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, Głębia 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. Rodzaje kamer i techniki używane do estymacji głębi 9

3.1.1. Monocular Vision 9

3.1.2. Stereo Vision 9

3.1.3. Multiview / Structure from Motion (SfM) 10

3.1.4. Time-of-Flight (ToF) Cameras 10

3.1.5. Structured Light 11

3.1.6. LIDAR (Light Detection and Ranging) 11

3.1.7. Ultrasound / Sonar 11

3.1.8. Event Cameras (Neuromorphic sensors) 12

3.2. Czym jest stereowizja? 12

3.2.1. Stereowizja w robotach do pomieszczeń czystych 12

3.3. Model kamery 12

3.3.1. Ogniskowa obiektywu 13

3.3.2. Zniekształcenie obiektywu 14

3.3.3. Kalibracja za pomocą OpenCV 15

3.4. Obrazowanie stereoskopowe 16

3.4.1. Wyjaśnienie 16

3.4.2. Triangulacja 16

3.4.3. Geometria epipolarna 18

3.4.4. Macierze podstawowe i fundamentalne 18

3.4.5. Macierz obrotu i wektor przesunięcia 19

3.4.6. Rektyfikacja stereo . . . 20
3.4.6.1. Algorytm Hartley'a . . . 20
3.4.6.2. Algorytm Bougueta . . . 20

4. Rozdział badawczy . . . 21
4.1. Raspberry Pi . . . 21
4.2. Camera . . . 22
4.3. Funkcjonalność programu do obrazowania stereo . . . 22
4.3.1. Wykorzystane pakiety . . . 23
4.3.2. Główna pętla . . . 23
4.3.3. Funkcjonalność programu „Take-images-for-calibration.py" . . . 24
4.3.3.1. Wektory kalibracyjne . . . 24
4.3.3.2. Pozyskiwanie obrazów do kalibracji . . . 24
4.3.4. Funkcjonalność programu „Main-Stereo-Vision-Prog.py" . . . 24
4.3.4.1. Kalibracja zniekształceń . . . 24
4.3.4.2. Kalibracja kamery stereo . . . 24
4.3.4.3. Obliczanie mapy rozbieżności . . . 24
4.3.4.4. Zastosowanie filtra WLS (ważonych najmniejszych kwadratów) . . . 24
4.3.4.5. Pomiar odległości . . . 24
4.3.4.6. Możliwe ulepszenia . . . 24

5. Zakończenie . . . 25

Bibliografia . . . 27

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 [2] lub [**?**]).

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

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

$$e = e' \circ e = e'. \tag{2.1}$$

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. Rodzaje kamer i techniki używane do estymacji głębi

3.1.1. Monocular Vision

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

Single camera.

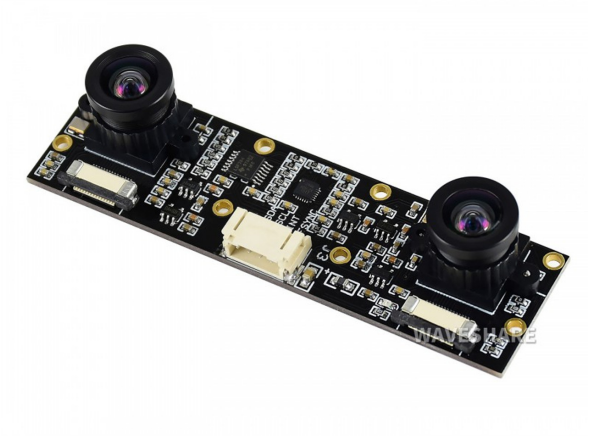
Relies on cues like motion, texture gradients, perspective, learned priors (deep learning).

Examples: Depth estimation from a single image using CNNs or transformers.

3.1.2. Stereo Vision

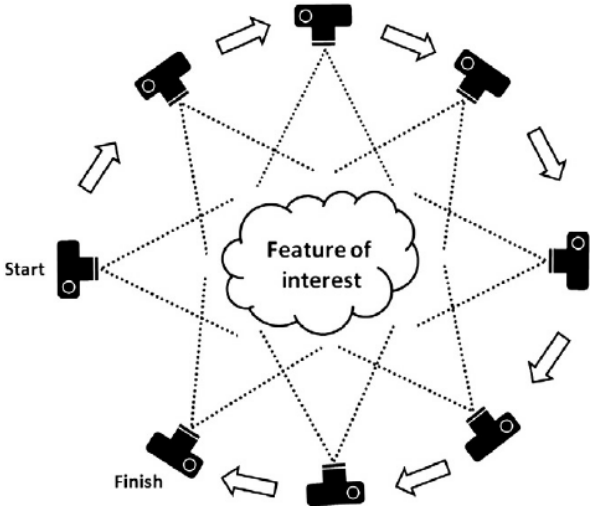
Two cameras with known relative positions.

Depth estimated via triangulation from pixel disparity.



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

3.1.3. Multiview / Structure from Motion (SfM)



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

Uses multiple images taken from different viewpoints (can be from a moving camera).

Reconstructs 3D structure by tracking features and estimating camera poses.

3.1.4. Time-of-Flight (ToF) Cameras

Emits infrared (IR) light and measures the time it takes to bounce back.

Direct depth measurement.

Common in smartphones (e.g., some iPhones, Kinect v2).

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

3.1.5. Structured Light

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

Projects a known light pattern (e.g., dots or stripes) onto a scene.
Deformation of the pattern gives depth info.
Used in original Microsoft Kinect.

3.1.6. LIDAR (Light Detection and Ranging)

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

Emits laser beams and measures return time.
Produces very accurate 3D point clouds.
Widely used in autonomous vehicles.

3.1.7. Ultrasound / Sonar

Common in robotics, underwater applications, and medical imaging.
Measures distance using the echo of sound waves.

3.1.8. 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. Czym jest stereowizja?

Stereoskopia to proces polegający na przechwyceniu 2 obrazów tej samej sceny w celu utworzenia mapy dysproporcji sceny. mapa rozbieżności sceny. Z tej mapy rozbieżności można zmierzyć odległość do obiektu i utworzyć mapę 3D sceny. do obiektu i utworzyć mapę 3D sceny.

3.2.1. Stereowizja w robotach do pomieszczeń czystych

Celem tego dużego projektu jest opracowanie robota do pomieszczeń czystych, który może mierzyć cząsteczki w całym pomieszczeniu czystym. w całym pomieszczeniu czystym. Aby przeprowadzić pomiar bez żadnych problemów, robot musi orientować się bez kolizji. robot musi orientować się bez kolizji. Do tej orientacji zdecydowaliśmy się użyć tylko kamer, a do pomiaru odległości od obiektu potrzebujemy co najmniej dwóch kamer co najmniej dwóch kamer (stereowizja). Dwie kamery znajdują się w odległości 110 mm między sobą.

Im większa jest ta odległość, tym lepiej można ocenić odległość do sceny dla obiektów znajdujących się daleko. obiektów, które są daleko. W tym projekcie zaimplementowano program Python wykorzystujący bibliotekę OpenCV do kalibracji kamer i pomiaru odległości do obiektów na scenie. obiektów w scenie. (Patrz załącznik)

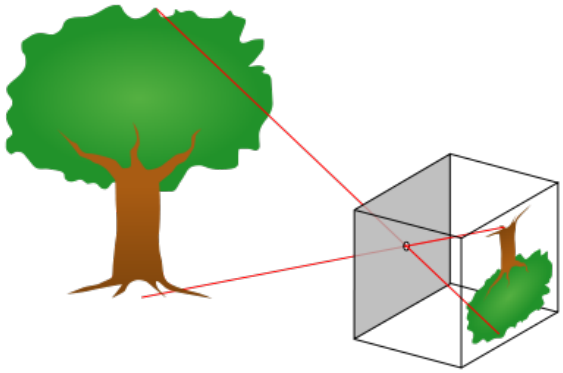
Kamera stereo składa się z dwóch kamer Logitech Webcam C170.

Widoki wideo są optymalne przy obrazach 640x480 pikseli. Ogniskowa: 2,3 mm. Wbudowana kamera stereo:

3.3. Model 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. Są one gromadzone na naszej siatkówce. „Kamera otworkowa” jest najprostszym modelem. Jest to dobry uproszczony model do zrozumienia zrozumieć, jak działa kamera. W tym modelu wszystkie promienie światła są zatrzymywane przez powierzchnię. powierzchnię. Tylko promienie przechodzące przez otwór są przechwytywane i rzutowane

w odwrotnej kolejności na powierzchnię w kamerze. powierzchnia w kamerze. Poniższa ilustracja wyjaśnia tę zasadę



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

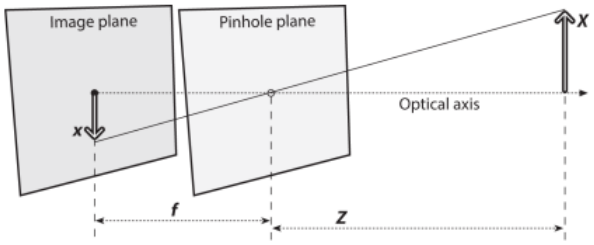
Zasada ta jest bardzo prosta, ale nie jest to dobry sposób na uchwycenie wystarczającej ilości światła przy szybkiej ekspozycji. szybkiej ekspozycji. Dlatego soczewki są używane do zbierania promieni światła w jednym miejscu. Problem polega na tym, że taki obiektyw powoduje zniekształcenia. zniekształcenia. 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. metod matematycznych. Proces kalibracji umożliwia stworzenie modelu geometrii kamery i modelu zniekształceń obiektywu. zniekształcenia obiektywu. Modele te tworzą parametry wewnętrzne kamery.

3.3.1. Ogniskowa obiektywu

Względny rozmiar obrazu rzutowanego na powierzchnię w kamerze zależy od ogniskowej. ogniskowa. 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

Ponieważ soczewka nie jest idealnie wyśrodkowana, wprowadzono dwa parametry, Cx i Cy, oznaczające odpowiednio poziome i pionowe przemieszczenie soczewki. odpowiednio



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

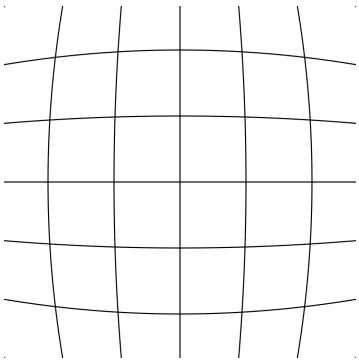
poziome i pionowe przemieszczenie soczewki. Ogniskowa na osiach X i Y są również różne, ponieważ obszar obrazu jest prostokątny. Daje to następujący wzór Daje to następujący wzór na położenie obiektu na powierzchni.

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

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

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

3.3.2. Zniekształcenie obiektywu



Źródło: Wikipedia

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

W centrum optycznym nie ma zniekształceń promieniowych, a przy zbliżaniu się do krawędzi stają się one coraz większe. większe, gdy zbliżamy się do krawędzi. W praktyce zniekształcenie to pozostaje niewielkie. W praktyce zniekształcenie to pozostaje niewielkie, wystarczy wykonać rozwinięcie Taylora do trzeciego członu. członu. Daje to następujący wzór.

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

x i y to współrzędne oryginalnego punktu na obszarze obrazu, które są używane do obliczenia pozycji skorygowanego punktu. skorygowanego punktu. Występuje również zniekształcenie styczne, ponieważ obiektyw nie jest idealnie idealnie równoległa do powierzchni obrazu. Aby to skorygować wprowadzane są dwa dodatkowe parametry, p1 i p2.

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

3.3.3. 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 rozpoznanym obrazem dla każdej kamery. z rozpoznanym 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.findChesssboardCorners

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):

3.4. Obrazowanie stereoskopowe

3.4.1. Wyjaśnienie

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

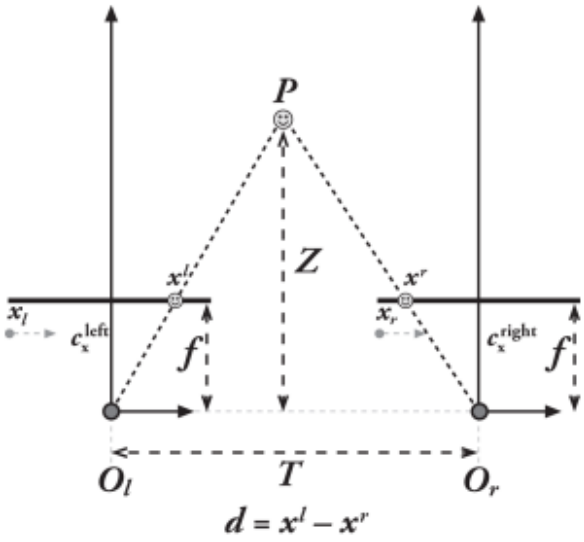
2. rektyfikacja kąta i odległości obrazów. Na tym etapie oba obrazy są obrazy współpłaszczyznowe na osi Y, co ułatwia wyszukiwanie korespondencji. łatwiejsze i wystarczy szukać tylko na jednej osi (osi X). (a mianowicie na osi X).

3. znajdź tę samą cechę na prawym i lewym obrazie. Daje to mapę mapę dysproporcji pokazującą różnice między obrazami na osi x.

4. Ostatnim krokiem jest triangulacja. Mapa rozbieżności jest przekształcana w odległości za pomocą triangulacji.

3.4.2. Triangulacja

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. są współpłaszczyznowe i poziomy rząd pikseli lewego obrazu jest wyrównany z odpowiednim poziomym rzędem pikseli lewego obrazu. poziomy rząd pikseli lewego obrazu jest wyrównany. Poniższy obraz można teraz skonstruować przy użyciu poprzednich hipotez.



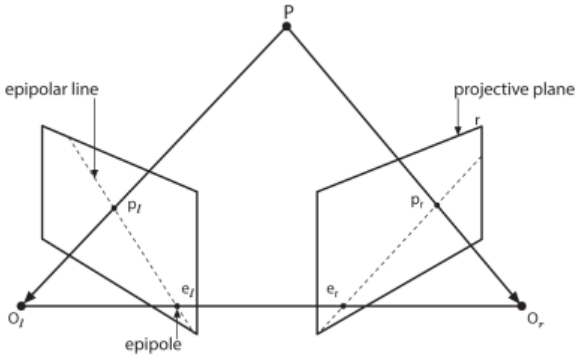
Źródło: Learning OpenCV 3, O'Reilly, Str. 705

Punkt P leży w środowisku i jest pokazany na lewym i prawym obrazie na pl i pr, z odpowiadającymi im współrzędnymi odpowiadającymi współrzędnymi xl i xr. To pozwala nam wprowadzić nową wielkość d = xl - xr. Można zauważyć, że im dalej punkt punkt P, tym mniejsza staje się wielkość d. Dysproporcja jest zatem odwrotnie proporcjonalna do odległości. odległości.
Do obliczenia odległości można użyć następującego wzoru można obliczyć:

$$Z = f * T/(xl - xr)$$

Można zauważyć, że istnieje nieliniowa zależność między rozbieżnością a odległością. istnieje. Jeśli rozbieżność jest bliska 0, małe różnice w rozbieżności prowadzą do dużych różnic w odległości. różnice w odległości. Zjawisko to ulega odwróceniu, gdy rozbieżność jest duża. Małe różnice 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. które znajdują 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 równolegle. Oczywiście kamery muszą być fizycznie ustawione równolegle. fizycznie ustawione równolegle. Zanim wyjaśnimy metodę matematycznego wyrównywania obrazów, musimy najpierw zrozumieć geometrię epipolarną. najpierw zrozumieć geometrię epipolarną.



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

3.4.3. Geometria epipolarna

Powyższy obrazek przedstawia model niedoskonałej kamery stereo składającej się z dwóch modeli kamer otworkowych. dwóch modeli kamer otworkowych. Przecięcie linii środków projekcji (Ol, Or) z płaszczyznami projekcji tworzone są punkty epipolarne el i er. Linie (pl, el) i (pr, er) 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 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). 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. drugiej płaszczyzny.

3.4.4. 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. są fizycznie rozmieszczone. Opisuje ona lokalizację drugiej kamery względem pierwszej za pomocą parametrów translacji i rotacji. względem pierwszej kamery za pomocą parametrów translacji i rotacji. Parametry te nie są bezpośrednio odczytywane w macierzy Parametrów tych nie można odczytać bezpośrednio w macierzy, ponieważ jest ona używana do planowania projektu. W sekcji Kalibracja stereo wyjaśnimy, jak obliczyć R i T (macierz rotacji i wektor translacji). Macierz F zawiera informacje z podstawowej macierzy E, fizyczny układ kamer i informacje o kamerach. kamer i informacje o wewnętrznych parametrach kamer. Relacja między rzutowanym punktem na lewym obrazie pl i rzutowanym punktem na prawym obrazie pr jest zdefiniowana następująco obraz pr jest zdefiniowany następująco:

prTEpl=0

Można by pomyśleć, że ta formuła w pełni opisuje związek między lewym i prawym punktem. prawym punktem. Należy jednak zauważyć, że macierz 3x3 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.

Pamiętamy, 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

3.4.5. Macierz obrotu i wektor przesunięcia

Teraz, gdy wyjaśniliśmy już macierz podstawową E i macierz podstawową F, musimy musimy zobaczyć, jak obliczyć macierz obrotu i wektor translacji. Zdefiniujemy następujące oznaczenia:

- Pl i Pr definiują pozycje punktu w układzie współrzędnych odpowiednio lewej i prawej kamery. prawa kamera
- Rl i Tl (lub Rr i Tr) 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 Pl=RlP+Tl i Pr=RrP+Tr
Mamy również: Pl=RT(Pr-T)
Z tych trzech równań ostateczny wynik to
R=RrRlT T=Tr-RTl

3.4.6. 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. kamery:

- wektor zniekształceń
- macierz rotacji Rrect, która musi zostać zastosowana do obrazu
- wyprostowana macierz kamery Mrect
- nierektyfikowana macierz kamery M

OpenCV pozwala nam obliczyć te warunki za pomocą dwóch algorytmów: algorytmu Hartleya i algorytmu Bougueta.

3.4.6.1. 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.

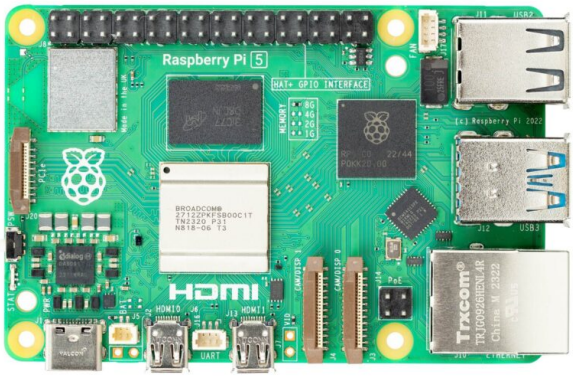
3.4.6.2. Algorytm Bougueta

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.

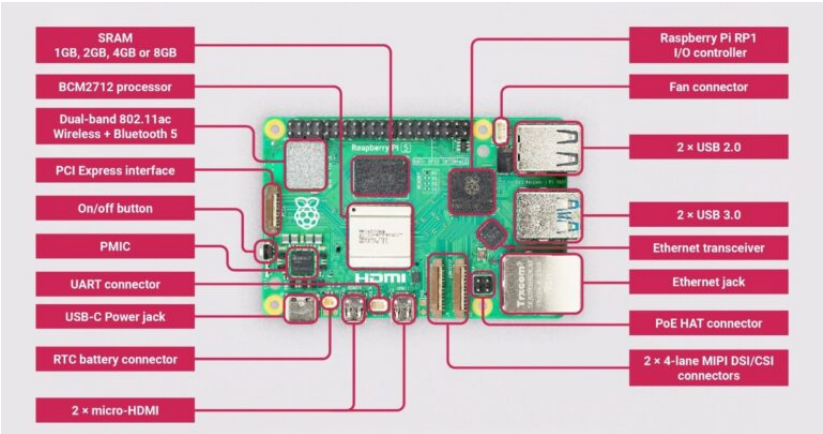
Rozdział 4

Rozdział badawczy

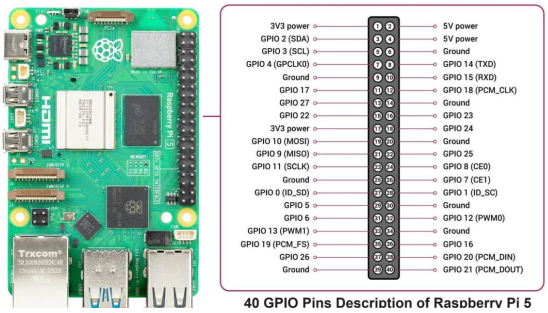
4.1. Raspberry Pi



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

4.2. Camera

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

4.3. 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.

4.3.1. 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

4.3.2. 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().

4.3.3. 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.

4.3.3.1. 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().

4.3.3.2. Pozyskiwanie obrazów do kalibracji

4.3.4. Funkcjonalność programu „Main-Stereo-Vision-Prog.py”

4.3.4.1. Kalibracja zniekształceń

4.3.4.2. Kalibracja kamery stereo

4.3.4.3. Obliczanie mapy rozbieżności

4.3.4.4. Zastosowanie filtra WLS (ważonych najmniejszych kwadratów)

4.3.4.5. Pomiar odległości

4.3.4.6. Możliwe ulepszenia

Rozdział 5

Zakończenie

Bibliografia

[1] OpenCV, *https://opencv.org*.

[2] Aleksiej Kostrykin, *Wstęp do algebry. Podstawy algebry*, Warszawa, Wydawnictwo Naukowe PWN, 2022.

[3] John Lambert, *https://johnwlambert.github.io/stereo/*, April 4, 2025.

[4] Rajesh Rao, *Lecture 16: Stereo and 3D Vision,https://courses.cs.washington.edu/courses/cse455/09wi/L* University of Washington.