



**WYDZIAŁ MATEMATYKI  
i INFORMATYKI**

Uniwersytet Łódzki

**Gabriel Ozeg**

Nr albumu: 395263

# **System antykolizyjny na mikroprocesorze Raspberry Pi**

Praca magisterska  
na kierunku Informatyka

Praca wykonana pod kierunkiem  
prof. dr hab. Paweł Zajączkowski  
Katedra Opiekuna

Łódź, 2025

**Słowa kluczowe:** Przetwarzanie obrazu, Głębina obrazu, Metody pomiaru odległości w czasie rzeczywistym, Zastosowania w robotyce

**Title in English:** Collision avoidance system on Raspberry Pi microprocessor

**Keywords:** Image Processing, Image Depth, Real-Time Distance Measurement Methods, Applications in Robotics

# Spis treści

<b>1. Wstęp</b>	5
<b>2. Podstawowe pojęcia</b>	7
2.1. Definicje i własności	7
2.2. Przykłady	7
<b>3. Część główna</b>	9
3.1. Przetwarzanie obrazu	9
3.1.1. Rodzaje kamer i techniki używane do estymacji głębi	9
3.1.1.1. Monocular Vision	9
3.1.1.2. Stereo Vision	10
3.1.1.3. Structure from Motion (SfM)	11
3.1.1.4. Time-of-Flight (ToF)	13
3.1.1.5. Structured Light	14
3.1.1.6. LIDAR (Light Detection and Ranging)	16
3.1.1.7. Event Cameras (Neuromorphic sensors)	17
3.2. Natura kamery	18
3.2.1. Ogniskowa obiektywu	18
3.2.2. Zniekształcenie obiektywu	19
3.3. Obrazowanie stereoskopowe	20
3.3.1. Triangulacja	21
3.3.2. Geometria epipolarna	22
3.3.3. Macierze podstawowe i fundamentalne	23
3.3.4. Macierz obrotu i wektor przesunięcia	24
3.3.5. Rektyfikacja stereo	24
3.3.5.1. Algorytm Hartley'a	25
3.3.5.2. Algorytm Bougueta	25

<b>4. Rozdział badawczy</b>	27
4.1. Opis projektu	27
4.2. Funkcjonalność programu do obrazowania stereo	28
4.2.1. Wykorzystane pakiety	29
4.2.2. Główna pętla	29
4.2.3. Funkcjonalność programu „Take-images-for-calibration.py”	29
4.2.3.1. Kalibracja za pomocą OpenCV	30
4.2.3.2. Wektory kalibracyjne	30
4.2.3.3. Pozyskiwanie obrazów do kalibracji	31
4.2.4. Funkcjonalność programu „Main-Stereo-Vision-Prog.py”	31
4.2.4.1. Kalibracja zniekształceń	31
4.2.4.2. Kalibracja kamery stereo	31
4.2.4.3. Obliczanie mapy rozbieżności	31
4.2.4.4. Zastosowanie filtra WLS (ważonych najmniejszych kwadratów)	31
4.2.4.5. Pomiar odległości	31
4.2.4.6. Możliwe ulepszenia	31
<b>5. Zakończenie</b>	33
<b>Bibliografia</b>	35

# Rozdział 1

## Wstęp

We wstępie pracy dyplomowej powinien znaleźć się opis wkładu własnego studenta w uzyskanie przedstawianych wyników a także informacje o podstawowych źródłach, na podstawie których student przygotował pracę.



## Rozdział 2

### Podstawowe pojęcia

#### 2.1. Definicje i własności

W niniejszej części pracy podane zostaną pojęcia niezbędne w późniejszych rozważaniach (patrz [?] lub [?]).

**Definicja 2.1.1.** *Niech  $G$  będzie niepustym zbiorem. Działaniem w  $G$  nazywamy dowolne odwzorowanie  $\circ : G \times G \rightarrow G$ .*

**Definicja 2.1.2.** *Niech  $G$  będzie niepustym zbiorem,  $\circ$  działaniem w  $G$ . Element  $e \in G$  nazywamy neutralnym (działania  $\circ$ ), jeśli dla każdego  $a \in G$  mamy  $a \circ e = e \circ a = a$ .*

**Lemat 2.1.1.** *Jeśli działanie  $\circ$  w  $G$  posiada element neutralny, to jest on jeden.*

**Dowód.** Niech  $e, e' \in G$  będą dwoma elementami neutralnymi. Wtedy

$$e = e' \circ e = e'. \quad (2.1)$$

Zatem element neutralny jest jeden. ■

#### 2.2. Przykłady

Działaniem w zbiorze liczb naturalnych jest dodawanie, natomiast działaniem w tym zbiorze nie jest odejmowanie.





# Rozdział 3

## Część główna

### 3.1. Przetwarzanie obrazu

Przetwarzanie obrazu to dziedzina informatyki i inżynierii zajmująca się analizą, modyfikacją i interpretacją obrazów cyfrowych za pomocą metod numerycznych i algorytmów komputerowych. Jej celem jest poprawa jakości obrazów, ekstrakcja informacji, segmentacja obiektów lub ich klasyfikacja. Przetwarzanie obrazu znajduje zastosowanie w wielu obszarach, takich jak medycyna (np. analiza zdjęć RTG), przemysł (np. kontrola jakości), bezpieczeństwo (np. rozpoznawanie twarzy), robotyka oraz systemy wizyjne pojazdów autonomicznych. Dzięki połączeniu technik matematycznych, statystycznych i sztucznej inteligencji możliwe jest coraz bardziej precyzyjne i automatyczne rozumienie zawartości obrazów.

#### 3.1.1. Rodzaje kamer i techniki używane do estymacji głębi

##### 3.1.1.1. Monocular Vision



Źródło: <https://cell-kom.com/inne/21454-kamera-internetowa-full-hd-b16-1080p-5900217390350.html>

Monocular vision (widzenie monokularne) to technika pozyskiwania informacji wizualnych przy użyciu tylko jednej kamery, czyli takiej, która rejestruje obraz z pojedynczego

punktu widzenia — podobnie jak jedno oko u człowieka.

#### **Zalety:**

- Niska cena: Wymaga tylko jednej kamery – tańsze niż systemy stereo lub LiDAR.
- Mniejsze zużycie energii: Idealne dla urządzeń mobilnych, dronów, robotów.
- Prosta instalacja: Nie trzeba kalibrować dwóch kamer ani martwić się synchronizacją.
- Łatwo dostępne dane: Każdy smartfon lub laptop może być źródłem obrazu.
- Możliwość działania w czasie rzeczywistym: Przy odpowiedniej optymalizacji algorytmu.

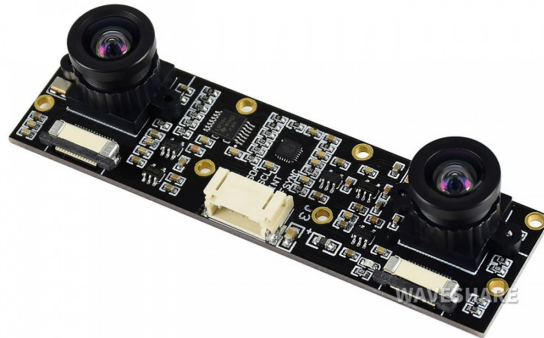
#### **Wady:**

- Brak skali bezwzględnej: Nie wiadomo, czy obiekt jest 1 metr czy 10 metrów od kamery bez dodatkowej informacji (np. IMU, GPS).
- Słabsza dokładność głębokości: W porównaniu do stereowizji lub sensorów ToF/LiDAR.
- Trudności w scenach jednorodnych: Powtarzalne tekstury lub słabe oświetlenie utrudniają analizę.
- Potrzeba ruchu kamery (np. w SLAM): Dla bardziej precyzyjnej rekonstrukcji sceny.

### **3.1.1.2. Stereo Vision**

Stereo vision (widzenie stereoskopowe) to technika komputerowego przetwarzania obrazu, która umożliwia określenie głębokości i kształtu obiektów w scenie poprzez analizę dwóch obrazów zarejestrowanych przez dwie kamery umieszczone w pewnej odległości od siebie (zwykle poziomej — jak oczy u człowieka).

Porównując różnice (czyli dysparycje) w położeniu tych samych punktów na lewym i prawym obrazie, można obliczyć odległość obiektów od kamer — dzięki znanym wzorom geometrycznym (triangulacja).



Źródło: <https://cell-kom.com/inne/21454-kamera-internetowa-full-hd-b16-1080p-5900217390350.html>

#### **Zalety:**

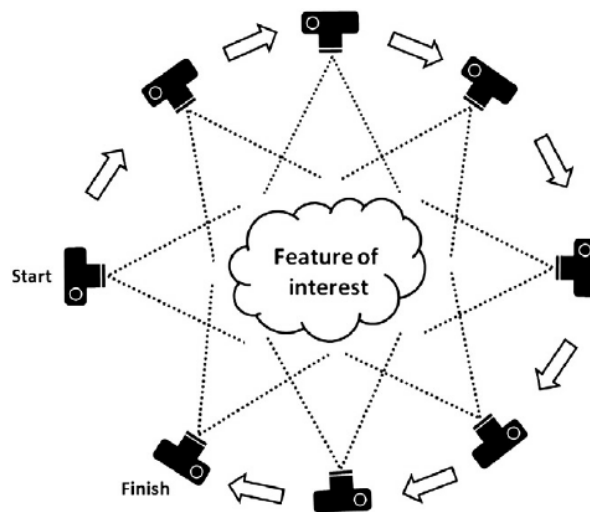
- Prawdziwa głębokość geometryczna — możliwa do przeliczenia na metry.
- Działa w czasie rzeczywistym — przy użyciu odpowiedniego sprzętu (np. ZED Camera, Intel RealSense).
- Nie wymaga poruszania kamerą — wystarczy dwa obrazy z dwóch stałych punktów.
- Nadaje się do zastosowań mobilnych i przemysłowych (robotyka, autonomiczne pojazdy, pomiary 3D).

#### **Wady:**

- Wrażliwość na warunki oświetleniowe — słabe światło, odbicia lub brak tekstury mogą uniemożliwić porównanie obrazów.
- Potrzebna kalibracja — kamery muszą być dokładnie skalibrowane (ustalenie ich pozycji i parametrów).
- Złożoność obliczeniowa — dopasowanie punktów między obrazami może być kosztowne.
- Martwe strefy — obiekty o jednolitej teksturze (np. ściana) mogą nie dawać użytecznych danych.

#### **3.1.1.3. Structure from Motion (SfM)**

Structure from Motion (SfM) to technika wizyjna, która pozwala zrekonstruować strukturę trójwymiarową sceny oraz trajektorię ruchu kamery na podstawie zwykłych zdjęć lub klatek



Źródło: <https://cell-kom.com/inne/21454-kamera-internetowa-full-hd-b16-1080p-5900217390350.html>

wideo wykonanych jedną lub wieloma kamerami, które poruszają się względem sceny.

SfM działa tak: Wykrywa punkty charakterystyczne (np. rogi, krawędzie) na zdjęciach. Śledzi ich pozycje na kolejnych ujęciach. Oblicza względne pozycje kamer i 3D współrzędne punktów w scenie – czyli rekonstruuje kształt i ruch.

#### **Zalety:**

- Umożliwia 3D rekonstrukcję z pojedynczej kamery – wystarczy poruszać kamerą wokół obiektu.
- Niski koszt sprzętowy – można wykorzystać zwykły aparat lub smartfon.
- Brak potrzeby kalibracji sprzętu z góry – SfM może działać nawet na niekalibrowanych zdjęciach.
- Wysoka dokładność w scenach z dobrą teksturą.
- Zastosowania w wielu dziedzinach: archeologia, kartografia, VR, robotyka, fotogrametria.

#### **Wady:**

- Wymaga wielu zdjęć z różnych perspektyw – nie działa dobrze, jeśli kamera stoi w miejscu.
- Problemy w scenach z małą teksturą – np. gładkie ściany, powierzchnie bez szczegółów.
- Czasochłonność obliczeniowa – rekonstrukcja może trwać długo, szczególnie dla dużych zbiorów zdjęć.
- Brak skali bezwzględnej – SfM daje model „w skali względnej”, czyli np. 1 m może być 1 cm, jeśli nie ma referencji.
- Wrażliwe na rozmycie, szum, ruch obiektów w tle.

#### 3.1.1.4. Time-of-Flight (ToF)



Źródło: <https://cell-kom.com/inne/21454-kamera-internetowa-full-hd-b16-1080p-5900217390350.html>

Time-of-Flight (ToF) to technologia, która służy do mierzenia odległości za pomocą światła, wysyłając impuls świetlny (zwykle w postaci podczerwieni) i rejestrując czas, jaki upływa, zanim światło odbije się od obiektu i wróci do czujnika. Dzięki temu możliwe jest uzyskanie dokładnych pomiarów odległości w czasie rzeczywistym.

### **Zalety:**

- Precyzyjne pomiary – ToF pozwala na bardzo dokładne pomiary odległości, nawet w trudnych warunkach, takich jak słabe oświetlenie.
- Szybkość – Technologie ToF oferują bardzo szybkie pomiary, co jest przydatne w systemach wymagających analizy danych w czasie rzeczywistym, np. w robotyce czy samochodach autonomicznych.
- Wielozadaniowość – Dzięki ToF możliwe jest uzyskanie mapy głębi w czasie rzeczywistym, co może być użyteczne w takich dziedzinach jak rozpoznawanie twarzy, zautomatyzowane rozpoznawanie obiektów, monitorowanie przestrzeni 3D itp.
- Wydajność przy różnych warunkach oświetleniowych – Technologia działa dobrze zarówno w pełnym słońcu, jak i w ciemności, co czyni ją bardziej wszechstronną w porównaniu do innych metod pomiarów odległości, takich jak lidar czy radar.

### **Wady:**

- Czułość na zakłócenia – ToF może być wrażliwa na różne źródła zakłóceń, takie jak silne światło zewnętrzne, np. słońce, które może wpłynąć na dokładność pomiaru.
- Ograniczenia w zasięgu – Chociaż ToF oferuje szybkie pomiary, jego zasięg jest ograniczony, a czujniki mogą mieć problemy w przypadku większych odległości.
- Jakość obrazu – W niektórych zastosowaniach, jak np. kamera 3D, ToF może generować dane o niższej jakości w porównaniu do innych technologii, takich jak stereovision, szczególnie w warunkach złożonych.
- Koszt – Zaawansowane czujniki ToF, zwłaszcza te o dużej rozdzielczości, mogą być dość drogie, co może stanowić barierę w masowym wdrożeniu.

#### **3.1.1.5. Structured Light**

Structured Light (światło strukturalne) to technologia wykorzystywana do pomiaru głębi i tworzenia map 3D obiektów. Polega na projekcji wzorca świetlnego (np. linii, siatki, wzorów) na powierzchnię obiektu, a następnie analizie deformacji tego wzorca przez czujnik kamery. Na podstawie tych deformacji obliczane są odległości punktów na powierzchni obiektu, co pozwala na stworzenie mapy głębi.



Źródło: <https://cell-kom.com/inne/21454-kamera-internetowa-full-hd-b16-1080p-5900217390350.html>

### **Zalety:**

- Wysoka dokładność – Dzięki projekcji specyficznych wzorców świetlnych, ta metoda może zapewnić bardzo dokładne pomiary głębi, szczególnie w porównaniu do innych technologii, takich jak stereo vision.
- Szybkość pomiarów – Structured Light pozwala na szybkie tworzenie map 3D, co jest szczególnie przydatne w aplikacjach wymagających analizy w czasie rzeczywistym, jak np. w systemach rozpoznawania twarzy, analizie powierzchni itp.
- Dobre działanie w kontrolowanych warunkach – W przypadku dobrze oświetlonych i kontrolowanych warunków (np. w laboratoriach, produkcji), structured light może zapewnić bardzo wysoką jakość obrazu 3D.
- Kompatybilność z kamerami RGB – Technologia może być używana w połączeniu z tradycyjnymi kamerami RGB, co pozwala na łatwiejsze połączenie danych o głębi z obrazami kolorowymi.

### **Wady:**

- Wrażliwość na oświetlenie – Structured Light jest wrażliwa na zmieniające się warunki oświetleniowe, zwłaszcza w przypadku silnego, niejednorodnego oświetlenia lub cieni. Może to prowadzić do błędów w pomiarach.
- Ograniczona skuteczność na dużych odległościach – Zasięg systemów Structured Light jest zwykle ograniczony do kilku metrów, co sprawia, że nie nadają się one dobrze do pomiarów w dużych przestrzeniach lub na dużych odległościach.
- Problemy z teksturami powierzchni – Obiekty o gładkich powierzchniach lub o jednorodnym kolorze mogą sprawiać trudności, ponieważ brak wyraźnych cech do analizy może prowadzić do mniej dokładnych pomiarów.
- Złożoność systemu – Aby uzyskać dokładne pomiary, systemy Structured Light wymagają zaawansowanego sprzętu do projekcji wzorców i rejestracji obrazu, co może podnieść koszt urządzenia.

### 3.1.1.6. LIDAR (Light Detection and Ranging)



Źródło: <https://cell-kom.com/inne/21454-kamera-internetowa-full-hd-b16-1080p-5900217390350.html>

LIDAR (Light Detection and Ranging) to technologia zdalnego pomiaru odległości, która działa poprzez wysyłanie impulsów laserowych i mierzenie czasu, jaki upływa od ich odbicia od obiektu do powrotu do sensora. Na tej podstawie LIDAR tworzy bardzo dokładne mapy 3D otoczenia.



### **Zalety:**

- Bardzo wysoka dokładność – LIDAR potrafi tworzyć szczegółowe modele 3D otoczenia z dokładnością rzędu centymetrów lub milimetrów.
- Długi zasięg – Niektóre systemy LIDAR mogą mierzyć odległości na setki metrów, co sprawia, że są bardzo przydatne w samochodach autonomicznych, dronach i mapowaniu terenu.
- Działa w nocy – Ponieważ LIDAR używa własnego źródła światła (lasera), działa niezależnie od oświetlenia zewnętrznego, także w całkowitej ciemności.
- Wysoka częstotliwość pomiaru – LIDAR może zbierać dane bardzo szybko, co jest istotne w systemach czasu rzeczywistego, np. podczas jazdy.
- Tworzenie map 3D – Idealnie nadaje się do tworzenia chmur punktów i modeli przestrzennych w 3D, np. w kartografii czy inżynierii.

### **Wady:**

- Wysoki koszt – Profesjonalne systemy LIDAR są kosztowne, co ogranicza ich zastosowanie w tanich produktach konsumenckich.
- Wrażliwość na warunki atmosferyczne – Deszcz, mgła, śnieg czy kurz mogą zaburzyć pomiary lub obniżyć ich dokładność.
- Rozmiar i waga – Choć istnieją miniaturowe wersje, wiele dokładnych systemów LIDAR jest nadal dość duże i ciężkie.
- Brak informacji o kolorze/teksturze – LIDAR dostarcza dane o odległości (głębi), ale nie zawiera informacji o kolorze czy wyglądzie powierzchni (w przeciwieństwie do kamer).
- Ograniczenia w odbiciach od niektórych materiałów – Niektóre powierzchnie, np. czarne lub bardzo błyszczące, mogą słabo odbijać laser, co pogarsza pomiar.

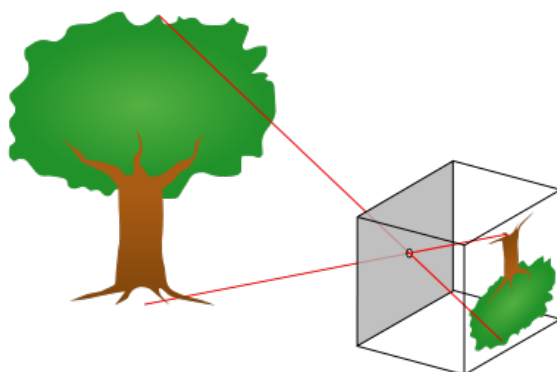
#### **3.1.1.7. Event Cameras (Neuromorphic sensors)**

Capture changes in intensity over time (not full frames).

Can be used for depth estimation when combined with motion.

## 3.2. Natura kamery

Kamery rejestrują promienie świetlne z naszego otoczenia. Zasadniczo kamera działa jak nasze oko, odbite promienie światła z naszego otoczenia docierają do naszego oka i są zbierane na siatkówce. „Kamera otworkowa” jest najprostszym modelem. Jest to dobry uproszczony model do zrozumienia, jak działa kamera. W tym modelu wszystkie promienie światła są zatrzymywane przez powierzchnię. Tylko promienie przechodzące przez otwór są przechwytywane i rzutowane w odwrotnej kolejności na powierzchnię w kamerze. Poniższa ilustracja wyjaśnia tę zasadę



Źródło: <https://funsizephysics.com/use-light-turn-world-upside/>

Zasada ta jest bardzo prosta, ale nie jest to dobry sposób na uchwycenie wystarczającej ilości światła przy szybkiej ekspozycji. Dlatego soczewki są używane do zbierania promieni światła w jednym miejscu. Problem polega na tym, że taki obiektyw powoduje zniekształcenia. Istnieją dwa różne rodzaje zniekształceń:

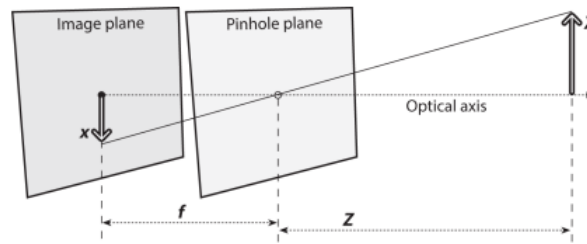
- zniekształcenie promieniowe
- zniekształcenie styczne

Zniekształcenie promieniowe wynika z kształtu samego obiektywu, a zniekształcenie styczne wynika z geometrii kamery. Obrazy można następnie skorygować za pomocą metod matematycznych. Proces kalibracji umożliwia stworzenie modelu geometrii kamery i modelu zniekształceń obiektywu. Modele te tworzą parametry wewnętrzne kamery.

### 3.2.1. Ogniskowa obiektywu

Względny rozmiar obrazu rzutowanego na powierzchnię w kamerze zależy od ogniskowej. W modelu otworkowym ogniskowa to odległość między otworem a obszarem, na który rzutowany jest obraz. Twierdzenie Talesa daje zatem:  $-x = f * (X/Z)$

- $x$ : obraz obiektu (znak minus wynika z tego, że obraz jest odwrócony)
- $X$ : rozmiar obiektu
- $Z$ : odległość od otworu do obiektu
- $f$ : ogniskowa, odległość od otworu do obrazu



Źródło: Learning OpenCV 3, O'Reilly, Str. 639

Ponieważ soczewka nie jest idealnie wyśrodkowana, wprowadzono dwa parametry,  $C_x$  i  $C_y$ , oznaczające odpowiednie poziome i pionowe przeszczenie soczewki. Ogniskowa na osiach  $X$  i  $Y$  są również różne, ponieważ obszar obrazu jest prostokątny. Daje to następujący wzór na położenie obiektu na powierzchni.

$$x_{\text{screen}} = f_x \left( \frac{X}{Z} \right) + c_x, \quad y_{\text{screen}} = f_y \left( \frac{Y}{Z} \right) + c_y$$

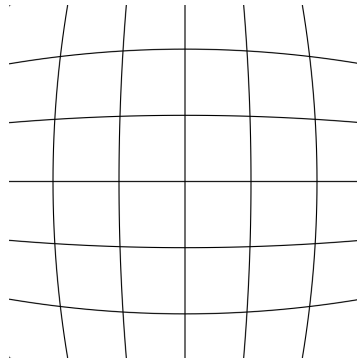
Rzutowane punkty świata rzeczywistego na powierzchnię obrazu można zatem modelować w następujący sposób.  $M$  jest tutaj macierzą wewnętrzną.

$$q = MQ, \quad \text{gdzie} \quad q = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, \quad M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

### 3.2.2. Zniekształcenie obiektywu

Teoretycznie możliwe jest zbudowanie obiektywu, który nie powoduje zniekształceń za pomocą soczewki parabolicznej. W praktyce jednak znacznie łatwiej jest stworzyć soczewkę sferyczną niż paraboliczną. Jak wspomniano wcześniej, istnieją dwa rodzaje zniekształceń. Zniekształcenie promieniowe, które wynikają z kształtu obiektywu i zniekształcenia styczne spowodowane procesem montażu kamery.

W centrum optycznym nie ma zniekształceń promieniowych, a przy zbliżaniu się do krawędzi stają się one coraz większe, gdy zbliżamy się do krawędzi. W praktyce zniekształcenie



Źródło: Wikipedia

to pozostaje niewielkie, wystarczy wykonać rozwinięcie Taylora do trzeciego członu. Daje to następujący wzór.

$$x_{\text{corrected}} = x (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{\text{corrected}} = y (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$x$  i  $y$  to współrzędne oryginalnego punktu na obszarze obrazu, które są używane do obliczenia pozycji skorygowanego punktu. Występuje również zniekształcenie styczne, ponieważ obiektyw nie jest idealnie idealnie równoległa do powierzchni obrazu. Aby to skorygować wprowadzane są dwa dodatkowe parametry,  $p_1$  i  $p_2$ .

$$x_{\text{corrected}} = x + [2p_1 y + p_2 (r^2 + 2x^2)]$$

$$y_{\text{corrected}} = y + [p_1 (r^2 + 2y^2) + 2p_2 x]$$

### 3.3. Obrazowanie stereoskopowe

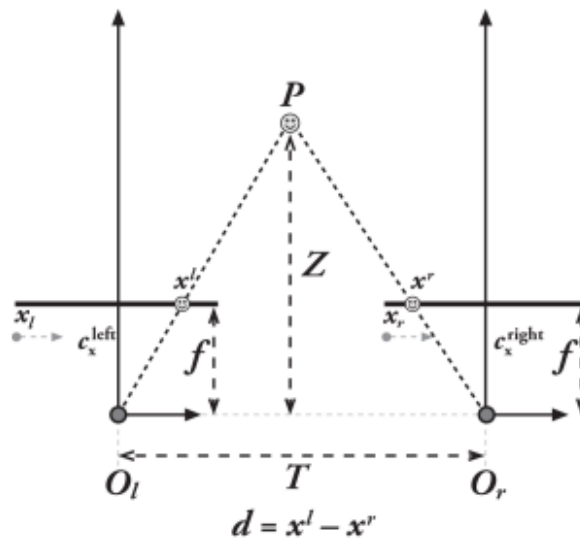
Stereo Vision umożliwia rozpoznawanie głębi na obrazie, wykonywanie pomiarów na obrazie i przeprowadzanie lokalizacji 3D. Między innymi należy znaleźć punkty, które pasują do siebie między dwiema kamerami. Można to następnie wykorzystać do odległości między kamerą a punktem. Wykorzystywana jest geometria systemu w celu uproszczenia obliczeń.

Te cztery kroki są wykonywane podczas obrazowania stereo:

1. usuwanie zniekształceń promieniowych i stycznych za pomocą obliczeń matematycznych obliczenia. W ten sposób powstają obrazy bez deformacji.
2. rektyfikacja kąta i odległości obrazów. Na tym etapie oba obrazy są obrazy współpłaszczyznowe na osi  $Y$ , co ułatwia wyszukiwanie korespondencji. łatwiejsze i wystarczy szukać tylko na jednej osi (osi  $X$ ).

3. znajdź tę samą cechę na prawym i lewym obrazie. Daje to mapę dysproporcji pokazującą różnice między obrazami na osi  $X$ .
4. Ostatnim krokiem jest triangulacja. Mapa rozbieżności jest przekształcana w odległości za pomocą triangulacji.

### 3.3.1. Triangulacja



Źródło: Learning OpenCV 3, O'Reilly, Str. 705

W ostatnim kroku, triangulacji, zakłada się, że oba obrazy projekcji są współpłaszczyznowe i że poziomy rząd pikseli lewego obrazu jest wyrównany z odpowiadającym mu obrazem prawego.

Poniższy obraz można teraz skonstruować przy użyciu poprzednich hipotez.

Punkt  $P$  leży w środowisku i jest pokazany na lewym i prawym obrazie na  $P_l$  i  $P_r$ , z odpowiadającymi im współrzędnymi odpowiadającymi współrzędnymi  $X_l$  i  $X_r$ . To pozwala nam wprowadzić nową wielkość  $d = X_l - X_r$ . Można zauważyć, że im dalej punkt  $P$ , tym mniejsza staje się wielkość  $d$ . Dysproporcja jest zatem odwrotnie proporcjonalna do odległości.

Do obliczenia odległości można użyć następującego wzoru można obliczyć:

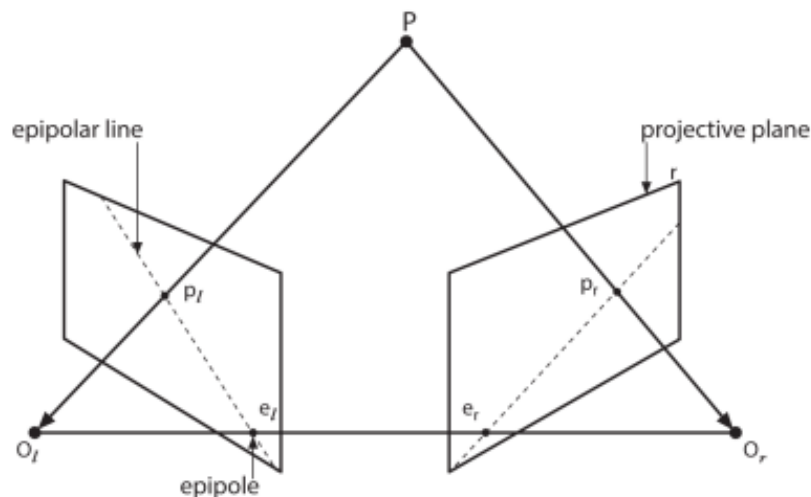
$$Z = f * T / (x_l - x_r)$$

Można zauważyć, że istnieje nieliniowa zależność między rozbieżnością a odległością. Jeśli rozbieżność jest bliska 0, małe różnice w rozbieżności prowadzą do dużych różnic w

odległości. Zjawisko to ulega odwróceniu, gdy rozbieżność jest duża. Małe różnice dysproporcji nie prowadzą do dużych różnic odległości. Na tej podstawie można wywnioskować, że stereowizja ma wysoką rozdzielczość głębi, tylko dla obiektów znajdujących się blisko kamery.

Metoda ta działa jednak tylko wtedy, gdy konfiguracja kamery stereo jest idealna. W rzeczywistości tak nie jest. Właśnie dlatego lewy i prawy obraz są matematycznie wyrównane równoległe. Oczywiście kamery muszą być fizycznie ustawione równoległe. Zanim zostanie wyjaśniona metoda matematycznego wyrównywania obrazów, trzeba najpierw zrozumieć geometrię epipolarną.

### 3.3.2. Geometria epipolarna



Źródło: Learning OpenCV 3, O'Reilly, Str. 709

Powyższy obrazek przedstawia model niedoskonałej kamery stereo składającej się z dwóch modeli kamer otworkowych. Przecięcie linii środków projekcji ( $O_l$ ,  $O_r$ ) z płaszczyznami projekcji tworzone są punkty epipolarne  $e_l$  i  $e_r$ . Linie  $(p_l, e_l)$  i  $(p_r, e_r)$  są nazywane liniami epipolarnymi. Obrazem wszystkich możliwych punktów punktu na płaszczyźnie rzutowania jest linia epipolarna, która leży na drugiej płaszczyźnie obrazu i przechodzi przez punkt epipolarny i szukany punkt. Umożliwia to ograniczenie wyszukiwania punktu na jednym wymiarze zamiast na całej płaszczyźnie.

Można zatem podsumować następujące punkty:

- Każdy punkt 3D w widoku kamery jest zawarty w planie epipolarnym
- Element w jednej płaszczyźnie musi znajdować się na odpowiednich liniach epipolar-

nych drugiej płaszczyzny (warunek epipolarny). drugiej płaszczyzny (warunek epipolarny)

- Dwuwymiarowe wyszukiwanie odpowiadającego elementu jest konwertowane na jednowymiarowe, jeśli znana jest geometria epipolarna.
- Kolejność punktów jest zachowana, tzn. dwa punkty  $A$  i  $B$  są w tej samej kolejności na liniach epipolarnych płaszczyzny.

ta sama kolejność na liniach epipolarnych jednej płaszczyzny, co na liniach epipolarnych drugiej płaszczyzny.

### 3.3.3. Macierze podstawowe i fundamentalne

Aby zrozumieć, w jaki sposób obliczane są linie epipolarne, musimy najpierw wyjaśnić macierze podstawowe i macierze fundamentalne (odpowiadające macierzom  $E$  i  $F$ ). Macierz podstawowa  $E$  zawiera informacje o tym, jak fizycznie rozmieszczone są obie kamery. Opisuje ona lokalizację drugiej kamery względem pierwszej za pomocą parametrów translacji i rotacji. Parametrów tych nie można odczytać bezpośrednio w macierzy, ponieważ jest ona używana do planowania projektu. W sekcji Kalibracja stereo wyjaśnione będzie, jak obliczyć  $R$  i  $T$  (macierz rotacji i wektor translacji). Macierz  $F$  zawiera informacje z podstawowej macierzy  $E$ , fizyczny układ kamer i informacje o wewnętrznych parametrach kamer. Relacja między rzutowanym punktem na lewym obrazie  $p_l$  i rzutowanym punktem na prawym obrazie  $p_r$  jest zdefiniowana następująco:

$$p_r^T E p_l = 0$$

Można by pomyśleć, że ta formuła w pełni opisuje związek między lewym i prawym punktem. prawym punktem. Należy jednak zauważyć, że macierz  $3 \times 3$   $E$  jest rzędu jest rangi 2. Oznacza to, że wzór ten jest równaniem prostej. Aby w pełni zdefiniować relację między punktami, parametry wewnętrzne. parametry wewnętrzne. Pamiętajmy, że  $q = Mp$ , z macierzą wewnętrzną  $M$ . Podstawienie do poprzedniego równania daje wynik:

$$q^T (Ml - 1)^T E Ml - 1 q = 0$$

Podstawienie:

$$F = (Ml - 1)^T E Ml - 1$$

W ten sposób otrzymujemy

$$qr^T F ql = 0$$

### 3.3.4. Macierz obrotu i wektor przesunięcia

Teraz, gdy została wyjaśniona macierz podstawowa  $E$  i macierz podstawowa  $F$ , trzeba zobaczyć, jak obliczyć macierz obrotu i wektor translacji. Zdefiniujemy następujące oznaczenia:

- $P_l$  i  $P_r$  definiują pozycje punktu w układzie współrzędnych odpowiednio lewej i prawej kamery.
- $R_l$  i  $T_l$  (lub  $R_r$  i  $T_r$ ) definiują obrót i translację z kamery do punktu w otoczeniu dla lewej (lub prawej) kamery.
- $R$  i  $T$  to obrót i translacja układu współrzędnych prawej kamery w układzie współrzędnych lewej kamery.

Daje to następujące wyniki

$$P_l = R_l P + T_l \quad P_r = R_r P + T_r$$

Mamy również:

$$Pl = RT(P_r - T)$$

Z tych trzech równań ostateczny wynik to

$$R = R_r R_l^T \quad T = T_r - R T_l$$

### 3.3.5. Rektyfikacja stereo

Dotychczas zajmowaliśmy się tematem „kalibracji stereo”. Chodziło o opis geometrycznego rozmieszczenia obu kamer. Zadaniem rektyfikacji jest rzutowanie dwóch obrazów tak, aby leżały dokładnie w tej samej płaszczyźnie i precyzyjne wyrównanie rzędów pikseli tak, aby linie epipolarne stały się poziome w celu zapewnienia zgodności punktu na dwóch obrazach. aby znaleźć zgodność punktu na dwóch obrazach w sposób bardziej losowy. W wyniku procesu wyrównywania obu obrazów uzyskuje się 8 wyrażeń, po 4 dla każdej kamery:

- wektor zniekształceń
- macierz rotacji  $R_{rect}$ , która musi zostać zastosowana do obrazu



- wyprostowana macierz kamery  $M_{rect}$
- nierektyfikowana macierz kamery  $M$

OpenCV pozwala nam obliczyć te warunki za pomocą dwóch algorytmów: algorytmu Hartley'a i algorytmu Bouguet'a.

#### 3.3.5.1. Algorytm Hartley'a

Algorytm Hartleya wyszukuje te same punkty na obu obrazach. Próbuje on stara się zminimalizować rozbieżności i znaleźć homografie, które ustawiają epipole w nieskończoności. nieskończoność. Dzięki tej metodzie nie jest więc konieczne obliczanie parametrów wewnętrznych dla każdej kamery. dla każdej kamery. Zaletą tej metody jest to, że kalibracja jest możliwa tylko dzięki obserwacji punktów w scenie. punktów na scenie. Główną wadą jest jednak brak skalowania obrazu Masz tylko informacje o względnej odległości. Nie można dokładnie zmierzyć jak daleko obiekt znajduje się od kamer.

#### 3.3.5.2. Algorytm Bouguet'a

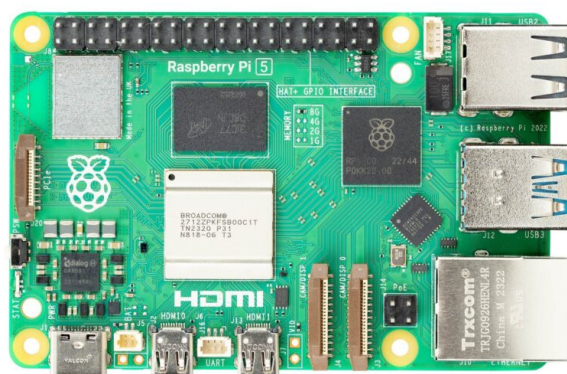
Algorytm Bougueta wykorzystuje obliczoną macierz obrotu i wektor translacji aby obrócić obie rzutowane płaszczyzny o pół obrotu, tak aby znalazły się w tej samej płaszczyźnie. tej samej płaszczyźnie. Sprawia to, że główne promienie są równoległe, a płaszczyzny współpłaszczyznowe. ale nie są jeszcze wyrównane w rzędach. Zostanie to zrobione później. W projekcie wykorzystaliśmy algorytm Bougueta.



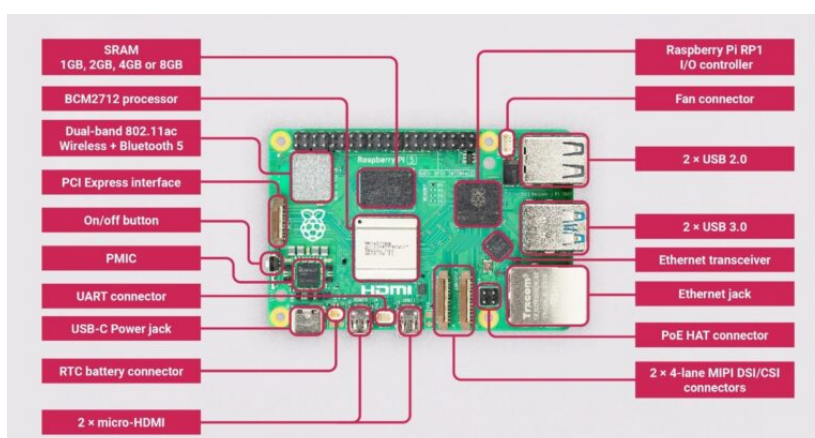
# Rozdział 4

## Rozdział badawczy

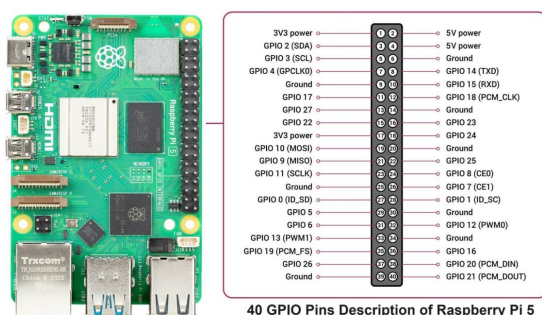
### 4.1. Opis projektu



Źródło: <https://www.hackatronic.com/wp-content/uploads/2023/11/Raspberry-5-pi-.jpg>



Źródło: <https://www.hackatronic.com/wp-content/uploads/2024/03/Raspberry-Pi-5-Pinout-1210x642.jpg>



Źródło: <https://www.hackatronic.com/wp-content/uploads/2024/03/Raspberry-Pi-5-Pinout-1210x642.jpg>



Źródło: <https://pl.aliexpress.com/i/1005006618001887.html>

## 4.2. Funkcjonalność programu do obrazowania stereo

Jak już wspomniano, program jest kodowany w Pythonie i wykorzystywana jest biblioteka OpenCV. jest używana. Zdecydowaliśmy się na język Python i bibliotekę OpenCV, ponieważ mieliśmy już z nimi doświadczenie i ponieważ istnieje wiele dokumentacji na ich temat. Innym argumentem za tą decyzją jest to, że chcieliśmy pracować tylko z bibliotekami „open source”. biblioteki. Na potrzeby tego projektu opracowano dwa programy w języku Python. Pierwszy z nich, „Take-images-for-calibration.py”, służy do robienia dobrych zdjęć, które są później wykorzystywane do kalibracji obu kamer. Później są one wykorzystywane do kalibracji obu kamer (kalibracja zniekształceń i kalibracja stereo). kalibracja. Drugi program, a tym samym główny program „Main-Stereo-Vision-Prog.py” jest używany do obrazowania stereo. jest używany do obrazowania stereo. W tym programie kalibrujemy kamery za pomocą wykonanych zdjęć, generujemy mapę dysproporcji i dzięki doświadczalnemu równaniu

równania, które zostało znalezione eksperymentalnie, możemy zmierzyć odległość dla każdego piksela. pomiar. Na końcu używany jest filtr WLS, aby lepiej rozpoznawać krawędzie obiektów. rozpoznać krawędzie obiektów.

#### **4.2.1. Wykorzystane pakiety**

Do programu zaimportowano następujące pakiety: - Wersja OpenCV.3.2.0 z opencv-contrib (zawiera funkcje stereo) jako „cv2” w Pythonie, zawiera: o bibliotekę do przetwarzania obrazów o funkcje do stereowizji - Numpy.1.12. o Używany do operacji na macierzach (obrazy składają się z macierzy) - Skoroszyt z openpyxl o Pakiet umożliwiający zapisywanie danych w pliku Excel - „normalizacja” biblioteki sklearn 0.18.1 o sklearn umożliwia uczenie maszynowe, ale w tym projekcie używany jest tylko filtr WLS

#### **4.2.2. Główna pętla**

Aby pracować z kamerami, należy je najpierw aktywować. Funkcja `cv2.VideoCapture()` aktywuje obie kamery poprzez wprowadzenie numeru portu każdej z nich. kamery (w programie tworzone są dwa obiekty korzystające z metod klasy `cv2.VideoCapture()`). klasy `cv2.VideoCapture()`.

Aby uzyskać obraz z kamer, używana jest metoda `cv2.VideoCapture().read()` Wyjściem jest obraz sceny, którą kamera ogląda w momencie wywołania tej funkcji. funkcja jest wywoływana. Aby uzyskać obraz wideo, należy wywołać tę metodę w nieskończonej pętli. Aby być bardziej wydajnym, zaleca się aby przekonwertować obrazy BGR na obrazy w odcieniach szarości, odbywa się to za pomocą funkcji `cv2.cvtColor()` funkcja.

Aby wyświetlić wideo na komputerze, używana jest funkcja `cv2.imshow()`. aby otworzyć okno, w którym można wyświetlić wideo.

Przerwanie służy do wyjścia z nieskończonej pętli. Staje się ona aktywna za każdym razem, gdy użytkownik naciśnie spację. Rozpoznanie, że klawisz został naciśnięty jest rozpoznawane dzięki funkcji `cv2.waitKey()`. Wreszcie, dwie używane kamery muszą zostać dezaktywowane za pomocą metody `cv2.VideoCapture().release()`, a otwarte okna są niszczone za pomocą funkcji `cv2.destroyAllWindows()`.

#### **4.2.3. Funkcjonalność programu „Take-images-for-calibration.py”**

Po uruchomieniu tego programu obie kamery stają się aktywne i otwierane są dwa okna. aby użytkownik mógł zobaczyć, gdzie na obrazach znajduje się szachownica.

#### 4.2.3.1. Kalibracja za pomocą OpenCV

Biblioteka OpenCV pozwala nam obliczyć parametry wewnętrzne za pomocą określonych funkcji. Proces ten nazywany jest kalibracją. Jest to możliwe dzięki różnym widoków szachownicy.

Program do robienia zdjęć do późniejszej kalibracji nosi nazwę „Take-images-for-calibration.py”. Jeśli rogi szachownicy zostaną rozpoznane na obu kamerach, otworzą się dwa okna z rozpoznaniem obrazem dla każdej kamery. z rozpoznaniem obrazem dla każdej kamery. Obrazy są następnie zapisywane lub usuwane zapisywane lub usuwane. Można rozpoznać dobre obrazy, na których narożniki są bardzo wyraźnie rozpoznawalne. Obrazy te są później używane do kalibracji w głównym programie programie „Main-Stereo-Vision-Prog.py”. OpenCV zaleca posiadanie co najmniej 10 obrazów dla każdej kamery, aby uzyskać dobrą kalibrację. My uzyskaliśmy dobre wyniki przy 50 obrazów dla każdej kamery. Aby skalibrować kamery, kod Pythona wyszukuje narożniki szachownicy na każdym obrazie dla każdej kamery przy użyciu funkcji OpenCV: `cv2.findChessboardCorners`

Pozycja narożników dla każdego obrazu jest następnie zapisywana w wektorze obrazu, a punkty obiektu dla sceny 3D są zapisywane w innym wektorze. dla sceny 3D są zapisywane w innym wektorze. Następnie należy użyć następnie użyć tych `Imgpoints` i `Objpoints` w funkcji `cv2.calibeCamera()`, której wynikiem jest macierz kamery, współczynniki zniekształceń, wektory obrotu i translacji są zwracane.

Funkcja `cv2.getOptimalNewCameraMatrix()` umożliwia nam uzyskanie dokładnych macierzy kamer. które później wykorzystamy w funkcji `cv2.stereoRectify()`.

Po kalibracji za pomocą OpenCV otrzymujemy następującą macierz  $M$  dla naszej kamery:

Matryca  $M$  bez rektyfikacji (prawa kamera):

Matrix  $M_{rekt}$  Rectified (prawa kamera):

Matryca  $M$  bez rektyfikacji (lewa kamera):

Matryca  $M$  wyprostowana (lewa kamera):

#### 4.2.3.2. Wektory kalibracyjne

Funkcja `cv2.findChessboardCorners()` wyszuka określoną liczbę narożników szachownicy i wygenerowane zostaną następujące wektory:

- `imgpointsR`: zawiera współrzędne narożników na prawym obrazie (w przestrzeni obrazu)
- `imgpointsL`: zawiera współrzędne narożników na lewym obrazie (w przestrzeni obrazu)
- `objpoints`: zawiera współrzędne narożników w przestrzeni obiektu.

Precyzja współrzędnych znalezionych narożników jest zwiększana za pomocą funkcji `cv2.cornerSubPix()`.

#### **4.2.3.3. Pozyskiwanie obrazów do kalibracji**

### **4.2.4. Funkcjonalność programu „Main-Stereo-Vision-Prog.py”**

#### **4.2.4.1. Kalibracja zniekształceń**

#### **4.2.4.2. Kalibracja kamery stereo**

#### **4.2.4.3. Obliczanie mapy rozbieżności**

#### **4.2.4.4. Zastosowanie filtra WLS (ważonych najmniejszych kwadratów)**

#### **4.2.4.5. Pomiar odległości**

#### **4.2.4.6. Możliwe ulepszenia**





## **Rozdział 5**

### **Zakończenie**



# Bibliografia

[1] OpenCV, <https://opencv.org>.

[2] Stereo and Disparity, John Lambert, <https://johnwlambert.github.io/stereo/>.

[3] Stereo and 3D Vision, Rajesh Rao, <https://courses.cs.washington.edu/courses/cse455/09wi/Lects/lect16>.

[4] Learning OpenCV 3, O'Reilly.