnętrzna i zewnętrzna kamer

Celem poprawnego wyznaczenia trajektorii należy wpięrow przeprowadzić kalibrację kamer dzięki czemu uzyskamy ich parametry pozwalające na późniejsze wyznaczenie punktów 3D trajektorii piłki na podstawie punktów położenia piłki w 2D oraz parametrów kamer.

Kalibracja wewnętrzna ma na celu wyznaczenie właściwości optycznych kamery. Ich liczba zależy od przyjętego modelu układu optycznego i kamery. Najprostszym z nich jest model kamery otworkowej, który został użyty w tym zadaniu. W tym przypadku parametrami jakie uzyskujemy z kalibracji są: ogniskowa obiektywu(w osi x oraz osi y) oraz rzeczywiste współrzędne środka obrazu kamery( w osi x oraz osi y) jest to tzw. macierz kamery, a także parametry dystorsji. Kalibracji kamery dokonujemy w tym przypadku z użyciem płaskiego obiektu(wzorca) o znanej geometrii umieszczanego w różnych położeniach względem kamery. Pozycja wzorca nie jest w tym przypadku potrzebna. Parametry wzorca wykorzystanego do kalibracji to: wymiar wzorca 8x6 oraz rozstaw okręgów wynoszący 0.12 m.

W projekcie za kalibrację odpowiada metoda `calibrateCamera()` zdefiniowana w module `calibration`. Na rysunku nr 4 przedstawiono wykryty wzorec użyty do kalibracji.

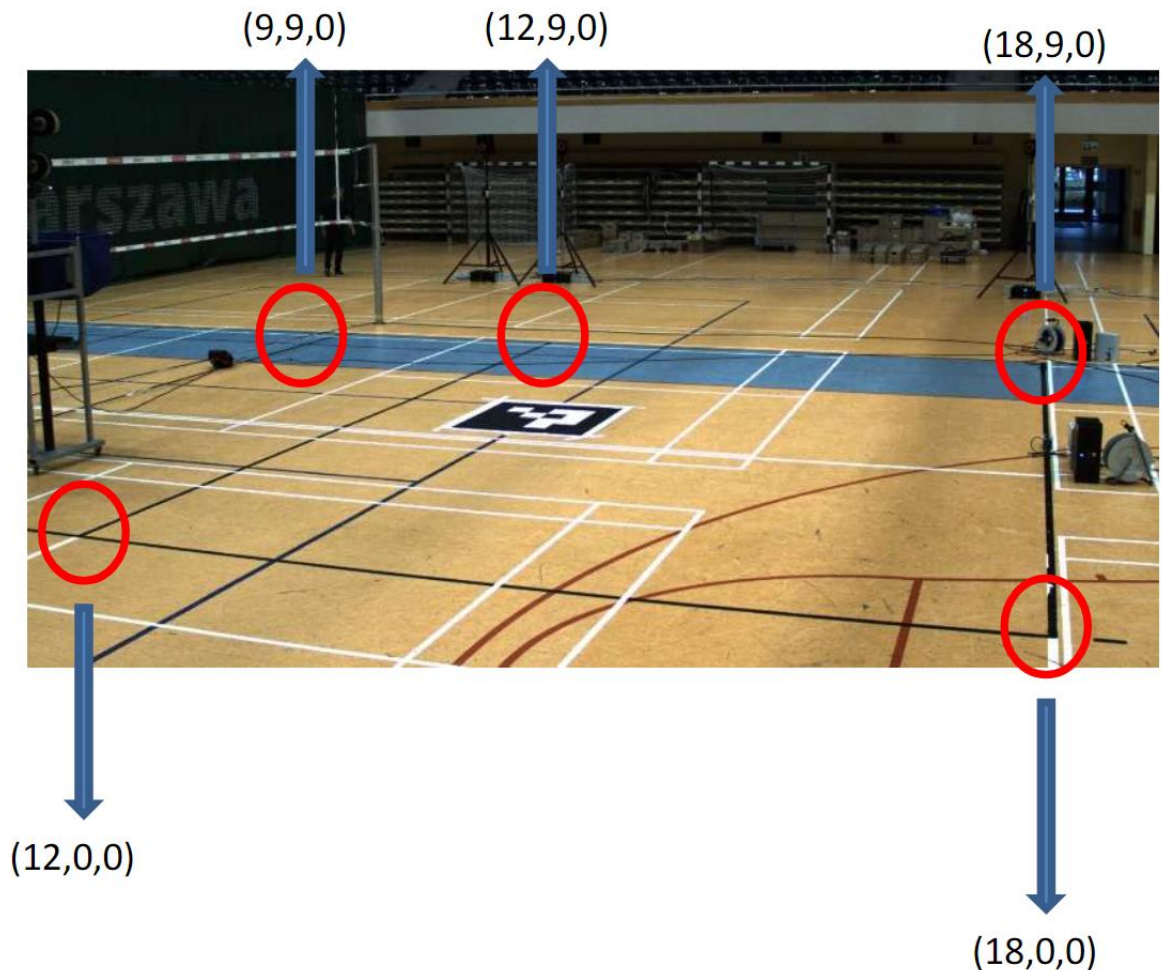


Rysunek 4. Wykryty wzorec okręgów.



Kalibracja zewnętrzna pozwala nam powiązać układy związane z kamerami z globalnym układem odniesienia. W tym przypadku jest to układ odniesienia związany z płaszczyzną boiska. Celem powiązania układów kamer z układem globalnym należy wyznaczyć dla każdej z kamer tzw. parametry zewnętrzne czyli macierze rotacji oraz translacji.

W projekcie punkty 2D dla każdej z kamer zostały wyznaczone ręcznie z użyciem programu paint. Na rysunku nr 5 zaznaczone zostało kilka punktów boiska o znanych współrzędnych w 3D. Punkty te zostały zdefiniowane na stałe na początku programu. Kalibrację zewnętrzną realizuje metoda `solvePnP()` będąca częścią biblioteki OpenCV.



Rysunek 5. Punkty użyte do kalibracji kamery.

### 3.2. Detekcja piłki w 2D

W projekcie do detekcji piłki został użyty algorytm Mixture of Gaussian (MoG), ponieważ tło jest w większości niezmiennie lub występują w nim niewielkie zmiany. Algorytm używa Gaussowskiego modelu statystycznego. Model ten jest ważoną sumą kilku modeli Gaussowskich opisujących charakterystykę tła. W ten sposób pojawiające się obiekty na obrazie wprowadzają znaczące różnice w stosunku do modelu tła dzięki czemu zostają wykryte.

Za detekcję trajektorii piłki w 2D odpowiedzialna jest funkcja `getCoordinates()` zdefiniowana w module `detect_2D`. W przypadku nie znalezienia współrzędnych funkcja wstawia do trajektorii punkt `(-1,-1)` a więc punkt który na obrazie nie istnieje.

Przy użyciu MoG wpierw wczytane zostaje pierwsze 5 obrazów z kamery celem estymacji tła. Nie przeszkadza nam to w detekcji piłki, ponieważ na kilkunastu pierwszych obrazach piłka nie porusza się lub jest nieobecna.

Następnie zaczynamy właściwą detekcję wczytując i przetwarzając kolejne obrazy. Obraz uzyskany z algorytmu MoG poddajemy dylatacji celem połączenia małych regionów oraz redukcji szumu z użyciem algorytmu Gaussian Blur. Następnie obraz MoG poddajemy binaryzacji metodą Otsu co zwalnia nas z konieczności określenia stałego progu binaryzacji, który może okazać się wadliwy przy lekkich zmianach oświetlenia. Rysunek 6 przedstawia przykładowy efekt zastosowania wyżej wspomnianych algorytmów. Rysunek 7 przedstawia odpowiadający rysunkowi 6 obraz z zaznaczoną czerwonym obramowaniem piłką.



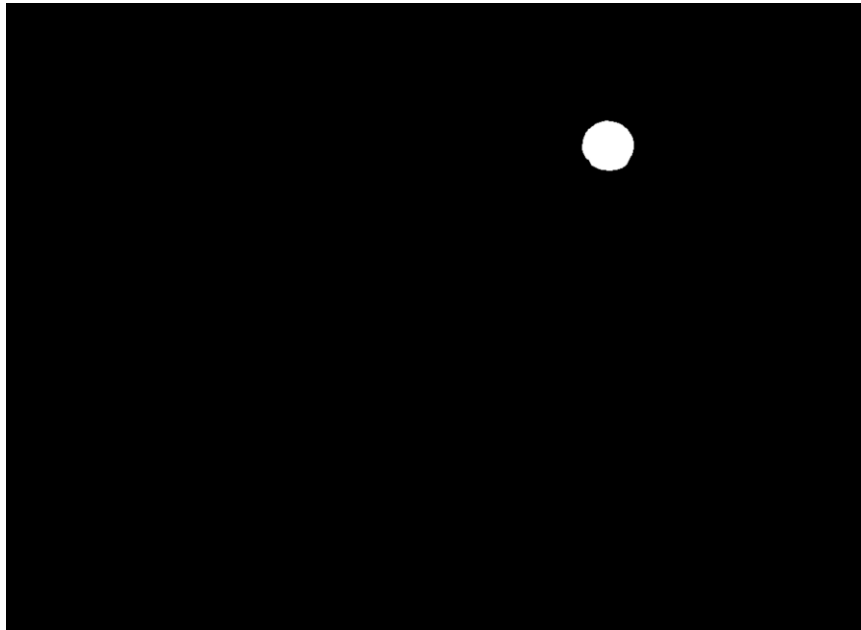
*Rysunek 6. Efekt zastosowania MoG po wstępnej filtracji.*



*Rysunek 7. Obraz z zaznaczoną piłką.*

Jak możemy zauważyć pole powierzchni zajmowane na obrazie przedstawionym na rysunku 6 przez piłkę jest dużo większe od innych przemieszczających się obiektów. W celu pozbycia się niechcianych obiektów innych niż piłka – wszystkie obiekty zajmujące mniejszą niż 1000 pikseli powierzchnię zostaną usunięte.

Efekt przeprowadzenia operacji usunięcia małych obiektów przedstawiono na rysunku 8.



*Rysunek 8. Efekt usunięcia obiektów o małej powierzchni.*

Niekiedy może zdarzyć się, że na obrazie znajdzie się więcej dużych obiektów. Na tę okoliczność w celu odróżnienia piłki od innego obiektu użyta została detekcja krawędzi algorytmem Sobla. Przy odjęciu poprzedniej ramki obrazu z Soblem od obecnej ramki otrzymujemy dość wyraźny obiekt jak przedstawiono na rysunku 9. Warto zaznaczyć, że jedynym dużym obiektem na obrazie jest piłka. Dzieje się tak za sprawą wzorca obecnego na piłce oraz ruchu obrotowego piłki podczas lotu przez co wzorzec nie pozostaje na tym samym miejscu. Niestety piłka jest na tym obrazie dość zniekształcona przez co trudnym będzie uzyskanie współrzędnych jej środka.



*Rysunek 9. Efekt różnicy: poprzedni obraz Sobla - obecny obraz Sobla*



Niemniej jednak po usunięciu wszystkich obiektów mniejszych niż 120 pikseli z obrazu z różnicą Sobli otrzymujemy obraz z niekształtną plamą, która znajduje się w miejscu występowania piłki jak pokazano na rysunku 10.



*Rysunek 10. Efekt filtracji obrazu różnicy Sobli.*

Stosując algorytm MoG w celu detekcji potencjalnych miejsc występowania piłki oraz metody odjęcia Sobli w celu weryfikacji czy dany obiekt jest piłką możemy uzyskać współrzędne środka piłki będących środkiem prostokąta opisanego na obiekcie z algorytmu MoG.

MoG wykrywa także cień rzucany przez lecącą piłkę. Cień ten w momencie styku piłki z ziemią powoduje wykrycie na obrazie MoG kilku dużych obiektów obok siebie. Za piłkę zostanie wówczas obrany obiekt posiadający w swoim obramowaniu(będącym prostokątem opisanym na obiekcie) największą ilość pikseli obiektu z obrazu odejmowania Sobla.

### 3.3. Obliczenie linii przechodzącej przez środek piłki

Na podstawie parametrów z kalibracji zewnętrznej, wewnętrznej oraz współrzędnych środka piłki, możemy wyznaczyć prostą przechodzącą przez ów piksel.

Najpierw dla współrzędnych środka piłki (u,v) wyznaczamy ich znormalizowane odpowiedniki x oraz y.

$$x_{norm} = \frac{(u - c_x)}{f_x} \quad y_{norm} = \frac{(v - c_y)}{f_y}$$

Następnie obliczamy macierze R oraz T gdzie:

T - wektor translacji danej kamery

R - macierz rotacji danej kamery

$$R_{inv} = R^T \quad T_{inv} = -R_{inv}T$$

Ostatnim krokiem jest wyznaczenie parametrycznego równania prostej.

$$A = T_{inv} \quad \alpha = R_{inv}P_{norm}$$

Gdzie prosta opisana jest równaniem:

$$l = A + t \cdot \alpha$$

W projekcie za obliczenie parametrów linii odpowiada metoda lineParams() zdefiniowana w module calculate\_3D\_points. Gdy środek piłki nie jest znany funkcja wpisuje None w wektor kierunkowy.

### 3.4. Obliczenie punktów w 3D

Do obliczenia najbliższego punktu leżącego między dwoma prostymi posłużył niżej przedstawiony pseudokod przedstawiony na wykładzie.

```
A = pos_vector_1
a = dir_vector_1
```

```
B = pos_vector_2
b = dir_vector_2
C = B-A
```

```
ab = Dot(a, b)
ac = Dot(a, c)
bc = Dot(b, c)
aa = Dot(a, a)
bb = Dot(b,b)
```

```
D = A + a * (ac*bb - ab*bc) / (aa*bb - ab*ab)
E = B + a * (ab*ac - bc*aa) / (aa*bb - ab*ab)
```

```
Punkt 3D = 0.5 * (D + E)
```

Dot() – iloczyn skalarny

Za obliczenie punktów w 3D odpowiada funkcja `calculate3DPoints()` zdefiniowana w module `calculate_3D_points`. Gdy punkt w 3D nie istnieje funkcja wpisuje w miejsce punktu `None`.

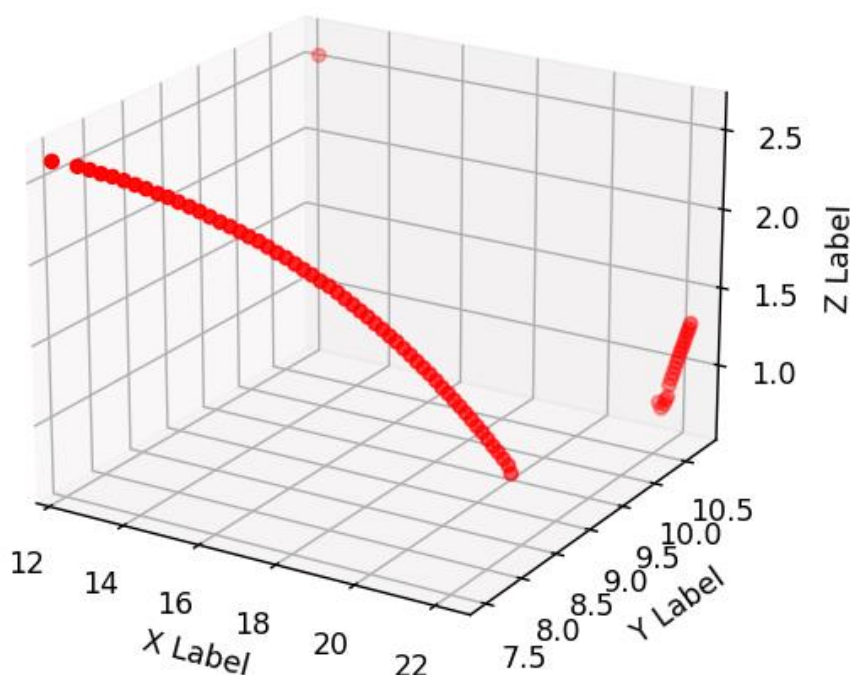
### 3.5. Obliczenie średniej wartości punktów

Obliczone punkty z kamer 1 i 2, 1 i 3 oraz 2 i 3 są następnie uśrednione w jeden punkt stosując średnią arytmetyczną.

Odpowiedzialna jest za to metoda `calculateMeanPoints()` zdefiniowana w module `calculate_3D_points`. Gdy punkt nie istnieje funkcja wpisuje w jego miejsce: `[None, None, None]`

### 3.6. Filtracja wyników

W każdej sekwencji piłka pojawia się i opuszcza pole widzenia każdej z kamer w innych momentach co powoduje nagłe odstępstwo w trajektorii. Ponieważ wiemy, że nasza piłka nie posiada możliwości teleportacji – potrzebna jest metoda, która odpowiednio zminimalizuje błąd „skoku w trajektorii” jak pokazano na rysunku 11.

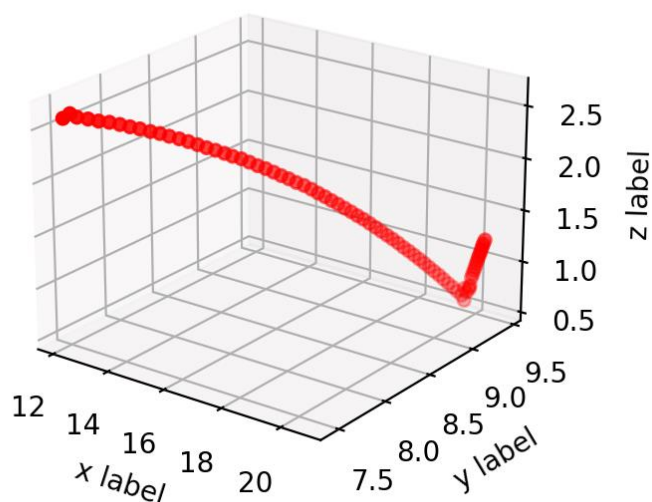


Rysunek 11. Błędy w ciągłości trajektorii.

Funkcja `filterSinglePoints()` zdefiniowana w module `filters` wyszukuje pojedyncze punkty posiadające odstępstwo od sąsiadów i zastępuje je średnią arytmetyczną sąsiadów.

Funkcja `filterDisplacementError()` zdefiniowana w module `filters` wyszukuje moment nieciągłości w trajektorii i przesuwa punkty o wektor w miejsce przewidywanej trajektorii.

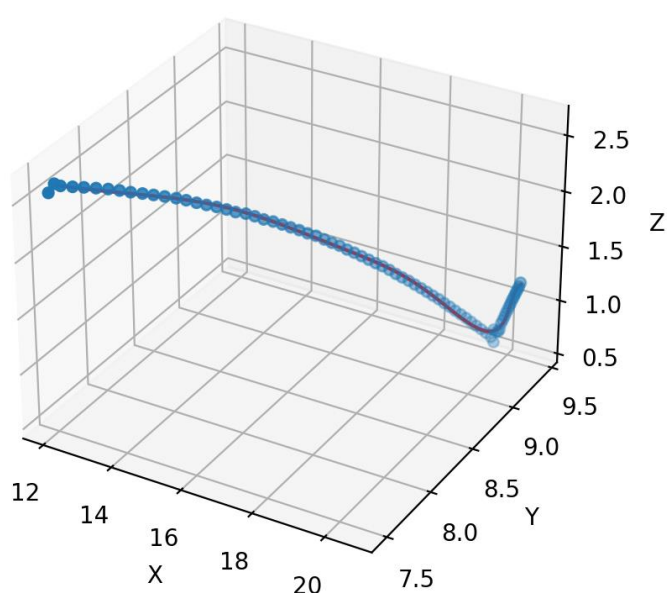
Efekt działania funkcji filtracji pokazano na rysunku 12.



Rysunek 12. Efekt po filtracji trajektorii.

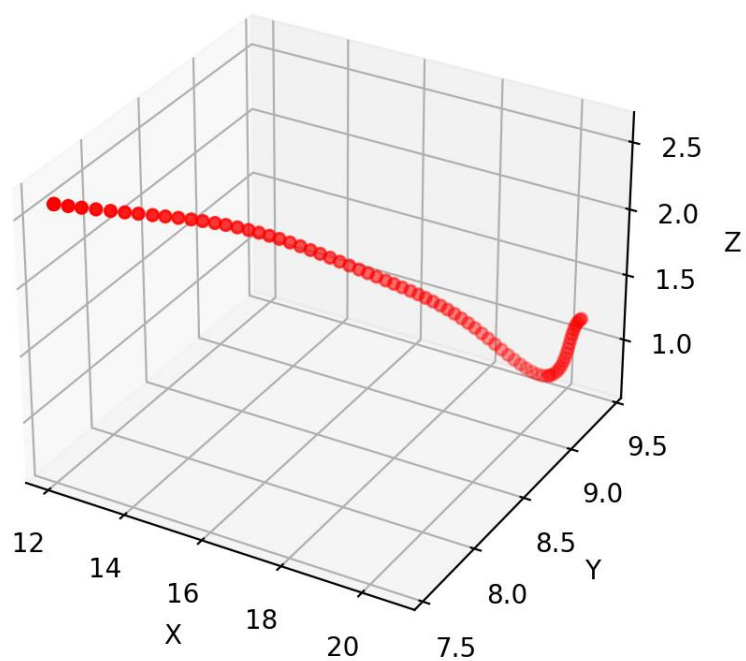
### 3.7. Dopasowanie trajektorii wielomianem

Funkcja `polynomialRegression3D()` zdefiniowana w module `fitting_curve` dopasowuje funkcję do punktów trajektorii 3D traktując współrzędną  $x$  jako zmienną niezależną, a współrzędne  $y$  i  $z$  jako zmienne zależne. Funkcja wykorzystuje metodę dopasowania zaimplementowaną w bibliotece `sklearn`. Funkcja zostaje dopasowana na podstawie wartości RSME(ang. Root Mean Square Error) czyli pierwiastka błędu średniokwadratowego. W efekcie otrzymujemy poprawioną trajektorię lotu piłki. Przykładowe działanie przedstawiono na rysunku numer 13 przedstawiającym czerwonym kolorem dopasowaną funkcję, natomiast kolorem niebieskim oznaczono oryginalne punkty trajektorii 3D.



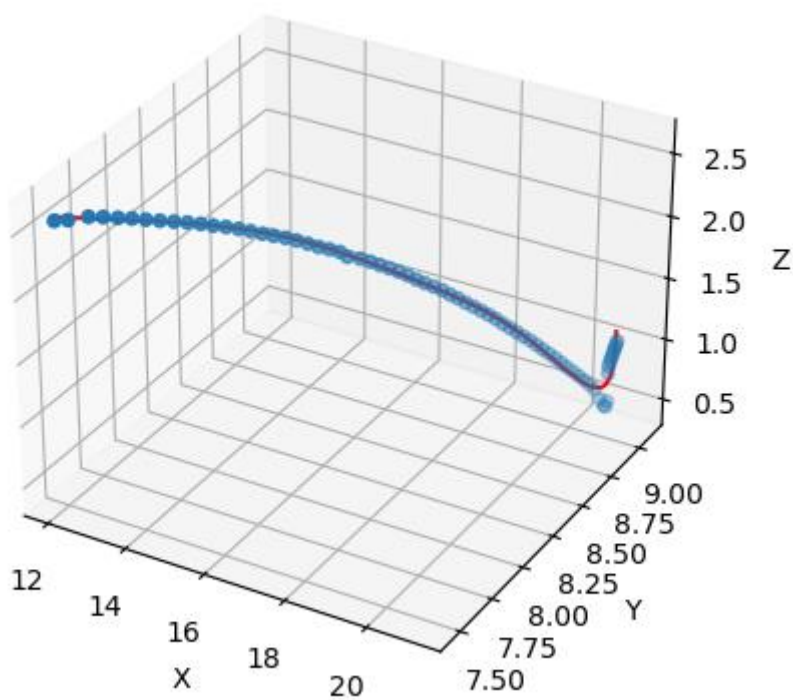
Rysunek 13. Dopasowanie punktów sekwencji 2 wielomianem.

Na rysunku numer 14 przedstawiono punkty sekwencji nr 2 po dopasowaniu do wyznaczonej funkcji.



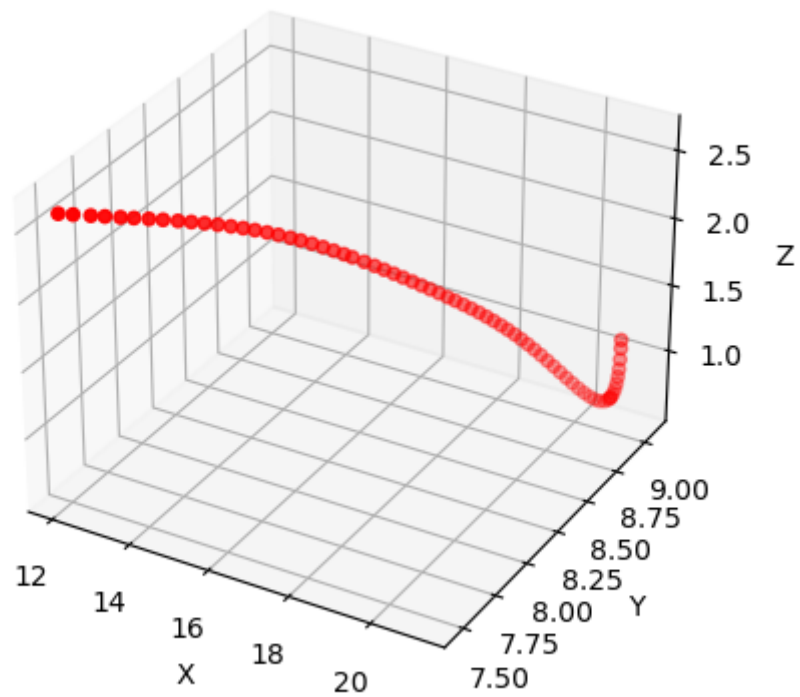
*Rysunek 14. Trajektoria piłki sekwencji 2 po dopasowaniu.*

Kolejne rysunki przedstawiają trajektorię oraz dopasowane punkty do sekwencji 1 i 3.

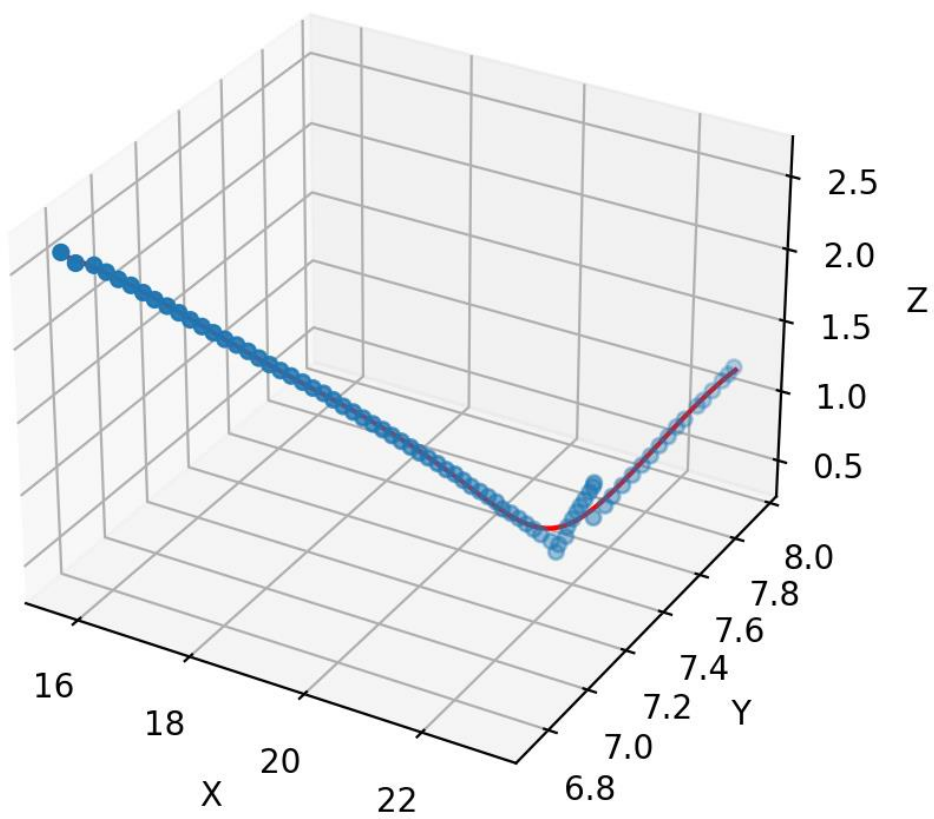


*Rysunek 15. Dopasowanie punktów sekwencji 1 wielomianem.*

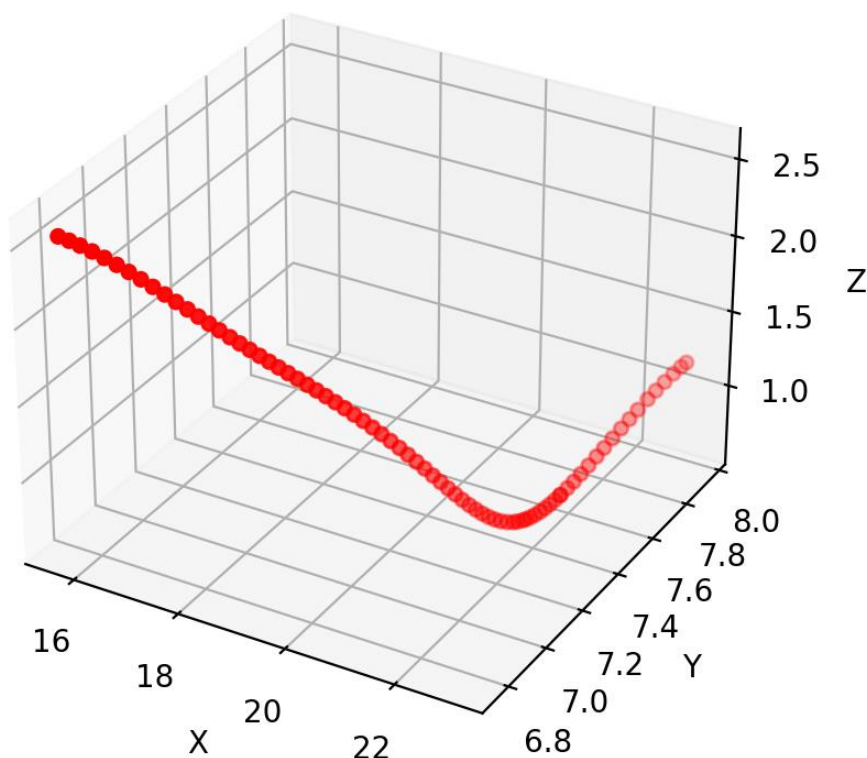




Rysunek 16. Trajektoria piłki sekwencji 1 po dopasowaniu.



Rysunek 17. Dopasowanie punktów sekwencji 3 wielomianem.



Rysunek 18. Trajektoria piłki sekwencji 3 po dopasowaniu.

#### 4. Subiektywna ocena projektu i algorytmów

Użyte algorytmy detekcji oraz ich parametry zostały dopasowane, aby z powodzeniem wyznaczyć trajektorię 3D piłki dla podanych trzech sekwencji. Algorytmy mogą nie zadziałać poprawnie dla innych scenarii, zwłaszcza gdy piłka może zostać na chwilę zasłonięta lub na trybunach będą zasiadać kibice co wprowadzi ruchy w tle. Dodatkowo piłka nie posiadająca na sobie wzoru może nie zostać wykryta algorytmem Sobla co uniemożliwi detekcję.

Zaproponowane w projekcie rozwiązanie niewątpliwie należy do najprostszych i prawdopodobnie prościej można rozwiązać zadanie z użyciem jedynie detekcji krawędzi algorytmem Sobla, jednak dokładność wyznaczenia trajektorii niewątpliwie na tym ucierpi. Obecnie algorytm skutecznie wykrywa i oblicza trajektorię piłki w 3D, a więc spełnia swoje zadanie.

Najwięcej czasu zajmuje obliczanie trajektorii 2D oraz zaraz po niej kalibracja kamer z uwagi na konieczność przetworzenia dużej ilości obrazów w rozdzielczości: 2048 x 1080. Warto jednak zaznaczyć, że kalibrację musimy przeprowadzić jedynie raz dla danego ustawienia kamery.