



**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

1. Wstęp	5
2. Podstawowe pojęcia	7
2.1. Definicje i własności	7
2.2. Przykłady	7
3. Część główna	9
3.1. Natura kamery	9
3.1.1. Ogniskowa obiektywu	10
3.1.2. Zniekształcenie obiektywu	11
3.2. Rodzaje kamer i technik używane do estymacji głębi	12
3.2.1. Monocular Vision	12
3.2.2. Stereo Vision	13
3.2.3. Structured Light	14
3.2.4. LIDAR (Light Detection and Ranging)	16
3.2.5. Kamery zdarzeniowe	17
3.3. Opis projektu	18
3.4. Obrazowanie stereoskopowe	18
3.4.1. Triangulacja	19
3.4.2. Geometria epipolarna	20
3.4.3. Macierze podstawowe i fundamentalne	21
3.4.4. Macierz obrotu i wektor przesunięcia	22
3.4.5. Rektyfikacja stereo	23
3.4.5.1. Algorytm Hartley’a	23
3.4.5.2. Algorytm Bouguet’a	23
4. Rozdział badawczy	25
4.1. Opis projektu	25

4.2. Funkcjonalność programu do obrazowania stereo	26
4.2.1. Kalibracja kamery	26
4.2.1.1. Kalibracja za pomocą OpenCV	27
4.2.1.2. Wektory kalibracyjne	27
4.2.1.3. Pozyskiwanie obrazów do kalibracji	28
4.2.2. Główny algorytm	28
4.2.2.1. Kalibracja zniekształceń	28
4.2.2.2. Kalibracja kamery stereo	28
4.2.2.3. Obliczanie mapy rozbieżności	28
4.2.2.4. Zastosowanie filtra WLS (ważonych najmniejszych kwa- dratów)	28
4.2.2.5. Pomiar odległości	28
4.2.2.6. Możliwe ulepszenia	28
5. Zakończenie	29
Bibliografia	31

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.

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.

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

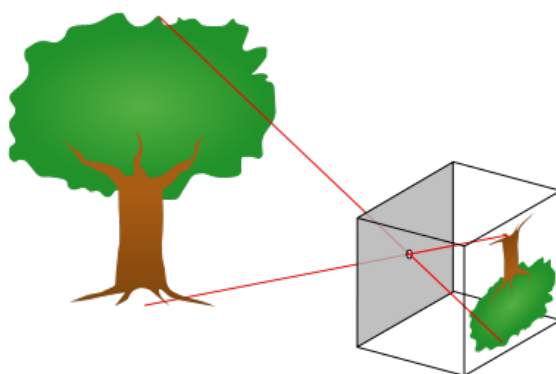
Działaniem w zbiorze liczb naturalnych jest dodawanie, natomiast działaniem w tym zbiorze nie jest odejmowanie.

Rozdział 3

Część główna

3.1. 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

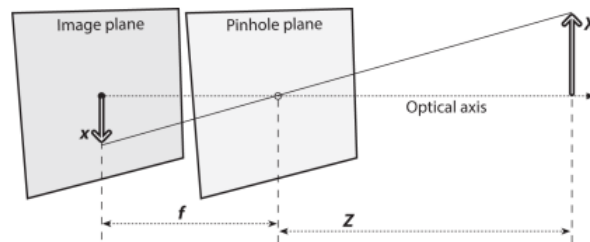
- zniekształcenie styczne

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.

3.1.1. 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

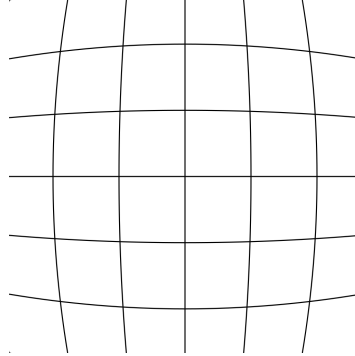
Ponieważ soczewka nie jest idealnie wyśrodkowana, wprowadzono dwa parametry, C_x i C_y , oznaczające odpowiednio poziome i pionowe przemieszczenie soczewki. Ogniskowa na osiach X i Y są również różne, ponieważ obszar obrazu jest prostokątny. Daje to następujący wzór na położenie obiektu na powierzchni.

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

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

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

3.1.2. Zniekształcenie obiektywu



Źródło: Wikipedia

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

$$\begin{aligned}x_{\text{corrected}} &= x (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\y_{\text{corrected}} &= y (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)\end{aligned}$$

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 .

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

3.2. Rodzaje kamer i technik używane do estymacji głębi

3.2.1. 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. W przypadku obrazu monokularnego, każdy piksel dostarcza jedynie informacji o jasności i kolorze w płaszczyźnie 2D. Brakuje natomiast bezpośredniej informacji o głębokości, czyli odległości od kamery. Z tego względu estymacja głębi z pojedynczego obrazu jest niedookreślonym problemem inwersyjnym – wiele różnych trójwymiarowych scen może prowadzić do identycznej projekcji 2D.

Aby rozwiązać ten problem, konieczne jest wprowadzenie priorytetów – dodatkowych założeń na temat struktury świata, geometrii sceny lub charakterystyki obiektów. Przykładowe priorytety to:

- Zakładanie horyzontalności podłoża i pionowości ścian.
- Regularność obiektów (np. ludzie mają podobną wysokość).
- Perspektywa (linie zbiegające się w punkcie zbiegu sugerują głębię).
- Znajomość statystycznych regularności obrazów.

Z matematycznego punktu widzenia, obraz monokularny powstaje na skutek rzutowania sceny trójwymiarowej na płaszczyznę dwuwymiarową za pomocą rzutowania perspektywicznego. Każdy punkt $P = (X, Y, Z)$ w przestrzeni 3D odwzorowany jest na punkt $p = (x, y)$ w obrazie 2D zgodnie z wzorami:

$$x = f \cdot \frac{X}{Z}, \quad y = f \cdot \frac{Y}{Z}$$

gdzie f to ogniskowa kamery, a Z to głębokość. Zauważmy, że głębokość Z znajduje się w mianowniku, co oznacza, że jej zmiany mają kluczowy wpływ na rozmiar i położenie obiektów na obrazie.

W kontekście wizji monokularnej, głębokie uczenie odgrywa kluczową rolę, umożliwiając estymację głębi, rekonstrukcję scen 3D czy detekcję obiektów na podstawie pojedynczego obrazu. Zamiast polegać na klasycznych metodach geometrycznych, takich jak triangulacja czy analiza ruchu, sieci neuronowe uczą się z dużych zbiorów danych, wychwytyując złożone wzorce i zależności przestrzenne. Mimo wysokiej skuteczności, tego typu podejścia mają swoje ograniczenia – wymagają dużej mocy obliczeniowej, są podatne na błędy przy nietypowych danych wejściowych, a ich efektywność jest ściśle związana z jakością i zakresem danych użytych do treningu.

Wady i ograniczenia:

- Brak absolutnej skali – z jednego obrazu nie można jednoznacznie wywnioskować rzeczywistej odległości.
- Trudności w teksturowo jednorodnych obszarach – gdzie brak cech uniemożliwia dobre przewidywanie.
- Problemy z generalizacją – modele trenowane na jednej dziedzinie mogą słabo działać na innych.
- Dynamiczne sceny i obiekty poruszające się niezależnie od kamery – zaburzają proces estymacji.

3.2.2. Stereo Vision



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

Wizja stereoskopowa (lub stereowizja) to technika polegająca na wykorzystaniu dwóch (lub więcej) obrazów tej samej sceny, uchwyconych z nieco innych punktów widzenia, do wyodrębnienia informacji przestrzennych. Jest to jedno z najstarszych i najintensywniej badanych podejść do estymacji głębi, inspirowane sposobem, w jaki ludzkie oczy – jako dwa przesunięte względem siebie punkty obserwacyjne – postrzegają świat trójwymiarowy.

W odróżnieniu od wizji monokularnej, stereowizja oferuje geometrycznie uzasadnioną możliwość bezpośredniego obliczenia głębokości, co czyni ją bardzo atrakcyjną w aplikacjach wymagających wysokiej dokładności. W tym rozdziale przedstawiono podstawy matematyczne wizji stereo, klasyczne i współczesne metody obliczania głębi, a także omówiono

praktyczne zastosowania i ograniczenia tej technologii.

W systemie stereowizyjnym wykorzystuje się dwa obrazy uchwycone przez kamery umieszczone w znanej odległości od siebie (baza stereo). Podstawowym pojęciem jest paralaksa – przesunięcie obrazu tego samego punktu sceny pomiędzy obrazami lewego i prawego oka/kamery.

Zakładając idealną konfigurację (kamery wyrównane, płaszczyzny obrazu równoległe), głębokość Z danego punktu sceny można obliczyć ze wzoru:

$$z = \frac{f \cdot B}{d}$$

- f - ogniskowa kamery.
- B - odległość między kamerami.
- d - przesunięcie danego punktu w obrazie lewym względem prawego.

Im większe d , tym mniejsza głębokość – obiekty bliżej kamery mają większe przesunięcie między obrazami.

Wady i ograniczenia:

- Wymóg kalibracji i synchronizacji kamer – błędy w tym zakresie przekładają się bezpośrednio na błędną głębokość.
- Brak dopasowania w teksturowo ubogich obszarach – np. białe ściany, niebo.
- Problemy przy silnym oświetleniu i odbiciach – zmienność intensywności zaburza dopasowanie.
- Duży koszt obliczeniowy – szczególnie w przypadku metod globalnych lub opartych na deep learningu.
- Widzenie tylko z jednej perspektywy – martwe strefy między kamerami lub poza polem widzenia jednej z nich.

3.2.3. Structured Light



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

Structured Light (pol. światło strukturalne) to technika aktywnej wizji komputerowej wykorzystywana do precyzyjnego pomiaru kształtu i głębokości obiektów. Polega na projekcji znanego wzorca świetlnego (np. siatki, kropek, pasków) na powierzchnię sceny, a następnie analizie deformacji tego wzorca za pomocą kamery. Technologia ta jest szeroko stosowana w systemach 3D skanowania, rozpoznawania gestów, robotyce, a także w popularnych urządzeniach konsumenckich, takich jak Kinect v1

System Structured Light składa się zazwyczaj z dwóch komponentów:

- Projektora – emituje wzorec świetlny (np. siatkę punktów lub paski) na obserwowaną scenę.
- Kamery – rejestruje zniekształcony wzorec po odbiciu od obiektów w przestrzeni.

Proces działa następująco:

1. Znany wzorec zostaje wyświetlony na scenie.
2. Gdy wzorec napotyka obiekty o różnych kształtach i odległościach, zostaje geometrycznie zniekształcony.
3. Kamera rejestruje te deformacje.
4. System porównuje zarejestrowany obraz wzorca ze wzorcem referencyjnym, który byłby widoczny na płaskiej powierzchni.
5. Na podstawie różnic (tzw. disparity) obliczana jest głębokość – za pomocą triangulacji.

Wady i ograniczenia:

- W jasnym świetle dziennym (szczególnie na zewnątrz), wzorec świetlny może zostać zaburzony lub całkowicie zaniknąć – szczególnie jeśli działa w paśmie IR.
- Technika najlepiej sprawdza się na krótkich dystansach (0,5–2 m). Dalsze obiekty dają mniej wyraźne zniekształcenia wzorca.
- Szkło, lustra, woda lub powierzchnie metaliczne mogą zaburzyć wzorec lub wprowadzać wielokrotne odbicia.
- Projektor i kamera muszą być precyzyjnie skalibrowane względem siebie – błędy kalibracji mogą znacząco wpłynąć na jakość głębi.
- Gdy wzorec nie dotrze do części sceny (np. w załomach, pod kątem), pomiar głębokości w tych miejscach będzie niemożliwy.

3.2.4. 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.

Podstawowy mechanizm działania LiDAR opiera się na bardzo prostej zasadzie:

Sensor emituje impuls laserowy w kierunku otoczenia.

Światło odbija się od powierzchni obiektów i wraca do detektora.

System mierzy czas, jaki upłynął od wysłania do odebrania sygnału (Time-of-Flight, ToF).

Znając prędkość światła, obliczana jest dokładna odległość:

- Sensor emituje impuls laserowy w kierunku otoczenia.
- Światło odbija się od powierzchni obiektów i wraca do detektora.
- System mierzy czas, jaki upłynął od wysłania do odebrania sygnału (Time-of-Flight, ToF).
- Znając prędkość światła, obliczana jest dokładna odległość:

$$d = \frac{c \cdot \Delta t}{2}$$

gdzie:

d – odległość do obiektu

c – prędkość światła (ok. $3 \cdot 10^8$ m/s)

Δt – czas przelotu sygnału

LiDAR-y mogą wykonywać takie pomiary miliony razy na sekundę, skanując środowisko w 2D (jeden plan) lub 3D (pełna chmura punktów).

Wady i ograniczenia:

- Mgła, deszcz, śnieg i kurz mogą zakłócać odbicie promieni lasera, co wpływa na dokładność pomiarów.
- Brak dopasowania w teksturowo ubogich obszarach – np. białe ściany, niebo.
- LiDAR rejestruje wyłącznie dane geometryczne – nie dostarcza żadnych informacji o kolorze czy teksturze powierzchni.
- W porównaniu do kamer, LiDAR-y mają stosunkowo rzadką siatkę pomiarową, co skutkuje niższą rozdzielczością przy dużych odległościach (np. obiekt 100 m dalej może być opisany przez kilka punktów).
- Bardzo ciemne lub przezroczyste powierzchnie (np. szyby) mogą słabo odbijać impulsy lasera lub w ogóle je przepuszczać.

3.2.5. Kamery zdarzeniowe

Kamery zdarzeniowe (ang. Event Cameras) to innowacyjne sensory wizyjne, które różnią się fundamentalnie od tradycyjnych kamer opartych na matrycy CMOS. Zamiast przechwytywać obraz w sposób klatkowy (frame-based), rejestrują one zmiany jasności na poziomie pojedynczych pikseli, co pozwala na znacznie wyższą rozdzielczość czasową i lepszą reakcję na dynamiczne sceny. Dzięki temu technologia ta znajduje coraz szersze zastosowanie w systemach robotycznych, autonomicznych pojazdach, AR/VR i przetwarzaniu sygnałów w czasie rzeczywistym.

W tradycyjnych kamerach każda klatka rejestrowana jest w określonym interwale czasowym, co powoduje powstawanie rozmycia ruchu i dużego opóźnienia w dynamicznych scenach. Kamery zdarzeniowe działają zupełnie inaczej:

Każdy piksel działa niezależnie i stale monitoruje zmiany lokalnej jasności.

Gdy zmiana przekroczy ustalony próg (np. 10

Zdarzenie zawiera informację o:

- położeniu piksela (x, y),
- czasie zdarzenia (z dokładnością do mikrosekund),
- polaryzacji zmiany (jasność wzrosła lub zmalała).

Dzięki temu kamera generuje strumień asynchronicznych zdarzeń, a nie szereg klatek. Przykładami takich kamer są m.in. DVS (Dynamic Vision Sensor), DAVIS (łączy klasyczną kamerę z kamerą zdarzeniową) oraz CeleX.

Brak informacji o statycznych obiektach: jeśli scena się nie zmienia, kamera nie generuje zdarzeń – co utrudnia pełną rekonstrukcję otoczenia.

Trudności w przetwarzaniu danych: strumień zdarzeń ma inną strukturę niż klasyczne obrazy – wymaga specjalnych algorytmów i często dedykowanego sprzętu (np. FPGA).

Niska rozdzielczość przestrzenna: w porównaniu do tradycyjnych kamer, choć technologia ta dynamicznie się rozwija.

Szum przy słabym oświetleniu: niektóre sensory są bardziej podatne na fałszywe zdarzenia w nocy lub w ciemnych pomieszczeniach.

Koszt: kamery zdarzeniowe są wciąż relatywnie drogie i mniej dostępne komercyjnie.

Wady i ograniczenia:

- Wymóg kalibracji i synchronizacji kamer – błędy w tym zakresie przekładają się bezpośrednio na błędną głębokość.
- Brak dopasowania w teksturowo ubogich obszarach – np. białe ściany, niebo.
- Problemy przy silnym oświetleniu i odbiciach – zmienność intensywności zaburza dopasowanie.
- Duży koszt obliczeniowy – szczególnie w przypadku metod globalnych lub opartych na deep learningu.
- Widzenie tylko z jednej perspektywy – martwe strefy między kamerami lub poza polem widzenia jednej z nich.

3.3. Opis projektu

3.4. Obrazowanie stereoskopowe

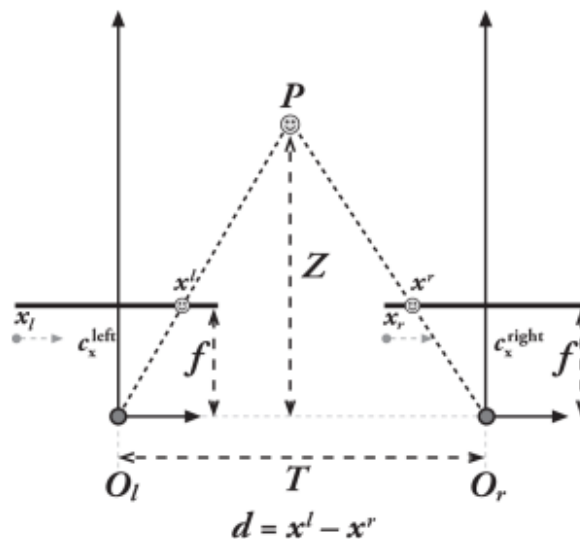
Stereo Vision umożliwia rozpoznawanie głębi na obrazie, wykonywanie pomiarów na obrazie i przeprowadzanie lokalizacji 3D. Między innymi należy znaleźć punkty, które pasują

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 obrazami 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.

3.4.1. 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.

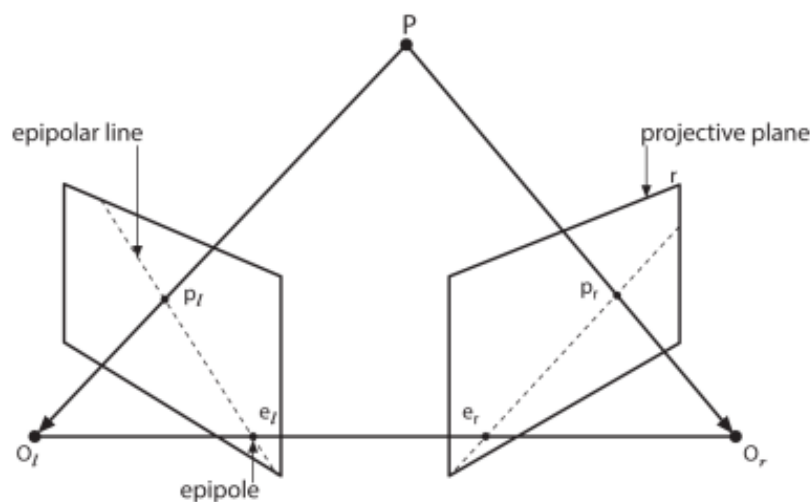
Do obliczenia odległości można użyć następującego wzoru:

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

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.

3.4.2. 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 epipolarnych 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.

3.4.3. 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. Należy jednak zauważyć, że macierz 3×3 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:

$$qrT(Ml - 1)TEml - 1ql = 0$$

Podstawienie:

$$F = (Ml - 1)TEml - 1$$

W ten sposób otrzymujemy

$$qrTFql = 0$$

3.4.4. Macierz obrotu i wektor przesunięcia

Teraz, gdy została wyjaśniona macierz podstawową E i macierz podstawową 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:

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

Z tych trzech równań ostateczny wynik to

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

3.4.5. 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 wyrażań, 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.

3.4.5.1. 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.

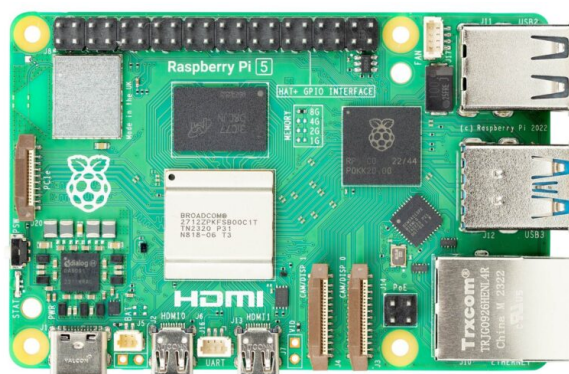
3.4.5.2. Algorytm Bouguet’a

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.

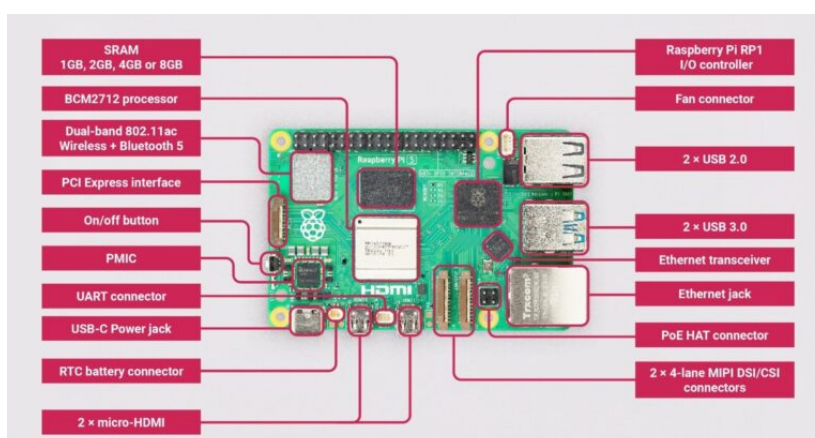
Rozdział 4

Rozdział badawczy

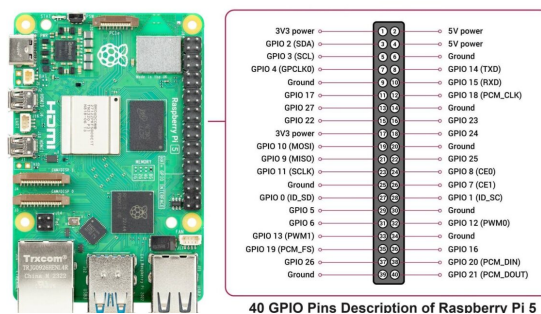
4.1. Opis projektu



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. Kalibracja kamery

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

4.2.1.1. 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 `Mrekt Rectified` (prawa kamera):

Matryca `M` bez rektyfikacji (lewa kamera):

Matryca `M` wyprostowana (lewa kamera):

4.2.1.2. 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()`.

4.2.1.3. Pozyskiwanie obrazów do kalibracji

4.2.2. Główny algorytm

4.2.2.1. Kalibracja zniekształceń

4.2.2.2. Kalibracja kamery stereo

4.2.2.3. Obliczanie mapy rozbieżności

4.2.2.4. Zastosowanie filtra WLS (ważonych najmniejszych kwadratów)

4.2.2.5. Pomiar odległości

4.2.2.6. 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.