

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Instytut Mikroelektroniki i Optoelektroniki

Praca dyplomowa inżynierska

na kierunku Elektronika i Telekomunikacja
w specjalności Inżynieria Komputerowa

Realizacja metody stereoskopowego widzenia maszynowego na
przykładzie prostego robota

Szymon Kieloch
Numer albumu 255779

promotor
prof. dr hab. inż. Artur Przelaskowski

Warszawa, 2019 r.

Streszczenie

Celem niniejszej pracy inżynierskiej było zbadanie realizacji stereoskopowego widzenia maszynowego w kontekście prostego robota. Został przedstawiony kompletny proces widzenia maszynowego od kalibracji kamer, aż po generowanie obrazu głębi, a także użycie uzyskanego obrazu głębi do procesu unikania kolizji w prototypie pojazdu autonomicznego. W celu zbadania procesów widzenia maszynowego i unikania kolizji zostały zbudowane odpowiednie narzędzia weryfikujące. Praca została zwieńczona wdrożeniem modułu stereowizji na prototyp pojazdu autonomicznego.

Słowa kluczowe: widzenie maszynowe, stereowizja, unikanie kolizji, samochody autonomiczne

Abstract

Title: Stereoscopic machine vision method implementation on the example of a simple robot

The goal of the thesis was to examine the implementation of the stereoscopic machine vision in the context of a simple robot. The complete machine vision process was presented starting from the cameras' calibration, up to generating the depth image, as well as using the generated depth image for collision avoidance in an autonomous vehicle prototype. Appropriate tools were developed to examine the machine vision and collision avoidance processes. The thesis was finalized by implementing the stereo vision and collision avoidance module on a vehicle prototype.

Keywords: machine vision, stereovision, collision avoidance, autonomous vehicles

Oświadczenie o autorstwie pracy z USOS - do podmiany

Spis treści

Wstęp	1
Cele pracy i problem badawczy	2
Zarys metodologii	2
Przybliżenie dalszych rozdziałów	2
1. Teoretyczny rozważania	4
1. Pojazdy autonomiczne	4
2. Sensory otoczenia	5
3. Stereowizja	8
3.1 Proces stereowizji	8
3.2 Model kamery otworkowej	9
3.3 Parametry wewnętrzne kamery	10
3.4 Zniekształcenia kamery	11
3.5 System kamer stereo	12
3.6 Geometria epipolarna	14
3.7 Kalibracja systemu stereo	15
3.8 Rektyfikacja	15
3.9 Korespondencja stereo	17
3.10 Triangulacja	18
4. Unikanie kolizji	18
2. Metody	21
1. Metoda kalibracji kamery	21
2. Metoda kalibracji układu stereo	25
3. Metoda obliczenia macierzy rektyfikujących	26
4. Metoda znajdowania korespondencji i generowania macierzy rozbieżności między obrazami	27
5. Metoda generowania obrazu głębi	29
6. Metoda generowania decyzji o kierunku pojazdu	29
3. Ekspertyzy	34
1. Eksperyty kalibracji kamery	36
2. Eksperyty kalibracji układu stereo	39
3. Eksperyty metody obliczającej macierze rektyfikujące	41
4. Eksperyty generowania macierzy rozbieżności oraz obrazu głębi	43
5. Eksperyty generowania decyzji o dalszym kierunku pojazdu	48
6. Eksperyty praktyczne prototypu pojazdu	52
4. Zakończenie	58

Bibliografia	61
Spis rysunków	65
Spis tabel	66
Spis załączników	68

Wstęp

Podjęty temat stereowizji cyfrowej na przykładzie prostego robota jest wynikiem zainteresowania autora tematem postrzegania i interpretacji otoczenia przez maszyny. Interpretacja otoczenia jest niezwykle istotnym aspektem maszyn, które docelowo powinny dobrze współpracować z ludźmi, aby wspomagać ich w wykonywanych pracach. Samochody autonomiczne, wielozadaniowe roboty, mobilne roboty na halach produkcyjnych to zaledwie mały wycinek zastosowań maszyn do ułatwienia życia.

Widzenie jest sposobem na pozyskiwanie informacji o otoczeniu, które z pewnością brało udział we wspinaczce człowieka na wyżyny rozwoju i przejmowaniu kontroli nad innymi gatunkami. W ciągu dnia, podczas gdy jest dostępna duża ilość światła, wzrok jest bardzo istotnym elementem pozwalającym na poruszanie się w zawiłym środowisku. Człowiek posiada dwa rejestratory obrazu, które odbierają fale elektromagnetyczne odbite od różnych przedmiotów. Zarejestrowane fale, przetworzone na impulsy nerwowe, przekazane są do mózgu. Człowiek w rezultacie ma wrażenie widzenia oraz głębi otoczenia. Jest to proces obliczeniowo kosztowny, ale jednocześnie pozwala określić relację przeszkód stojących na drodze do miejsca docelowego.

Maszynowe odzwierciedlenie procesu zachodzącego w ludzkim wzroku można uzyskać na kilka sposobów. Jednym z nich, który jest głównym motywem niniejszej pracy, jest akwizycja obrazów za pomocą dwóch równolegle umieszczonej kamer. Kamery, rejestrujące obraz niemalże w tym samym momencie, pozwalają na uzyskanie par zdjęć otoczenia. Dzięki uchwyceniu otoczenia na zdjęciach z pewnym przesunięciem w poziomie oraz w wyniku dalszej pracy na parze zdjęć powstaje trójwymiarowa przestrzeń punktów otoczenia, zwana dalej obrazem głębi.

W niniejszej pracy autor zaproponuje realizację procesu przemieszczania się prototypu pojazdu autonomicznego. Realizacja będzie składała się z procesów stereowizji oraz unikania kolizji. Autor przedstawi teorię dotyczącą obu procesów, a następnie wykona implementację oraz weryfikację na małym prototypie pojazdu autonomicznego. Obszar zainteresowania autora to podstawowa percepja otoczenia przez maszynę: wykrywanie i unikanie kolizji z przeszkodami podczas eksploracji wnętrza budynku przy użyciu tylko i wyłącznie lokalnych informacji, w tym przypadku, obrazu głębi. Praca ma na celu przyjrzeniu się stereowizji jako istotnemu receptorowi, który dzięki dostarczonym informacjom o otoczeniu umożliwia wykrywanie przeszkód i wobec tego, podjęcie decyzji o zmianie kierunku przemieszczania.

W obrębie pracy będzie zbudowany prototyp pojazdu autonomicznego, który zademonstruje omijanie fizycznych przeszkód na podstawie analizy obrazów pochodzących z pary kamer. W wyniku praktycznych eksperymentów zostanie zbadane unikanie oraz nawigacja między przeszkodami na drodze prototypu. Przyjęte będzie założenie o ograniczeniu przestrzeni sprawdzającej pojazd do wnętrza budynku.

Cele pracy i problem badawczy

Główny zamysł pracy skupia się na łączeniu procesu generowania obrazu głębi z metodą decydującą o następnym kierunku przemieszczenia się prototypu pojazdu na podstawie obrazu głębi, wraz z realizacją obu metod w prototypie pojazdu autonomicznego.

Niniejsza praca ma charakter projektowy. Praktycznymi aspektami pracy są:

- implementacja procesu generowania obrazu głębi,
- implementacja metody unikania kolizji z przeszkodami,
- zbudowanie prototypu pojazdu autonomicznego.

Pytania postawione przez autora pracy:

- Jak dokładnie zostanie odzwierciedlone otoczenie w procesie generującym obraz głębi?
- Czy jest możliwe generowanie obrazu głębi na taniem komputerze pokładowym?
- Czy jest możliwe zrealizowanie procesu wykrywania przeszkód w czasie rzeczywistym na taniem komputerem pokładowym?
- Czy jest możliwa analiza otoczenia w czasie rzeczywistym na taniem komputerze pokładowym?

Zarys metodologii

Przeprowadzone badania zostaną podzielony na trzy etapy: generowanie obrazu głębi, unikanie kolizji oraz realizacja obu w prototypie pojazdu autonomicznego.

1. Pierwszy etap będzie weryfikował proces generowania obrazu głębi. Zostanie użyty program do wyświetlania obrazu głębi w przestrzeni dwuwymiarowej i na tej podstawie zostanie dokonana weryfikacja.
2. Drugi etap to ocena metody służącej do generowania decyzji w celu uniknięcia kolizji. Metoda zostanie zbadana przez sprawdzenie jaka decyzja została wygenerowana do zadanej sytuacji w obrazie głębi.
3. Trzeci etap to praktyczne zbadanie zachowania się prototypu pojazdu autonomicznego w przygotowanym środowisku testowym zawierającym fizyczne przeszkody.

Przybliżenie dalszych rozdziałów

Rozdział I. Teoretyczne rozważania

Rozdział ten zapozna czytelnika z literaturą i teorią w obrębie pojazdów autonomicznych, stereowizji oraz unikania kolizji. Rozdział będzie zawierał objaśnienia wstępne potrzebne na rzecz dalszej pracy.

Rozdział II. Metody

Bazując na teoretycznym opisie zagadnienia, autor przedstawi proces generowania obrazu głębi oraz metodę unikania przeszkód na drodze pojazdu autonomicznego.

Rozdział III. Eksperymenty

Autor przedstawi przygotowane środowisko do weryfikacji procesu generowania obrazu głębi oraz unikania kolizji. Następnie korzystając ze wdrożenia na pojeździe autonomicznym, autor przedstawi wyniki zachowania się pojazdu podczas eksploracji otoczenia.

Rozdział IV. Zakończenie

Zostaną przedstawione końcowe myсли oraz wnioski odnosząc się do całej pracy. Rozdział ten zostanie zakończony przedstawieniem propozycji dalszego wykorzystania wyników pracy oraz nakreślenia kierunku rozwoju.

Rozdział 1. Teoretyczny rozważania

W tym rozdziale czytelnik zapozna się z teorią dotyczącą stereowizji w kontekście pojazdów autonomicznych. Dyskusja rozpocznie się od przedstawienia idei pojazdów autonomicznych wraz z ich sposobami na interpretację otoczenia. Zostanie uszczegółowiony sposób percepcji otoczenia do procesu stereowizji, które będzie jedynym źródłem informacji użytym do wykrywania przeszkód na drodze pojazdu. Ostatnim aspektem poddanym dyskusji w tym rozdziale będzie teoria dotycząca metody unikania kolizji, skupiająca się na użyciu lokalnych informacji o otoczeniu z uzyskanej obrazu głębi.

1. Pojazdy autonomiczne

Zanysł pojazdów autonomicznych powstał już w 1925 roku, w którym Francis P. Houdina przedstawił swój model pojazdu sterowanego przez radio. Przez niecałe sto lat, wykonano wiele badań prowadzących do dalszego rozwoju autonomii pojazdów [1].

Pojazdy autonomiczne przemieszczają się, samodzielnie podejmując decyzje o krokach prowadzących do miejsca docelowego. W [2] autorzy zdefiniowali pojęcie samochodu intelligentnego, który rozumie i odnajduje się w otoczeniu, sterując w bezpieczny sposób po wyznaczonej trasie do celu. Zostały tam również wspomniane problemy prawne, w wyniku których aktualny rozwój samochodów autonomicznych podąża w kierunku szerszego wspomagania kierowcy podczas jazdy, aby podwyższyć bezpieczeństwo, zmniejszyć zużycie paliwa i negatywny wpływ na środowisko, zamiast skupiania się na budowaniu całkowicie autonomicznego komercyjnego pojazdu. W [3] autonomia pojazdu została podzielona na pięć poziomów:

- Poziom 0 - brak automatyzacji, każdy aspekt kierowania jest wykonany przez kierowcę.
- Poziom 1 - automatyzacja tylko niektórych elementów kontroli pojazdu wspomagania osoby kierującej samochodem podczas jazdy.
- Poziom 2 - automatyzacja wielu elementów kontroli pojazdu, kierowca jest wymagany do ciągłego monitorowania samochodu.
- Poziom 3 - wysoka autonomia, kierowca nie ma potrzeby ciągłego monitorowania samochodu, kierowca jest jedynie powiadomiony o potrzebie podjęcia działania.
- Poziom 4 - całkowita autonomia, pojazd nie wymaga kontroli ani monitorowania przez człowieka w żadnym momencie jazdy.

Jednakże z inżynierskiego punktu widzenia, można zastosować alternatywny, dwupoziomowy podział [4]:

1. Autonomia skoncentrowana na człowieku (ang. human-centered autonomy) - człowiek bierze pełną odpowiedzialność za samochód oraz otrzymuje wsparcie od maszyny i w razie potrzeby, ma zapewniony odpowiedni czas na komfortowe przejęcie kontroli.
2. Pełna autonomia (ang. full autonomy) - osoba nie jest odpowiedzialna za samochód i nie ma potrzeby kontrolowania samochodu, poza jedynie wyrażeniem chęci skorzystania z samochodu i wyznaczenia celu podróży.

W [4] również przedstawiono problem natury ludzkiej, który w przypadku autonomii skoncentrowanej na człowieku objawia się przyzwyczajeniem i złudnym poleganiem na maszynie, w wyniku czego człowiek zmniejsza czujność wtedy, kiedy jest istotna.

Zadania pojazdu autonomicznego można podzielić na następujące [5]:

- Lokalizacja i mapowanie - lokalne i globalne lokalizowanie pojazdu.
- Percepcja i zrozumienie sceny - lokalizowanie obiektów oraz przeszkód na drodze i wokół pojazdu.
- Przewidywanie zachowania - zrozumienie znaczenia obiektów wokół pojazdu i antycypacja ich ruchu względem pojazdu.
- Planowanie trasy - planowanie ścieżki prowadzącej do zadanego celu.
- Kontrola pojazdu - sterowanie napędem pojazdu do przemieszczenia.
- Odczyt sensorów - interpretacja informacji pochodzących z różnych sensorów umieszczonych na pojeździe.

Percepcja i interpretacja otoczenia jest jednym z najistotniejszych elementów pojazdów autonomicznych, które decydują o obdarzeniu zaufaniem maszyny przez człowieka [4]. Pojazdy autonomiczne potrzebują systemów, które umożliwiają komunikację ze światem zewnętrznym. Komunikacja i postrzeganie otoczenia zostaje uwzględniona już na pierwszym poziomie automatyzacji, przykładowo jako asystent parkowania. Do tego, jak i wielu innych zadań, stosuje się czujniki, które zostaną nakreślone dalej.

2. Sensory otoczenia

Zbieranie informacji potrzebnych do percepcji otoczenia jest wykonane za pomocą wielu urządzeń, które na różne sposoby rejestrują informacje o otoczeniu. W [6] sensory zostały podzielone na trzy grupy opierające się o kamery, radary i lidary. Te trzy grupy rozszerzone są o czujniki ultradźwiękowe. Kamery, radary i czujniki ultradźwiękowe wykorzystuje się komercyjnie w samochodach, natomiast lidary nadal są w fazie badań.

Czujnik ultradźwiękowy

Czujnik ultradźwiękowy [7] emisuje fale ultradźwiękowe, następnie mierzy czas potrzebny na powrót odbitych fal od powierzchni napotkanych przedmiotów. Cechuje się niewielkim kosztem, małymi rozmiarami, dobrą oceną odległości, pracą w niezależną od warunków pogodowych oraz ilości dostępnego światła. Czujniki ultradźwiękowe pracują na krótkich dystansach, do kilku metrów, nie dostarczają informacji o prędkości obserwowanego przedmiotu [8]. Praktyczne wykorzystanie to mierzenie bliskich odległości, przykładowo jako mierzenie odległości od przeszkody podczas parkowania samochodu.

Radar

Radar (od ang. Radio detection and ranging), korzysta z fal radiowych. Radar wysyła fale radiowe, które odbijają się od powierzchni przedmiotów, następnie są rejestrowane podczas powrotu do urządzenia odbiorczego [9]. Aktualnie, radary korzystają z sygnałów świergotowych, które pozwalają na większą precyzję i szybszy czas pomiaru. Za pomocą radaru uzyskuje się informacje o odległości, kącie i prędkości przedmiotu. Radar cechuje się dobrą rozdzielczością i zasięgiem, małym kosztem i rozmiarem, pracuje w różnych warunkach pogodowych. Aktualnie, radary krótkich zasięgów powoli wypierają czujniki ultradźwiękowe.

Lidar

Lidar (od ang. Light detection and ranging) [9] to urządzenie, które mierzy czas przelotu fotónów, które są wysyłane ze źródła, odbite od powierzchni przedmiotu do momentu zarejestrowana przez odbiornik. Lidar cechuje się dobrym zasięgiem i rozdzielczością, pracą niezależną od ilości światła i możliwością określenia prędkości zarejestrowanego przedmiotu. Aktualnie jednak koszt i wielkość urządzenia jest zbyt duża na praktyczne i popularne wykorzystanie w komercyjnych pojazdach.

Kamera

Kamera pasywnie rejestruje odbite fale światła widzialnego od przedmiotów w otoczeniu. Istotne cechy to zasięg do 300 m w dobrych warunkach pogodowych, niewielkiej wielkości sensor, mały koszt, wysoka rozdzielcość danych oraz popularność wykorzystania [5]. Pojedyncza kamera nie dostarcza informacji o odległościach zarejestrowanych przedmiotów. Ograniczeniem kamer jest wpływ zewnętrznych czynników na rejestrowany obraz, takich jak warunki pogodowe, zbyt mała ilość odbitego światła od powierzchni przedmiotów. Innymi ograniczeniami kamery są ograniczenia stabilności widzenia maszynowego podczas interpretacji wraz z szybkością akwizycji obrazu.

Połączenie czujników

Kamery, radary i czujniki ultradźwiękowe używane są komplementarnie, integrując i uzupełniając się wzajemnie danymi z obszarów, w których pracują najlepiej [8]:

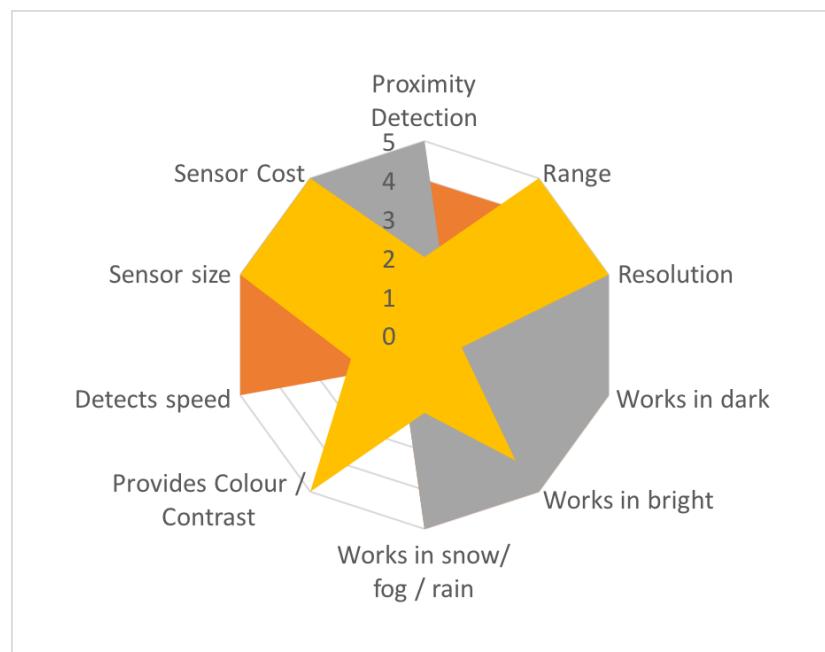
- Czujnik ultradźwiękowy:

- mierzenie dystansu od przedmiotów na bliskich odległościach, do dwóch metrów.
- Radar:
 - mierzenie dystansu od przedmiotów na dalszych odległościach, powyżej dwóch metrów,
 - obserwacja martwych stref wokół pojazdu,
 - mierzenie prędkości otaczających przedmiotów.
- Kamera:
 - identyfikacja i rozpoznawanie przedmiotów.

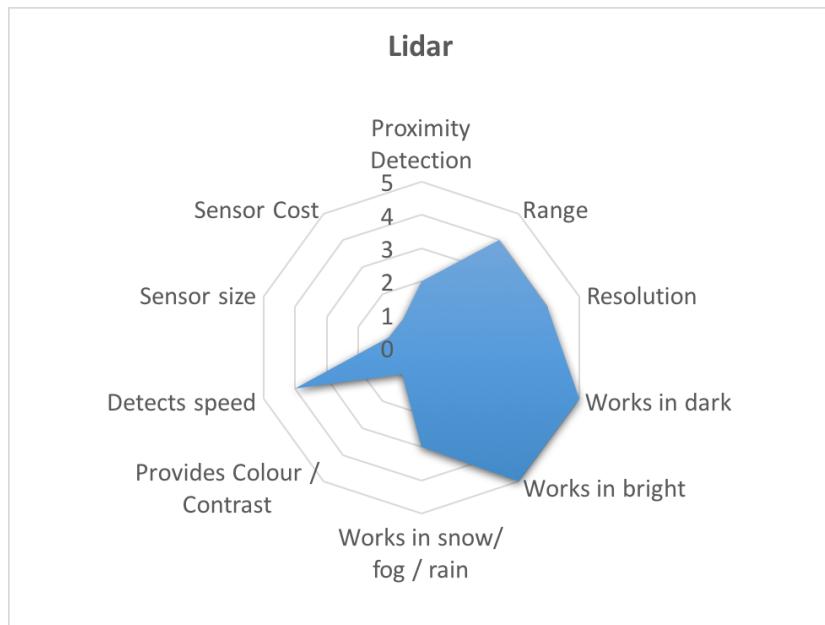
Kamera stereowizyjna

Alternatywnym rozwiązańiem do łączenia trzech sensorów: kamery, radaru i czujnika ultradźwiękowego może być jednoczesne użycie dwóch kamer, tworząc układ kamer stereo. Kamera taka dostarcza informacje o barwie i kontraste, a dodatkowo korzystając z stereowizji [10] jest to uzupełnione informacjami o głębi zarejestrowanego obrazu. Jednakże sama kamera stereowizyjna dziedziczy ograniczenia wynikające z pojedynczych kamer, takich jak negatywny wpływ warunków atmosferycznych, przykładowo: deszcz, śnieg, mgła. W [11] został przedstawiony problem fałszywych granic (z ang. false boundary problem) w stereowizji, który można uniknąć przez zastosowanie dodatkowego światła strukturalnego.

Porównanie sensorów



Rysunek 1.1: Cechy kamery (żółty), radaru (pomarańczowy) i czujnika ultradźwiękowego (szary) ocenione w skali 0 - 5 [8].



Rysunek 1.2: Cechy lidaru ocenione w skali 0 - 5 [8].

3. Stereowizja

W [12] stereowizja cyfrowa została zdefiniowana jako proces uzyskiwania informacji o trójwymiarowej strukturze zarejestrowanej przestrzeni sceny. W procesie biorą udział minimum dwa urządzenia elektroniczne rejestrujące obraz (kamery). Mając zarejestrowane dwa obrazy z różnych punktów sceny, przez wyznaczenie punktów wspólnych między zarejestrowanymi obrazami (punkty korespondujące), wyznaczona zostaje lokalizacja korespondujących punktów w przestrzeni trójwymiarowej (3W). Wyłoniono trzy podstawowe problemy do rozwiązania: kalibracja układu kamer stereo, znajdowanie punktów wspólnych (korespondujących, dopasowanych) oraz rekonstrukcja sceny. Historycznie prace nad stereowizją cyfrową rozpoczęły się już w latach siedemdziesiątych ubiegłego stulecia [12]. W [13] stereowizja została ujęta w grupie pasywnej triangulacji 3W, która pochodzi pod grupę fotogrametrii, techniki geodezyjnej.

3.1 Proces stereowizji

W [14] proces stereowizji został podzielony na następujące kroki:

1. likwidacja zniekształceń wynikających z poszczególnych kamer,
2. rektyfikacja obrazów,
3. znalezienie punktów korespondujących między obrazami,
4. uzyskanie obrazu głębi za pomocą triangulacji.

Uwzględniając wszystkie elementy procesu stereowizji, można podzielić na dwa etapy [15]:

1. Off-line: kalibracja poszczególnych kamer oraz kalibracja systemu stereo.
2. On-line: rejestracja obrazów oraz generowanie obrazu głębi.

W pierwszym etapie, kalibracji, zostanie wykonane:

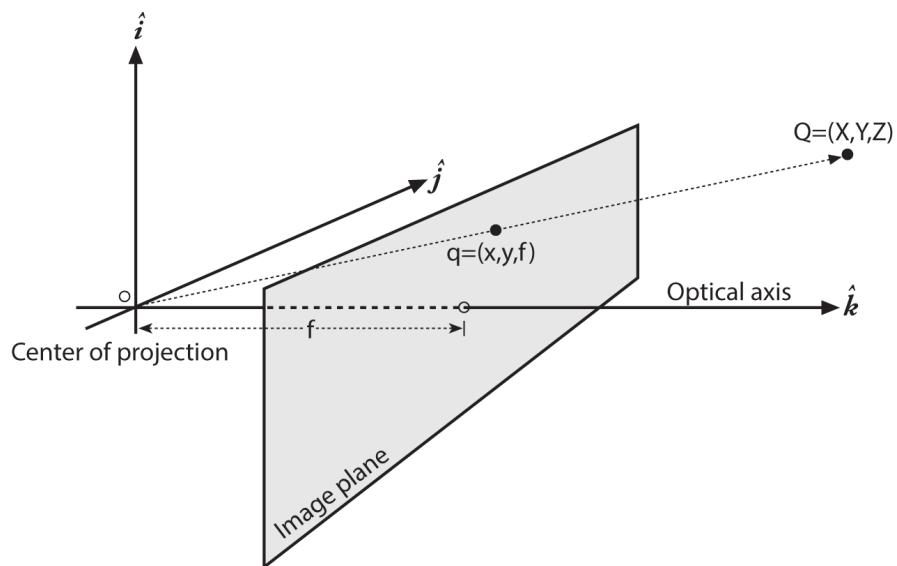
1. obliczenie parametrów wewnętrznych i zniekształceń poszczególnych kamer,
2. obliczenie wzajemnego położenia kamer,
3. obliczenie matrycy rektyfikującej obrazy.

W drugim etapie:

1. jednoczesna rejestracja obrazu w dwóch kamerach,
2. matematyczna likwidacja zniekształceń w obrazach,
3. rektyfikacja obrazów,
4. wyznaczenie punktów korespondujących,
5. generowanie głębi.

3.2 Model kamery otworkowej

Idea kamery otworkowej bazuje na pudełku z małym otworem, przez który przechodzą promienie światła widzialnego, rzutowane na powierzchnię światłoczułą. Istotną zaletą, jednocześnie wadą takiej kamery jest mała ilość światła przechodząca przez otwór. Zaleta wynika z faktu, że zarejestrowany obraz na powierzchni światłoczułej nie posiada zniekształceń, a wadą jest długi czas naświetlania tejże powierzchni [16]. Dzisiejsze kamery korzystają z soczewek, aby zniwelować problem małej ilości światła docierającego do powierzchni światłoczułej kamery. Użycie soczewek wiąże się ze zniekształceniami uchwyconymi na obrazie [14]. Na podstawie kamery otworkowej, został przyjęty następujący model rzutowania punktów z otoczenia kamery na płaszczyznę obrazowania (ang. image plane) w rys. 1.3.



Rysunek 1.3: Przekształcona płaszczyzna modelu kamery otworkowej [14].

W rysunku 1.3 przedstawiony jest rzeczywisty punkt $Q = (X, Y, Z)$, punkt na płaszczyźnie obrazowania $q = (x, y, f)$, odległość płaszczyzny obrazowania od centralnego punktu projekcji f , zwaną dalej odlegością rzutowania. Punkt O jest centralnym punktem projekcji, reprezentuje otwór kamery otworkowej, przez który przechodzą promienie światła widzialnego.

Relacja punktu q z punktem Q wyłania się z podobieństw trójkątów:

$$x = f \frac{X}{Z} \quad (3.1)$$

$$y = f \frac{Y}{Z} \quad (3.2)$$

3.3 Parametry wewnętrzne kamery

Macierz M parametrów wewnętrznych kamery składa się na odległości rzutujące f_x , f_y oraz wektor przeniesienia centralnego środka płaszczyzny obrazowania od głównej osi optycznej $[c_x, c_y]$. Wszystkie składniki wyrażone są w pikselach [14]:

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

W nowszych kamerach, często przyjmuje się założenie, że piksele są kwadratowe, wówczas: $f_x = f_y$, prowadząc do uproszenia obliczeń. W starszych modelach kamer odległość rzutująca osi X różniła się od odległości rzutującej Y , wobec czego piksele obrazu były prostokątne [17].

Oznaczając wektor p , jako reprezentację punktu znajdującego się na płaszczyźnie obrazującej, korzystając z geometrii rzutowej, zostaje ustalona relacja wektora p przedstawiającego punkt q do rzeczywistego punktu Q w postaci iloczynu macierzowego [14]:

$$p = M \cdot Q^T = \begin{bmatrix} f_x X + c_x Z \\ f_y Y + c_y Z \\ Z \end{bmatrix} \quad (3.4)$$

W celu uzyskania współrzędnych pikselowych punktu reprezentowanego za pomocą wektora p , oznaczonych dalej jako (x_{obr}, y_{obr}) , korzysta się ze współrzędnych jednorodnych, dzieląc pierwsze dwa składniki wektora p przez ostatni:

$$x_{obr} = f_x \frac{X}{Z} + c_x \quad (3.5)$$

$$y_{obr} = f_y \frac{Y}{Z} + c_y \quad (3.6)$$

Do obliczenia macierzy parametrów wewnętrznych kamer można zastosować metodę największej wiarygodności przedstawionej w [18], która również pozwala na obliczenie współczynników zniekształceń radialnych omówionych dalej.

3.4 Znieksztalcenia kamery

Zastosowane soczewki zwiększą ilość światła dostającego się na powierzchnię światłoczułą kamery, kosztem znieksztalcen radialnych w obrazach. Proces składania kamery nie jest idealny, umieszczając powierzchnię światłoczułą delikatnie przesuniętą oraz pod delikatnym kątem względem soczewki, w wyniku czego, zarejestrowane obrazy posiadają znieksztalce zwane tangensowymi. Obie grupy znieksztalcen: radialne i tangensowe zostają usunięte po matematycznym przekształceniu obrazów [19].



Rysunek 1.4: Znieksztalcenia obrazu [20].

W [21] został przedstawiony model znieksztalcen obrazu Browna łączący znieksztalce radialne oraz tangensowe.

$$x_u = x_d + (x_d - x_c)(k_1 r^2 + k_2 r^4 + \dots) + \\ (p_1(r^2 + 2(x_d - x_c)^2) + 2p_2(x_d - x_c)(y_d - y_c))(1 + p_3 r^2 + \dots) \quad (3.7)$$

$$y_u = y_d + (y_d - y_c)(k_1 r^2 + k_2 r^4 + \dots) + \\ (2p_1(x_d - x_c)(y_d - y_c) + p_2(r^2 + 2(y_d - y_c)^2))(1 + p_3 r^2 + \dots) \quad (3.8)$$

Gdzie:

x_u, y_u	– współrzędne nieznieksztalconego punktu obrazu
x_d, y_d	– współrzędne znieksztalconego punktu obrazu
x_c, y_c	– współrzędne środek znieksztalce
k_n	– n-ty współczynnik znieksztalce radialnego
p_n	– n-ty współczynnik znieksztalce tangensowego
$r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$	– odległość znieksztalconego punktu od środka znieksztalce
...	– kolejne wyrazy nieskończonego ciągu

Powyższy model został również przedstawiony w [14], gdzie wprowadzono ograniczenia na rozwinięcia serii k_n oraz p_n do dwóch pierwszych wyrazów. Trzeci oraz kolejne wyrazy k_n znajdują zastosowanie w przypadku soczewek ze znieksztalcem typu rybie oko.

Podczas usuwania zniekształceń radialnych i tangensowych obrazu, stosowany jest odwrotny proces znajdowania właściwej wartości danego punktu obrazu. Iterując po pikselach obrazu wynikowego, oblicza się lokalizację punktu w oryginalnym obrazie i na podstawie otoczenia danego punktu, interpolując, uzyskuje się właściwą wartość piksela wyjściowego [21].

3.5 System kamer stereo



Rysunek 1.5: Przykładowy system stereo w postaci kanonicznej [22].

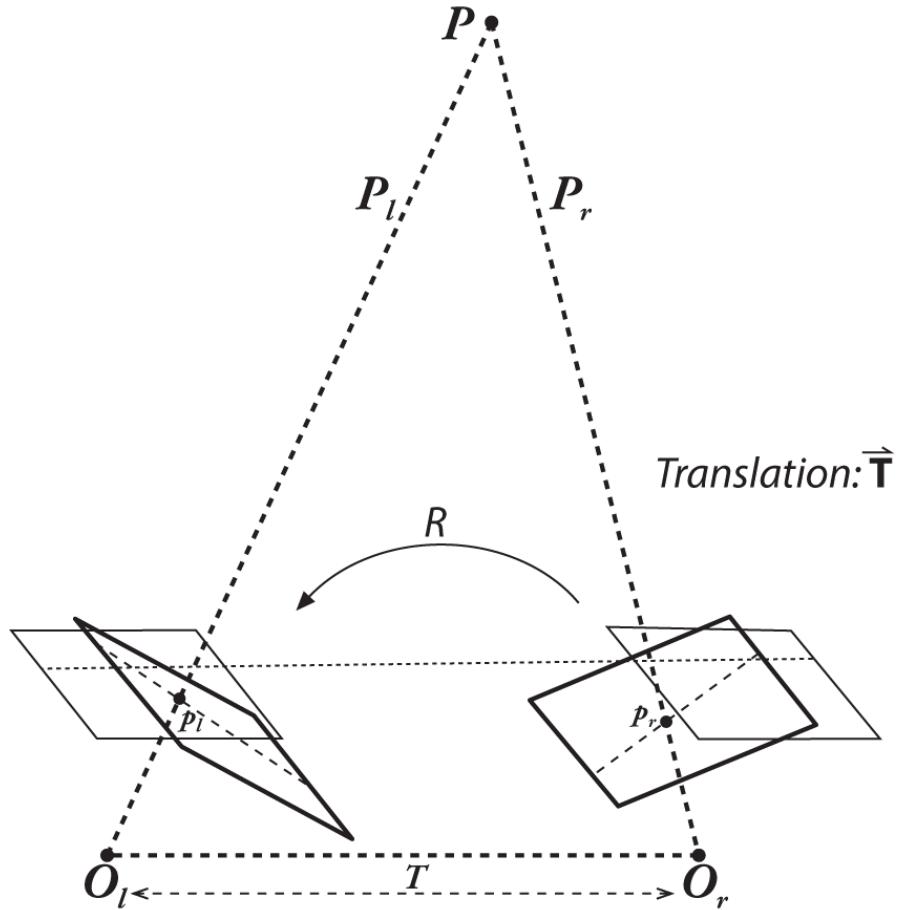
System stereo tworzą dwie kamery ustawione równolegle, oddalone od siebie na pewną odległość. Odległość między kamerami jest dostosowana tak, aby dane zarejestrowane cechy szczegółowe były widoczne na obu obrazach. Punkty obrazów, które odnoszą się do tych samych fizycznych punktów to punkty wspólne lub korespondujące. Widoczność i pokrywanie się punktów wspólnych jest istotne do określenia trójwymiarowej reprezentacji sceny. Relacja między kamerami w globalnej przestrzeni fizycznej jest zawarta w macierzy rotacji R oraz wektorze translacji T , zobrazowane w 1.6.

Macierz rotacji R odpowiada za rotację pomiędzy układami globalnych współrzędnych kamer [22]. Jest to macierz 3×3 , składająca się z trzech wektorów, określające rotację każdej osi współrzędnej X , Y i Z [14].

$$R = [R_X \quad R_Y \quad R_Z] \quad (3.9)$$

Gdzie:

R_X, R_Y, R_Z – trójskładnikowe wektory określające rotację w osi X , Y i Z jako długość danego wektora



Rysunek 1.6: Relacja dwóch płaszczyzn obrazujących za pomocą macierzy rotacji R oraz wektora translacji T [14].

W rys. 1.6 przedstawiono:

- punkt P - zaobserwowany rzeczywisty punkt,
- wektory P_r i P_l - wektory reprezentujące punkt P odpowiednio w prawej i lewej kamerze,
- punkty O_l i O_r - środki projekcji,
- punkty p_l , p_r - powstają w wyniku przecięcia prostych odpowiednio $|O_lP_l|$ i $|O_rP_r|$ przez powierzchnie obrazujące.

Wektor translacji T określa przesunięcie pomiędzy środkami układów współrzędnych obu kamer. Składa się z trzech składników określających przesunięcie w każdej osi współrzędnych.

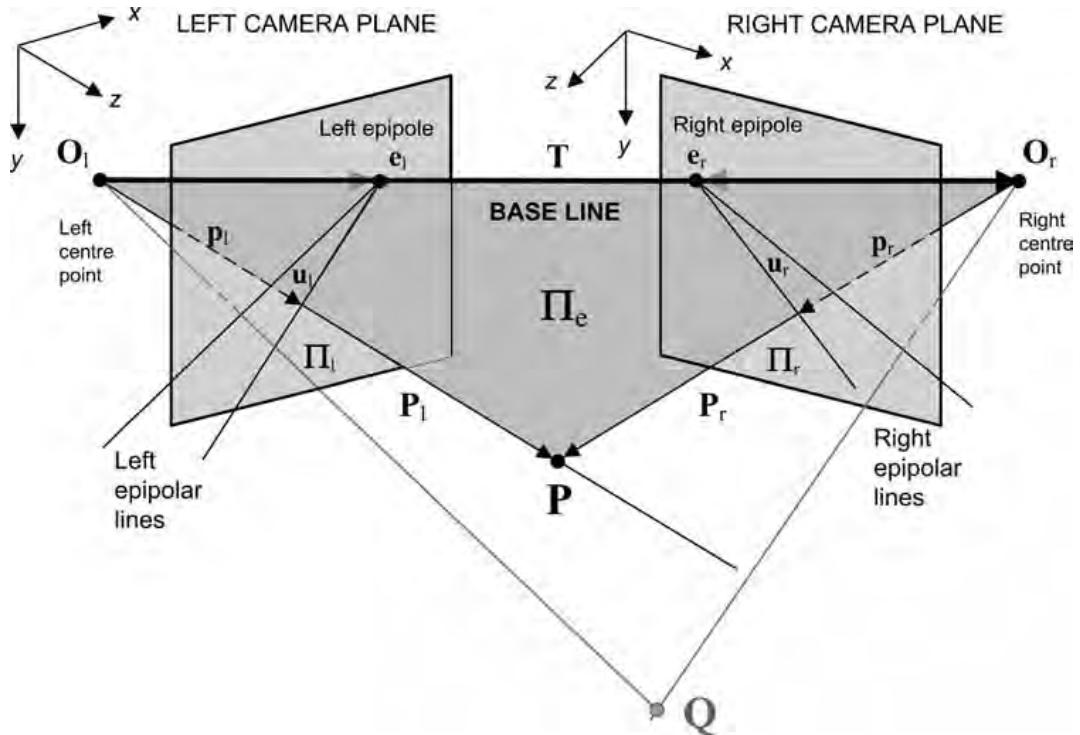
$$T = [t_X, t_Y, t_Z] \quad (3.10)$$

Relacja między wektorami P_r i P_l we współrzędnych globalnych [22]:

$$P_r = R(P_l - T) \quad (3.11)$$

3.6 Geometria epipolarna

Geometria epipolarna [22] łączy dwa systemy kamer otworkowych. Istotnymi elementami w geometrii epipolarnej są punkty epipolarne e_l oraz e_r wyznaczone przez przecięcie płaszczyzn obrazowania odcinkiem $|O_l O_r|$. Przecięcie płaszczyzny Π_e z płaszczyznami obrazowania tworzy linie epipolarne zawierające punkty epipolarne oraz punkty p_l i p_r . Linie epipolarne odgrywają ważną rolę w stereowizji, gdyż ograniczają wyszukiwanie pary punktów korespondujących między obrazami do przeszukania prostej zamiast całej płaszczyzny obrazującej. To ograniczenie nazwano warunkiem epipolarnym [14].



Rysunek 1.7: Układ geometrii epipolarnej [22].

W rys. 1.7 przedstawiono:

- punkt P jest obserwowanym punktem z obu kamer
- punkty O_l i O_r to środki projekcji
- P_r i P_l to wektory reprezentujące punkt P odpowiednio w prawej i lewej kamerze
- Π_l oraz Π_r to płaszczyzny obrazowania
- Π_e to płaszczyzna epipolarna powstała z trzech punktów: P , O_l i O_r
- punkty e_l i e_r są punktami epipolarnymi

W [16] zdefiniowano macierz zasadniczą (ang. essential matrix) E oraz macierz fundamentalną (ang. fundamental matrix) F . Obie macierze połączone są przez macierz rotacji R oraz wektor translacji T . Różnica między tymi dwiema macierzami leży w systemie współrzędnych, do którego dana macierz się odnosi. W macierzy E wzajemne położenie kamer odnosi się do współrzędnych globalnych, a z kolei w macierzy F jest wyrażona we współrzędnych pikselowych.

Zależność macierzy F od macierzy E :

$$F = (M_r^{-1})T \cdot E \cdot M_l^{-1} \quad (3.12)$$

Gdzie:

M_r, M_l – parametry wewnętrzne odpowiednio prawej i lewej kamery

3.7 Kalibracja systemu stereo

W [14] zdefiniowano kalibrację systemu stereo jako ustalenie wzajemnego położenia kamer w przestrzeni przez określenie macierzy rotacji R i wektora translacji T lub określenie macierzy fundamentalnej F , zawierającej obie macierze. Określenie relacji kamer jest istotne do przekształcenia jednej z płaszczyzn obrazujących, tak aby ta płaszczyzna zawierała się w drugiej płaszczyźnie. Po przekształceniu płaszczyzny, długości rzutowania kamer f_x oraz f_y są takie same dla obu kamer.

Do obliczenia macierzy rotacji oraz wektora translacji można użyć algorytmu iteracyjnego Levenberga-Marquardta szukającego lokalnego minimum błędu reprojekcji punktów kalibracyjnych. Natomiast do obliczenia macierzy fundamentalnej można skorzystać z jednego z następujących algorytmów:

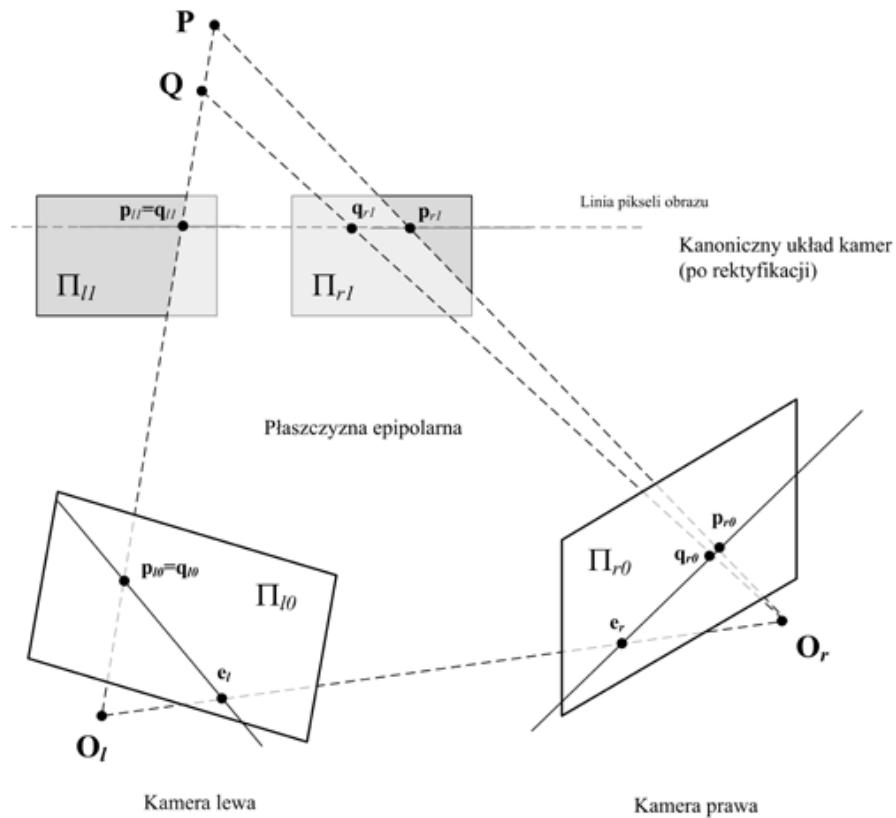
- algorytm siedmiopunktowy,
- algorytm ósmiopunktowy,
- algorytm RANSAC,
- algorytm LMedS.

Najkorzystniejsze z tych algorytmów są RANSAC i LMedS, które eliminują wartości oddalone, negatywnie wpływające na stabilność obliczeń i końcowy rezultat [14].

3.8 Rektyfikacja

Rektyfikacja obrazów jest procesem poprawiającym parę obrazów, w wyniku którego uzyskuje się wyrównanie rzędowe obu obrazów [14]. Wyrównanie rzędowe jest to przekształcenie obrazu, po którym w jednym rzędzie pikseli pary obrazów, będą leżały piksele odpowiadające tym samym rzeczywistym punktom. To oznacza także, że linie epipolarne stają się równoległe do wierszy skanowania obrazu, a punkty epipolarne są w nieskończoności. Taka poprawa obrazu znacznie ułatwia wyszukiwanie punktów korespondujących między parą obrazów, gdyż wyszukiwanie odbywa się w linii poziomej.

Rys. 1.8 obrazuje sens rektyfikacji - wyrównania rzędowego obrazów w celu szybszego znajdowania pasującego punktu. Do przeprowadzenia rektyfikacji oblicza się nowe macierze rotacji R oraz wektor translacji T . Oba składniki mogą zostać wyliczone za pomocą algorytmu Hartleya [24] lub algorytmu Bougueta [14]. Algorytm Hartley'a polega na macierzy fundamentalnej, a Bougueta na macierzach rotacji oraz wektorach translacji obu kamer.

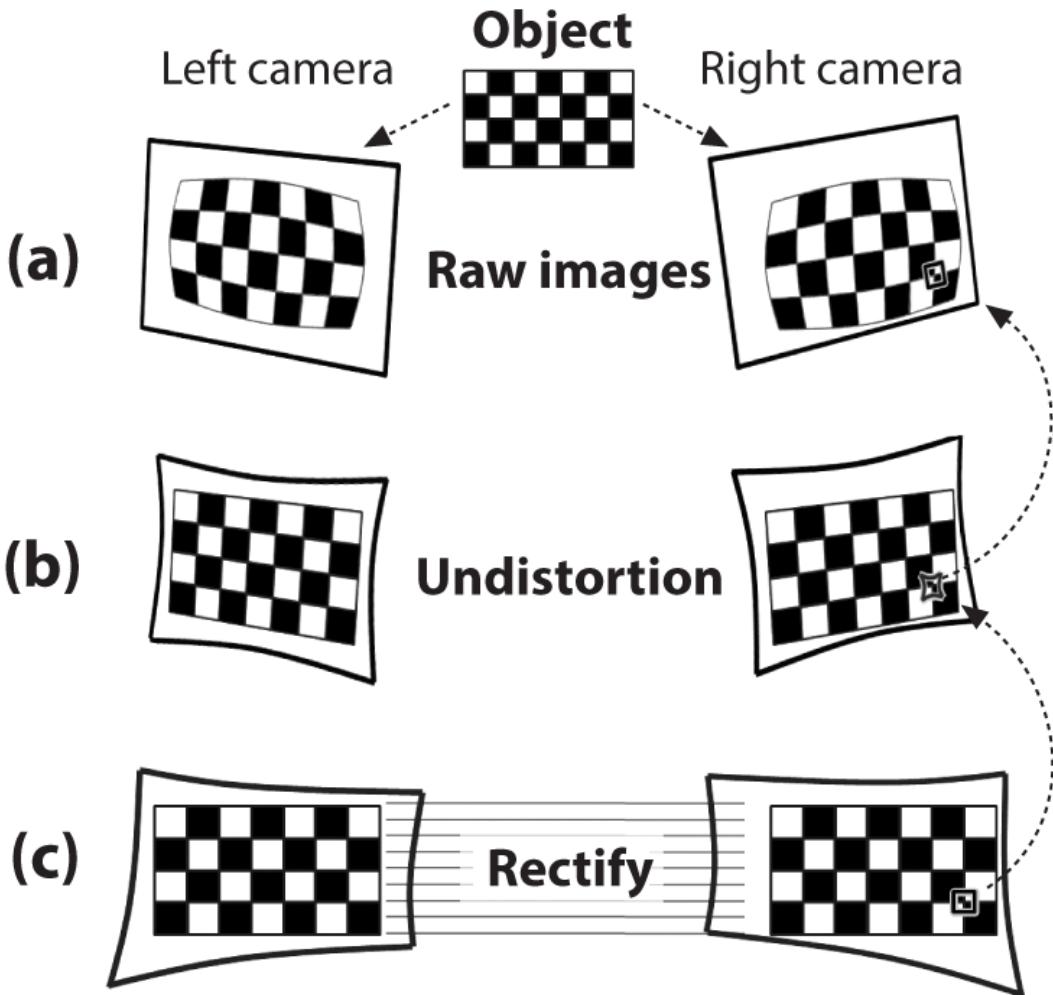


Rysunek 1.8: Płaszczyzny obrazowania przed i po rektyfikacji [23].

W rys. 1.8 przedstawiono:

- punkt P jest rzeczywistym punktem widzianym z obu kamer
- punkt Q jest punktem leżącym na linii wyszukiwania punktu odpowiadającego P
- płaszczyzny Π_{l0} i Π_{r0} to prawa i lewa płaszczyzna obrazowania przed rektyfikacją
- płaszczyzny Π_{l1} i Π_{r1} to prawa i lewa płaszczyzna obrazowania po rektyfikacji

W [14] zdefiniowano macierz rektyfikacji, które określają nowe lokalizacje punktów obrazu, uwzględniając: korekcję zniekształceń kamer, przekształcenie obrazu do tej samej płaszczyzny oraz wyrównanie rzędowe obrazów. Proces poprawy obrazu w ramach rektyfikacji przebiega w następujący sposób: dla każdego punktu obrazu wynikowego wyszukiwany jest punkt, który odpowiada zadanemu punktowi w oryginalnych obrazie, następnie przez odwzorowanie odwrotne i interpolację wartości piksela bazując na otoczeniu punktu w obrazie oryginalnym, jak zobrazowane na 1.9.



Rysunek 1.9: Odwzorowanie odwrotne na przykładzie punktu znajdującego się w prawym zrektyfikowanym obrazie [14].

a) Obrazy oryginalne, b) Obrazy po naprawie zniekształceń, c) Obrazy zrektyfikowane.

W rys. 1.9 na przykładzie prawego obrazu, dla każdego punktu obrazu z poziomu c) jest wyszukiwana lokalizacja danego punktu w obrazie z poziomu b), a z kolei dla każdego punktu z poziomu b) jest wyszukiwana lokalizacja tego punktu w obrazie z poziomu a).

3.9 Korespondencja stereo

Para punktów, w której każdy punkt należy do osobnego obrazu, można nazwać parą korespondującą, gdy reprezentuje ten sam rzeczywisty punkt na zarejestrowanych obrazach. Proces, który wyszukuje takie pary punktów, nazywamy korespondencją stereo [14]. Proces ten odbywa się na obrazach, które należą do tej samej płaszczyzny oraz są zrektyfikowane, aby uskutecznić wyszukane dopasowanie. Identyfikacja punktów korespondencji w [20] została podzielona na gęste oraz rzadkie dopasowanie. W [12] wyszukiwanie dopasowania zostało podzielone na grupy metod:

1. Lokalnych: korespondencja blokowa, korespondencja cech, optymalizacja gradientowa.
2. Globalnych: programowania dynamicznego, krzywe wewnętrzne i cięcia grafów.

Pomiędzy danymi punktami korespondującymi, dla wybranego punktu zostaje przypisana wartość rozbieżności d . Rozbieżność określa różnicę pomiędzy lokalizacją lewego x_l i prawego punktu x_p :

$$d = x_l - x_p \quad (3.13)$$

3.10 Triangulacja

W procesie triangulacji zostaje ustalona fizyczna odległość Z każdego punktów korespondujących wynikających z rozbieżności [14]:

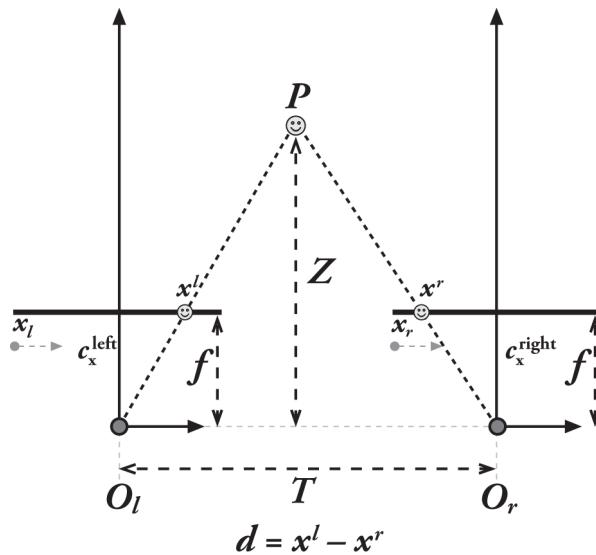
$$Z = \frac{f \cdot T}{d} \quad (3.14)$$

Gdzie:

f – odległość rzutowania

T – odległość między środkami powierzchni obrazującej

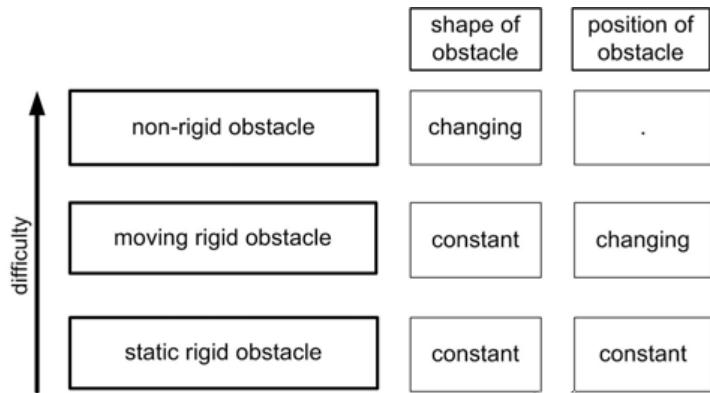
d – rozbieżność punktu obrazu o współrzędnych (x, y)



Rysunek 1.10: Fizyczna odległość Z punktu P [14].

4. Unikanie kolizji

Unikanie kolizji z obiektami napotkanymi na drodze pojazdu autonomicznego jest niezwykle istotnym elementem. W [25] autorzy pracy dokonali podziału przeszkód na sztywne (ang. rigid) oraz niesztywne (ang. non-rigid). Sztywne przeszkody mają znane, niezmieniające się wymiary. Przeszkody niesztywne mają kształt, który ulega zmianie w czasie - mogą się przemieszczać i dynamicznie ograniczać przemieszczanie pojazdu. W [25], przedstawiono przykład śledzenia obiektu zainteresowania w sposób niewykrywalny przez dany obiekt.



Rysunek 1.11: Klasyfikacja trudności przeszkód ze względu na podział sztywne - niesztwyne przeszkody [25].

W [26] przedstawiono zadanie powiązane z unikaniem kolizji w zmiennym środowisku. Zaprezentowano trzy sposoby na unikanie kolizji z obrazu głębi:

1. Metoda oszacowania średniej wartość (ang. the mean estimation method) - bazuje na pionowym podziale obrazu głębi na trzy okna i dla każdego z nich jest obliczona średnia wartość rozbieżności. Wybiera się okno, dla którego jest najmniejsza średnia wartość.
2. Metoda estymacji progu (ang. the threshold estimation method) - polega na pionowym podziale obrazu głębi na trzy okna. Dla środkowego jest sprawdzane, czy jest przekroczena zadana procentowa ilość pikseli, dla których został przekroczyony próg rozbieżności. Jeśli nie przekroczeno, to jest wybrany kierunek na wprost, jeśli tak, to zostaje wybrane okno prawe lub lewe, w zależności, które ma mniejszą wartość średnio obrazu rozbieżności.
3. Metoda wieloprogowa (ang. the multi-thresholds method) - polega na pionowym podziale obrazu rozbieżności na trzy okna. Dla każdego okna jest obliczona ilość pikseli, dla których jest przekroczeno zadany próg. Zostaje wybrane okno, dla którego jest najmniejsza ilość pikseli, które przekroczyły zadany próg. Jeśli procentowa ilość pikseli przekraczających próg w wybranym oknie jest większa niż zadany wskaźnik, oznacza to, że pojazd powinien się wycofać, gdyż znajduje się blisko szerokiej przeszkody.



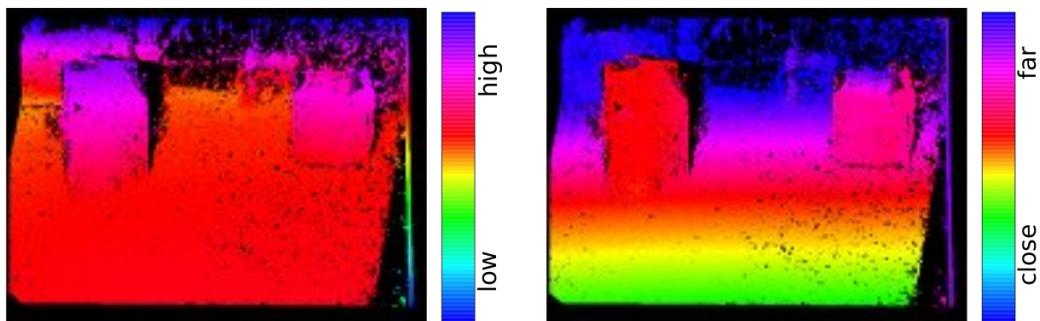
Rysunek 1.12: Podział obrazu głębi na trzy okna w celu oceny możliwego kierunku [26].

W [27] autorzy przedstawili unikanie kolizji przez ograniczenie obszaru zainteresowania w obrazie rozbieżności do wąskiego poziomego paska, na którym zostają przeprowadzone obliczenia do decyzji o zmianie kierunku w celu uniknięcia kolizji. Niesie to pewnego rodzaju ograniczenia, chociażby o przeszkodach, które są poniżej określonego paska, a jednocześnie na tyle wysokie, aby przeszkodzić w czasie przemieszczania. W 1.13 jest przedstawiony obszar decyzyjny ograniczony zielonymi poziomymi liniami.



Rysunek 1.13: Ograniczony obszar zainteresowania wyznaczony przez wnętrze między zielonymi poziomymi liniami [27].

W [28] w odróżnieniu od [27] przedstawiono problem związanego z uwzględnieniem podłogi podczas podejmowania decyzji o kierunku pojazdu. Określono ograniczenie obszaru zainteresowania w obrazie głębi przez określenie progu, który decyduje, czy punkt należy do obszaru zainteresowania, czy też powinien być wykluczony, gdyż należy do podłogi. W 1.14 po lewej jest obraz wzniesienia terenu, natomiast po prawej informacja o wzniesieniu jest naniesiona na obraz głębi w celu wykluczenia podłogi.



Rysunek 1.14: Użycie wzniesienia do identyfikacji punktów podłogi oraz punktów zidentyfikowanych jako punkty przeszkód [28].

Rozdział 2. Metody

Po wstępnie teoretycznym do zagadnienia pojazdów autonomicznych, stereowizji oraz sposobów na unikanie kolizji, jest kolejna przejście do zaprezentowania wybranych metod realizujących elementy procesu stereowizji oraz metody unikania kolizji. W tym rozdziale zostaną przedstawione szczegóły metod i propozycje ich ewaluacji, których eksperymenty zostaną przedstawione w kolejnym rozdziale.

Omówione zostaną następujące metody:

1. Metoda kalibracji kamery
2. Metoda kalibracji układu stereo
3. Metoda obliczenia macierzy rektyfikujących
4. Metoda generowania macierzy rozbieżności
5. Metoda generowania obrazu głębi
6. Metoda unikania kolizji

Powyższe punkty dotyczące wizję maszynową, zostaną omówione i zbadane w kontekście biblioteki OpenCV, która realizuje tę część.

1. Metoda kalibracji kamery

Celem kalibracji kamery jest obliczenie parametrów wewnętrznych oraz współczynników zniesztalceń kamer:

- współczynniki zniesztalceń radialnych: k_1 i k_2 ,
- współczynniki zniesztalceń tangensowych: p_1 i p_2 ,
- parametry wewnętrzne kamery: f_x , f_y , c_x i c_y .

Metoda kalibracji rozpoczyna się od wykonania wielu zdjęć obiektu kalibrującego. Obiekt kalibrujący może być dowolny, jednak ze względu na praktyczne zastosowanie, stosuje się czarno-białą szachownicę o znanych wymiarach. Z dokładnością subpixselową określa się lokalizacje narożników pól wewnętrznych szachownicy i dla tychże punktów ustala się serię równań, które pozwalają na wyliczenie powyższych parametrów.

Biblioteka OpenCV implementuje następujące funkcje i procedury, używane do wyliczenia powyższych parametrów:

- *findChessboardCorners* - znajduje wstępne lokalizacje wewnętrznych narożników szachownicy,
- *cornerSubPix* - określa lokalizacje narożników z dokładnością subpixelsową,
- *findChessboardCornersSB* - za jednym razem znajduje lokalizacje czarnych pól z dokładnością subpixelsową,
- *calibrateCamera* - oblicza parametry wewnętrzne kamer oraz współczynniki zniekształceń.

Określanie lokalizacji punktów przecięcia pól szachownicy

Lokalizowanie przecięć pól szachownicy jest zrealizowane dwoma etapami. Wstępnie określa się przybliżone lokalizacje, następnie na podstawie otrzymanych lokalizacji obliczane są subpixelsowe współrzędne przecięć. Alternatywnie, można zastosować metodę, która w jednym wykonaniu zwraca lokalizacje narożników z subpixelsową dokładnością.

Funkcja *findChessboardCorners*

Funkcja określa wstępne lokalizacje przecięć na podstawie zdjęcia tablicy kalibrującej.

Funkcja przyjmuje 8-bitowy czarno-biały lub kolorowy obraz, wymiar szachownicy oraz następujące flagi decydujące o przebiegu wykonania:

- *CALIB_CB_ADAPTIVE_THRESH* - wykorzystane jest progowanie adaptacyjne do konwersji obrazu na biało-czarny,
- *CALIB_CB_NORMALIZE_IMAGE* - wykonanie korekcji gamma obrazów przed przystąpieniem do progowania stałego lub adaptacyjnego,
- *CALIB_CB_FILTER_QUADS* - użycie dodatkowych warunków do eliminacji nieoprawnych kwadratów, np. kształt, w etapie wyszukiwania konturowego,
- *CALIB_CB_FAST_CHECK* - szybkie sprawdzenie obrazu, czy obraz nie zawiera wzoru tablicy kalibrującej.

Działanie funkcji określone jest przez flagi przekazane do metody. Domyślnie funkcja używa progowania z ustalonym poziomem na podstawie jasności obrazu podczas konwersji na czarno-biały, następnie dokonane jest przeszukanie obrazu w celu znalezienia szachownicy o przekazanych wymiarach.

Funkcja zwraca wartość logiczną, określającą, czy znaleziono wzór szachownicy w danym obrazie. Ponadto funkcja przyjmuje jeden parametr, macierz, której wartości zostaną zmienione na lokalizacje punktów przecięć poszczególnych linii tworzących wzór szachownicy.

Określenie dokładności tej funkcji jest wykonane przez wizualną weryfikację współrzędnych punktów przecięć dla przekazanych obrazów, nanosząc współrzędne przecięć na oryginalne obrazy.

Procedura *cornerSubPix*

Procedura wyszukuje lokalizacje przecięć z dokładnością subpixelsową dla wstępnie określonych punktów otrzymanych z metody *findChessboardCorners*.

Procedura przyjmuje następujące parametry:

- jednokanałowy ośmiobitowy obraz,
- wstępnie określone punkty przecięć,
- wymiary okna, w którego obszarze określana jest subpixelsowa lokalizacja przecięcia dla każdego punktu przekazanego do metody,
- kryteria ograniczające wyszukiwanie dokładnych lokalizacji przecięć składające się na ilość kroków pętli wyszukującej dokładne lokalizacje lub minimalny błąd, który ma być osiągnięty podczas minimalizowania błędu reprojekcji punktów przecięć.

Parametr metody zawierający początkowe lokalizacje punktów przecięć szachownicy zostaje zmieniony podczas działania procedury. Po wykonaniu zostaną tam ustawione wartości punktów przecięć z dokładnością subpixelsową.

Weryfikacja procedury wykonana jest przez wizualne sprawdzenie współrzędnych przecięć oraz przez określenie odległości odpowiednich par punktów przed i po wykonaniu metody. Uzyskane długości wyznaczają, jak duża jest różnica przesunięcia w celu zidentyfikowania punktów, które zostały znacznie przesunięte.

Funkcja *findChessboardCornersSB*

Funkcja znajduje lokalizacje przecięć szachownicy z dokładnością subpixelsową w jednym wykonaniu - w przeciwnieństwie do dwóch wcześniejszych metod.

Funkcja przyjmuje:

- jedno- lub trój-kanałowy ośmiobitowy obraz,
- wymiary szachownicy,
- flagi decydujące o przebiegu procesu ustalania dokładnych współrzędnych przecięć pól szachownicy.

Flagi decydujące o przebiegu wykonania:

- *CALIB_CB_NORMALIZE_IMAGE* - wykonanie korekcji gamma obrazów przed przyjęciem do progowania stałego lub adaptacyjnego,
- *CALIB_CB_EXHAUSTIVE* - zwiększenie ilości prób wyszukiwania dokładnych lokalizacji przecięć,
- *CALIB_CB_ACCURACY* - zwiększenie dokładności subpixelsowej przez nadpróbkowanie obrazu.

Metoda zwraca wartość logiczną, definiującą, czy znaleziono wzór szachownicy w danym obrazie. Co więcej, funkcja przyjmuje jeden parametr, który zostanie zmieniony podczas wykonania: macierz z dokładnymi lokalizacjami punktów przecięć tworzących wzór szachownicy.

Wyniki działania tej funkcji zostaną porównane z wynikami wcześniejszej funkcji i procedury przez określenie odległości odpowiednich par punktów z tej funkcji oraz odpowiednich punktów zwróconych przez wcześniejsze. Ponadto zostanie dokonana wizualna weryfikacja w celu określenia lokalizacji przecięć.

Na rzecz powyższych trzech metod, wykorzystuje się procedurę *drawChessboardCorners*, która rysuje określone punkty na obrazach tablic kalibrujących w celu wizualnej weryfikacji.

Określenie parametrów wewnętrznych i współczynników znieksztalceń kamery

Funkcja *calibrateCameraRO*

Funkcja *calibrateCameraRO* działa w sposób iteracyjny. Określenie parametrów wewnętrznych oraz współczynników znieksztalceń jest realizowane w następujących krokach:

- określenie początkowych parametrów wewnętrznych kamery na podstawie wielu punktów z tablic kalibrujących,
- określenie pozycji kamery we współrzędnych globalnych względem tablic kalibrujących na podstawie wielu punktów z tablic,
- iteracyjne optymalizowanie współczynników znieksztalceń oraz parametrów wewnętrznych przez minimalizację odległości między zaobserwowanymi punktami przecięć a projekcją tych punktów z użyciem algorytmu Levenberg-Marquardta.

Funkcja przyjmuje następujące parametry:

- listę współrzędnych punktów przecięć na tablicy kalibrujących w wybranej jednostce miary,
- listę znalezionych wcześniej współrzędnych punktów przecięć widzianych na obrazach tablic kalibrujących,
- rozmiar obrazu użyty do ustalenia współrzędnych punktu głównego kamery,
- opcjonalny indeks punktu przecięcia, który ma być ustawiony jako stały punkt odniesienia - gdy jest przekazany, funkcja używa algorytmu uwzględniającego niedoskonałości wykonania tablicy kalibrującej,
- flagi określające wewnętrzne działanie metody,
- kryteria zatrzymania pętli optymalizacji bazujące na ilości iteracji optymalizacji lub minimalnego błędu reprojekcji.

Funkcja również przyjmuje serię parametrów, których wartości zostają zmienione w czasie wykonania działania metody:

- macierz z parametrami wewnętrznymi kamery,
- wektor współczynników znieksztalceń kamery,
- błąd reprojekcji dla każdego obrazu tablicy kalibrującej.

Z istotniejszych flag, które sterują metodą, to:

- *CALIB_USE_INTRINSIC_GUESS* - użycie przekazanych parametrów wewnętrznych kamery,
- *CALIB_ZERO_TANGENT_DIST* - ustawienie współczynników znieksztalceń tangentialnych na zero,
- *CALIB_FIX_PRINCIPAL_POINT* - użycie przekazanych współrzędnych punktów głównego, który nie zostanie zmieniony podczas optymalizacji.

Wartością zwróconą przez funkcję jest całkowity błąd reprojekcji punktów tablic kalibrujących.

Ocena funkcji zostanie wykonana poprzez sprawdzenie błędu reprojekcji dla wszystkich obrazów oraz każdego z osobna, a także weryfikację wyliczonych parametrów i współczynników.

2. Metoda kalibracji układu stereo

Funkcja *stereoCalibrate*

Kalibracja układu stereo rozpoczyna się od ustawienia kamer na tej samej linii, aby osie optyczne obu kamer były równoległe względem siebie oraz aby obrazy z obu kamery się pokrywały. Następnie używając obu kamer, jednocześnie fotografuje się szachownice kalibrujące z różnych perspektyw. Z obliczonych wcześniej parametrów wewnętrznych, współczynników znieksztalcień i otrzymanych par zdjęć jest obliczana relacja geometryczna obu kamer: macierz rotacji R oraz wektor translacji T określająca wzajemną relację obu kamer we współrzędnych globalnych.

Funkcja przyjmuje wiele parametrów, z których ważniejsze to:

- listę współrzędnych punktów przecięć na tablicy kalibrujących w wybranej jednostce miary,
- listy znalezionych wcześniej współrzędnych punktów przecięć widzianych na obrazach tablic kalibrujących z prawej i lewej kamery,
- parametry wewnętrzne oraz współczynniki znieksztalczeń obu kamer,
- rozmiar obrazu użyty do ustalenia współrzędnych punktu głównego,
- flagi określające działanie metody,
- kryteria zatrzymania pętli optymalizującej składającej się na ilość iteracji lub minimalny błąd reprojekcji.

Funkcja również przyjmuje serię parametrów, których wartości zostają zmienione w czasie wykonania działania metody:

- macierz rotacji R ,
- wektor translacji T ,
- macierz zasadnicza E oraz fundamentalna F ,
- błąd reprojekcji dla każdego obrazu tablicy kalibrującej.

Z istotniejszych flag, które sterują funkcją, to:

- *CALIB_FIX_INTRINSIC* - użycie dostarczonych parametrów wewnętrznych oraz współczynników znieksztalczeń kamer,
- *CALIB_USE_EXTRINSIC_GUESS* - użycie dostarczonej macierzy rotacji R oraz wektora translacji T , które zostaną dalej optymalizowane.

Wartością zwróconą przez funkcję, podobnie jak we wcześniejszych metodach kalibrujących, jest całkowity błąd reprojekcji punktów tablic kalibrujących.

Ocena funkcji zostanie dokonana przez weryfikację błędu reprojekcji dla wszystkich obrazów oraz weryfikację macierzy rotacji i wektora translacji.

3. Metoda obliczenia macierzy rektyfikujących

Procedura *stereoRectify*

Procedura *stereoRectify* oblicza macierze przekształcające obrazy do tej samej płaszczyzny wraz z wyrównaniem rzutowym, bazując na algorytmie Bougueta.

Procedura przyjmuje parametry, z których istotne to:

- macierze parametrów wewnętrznych oraz współczynników znieksztalceń obu kamer,
- macierz rotacji R oraz wektor translacji T łączące obie kamery we współrzędnych globalnych,
- parametr skalujący, określający zbliżenie na obszary obrazów, które się pokrywają po dokonanym przekształceniu,
- flaga określająca, czy po przekształceniu, współrzędne x oraz y punktów głównych obu kamer, zmieniają się na wartości średnie współrzędnych x oraz y punktów głównych obu kamer.

Parametry metody, które zostaną zmienione w czasie wykonania:

- nowe macierze rotacji dla prawej i lewej kamery,
- nowe macierze projekcji zawierające parametry wewnętrzne oraz wektor translacji dla obu kamer,
- macierz Q określająca relację poszczególnych punktów obrazu (x, y, d) do rzeczywistych punktów 3W (X, Y, Z) .

Ocena procedury zostanie dokonana przez sprawdzenie wartości otrzymanych w macierzach rotacji oraz projekcji.

Procedura *initUndistortRectifyMap*

Drugą wykorzystaną procedurą w procesie rektyfikacji jest *initUndistortRectifyMap*, która jest odpowiedzialna za obliczenie macierzy rektyfikującej, tj. macierzy, w której zostaną określone nowe współrzędne poszczególnych pikseli oryginalnych obrazów. Operację tę wykonuje się osobno dla obrazów prawej i lewej kamery.

Procedura przyjmuje następujące parametry:

- macierz parametrów wewnętrznych kamery,
- wektor współczynników znieksztalceń kamery,
- macierz rotacji kamery otrzymanej z metody *stereoRectify*,
- nowa macierz z parametrami wewnętrznymi kamery otrzymanych z metody *stereoRectify*.

Procedura przyjmuje również dwie macierze, których wartości zostają zmienione w czasie działania. Obie macierze po wykonaniu będą miały lokalizacje poszczególnych punktów obrazów rektyfikowanych w oryginalnych obrazach.

Sprawdzeniem działania tej procedury jest wizualna weryfikacja poprawy obrazów z użyciem metody pomocniczej *remap*, która przekształca obrazy, korzystając z otrzymanych macierzy.

4. Metoda znajdowania korespondencji i generowania macierzy rozbieżności między obrazami

Metoda znajdowania korespondencji między obrazami jest połączona z metodą generowania macierzy rozbieżności. Pierwsza metoda dla każdego punktu jednego obrazu wyszukuje odpowiadający punkt na obrazie drugim, natomiast druga zwróci wartość przesunięcia punktów dla danej pary obrazów.

Do realizacji tego zadania zostały wybrane dwie metody dopasowania:

- metoda lokalnego dopasowania blokowego BM (ang. Block Matching),
- metoda częściowo-globalnego dopasowania blokowego SGBM (ang. Semi-Global Block Matching).

Metoda lokalnego dopasowania blokowego BM

Jest to szybka i efektywna metoda wyszukiwania dopasowania między obrazami. Wykorzystuje okienka sum różnic bezwzględnych SAD (ang. Sum of Absolute Differences) do wyszukiwania pasujących do siebie pikseli obu obrazów w jednym wierszu. Działa bardzo dobrze w przypadku obrazów zawierających wzorzyste otoczenie, przykładowo drzewa lasu. Podczas pracy ze słabo wzorzystymi zdjęciami, może zwracać mniej punktów korespondujących. Metoda jest zrealizowana za pomocą klasy *StereoBM*.

Algorytm dopasowania blokowego, zaimplementowany w OpenCV działa w trzech etapach:

1. Wstępne filtrowanie w celu poprawy obrazu.
2. Wyszukanie korespondencji przy użyciu okna SAD.
3. Końcowe filtrowanie w celu wyeliminowania niepoprawnych dopasowań.

Do wstępnego filtrowania może zostać wybrana normalizacja jasności obrazu oraz poprawienie tekstur lub alternatywnie konwersja obrazów na pierwsze pochodne Sobela względem osi X, co jest pomocne podczas usuwania artefaktów oświetleniowych.

Metoda częściowo-globalnego dopasowania blokowego SGBM

Metoda wykorzystuje informacje o pikselach otaczających dany piksel w celu wygenerowania gładziej macierzy rozbieżności. Główną różnicą między SGBM a algorytmem lokalnym dopasowania blokowego jest śledzenie ciągłości rozbieżności w kilku kierunkach, aniżeli tylko w kierunku poziomym.

W bibliotece OpenCV algorytm został zrealizowany wg następujących kroków:

1. Konwersja obrazu na pierwsze pochodne Sobela.

2. Obliczenie macierzy kosztów dopasowania $C(x, y, d)$ korzystając z miary Birchfielda-Tomasiego.
3. Ustawienie zer jako wartości początkowych w macierzy $S(x, y, d)$.
4. Iteracyjne obliczanie $S_r(x, y, d)$ dla r , który jest kierunkiem przetwarzania.
5. Dodanie $S_r(x, y, d)$ do $S(x, y, d)$.
6. Znalezienie właściwej rozbieżności $d(x, y) = \text{argmin} S(x, y, d)$.
7. Sprawdzenie spójności macierzy rozbieżności przez ponowne wyznaczenie tejże macierzy dla odwrotnej pary obrazów. Piksele, dla których wartość rozbieżności się nie zgadzają, są oznaczone jako nieznane.
8. Końcowe filtrowanie w celu wyeliminowania niepoprawnych dopasowań.

Istotne parametry dwóch powyższych metod

Dla lokalnej oraz częściowo globalnej metody dopasowania blokowego zostały zdefiniowane istotne parametry.

Istotne parametry metody lokalnego dopasowania blokowego

- minimalna rozbieżność - określa minimalną rozbieżność, "0", gdy linie przecinają się w nieskończoności, może być ujemna, gdy kamery są zbieżne,
- liczba rozbieżności - definiuje zakres przeszukiwania korespondencji, rozpoczynając od minimalnej rozbieżności,
- wielkość bloku - rozmiar obszaru, dla którego obliczana jest miara znaku absolutnych różnic dla każdego piksela,
- współczynnik unikalności - współczynnik określający różnicę pomiędzy najlepszym dopasowaniem a drugim najlepszym dopasowaniem.

Istotne parametry metody częściowo-globalnej dopasowania blokowego

- minimalna rozbieżność - określa minimalną rozbieżność, wartość "0" gdy linie przecinają się w nieskończoności, może być ujemna, gdy kamery są zbieżne,
- liczba rozbieżności - definiuje zakres przeszukiwania korespondencji, rozpoczynając od minimalnej rozbieżności,
- wielkość bloku - rozmiar obszaru, dla którego obliczana jest miara znaku absolutnych różnic dla każdego piksela,
- współczynnik unikalności - współczynnik określający różnicę pomiędzy najlepszym dopasowaniem a drugim najlepszym dopasowaniem,
- różnica między rozbieżnościami lewo-prawo i prawo-lewo - określa dozwoloną różnicę między rozbieżnościami powstałymi z dopasowania lewego z prawym i prawego z lewym obrazem jednego piksela, jeśli dla danego piksela różnica jest wyższa od zadanej, dany piksel jest wówczas oznaczony jako niewiadomy,
- tryb - ilość kierunków, dla których będzie badany koszt rozbieżności,

Parametry filtrowania obrazu rozbieżności w celu usunięcia fałszywych dopasowań

- zakres plamek - dopuszczalna różnica między wartością rozbieżności danego piksela a wartością rozbieżności innego piksela w pewnym obszarze podczas tworzenia komponentu wypełnienia (ang. flood fill) wychodząc od danego piksela,
- rozmiar plamki - minimalny obszar dla komponentu wypełnienia, powyżej którego dany piksel nie będzie uznany za plamkę.

Powыższe metody zostały zbadane przez wygenerowanie obrazu macierzy rozbieżności, w celu zbadania jej zmiany podczas zamiany jednego parametru, gdy inne parametry mają ustaloną stałą wartość.

5. Metoda generowania obrazu głębi

Generowanie obrazu głębi jest ustaleniem rzeczywistych współrzędnych każdego punktu obrazu jednej z kamer, dla którego została ustalona wartość rozbieżności. Działanie jest wykonane z użyciem macierzy reprojekcji punktów Q ustalonej przez procedurę *stereoRectify*. Do generowania obrazu głębi wykorzystana jest procedura *reprojectImageTo3D*.

Procedura *reprojectImageTo3D*

Istotne parametry, które przyjmuje ta procedura:

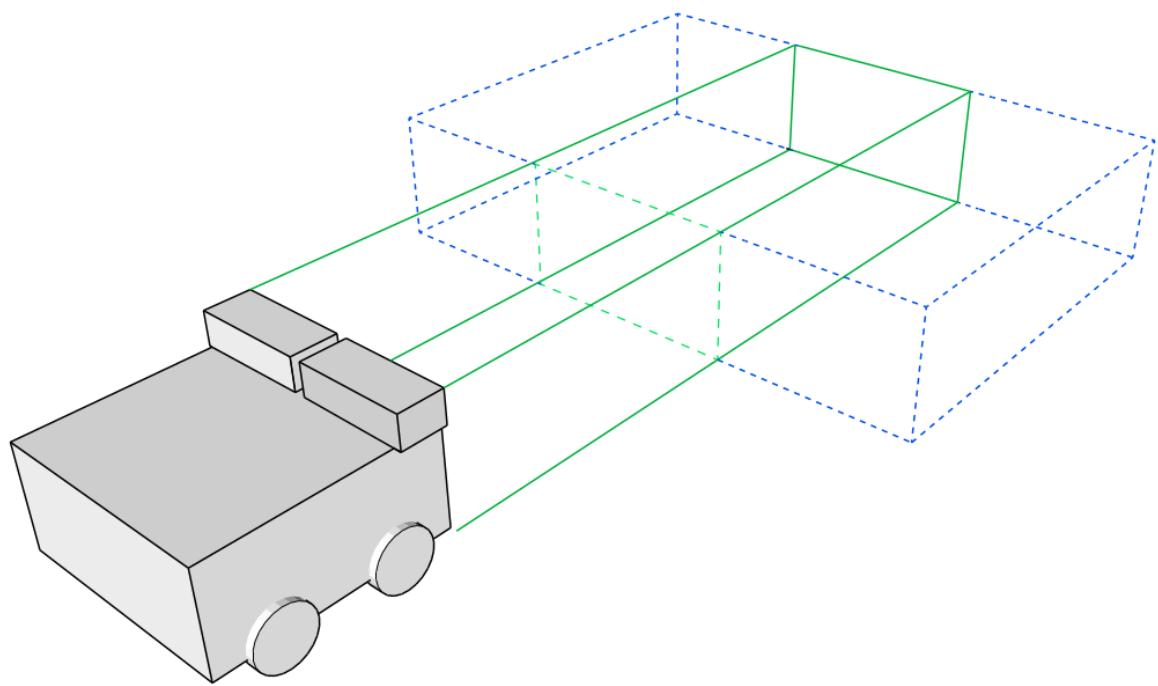
- macierz rozbieżności,
- macierz reprojekcji,
- flagę określającą, czy przypisać dużą wartość liczbową dla punktów, których rozbieżności są nieznane.

Procedura również przyjmuje macierz, której wartości zostaną zmienione na rzeczywiste współrzędne punktów.

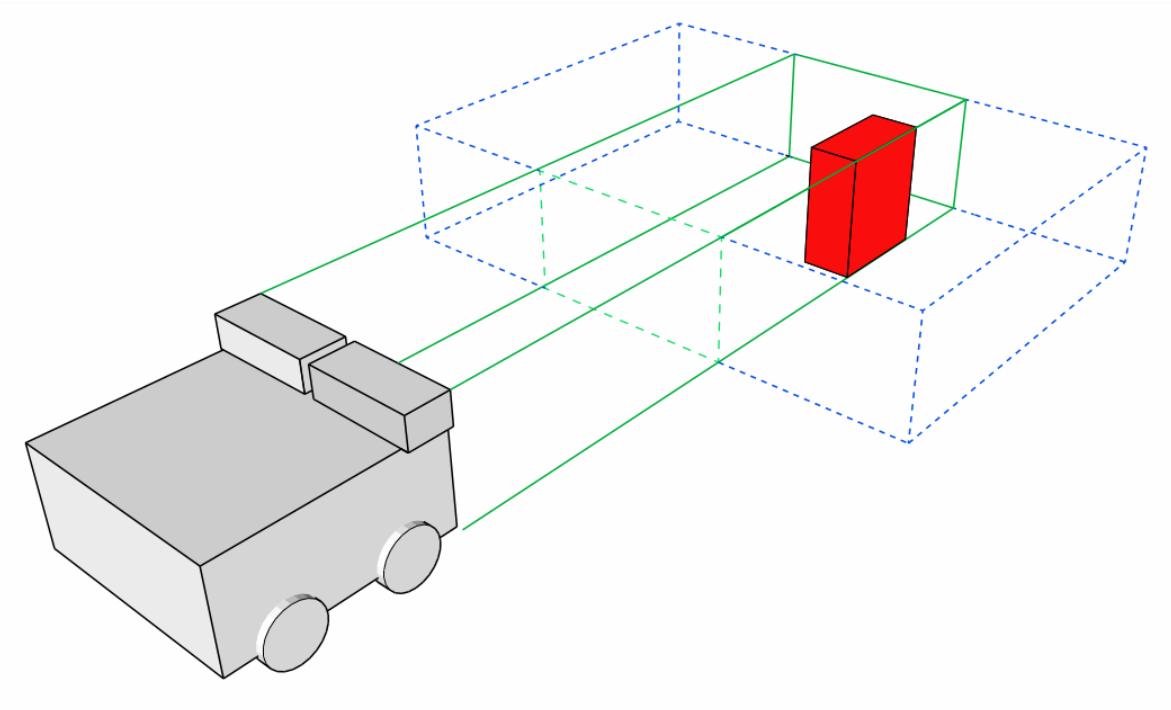
W celu zweryfikowania tej metody zostaną wygenerowane obrazy głębi, a następnie będą sprawdzone współrzędne określonych punktów ze znanimi współrzędnymi.

6. Metoda generowania decyzji o kierunku pojazdu

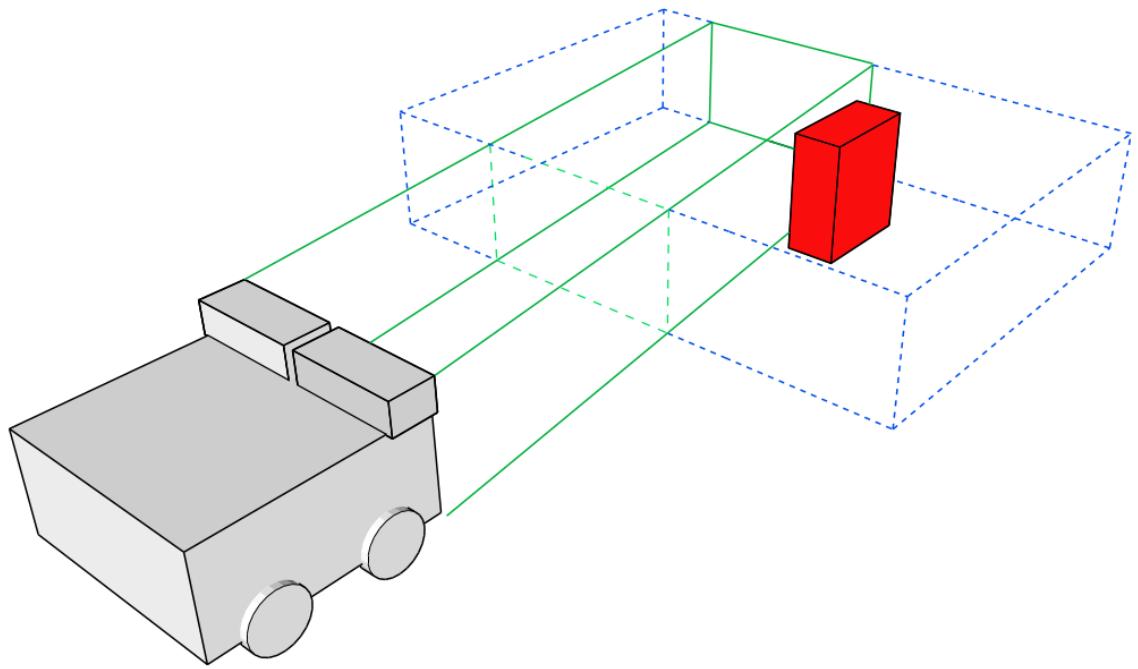
3W podprzestrzeń przedstawiona na zielono w rys. 2.1 ilustruje trójwymiarowy tor pojazdu umieszczony wewnątrz całej zarejestrowanej przestrzeni na niebiesko. Dla kontynuacji przejazdu, tor pojazdu powinien być pusty. Pomiędzy całkowitą zarejestrowaną przestrzenią na niebiesko a pojazdem, znajduje się przestrzeń, która nie jest zarejestrowana przez układ kamer stereowizyjnych. W momencie znalezienia się przeszkody na torze pojazdu zobrazowane na rys. 2.2, jest dokonywana analiza całkowitej przestrzeni 3W, która pozwala na określenie kąta skrętu potrzebnego do uniknięcia kolizji. W takim przypadku bezkolizyjny tor przedstawiony jest na rys. 2.3. W momencie zarejestrowania bardzo szerokich przeszkód w stosunku do wymiarów pojazdu, jak np. ściany, całkowita zarejestrowana przestrzeń jest podzielona na dwie części, które zostają użyte do określenia kierunku skrętu. Podział całej przestrzeni jest przedstawiony na rys. 2.4 w postaci ściany dzielącej przestrzeń na połowę.



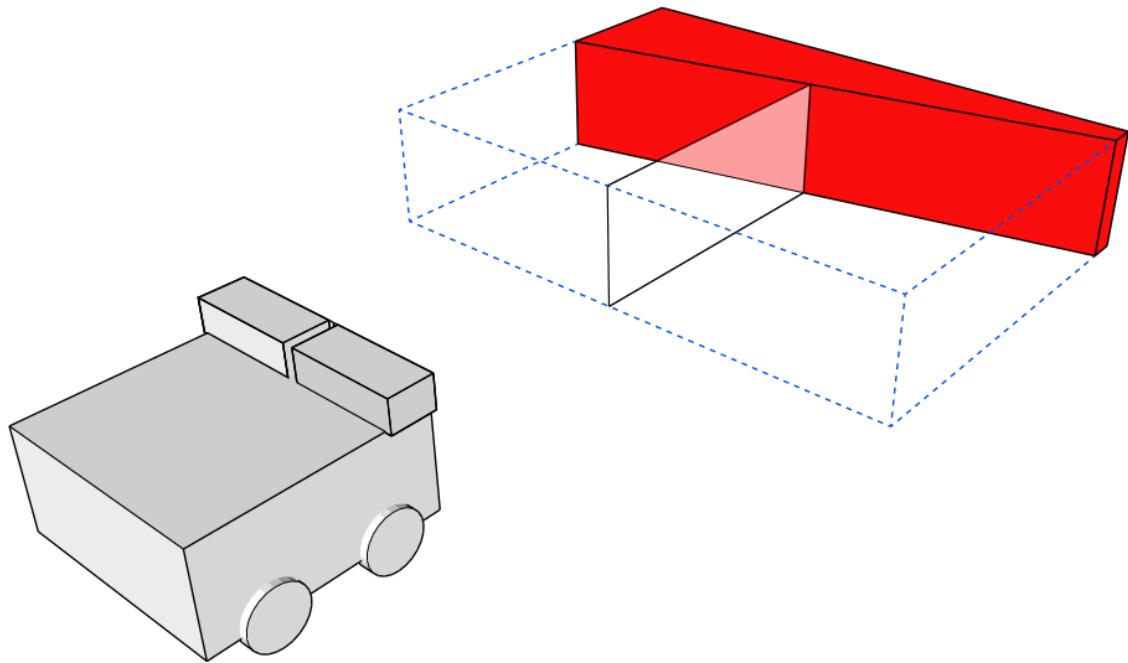
Rysunek 2.1: 3W zarejestrowana przestrzeń przed pojazdem (na niebiesko) wraz z jego torem (na zielono).



Rysunek 2.2: 3W przestrzeń przed pojazdem zawierająca przeszkodę na torze pojazdu (przeszkoda na czerwono).



Rysunek 2.3: Wyznaczony tor alternatywny w celu ominięcia przeszkody.



Rysunek 2.4: 3W przestrzeń zawierająca szeroką przeszkodę typu ściany (przeszkoda na czerwono).

Przyjęta metoda unikania kolizji czerpie inspirację z metody estymacji progu oraz metody uwzględniającej podłogę, dodając nowy element: przeszukiwanie całkowitej przestrzeni 3W dla optymalnego toru pojazdu. Metoda działa w trzech krokach:

- W pierwszym kroku jest dokonana weryfikacja, czy w przestrzeni 3W toru ruchu pojazdu znajdują się przeszkody: jeżeli nie, to wydane jest polecenie o kontynuacji jazdy na wprost, jeżeli tak, to jest przejście do drugiego kroku.
- W drugim kroku, jest weryfikowane, czy w całkowitej przestrzeni 3W otoczenia zarejestrowanego przez stereowizyjny układ kamer jest możliwość uniknięcia kolizji. Jeżeli tak, to obliczany jest kąt skrętu pojazdu, który pozwala na uniknięcie kolizji, a następnie wydane polecenie wykonania takiego skrętu i przejście do kroku pierwszego. Jeżeli w całkowitej przestrzeni 3W nie ma takiej możliwości, to jest przejście do kroku trzeciego.
- W trzecim kroku wydane jest polecenie o wykonaniu skrętu w kierunku, który jest optymalny w danym momencie o 30 deg. Optymalny kierunek skrętu jest zdefiniowany przez podzielenie całkowitej przestrzeni 3W na pół i wybraniu strony, dla której średnia odległość punktów zagrażających przejazdowi jest położona dalej od pojazdu.

Eksperymentalnie zostaje dobrana ilość punktów, które definiują, czy w danym torze ruchu jest przeszkoda.

W drugim kroku obranej metody, w momencie, kiedy jest możliwość wyznaczenia kąta skrętu w celu uniknięcia kolizji z zarejestrowanej przestrzeni 3W, kąt skrętu jest obliczany następująco:

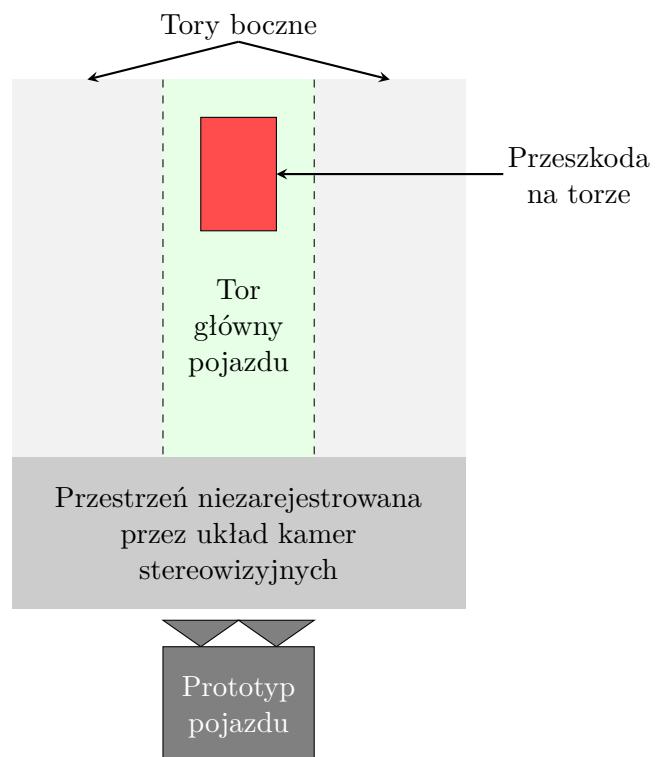
- Rozpoczynając od przestrzeni toru ruchu pojazdu, sprawdzając kolejne prawe oraz lewe przestrzenie toru przejazdu przesunięte w osi X o pewien dystans, wyszukuje się przestrzeń, w której ilość punktów zagrażających przejazdowi jest poniżej dobranego progu. W przypadku, kiedy ilość punktów zagrażających pojazdowi w lewej oraz prawej przestrzeni jest poniżej dobranego progu, wybiera się przestrzeń, w której jest mniejsza ilość punktów zagrażających. Rzut przestrzeni z góry przedstawiony na rys. 2.5.
- Jeżeli istnieje nowa bezkolizyjna przestrzeń toru ruchu pojazdu, obliczany jest tangens kąta zdefiniowany w 6.1, przedstawionego na rys. 2.6. Tangens kąta jest analogicznie obliczony w lewą stronę.

$$tg(\theta) = \frac{x_{nowej_przestrzeni}}{odl_przeszkody_{sr}} \quad (6.1)$$

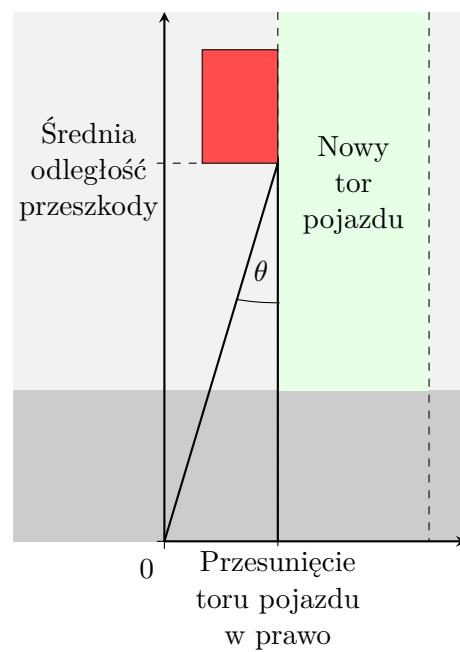
Gdzie:

- | | |
|--------------------------|---|
| θ | – kąt skrętu |
| $x_{nowej_przestrzeni}$ | – przesunięcie nowej podprzestrzeni toru ruchu pojazdu względem przestrzeni głównego toru ruchu pojazdu w osi X |
| $odl_przeszkody_{sr}$ | – średnia odległość punktów przeszkody zagrażających przejazdowi pojazdowi znajdującej się na głównym torze |

Na rys. 2.6 przedstawiono jeden układ współrzędnych podczas wyszukiwania nowej przestrzeni w prawo. Układ ten rozpoczyna się do lewej krawędzi toru głównego pojazdu. Analogiczny układ współrzędnych jest wprowadzony dla wyszukiwania nowego toru w lewo od głównego toru pojazdu, który rozpoczyna się od prawej krawędzi głównego toru pojazdu.



Rysunek 2.5: Rzut z góry przestrzeni z przeszkodą w torze głównym przed pojazdem.



Rysunek 2.6: Wyznaczenie kąta θ skrętu pojazdu.

Rozdział 3. Eksperymenty

W niniejszym rozdziale zostaną przedstawione wyniki eksperymentów dla metod oraz procedur przedstawionych w rozdziale drugim.

Eksperymenty zostały przeprowadzone dla następujących kroków:

1. Kalibracja poszczególnych kamer.
2. Kalibracja układu stereo.
3. Obliczenie macierzy rektyfikacji.
4. Generowanie macierzy rozbieżności.
5. Generowanie obrazu głębi.
6. Generowanie decyzji o kierunku pojazdu na podstawie obrazu głębi.
7. Poruszanie się prototypu pojazdu.

Środowisko testowe

W ramach części praktycznej pracy został wykorzystany następujący język programowania oraz biblioteki do pracy z obrazem komputerowym:

- Python, wersja: 3.7.1
- OpenCV, wersja: 4.0.0.21
- Numpy, wersja: 1.15.4

Model pojazdu autonomicznego

Finalizacja niniejszej pracy została wykonana na modelu pojazdu składającego się z następujących komponentów:

- podwozie z czterema kołami napędzanymi silnikami stałego prądu (Baron-4WD [29]):
 - wymiary: $230 \times 185 \times 110$ mm,
 - maksymalne obciążenie: 800 g,
 - własne zasilanie 6 V,
 - prędkość maksymalna bez obciążenia: $0.68 \frac{m}{s}$,
- układ kompatybilny z Arduino Uno ze sterownikiem silników podwozia (Romeo BLE [30]):
 - mikrokontroler ATmega328,
 - dwukanałowy sterownik silników L298 2A,
- komputer pokładowy do obliczania obrazu głębi oraz realizacja modułu decyzyjnego o przemieszczeniu pojazdu (Raspberry Pi 3 Model B+ [31])?

- procesor: Broadcom BCM2837B0 quad-core 64-bit ARM-8 Cortex-A53 1.4 GHz,
- pamięć RAM 1 GB LPDDR2 z taktowaniem 900 MHz,
- cztery interfejsy USB,
- dwie kamery w trybie stereo (Logitech HD Pro Webcam C920 [32]):
 - pole widzenia: 78°,
 - odległość ogniskowa: 3.67 mm,
 - dostępne rozdzielczości: 360p, 480p, 720p, 1080p.

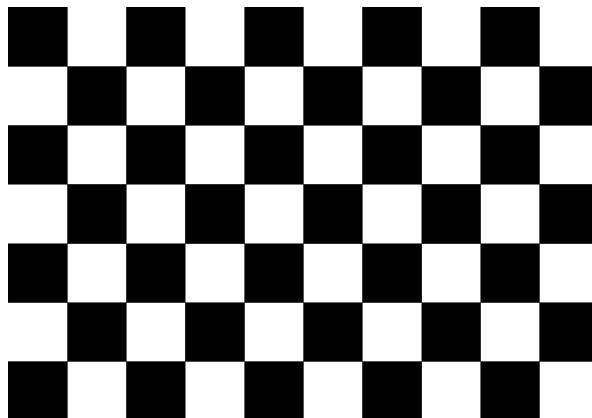


Rysunek 3.1: Wysokopoziomowy schemat komponentów pojazdu autonomicznego.

Przebieg procesu przemieszczania się w prototypie pojazdu autonomicznego:

1. Moduł kamer:
 - dwie kamery jednocześnie rejestrują obrazy,
 - zarejestrowane obrazy są przekazane do modułu komputera pokładowego (Raspberry Pi).
2. Komputer pokładowy:
 - generowana jest macierz rozbieżności i obraz głębi z otrzymanych obrazów,
 - generowana jest decyzja o kierunku przemieszczenia się pojazdu,
 - decyzja o kierunku jest przekazana do układu sterowania silnikami.
3. Układ sterowania silnikami:
 - uruchamia silniki podwozia i dokonuje przemieszczenia pojazdu.

1. Eksperymenty kalibracji kamery



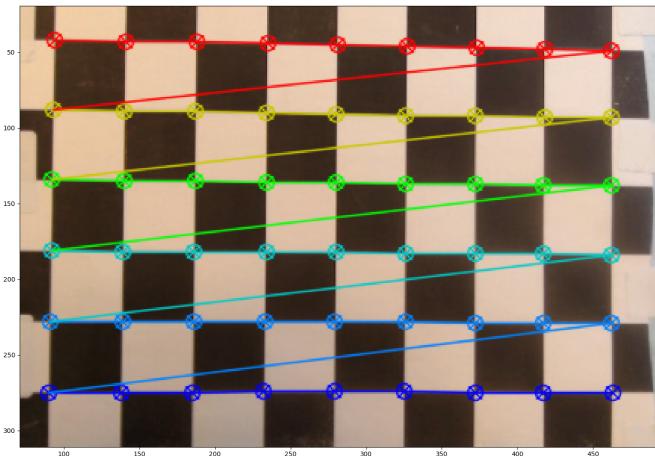
Rysunek 3.2: Wzór szachownicy kalibrującej użytej do kalibracji kamer [14].
Szerokość boku czarnego kwadratu to 23.3mm

Testowane obrazy były w rozdzielczościach (wartości wyrażone w pikselach):

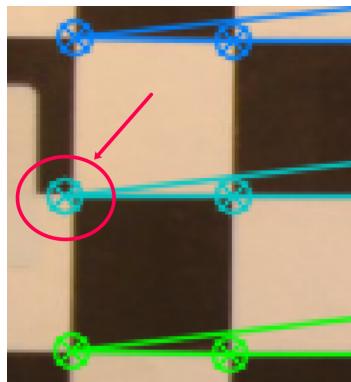
- 640×360 ,
- 1270×720 ,
- 1920×1080 .

Eksperymenty znajdowania punktów przecięć wewnętrznych szachownicy

W tym eksperymencie zostały poddane badaniom trzy procedury i funkcje, które pozwalają na określenie punktów przecięć szachownicy kalibrującej. Badanie zostało wykonane przez wizualizację punktów przecięć naniesionych na oryginalny obraz, dla którego były znalezione punkty. Następnie przeprowadzono badanie oddalenia się poszczególnych punktów przecięć pomiędzy wynikami z trzech metod w celu identyfikacji błędnie znalezionych punktów.



Rysunek 3.3: Wizualizacja punktów przecięć znalezionej funkcją *findChessboardCornersSB*.
Kolorowe linie przedstawiają połączenie kolejnych punktów przecięć szachownicy kalibracyjnej.



Rysunek 3.4: Zbliżenie na błędnie określony punkt przecięcia przez procedurę *cornerSubPix*.
Błędny punkt jest zaznaczony okręgiem i strzałką wskazującą.

Źródło pierwszego zestawu punktów	Źródło drugiego zestawu punktów	Odległości punktów
<i>findChessboardCorners</i>	<i>cornerSubPix</i>	2.3815796
<i>cornerSubPix</i>	<i>findChessboardCornersSB</i>	2.498099
<i>findChessboardCorners</i>	<i>findChessboardCornersSB</i>	1.5785608

Tabela 3.1: Suma odległości punktów pomiędzy poszczególnymi metodami.
Wartości liczbowe są wyrażone w pikselach.

Dokonanie wizualizacji punktów przecięć pozwoliło na identyfikację znacznie oddalonych punktów od ich rzeczywistych lokalizacji. Punkty znalezione z użyciem *findChessboardCornersSB* są bardziej zbliżone do rzeczywistych przecięć pól szachownicy niż pola znalezione z użyciem *cornerSubPix* lub *findChessboardCorners*. Procedura *cornerSubPix* cechowała się zwracaniem

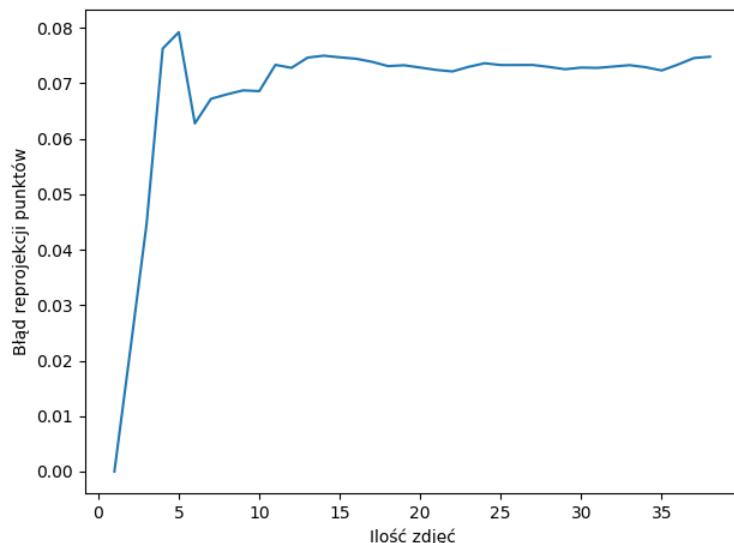
co najmniej jednego punktu, którego lokalizacja była znacznie oddalona od poprawnego miejsca, co nie zdarzało się w przypadku metody zwracającej przybliżone lokalizacje, lub metody określającej lokalizacje od razu. Można z tego wnioskować niestabilność procedury *cornerSubPix*.

Eksperymenty znajdowania parametrów wewnętrznych oraz współczynników zniekształceń kamery

Następujący eksperyment pozwolił na sprawdzenie działania metody obliczającej parametry wewnętrzne oraz współczynniki zniekształceń.

Metoda	$640 \times 360\text{px}$	$1280 \times 720\text{px}$	$1920 \times 1080\text{px}$
Brak stałego punktu	0.107	0.184	0.216
Stał punkt odniesienia	0.076	0.097	0.112

Tabela 3.2: Błąd reprojekcji dla zadanej metody oraz rozdzielczości.



Rysunek 3.5: Błąd reprojekcji punktów w zależności od ilości zastosowanych zdjęć dla metody z określonym punktem odniesienia w rozdzielczości 640x360px.

Metoda	$640 \times 360\text{px}$		$1280 \times 720\text{px}$		$1920 \times 1080\text{px}$	
	f_x	f_y	f_x	f_y	f_x	f_y
Brak stałego punktu	490.44	492.24	977.91	981.27	1451.05	1457.85
Stał punkt odniesienia	481.58	481.12	958.86	956.75	1428.08	1428.68

Tabela 3.3: Obliczone długości rzutowania f_x , f_y dla zadanej metody oraz rozdzielczości. Wartości liczbowe wyrażone w pikselach.

Metoda	640 × 360px		1280 × 720px		1920 × 1080px	
	c_x	c_y	c_x	c_y	c_x	c_y
Brak stałego punktu odniesienia	311.43	181.45	621.89	364.00	931.35	543.88
Stał punkt odniesienia	310.76	179.14	626.88	355.76	934.13	538.10

Tabela 3.4: Środka płaszczyzny obrazowania (c_x, c_y) dla zadanej metody oraz rozdzielczości. Wartości liczbowe wyrażone w pikselach.

Metoda	640 × 360px		1280 × 720px		1920 × 1080px	
	k_1	k_2	k_1	k_2	k_1	k_2
Brak stałego punktu	0.0538	-0.1684	0.0523	-0.1614	0.0490	-0.1664
Stał punkt odniesienia	0.0428	-0.1612	0.0420	-0.1556	0.0407	-0.1643

Tabela 3.5: Współczynniki zniekształceń radialnych k_1, k_2 dla zadanej metody oraz rozdzielczości.

Metoda	640x360px		1280x720px		1920x1080px	
	p_1	p_2	p_1	p_2	p_1	p_2
Brak stałego punktu	0.00074	-0.00077	0.00092	-0.00153	0.00029	-0.00135
Stał punkt odniesienia	-0.00034	-0.00098	-0.00098	-0.00054	-0.00018	-0.00097

Tabela 3.6: Współczynniki zniekształceń tangensowych p_1, p_2 dla zadanej metody oraz rozdzielczości.

Wykonanie powyższych eksperymentów pozwoliło na uzyskanie wiedzy na temat budowy wewnętrznej kamery oraz poznania ich zniekształceń. Dla dwóch metod kalibracji kamery, metoda ze stałym punktem odniesienia pomiędzy obrazami daje lepsze rezultaty, aniżeli metoda nieużywająca takiego punktu. Cechuje się to mniejszym błędem reprojekcji punktów oraz rosnącej wraz ze wzrostem rozdzielczości różnicą między błędami reprojekcji obu metod. Używając powyżej dziesięciu zdjęć do wyliczania parametrów kamery, błąd reprojekcji mieści się w otoczeniu wartości 0.07. Ciekawym wnioskiem jest mały współczynnik zniekształcenia tangensowego w porównaniu do współczynnika radialnego, z którego można wnioskować, iż kamera posiada soczewkę ułożoną w miarę równolegle względem powierzchni światłoczułej.

2. Eksperymenty kalibracji układu stereo

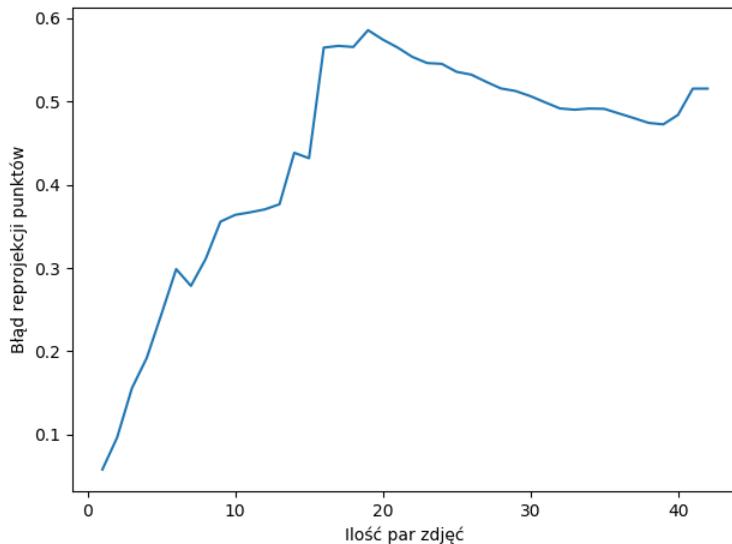
W kalibracji układu stereo zostały przeprowadzone eksperymenty dla metody *calibrateStereo*. Były zbadane macierze rotacji, wektory translacji oraz błędy reprojekcji punktów dla określonej ilości zdjęć tablicy kalibrującej. Na rzecz niniejszych eksperymentów użyto 43 pary zdjęć dla rozdzielczości 640x360px, 52 pary zdjęć dla rozdzielczości 1280x720px oraz 57 par zdjęć dla rozdzielczości 1920x1080px. Badanie zostało wykonane w dwóch etapach: w pierwszym metoda nie dokonywała optymalizacji parametrów wewnętrznych kamer, gdzie zostały użyte parametry wewnętrzne obliczone wcześniej, natomiast w drugim etapie metoda optymalizowała te parametry w czasie określania wzajemnej relacji dwóch kamer.

Metoda	640x360px	1280x720px	1920x1080px
Bez optymalizacji param. wew. kamery	0.551	0.505	1.361
Z optymalizacją param. wew. kamery	0.516	0.479	1.315

Tabela 3.7: Błąd reprojekcji stereo dla zastosowanej metody oraz rozdzielczości.

Rozdzielcość [piksele]	Macierz rotacji [bezwymiarowe]	Kąty rotacji [stopnie]	Wektor translacji [milimetry]
640x360	$\begin{bmatrix} 0.9999 & -0.0006 & -0.0155 \\ 0.0009 & 0.9998 & 0.0206 \\ 0.0155 & -0.0206 & 0.9997 \end{bmatrix}$	$\begin{bmatrix} -1.18 \\ -0.89 \\ 0.05 \end{bmatrix}$	$\begin{bmatrix} -96.003 \\ -1.007 \\ -4.501 \end{bmatrix}$
1280x720	$\begin{bmatrix} 0.9999 & -0.0009 & -0.0103 \\ 0.0010 & 0.9999 & 0.0169 \\ 0.0102 & -0.0169 & 0.9998 \end{bmatrix}$	$\begin{bmatrix} -0.97 \\ -0.59 \\ 0.06 \end{bmatrix}$	$\begin{bmatrix} -94.393 \\ -0.294 \\ 1.042 \end{bmatrix}$
1920x1080	$\begin{bmatrix} 0.9998 & -0.0016 & -0.0177 \\ 0.0018 & 0.9999 & 0.0128 \\ 0.0177 & -0.0128 & 0.9998 \end{bmatrix}$	$\begin{bmatrix} -0.74 \\ -1.02 \\ 0.1 \end{bmatrix}$	$\begin{bmatrix} -94.752 \\ 1.580 \\ -1.00 \end{bmatrix}$

Tabela 3.8: Macierz rotacji kamer, jej odpowiedniki w stopniach dla osi X , Y , Z oraz wektory translacji dla trzech rozdzielcości.



Rysunek 3.6: Błąd reprojekcji punktów w zależności od ilości zastosowanych par zdjęć w rozdzielcości 640x360px.

Na podstawie powyższych eksperymentów, metoda z pewną dokładnością zwraca kąty rotacji oraz translacji między kamerami. Kąty rotacji mają wartości od zera do jednego stopnia, odzwierciedlając położenie głównych osi optycznych kamer, które były ustawione równolegle względem siebie oraz na tej samej wysokości. Wektor translacji, który określa przeniesienie punktu głównego kamer w globalnych współrzędnych, przybliża wartość oddalenia kamer, których główne osie optyczne są rozstawione na szerokości 95 mm. Jednocześnie ten eksperyment zaprzeczył faktowi, o wiążaniu małego błędu reprojekcji z dobrymi wyliczeniami końcowymi, gdyż dla niewielkiej ilości zdjęć, otrzymana macierz rotacji oraz wektor translacji odbiegały od pożądanych wartości. Po serii eksperymentów, dodatkowym wnioskiem jest fakt, iż optymalizacja parametrów wewnętrznych oraz współczynników zniekształceń kamer pomaga w każdej rozdzielcości w celu zmniejszenia błędu reprojekcji.

3. Eksperymenty metody obliczającej macierze rektyfikujące

W niniejszym eksperymencie zostały uruchomione dwie procedury: *stereoRectify* oraz *initUndistortRectifyMap* dla rozdzielcości 640x360px. Pierwsza procedura otrzymała uprzednio obliczone macierze wewnętrzne kamer, współczynniki zniekształceń, rotacje i translacje między kamerami. Druga miała przekazane rezultaty macierzy rotacji i projekcji z *stereoRectify*, dla których zostają obliczone macierze wykorzystane do przekształcenia obrazów, aby były one w tej samej płaszczyźnie oraz wyrównane liniowo.

Procedura *stereoRectify*

Wykorzystane zostały następujące parametry:

1. Macierz rotacji kamer:

$$\begin{bmatrix} 0.99988 & -0.00062 & -0.01547 \\ 0.00094 & 0.99979 & 0.02058 \\ 0.01545 & -0.02059 & 0.99967 \end{bmatrix}, \text{ w stopniach: } \begin{bmatrix} -1.17979 \\ -0.88535 \\ 0.05384 \end{bmatrix}$$

2. Wektor translacji kamer [mm]:

$$\begin{bmatrix} -96.00281 \\ -1.0065 \\ -4.50113 \end{bmatrix}$$

3. Parametry wewnętrzne lewej kamery [px]:

$$\begin{bmatrix} 487.50928 & 0. & 329.21226 \\ 0. & 485.9041 & 178.09907 \\ 0. & 0. & 1. \end{bmatrix}$$

4. Parametry wewnętrzne prawej kamery [px]:

$$\begin{bmatrix} 483.79126 & 0. & 314.03802 \\ 0. & 482.51104 & 177.36137 \\ 0. & 0. & 1. \end{bmatrix}$$

Zwrócone wartości:

1. Macierz rotacji kamery lewej:

$$\begin{bmatrix} 0.99946 & 0.00889 & 0.03158 \\ -0.00922 & 0.9999 & 0.01036 \\ -0.03149 & -0.01065 & 0.99945 \end{bmatrix}, \text{ w stopniach: } \begin{bmatrix} -0.61049 \\ 1.80433 \\ -0.52833 \end{bmatrix}$$

2. Macierz rotacji kamery prawej:

$$\begin{bmatrix} 0.99885 & 0.01047 & 0.04683 \\ -0.01 & 0.9999 & -0.01037 \\ -0.04694 & 0.00989 & 0.99885 \end{bmatrix}, \text{ w stopniach: } \begin{bmatrix} 0.56711 \\ 2.69017 \\ -0.57342 \end{bmatrix}$$

3. Macierz projekcji lewej kamery:

$$\begin{bmatrix} 484.20757 & 0. & 294.28485 & 0. \\ 0. & 484.20757 & 177.62466 & 0. \\ 0. & 0. & 1. & 0. \end{bmatrix}$$

4. Macierz projekcji prawej kamery:

$$\begin{bmatrix} 484.20757 & 0. & 294.28485 & -46538.90482 \\ 0. & 484.20757 & 177.62466 & 0. \\ 0. & 0. & 1. & 0. \end{bmatrix}$$

5. Macierz rzutowania punktów obrazu z rozbieżnościami na homogeniczne punkty 3W:

$$\begin{bmatrix} 1. & 0. & 0. & -294.28485 \\ 0. & 1. & 0. & -177.62466 \\ 0. & 0. & 0. & 484.20757 \\ 0. & 0. & 0.0104 & -0. \end{bmatrix}$$

Powyższa procedura zwróciła dwie macierze rotacji, które zostały dobrane przez minimalizację błędu reprojekcji, tak aby ich wypadkowa była równa przekazanej macierzy rotacji kamer. Parametry wewnętrzne w macierzach projekcji są takie same dla obu kamer, natomiast macierz projekcji prawej kamery zawiera element w pierwszym wierszu, który łączy przesunięcie kamer w globalnej współrzędnej X oraz długości rzutowania.

Procedura *initUndistortRectifyMap*

Do procedury zostały przekazane wartości obliczone wcześniej. Poniższe wartości są dla lewej kamery oraz rozdzielczości 640x360px.

1. Parametry wewnętrzne lewej kamery [px]:

$$\begin{bmatrix} 487.50928 & 0. & 329.21226 \\ 0. & 485.9041 & 178.09907 \\ 0. & 0. & 1. \end{bmatrix}$$

2. Wektor Współczynników zniekształceń lewej kamery:

$$[0.06081 \quad -0.18648 \quad 0.00025 \quad -0.00175 \quad 0.10908]$$

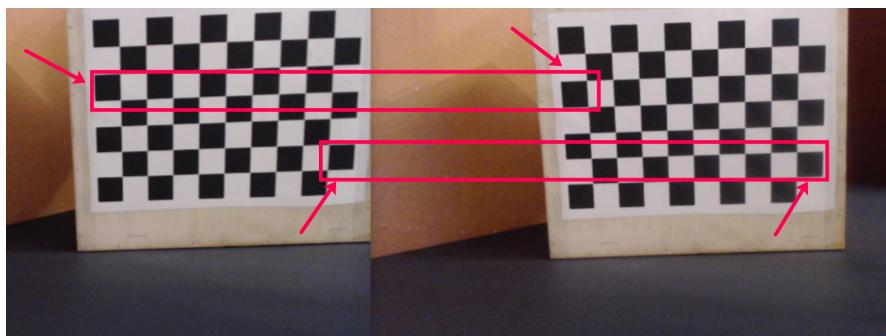
3. Macierz rotacji kamery lewej:

$$\begin{bmatrix} 0.99946 & 0.00889 & 0.03158 \\ -0.00922 & 0.9999 & 0.01036 \\ -0.03149 & -0.01065 & 0.99945 \end{bmatrix}, \text{ w stopniach: } \begin{bmatrix} -0.61049 \\ 1.80433 \\ -0.52833 \end{bmatrix}$$

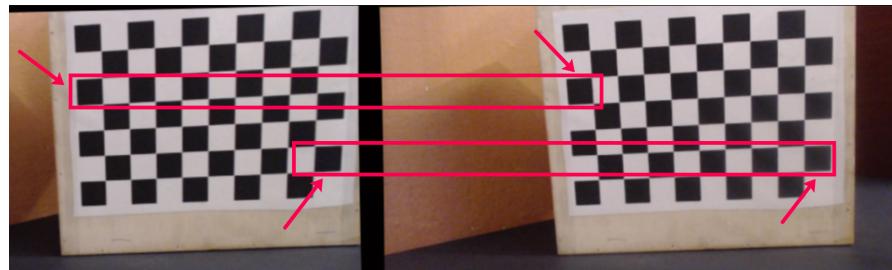
4. Macierz projekcji lewej kamery:

$$\begin{bmatrix} 484.20757 & 0. & 294.28485 & 0. \\ 0. & 484.20757 & 177.62466 & 0. \\ 0. & 0. & 1. & 0. \end{bmatrix}$$

Otrzymany wynik procedury - macierze transformacji obrazu - jest przekazany do procedury *remap*, która dokonuje przekształcenia obrazu, korzystając z tych macierzy.



Rysunek 3.7: Para zdjęć szachownicy kalibrującej przed rektyfikacją.



Rysunek 3.8: Para zdjęć szachownicy kalibrującej po rektyfikacji.

W powyższej parze zdjęć widać dokonaną zmianę w dwóch zaznaczonych obszarach, których początkowe pola nie były wyrównane wierszami, po rektyfikacji obraz, w szczególności zaznaczone pola szachownicy są wyrównane liniowo. Czarny pionowy obszar między obrazami wynika z przekształcenia obrazu. Obliczone macierze transformacji obrazu pozwalają na jego modyfikację, tak aby było wrażenie zdjęć wykonanych przez parę idealnie równolegle ułożonych kamer z takimi samymi parametrami wewnętrznymi. Obrazy wynikowe znacznie ułatwiają wyszukiwanie punktów wspólnych między obrazami.

4. Eksperymenty generowania macierzy rozbieżności oraz obrazu głębi

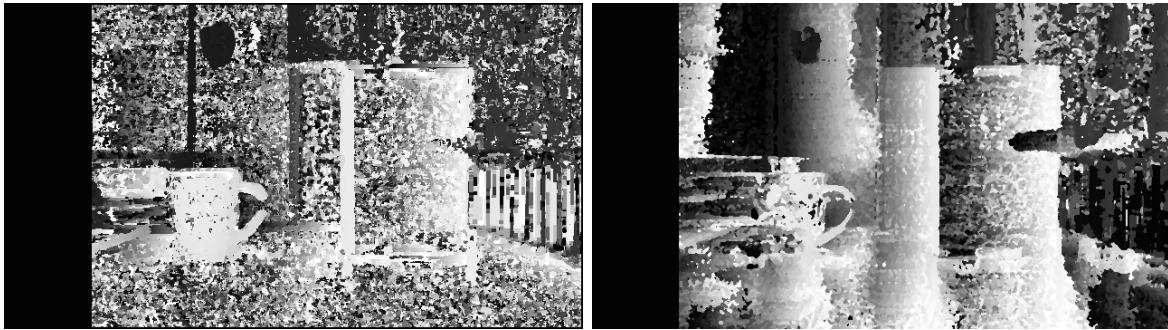
W tej sekcji zostały poddane eksperymantom dwie metody znajdowania punktów korespondujących oraz generujących macierze rozbieżności: lokalna, oraz częściowo-globalna metoda dopasowania blokowego. Następnie była przeprowadzona eksploracja wpływu różnych parametrów na generowanie macierzy rozbieżności lokalną metodą dopasowania blokowego. Kolejno została poddana badaniu metoda generująca obraz głębi.

Do eksperymentu wykorzystano parę zdjęć o rozdzielczości 640x360px, które przedstawiały dwa przedmioty umieszczone obok siebie: kubek oraz kartonowe pudełko w odległości 55cm od kamery. Główne osie optyczne kamer były oddalone od siebie o 95mm.



Rysunek 3.9: Para obrazów wykorzystanych do eksperymentów z metodami generowania obrazu rozbieżności.

Porównanie metod generowania macierzy rozbieżności: BM i SGBM



(a) Obraz rozbieżności z metody BM

(b) Obraz rozbieżności z metody SGBM

Rysunek 3.10: Wizualizacja macierzy rozbieżności dla metod BM oraz SGBM.

Komputer	640x360px		1280x720px		1920x1080px	
	BM	SGBM	BM	SGBM	BM	SGBM
Dell XPS	5286	3216	1655	693	797	259
Raspberry Pi	481	77	77	16	39	6

Tabela 3.9: Porównanie ilości wygenerowanych obrazów rozbieżności w ciągu jednej minuty dla zadanej rozdzielczości oraz metody na komputerze Dell XPS [33] oraz Raspberry Pi.

Metoda BM działa lepiej, gdy elementy były bardziej wzorzyste, co można zaobserwować w przypadku kubka na powyższej parze zdjęć, z drugiej strony SGBM lepiej dopasowuje powierzchnie posiadające jednolitą teksturę, co można zaobserwować na ścianie za przedmiotami oraz stoliku pod przedmiotami, przez co jest wrażenie wygładzenia obszarów.

Jeśli chodzi o czas generowania obrazu rozbieżności, z praktycznych powodów, lepiej wypada metoda BM, która generuje obraz rozbieżności powyższej ośmiu razy na sekundę dla rozdzielczości 640x360px. Przy przemieszczaniu się pojazdu z prędkością $11.33 \frac{mm}{s}$, umożliwia to analizę obrazu co 1.4 mm przemieszczenia się pojazdu w linii prostej.

Wpływ parametrów na generowanie macierzy rozbieżności dla wybranej metody BM

W niniejszym eksperymencie został zbadany wpływ wybranych parametrów metody BM na wizualizację macierzy rozbieżności. Wykorzystana została para zdjęć 3.9, która była użyta wyżej w celu porównania metod BM i SGBM.

Podczas tego badania, wszystkie parametry metody BM były stałe z wyjątkiem parametru, gdzie oceniano, jak jego zmiana wpływa na wygenerowany obraz rozbieżności.

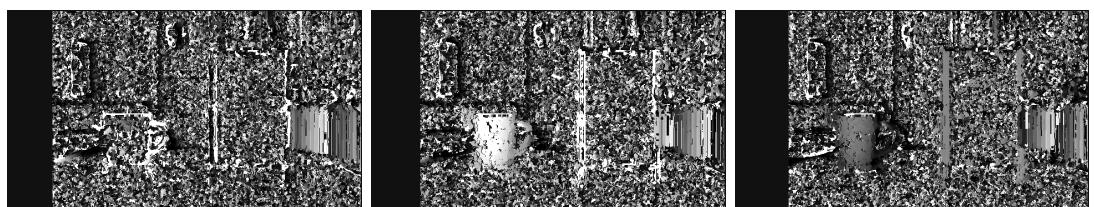


(a) 80px

(b) 96px

(c) 112px

Rysunek 3.11: Wpływ szerokości dopasowywania na wynikową macierz rozbieżności.



(a) 65px

(b) 70px

(c) 80px

Rysunek 3.12: Wpływ minimalnej rozbieżności na wynikową macierz rozbieżności.



(a) 5px

(b) 7px

(c) 11px

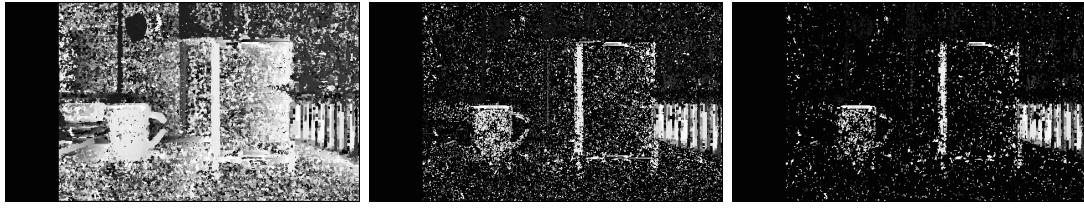
Rysunek 3.13: Wpływ rozmiaru bloku na uzyskaną macierz rozbieżności.



(a) Normalizowanie intensywności w obrębie okna dopasowania.

(b) Przekształcenie obrazów na pierwsze pochodne Sobela względem osi X.

Rysunek 3.14: Wpływ typu filtrowania wstępного par obrazów na wynikową macierz rozbieżności.

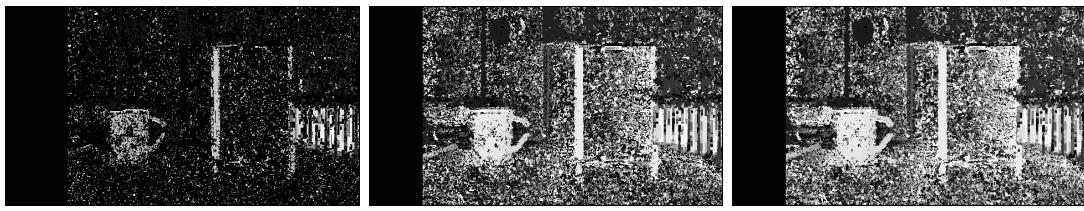


(a) Rozmiar okna 0px

(b) Rozmiar okna 1px

(c) Rozmiar okna 2px

Rysunek 3.15: Wpływ wielkości okna plamki na wynikową macierz rozbieżności przy dopuszczalnej zerowej różnicy między rozbieżnościami w obrębie okna plamki.



(a) Dopuszczalna różnica 0

(b) Dopuszczalna różnica 2

(c) Dopuszczalna różnica 4

Rysunek 3.16: Wpływ parametru określającego dopuszczalne różnice rozbieżności między kolejnymi pikselami w obrębie plamki o szerokości 1px na wynikową macierz rozbieżności.

Powysze eksperymenty zademonstrowały, jak poszczególne parametry wpływają na wygenerowaną macierz rozbieżności. Im dany piksel jest jaśniejszy, tym rzeczywista odległość punktu, który reprezentuje, jest bliżej kamery. Zauważalny czarny pionowy pasek z lewej strony każdego obrazu jest obszarem, który zmniejsza szerokość pola widzenia w celu uzyskania obrazu rozbieżności, kontrolowany za pomocą parametru minimalnej rozbieżności (rezultaty w 3.12) i zakresu wyszukiwanej rozbieżności (rezultaty w 3.11). Aby uzyskać informację o przedmiotach umieszczonych na dystansie minimum 55 cm od kamer, których główne osie optyczne są rozmieszczone na szerokość 95 mm, trzeba poświęcić minimum 96 pikseli z dostępnych 640 pikseli na szerokość, aby uzyskać dopasowanie między pikselami pary obrazów. Im bliższa jest oczekiwana odległość, tym większy musi być parametr minimalnej rozbieżności lub zakres wyszukiwanego dopasowania, jednocześnie pomniejszając szerokość uzyskanej macierzy rozbieżności. Przedmioty znajdujące się w obszarze bliższym niż 55 cm lub w przypadku, gdy szerokość dopasowania jest zbyt wąska w porównaniu do potrzebnej w celu znalezienia wspólnych punktów między obrazami, uzyskuje się szum, który jest widoczny na pierwszych obrazach w 3.11 oraz 3.12. Użycie obrazów, dla których zostały znormalizowane intensywności w obrębie okna dopasowania, wizualnie zwracało lepsze wyniki, niż użycie filtra Sobela względem osi X. Parametry kontrolujące rozmiar plamki, widoczne na 3.15 oraz 3.16 pozwalają skutecznie eliminować małe obszary rozbieżności, które znacznie odstają od otoczenia. Ciekawym parametrem był również parametr kontrolujący rozmiar bloku, pozwalający na kontrolę efektu wygładzenia ujęty na 3.13.

W A.1 został ukazany interfejs programu, który został wykonany w ramach niniejszej pracy, aby w praktyczny sposób uzyskiwać informacje o wpływie poszczególnych parametrów na wynikowy obraz rozbieżności.

Eksperymentalnie zostały wyznaczone następujące parametry generowania obrazu rozbieżności, aby umożliwić wykrywanie obiektu na odległości 30 cm od kamer:

- metoda: BM,

- liczba rozbieżności: 160 px,
- minimalna rozbieżność: 0 px,
- rozmiar bloku: 5 px,
- typ filtrowania wstępne: Znormalizowanie intensywności obrazu w oknie dopasowania,
- współczynnik unikalności: 0,
- zakres plamki: 2 px,
- wielkość okna plamki: 1 px,
- dopuszczalna różnica między rozbieżnościami par obrazów prawo-lewo i lewo-prawy: 0.

Metoda generująca obraz głębi

Celem tego eksperymentu było zbadanie, jak dokładnie odległości są zarejestrowane dla poszczególnych punktów, względem ich realnych odległości od kamer. Do eksperymentu wykorzystano zdjęcia przedmiotu umieszczonego w odległościach od 30 cm do 90 cm. Wykorzystując zdjęcie przedmiotu umieszczonego na dystansie 30 cm, został zbadany kąt percepcji głębi.



Rysunek 3.17: Przykładowa para obrazów oraz obraz rozbieżności wykorzystany do weryfikacji odległości przedmiotu od kamery.
Przedmiot umieszczony na dystansie 35cm.

Faktyczna odległość	Uśredniona odległość z głębi	Różnica
30cm	31.4cm	1.4cm
35cm	37.3cm	2.3cm
40cm	41.8cm	1.8cm
45cm	47.1cm	2.1cm
50cm	52.8cm	2.8cm
60cm	62.9cm	2.9cm
70cm	73.2cm	3.2cm
80cm	83.6cm	3.6cm
90cm	93.3cm	3.3cm

Tabela 3.10: Różnice między faktycznymi odległościami przedmiotu od kamer a dystansem wyznaczonym w obrazie głębi.

Otrzymane obrazy głębi przedstawiają przybliżone odległości przedmiotów od kamer z minimalnym błędem wynoszącym 1.4 cm na 30 cm, zwiększającym się aż do 3.3 cm na dystansie 90 cm. Szerokość percepcji na podstawie badania zarejestrowanej głębi na dystansie 60 cm to 25° względem głównej osi optycznej, uzyskując całkowicie 50°, przy rejestraniu przedmiotów oddalonych minimalnie na 33 cm od kamer. Kąt można zwiększyć przez zmniejszenie zakresu rozbieżności, jednocześnie zwiększając minimalną odległość przedmiotów od kamer.

Co więcej, przez projekcję punktów obrazu do 3W, poza informacją o odległości danego punktu od kamery, otrzymuje się również informacje o przemieszczeniu tego punktu w pozostałych osiach względem kamer, umożliwiając lokalizowanie przedmiotu w przestrzeni 3W w relacji do pojazdu.

5. Eksperymenty generowania decyzji o dalszym kierunku pojazdu

W niniejszej sekcji eksperimentalnej badaniu został poddany moduł odpowiedzialny za generowanie decyzji o następnym kierunku pojazdu na podstawie obrazu głębi. Wcześniej uzyskany obraz głębi charakteryzuje się tym, że każdy punkt posiada współrzędne traktujące zredukowany środek lewej kamery układu stereowizyjnego jako środek układu współrzędnych, wobec czego jest możliwość określenia punktów leżących na torze pojazdu oraz planowania alternatywnej trasy. Niniejsza seria eksperymentów przebiegała z użyciem pary kamer w trybie stereo, umieszczonych na modelu pojazdu. Główne osie optyczne kamer były umieszczone 10.8 cm ponad podłożem. Przedmioty użyte jako przeszkody do badania to: kartonowe pudełko o wymiarach 15x23x6 cm, przenośna ładowarka o wymiarach 13.9x6.8x1.5 cm oraz ściana. Głównym założeniem badania było sprawdzenie, czy moduł decyzyjny jest w stanie określić, czy w ustalonym zasięgu przed pojazdem znajduje się przeszkoda oraz jaki skręt należy wykonać, aby uniknąć kolizji. Dla przypadku, gdzie przeszkodą była ściana, nastąpiła weryfikacja, jak pojazd reaguje na zbliżenie się do przeszkody, która całkowicie blokuje dalszy przejazd. Sekcja ta rozpocznie się od przedstawienia wizualizacji, w której moduł decyzyjny etapowo przygotowuje obraz głębi.

Wizualizacja ograniczenia obrazu głębi



(a) Obraz lewej kamery.



(b) Całkowity obraz głębi.



(c) Ograniczenie obrazu głębi na wysokość.



(d) Ograniczenie obrazu głębi do toru pojazdu.

Rysunek 3.18: Ograniczenia naniesione na obraz głębi wykorzystane w module decyzyjnym.

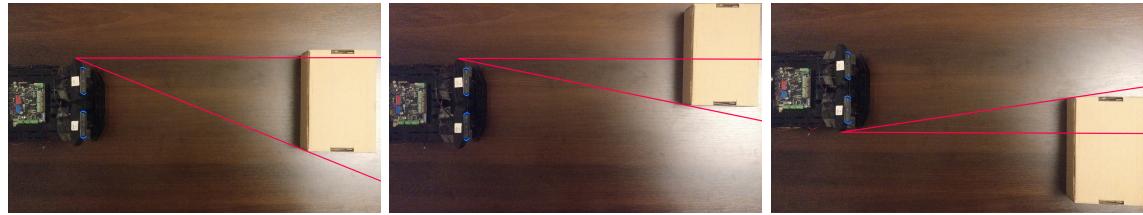
W 3.18, zostały przedstawione etapy, w których obrazu głębi jest stopniowo ograniczany. Z całkowitego obrazu głębi, najpierw zostaje dokonane ograniczenie na wysokość, eliminując punkty, które nie zagrażają pojazdowi: są ponad pojazdem lub zawierają się w podłożu. Następnie jest dokonane dalsze ograniczenie na szerokość, eliminując punkty, które nie zawierają się w drodze pojazdu na wprost. Takie ograniczenia pozwalają na wnioskowanie, gdzie znajduje się przeszkoda oraz jaki jest potrzebny kierunek i kąt skrętu, aby uniknąć kolizji z przeszkodą.

Weryfikacja modułu decydującego o kierunku pojazdu

Dla niniejszej weryfikacji zostało rozpatrzonych pięć przypadków, będących istotnym wycinkiem spektrum sytuacji, które pojazd może spotkać na swojej drodze:

1. Przeszkoda o szerokości 23 cm stojąca na drodze pojazdu.
2. Przeszkoda o szerokości 5 cm stojąca na drodze pojazdu.
3. Przeszkoda o szerokości 1.5 cm stojąca na drodze pojazdu.
4. Okno umożliwiające przejazd o szerokości szerszej 25 cm, zbliżonej do szerokości pojazdu.
5. Przeszkoda uniemożliwiająca całkowicie jazdę na wprost.

Dalej zostały przedstawione konfiguracje przeszkód w relacji do modelu pojazdu oraz czerwonymi liniami zaznaczony minimalny kąt skrętu, który należy wykonać, aby bezpiecznie przejechać obok przeszkody. Dla każdej sytuacji z wyjątkiem ściany zostało przedstawiony kąt skrętu wyznaczony przez moduł decyzyjny.

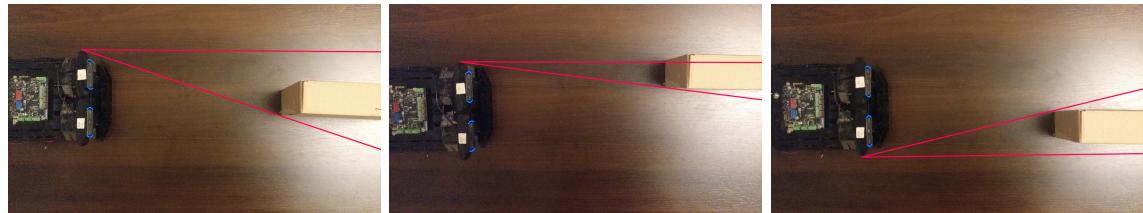


(a) Przeszkoda centralnie przed pojazdem.
 (b) Przeszkoda blokuje lewą część toru.
 (c) Przeszkoda blokuje prawą część toru.

Rysunek 3.19: Zdjęcie z góry toru pojazdu zawierającego przeszkodę o szerokości 23 cm znajdującą się 50 cm przed pojazdem.

Sytuacja	Minimalny kąt skrętu	Uzyskany kąt
Przeszkoda centralnie przed pojazdem	23 stopnie	25 stopni
Przeszkoda blokuje lewą część toru	9 stopni	12 stopni
Przeszkoda blokuje prawą część toru	9 stopni	11 stopni

Tabela 3.11: Kąty skrętu dla przeszkody o szerokości 23 cm znajdującej się 50 cm przed pojazdem, przedstawionej w rys. 3.19.

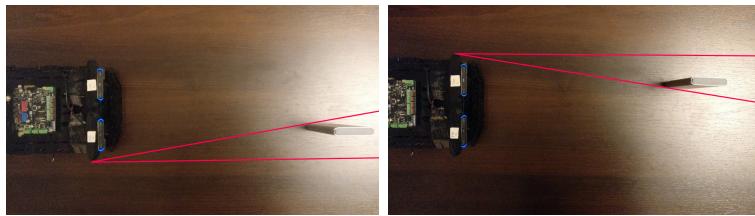


(a) Przeszkoda centralnie przed pojazdem.
 (b) Przeszkoda blokuje lewą część toru.
 (c) Przeszkoda blokuje prawą część toru.

Rysunek 3.20: Zdjęcie z góry toru pojazdu zawierającego przeszkodę o szerokości 5 cm znajdującą się 40 cm przed pojazdem.

Sytuacja	Minimalny kąt skrętu	Uzyskany kąt
Przeszkoda centralnie przed pojazdem	18 stopni	18 stopni
Przeszkoda blokuje lewą część toru	7 stopni	7 stopni
Przeszkoda blokuje prawą część toru	13 stopni	15 stopni

Tabela 3.12: Kąty skrętu dla przeszkody o szerokości 5 cm znajdującej się 40 cm przed pojazdem, przedstawionej w rys. 3.20.

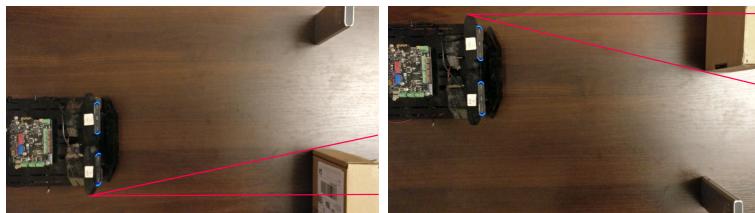


(a) Przeszkoda blokuje prawą część toru pojazdu. (b) Przeszkoda blokuje lewą część toru pojazdu.

Rysunek 3.21: Zdjęcie z góry toru pojazdu zawierającego przeszkodę o szerokości 1.6 cm znajdującą się 43 cm przed pojazdem.

Sytuacja	Minimalny kąt skrętu	Uzyskany kąt
Przeszkoda blokuje lewą część toru	10 stopni	13 stopni
Przeszkoda blokuje prawą część toru	8 stopni	8 stopni

Tabela 3.13: Kąty skrętu dla przeszkody o szerokości 1.6 cm znajdującej się 43 cm przed pojazdem, przedstawionej w rys. 3.21.

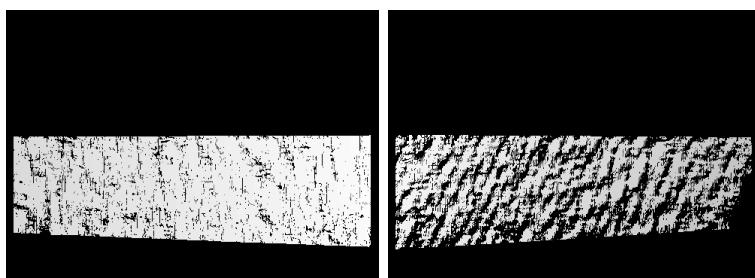


(a) Okno umiejscowione z lewej strony. (b) Okno umiejscowione z prawej strony.

Rysunek 3.22: Zdjęcie z góry toru pojazdu, z ulokowanym oknem umożliwiającym przejazd między przeszkodami.

Sytuacja	Minimalny kąt skrętu	Uzyskany kąt
Okno umiejscowione z lewej strony	13 stopni	14 stopni
Okno umiejscowione z prawej strony	12 stopni	13 stopni

Tabela 3.14: Kąty skrętu dla okna ulokowanego między przeszkodami, przedstawionego w rys. 3.22.



(a) W lewo. (b) W prawo.

Rysunek 3.23: Wizualizacja obrazu głębi ograniczonej z góra i dołu dla sytuacji, gdy pojazd znajduje się 45 cm przed ścianą z 5-stopniowym skrętem w lewo oraz prawo.

Sytuacja	Decyzja
Podjazd do ściany z lewej strony	Decyzja skrętu w prawo
Podjazd do ściany z prawej strony	Decyzja skrętu w lewo

Tabela 3.15: Decyzje o dalszej drodze pojazdu dla przeszkody o dużej szerokości względem pojazdu na postawie obrazów macierzy rozbieżności przedstawionych w rys. 3.23.

Komputer	Średni czas	Najgorszy przypadek
Dell XPS	0.01	0.026
Raspberry Pi	0.17	0.54

Tabela 3.16: Czas generowania decyzji na komputerze Dell XPS [33] oraz Raspberry Pi. Wartości liczbowe podane w sekundach.

Mając na uwadze powyższe wyniki z sytuacji, które pojazd może spotkać na swojej drodze, jest duże prawdopodobieństwo, iż moduł decyzyjny dobierze odpowiednią decyzję o dalszym kroku. W eksperymentach weryfikujących unikanie kolizji z przeszkodami, na rys. 3.19, 3.20 oraz 3.21, moduł decyzyjny korzystając z weryfikacji alternatywnych kierunków, wybrał minimalny kąt skrętu, aby swobodnie się przemieścić obok przeszkody. W przypadku przejazdu przez okno, w rys. 3.22, pojazd również dobrał taki kąt skrętu, który umożliwiłby swobodny przejazd przez okno. Dwa przypadki eksperymentów ze ścianą pokazały, iż moduł decyzyjny właściwie stwierdził, że optymalnym kierunkiem skrętu w przypadku, gdy w całej zarejestrowanym obrazie głębi nie ma możliwości przejazdu, jest odpowiednio skręt w prawo lub lewo w zależności od ustawienia pojazdu względem ściany.

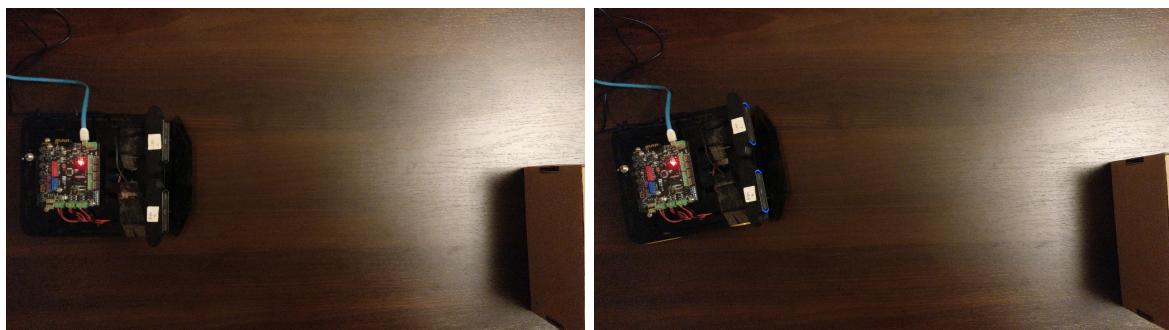
Po weryfikacji czasów generowania decyzji o kierunku, przedstawionych w 3.16, można stwierdzić, że moduł na Raspberry Pi powinien być w stanie wolno się przemieszczać, znacznie zwalniając prędkość w momencie podjeżdżania do ścian lub przeszkód okupujących całą mapę głębi. Ze względu na wolny czas generowania decyzji na komputerze Raspberry Pi jako komputerze pokładowym, ciągła analiza otoczenia jest raczej ograniczona do obiektów statycznych, które się nie przemieszczają względem pojazdu.

6. Eksperymenty praktyczne prototypu pojazdu

Poniższa, ostatnia sekcja eksperimentalna przedstawia badanie prototypu pojazdu autonomicznego, na którym został umieszczony wcześniej opisany stereowizyjny układ kamer, komputer pokładowy Raspberry Pi oraz układ sterujący silnikami pojazdu. W pierwszej podsekcji zostaną zaprezentowane rezultaty skrętów wykonanych przez prototyp pojazdu po komendzie z komputera pokładowego na podstawie analizy umieszczonej przed nim przeszkody. W drugiej podsekcji zostaną przedstawione wyniki przemieszczania się pojazdu w zaaranżowanym otoczeniu w celu zbadania zachowania w obliczu napotkanych przeszkód na swojej drodze.

Eksperymenty skrętu pojazdem wykonanego przez układ sterowania silnikami pojazdu

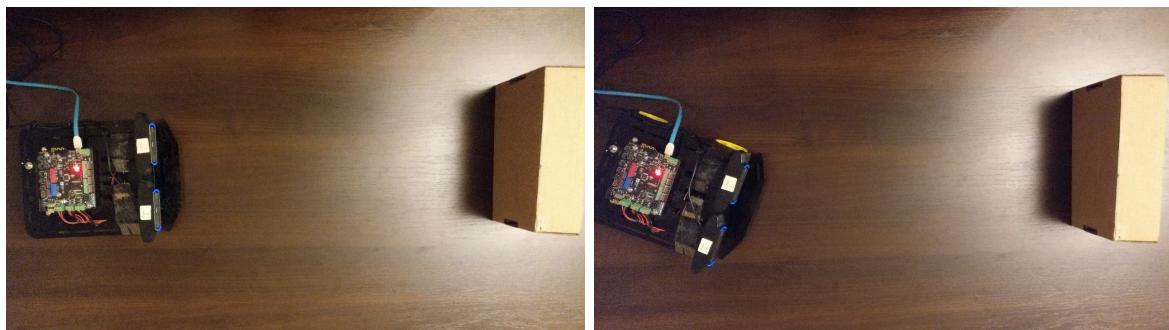
W tej podsekcji został zbadany układ sterujący silnikami pojazdu, aby przetestować jego działanie podczas wykonywania komendy skrętu otrzymanej z komputera pokładowego. Do badania wykorzystano dwa typy przeszkód: szerokiego pudełka na 23 cm oraz przenośnej ładowarki o szerokości 1.6 cm. Badanie przebiegało przez ustawienie przeszkody przed pojazdem, która następnie była przeanalizowana przez komputer pokładowy, wydający polecenie skrętu do układu sterowania silnikami pojazdu. W tym badaniu ruchu pojazdu były ograniczone do obracania się wokół własnej osi, bez przemieszczania się do przodu. Na następujących zdjęciach z przebytych eksperymentów jest przedstawione początkowe oraz końcowe ustawienie pojazdu.



(a) Początkowe ustawienie.

(b) Po wykonaniu skrętu.

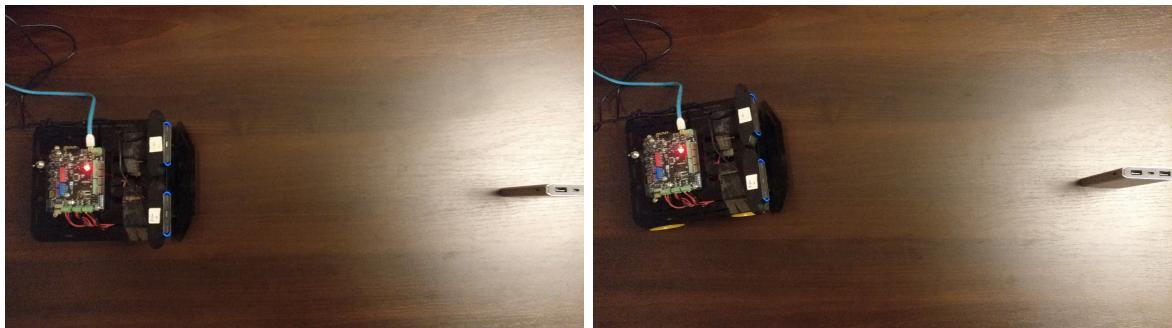
Rysunek 3.24: Sytuacja, w której szersza przeszkoda jest umieszczona z prawej strony pojazdu.



(a) Początkowe ustawienie.

(b) Po wykonaniu skrętu.

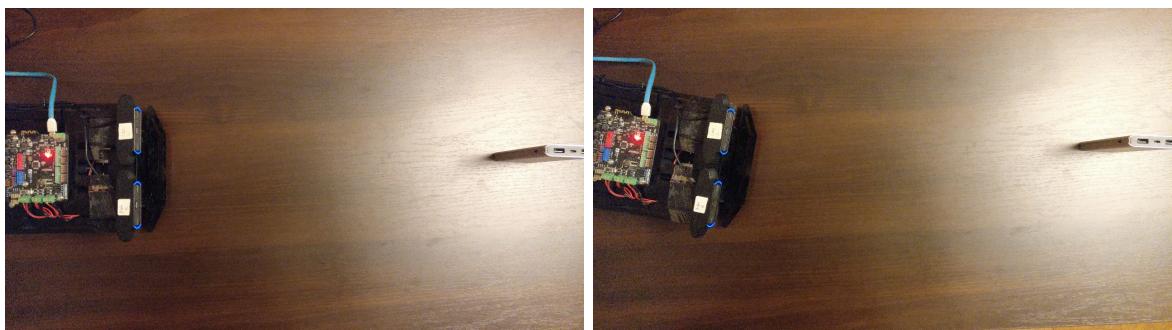
Rysunek 3.25: Sytuacja, w której szersza przeszkoda jest umieszczona z lewej strony pojazdu.



(a) Początkowe ustawienie.

(b) Po wykonaniu skrętu.

Rysunek 3.26: Sytuacja, w której wąska przeszkoda jest umieszczona z prawej strony pojazdu.



(a) Początkowe ustawienie.

(b) Po wykonaniu skrętu.

Rysunek 3.27: Sytuacja, w której wąska przeszkoda jest umieszczona z lewej strony pojazdu.

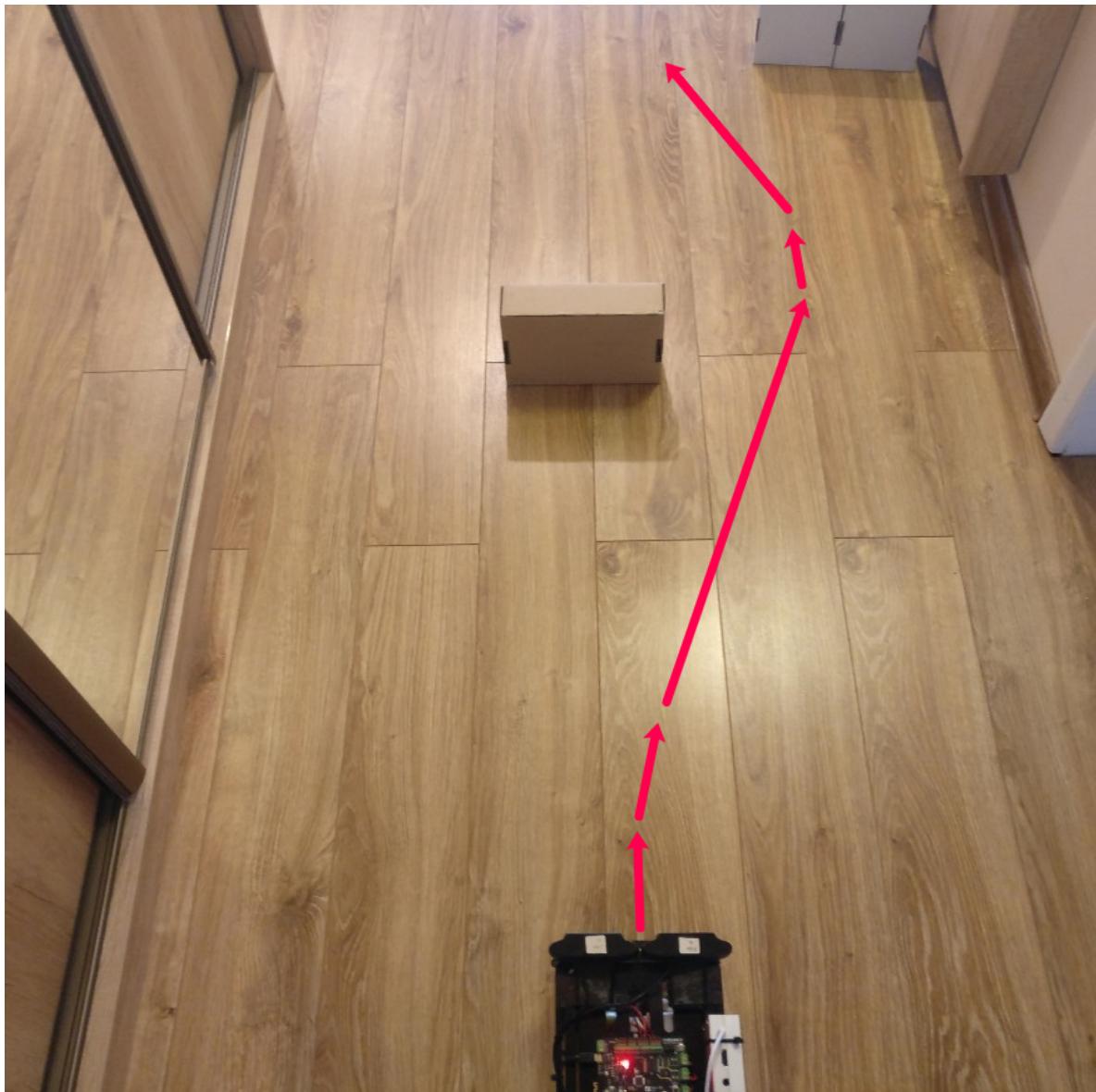
W powyższych czterech przypadkach, które były wybranymi przykładami spośród wielu, pojazd wykonał skręt pozwalający na uniknięcie kolizji. Podczas testów, od zarejestrowania obrazów do wykonania skrętu, średni czas to 0.71 sekundy. Oceniając współpracę modułu decyzyjnego oraz układu sterującego podwoziem, w tym eksperyencie moduł decyzyjny dobierał kąt skrętu, a układ sterujący silnikami wykonał skręt, który z sukcesem pozwalał na uniknięcie kolizji. Podczas manualnego eksperimentu, który polegał na zbliżaniu przeszkody z prawej lub lewej strony, pojazd dokonał odpowiednio kilkustopniowego skrętu w lewo lub prawo, tak aby jego główny tor ruchu był przejezdny.

Eksperymenty przemieszczania się prototypu pojazdu

Ostateczna seria eksperymentów była zrealizowana z samodzielnym prototypem pojazdu autonomicznego w zaaranżowanym torze przeskóde. Spośród wykonanych eksperymentów, zostały przedstawione dwa, które pokrywały szerszy zakres przemieszczania się między przeskodami. Jako przeskody zostały użyte kartonowe pudełka oraz elementy holu mieszkania. Na poniższych rysunkach, czerwone strzałki oznaczają drogę przebytą przez pojazd. W przerwach między strzałkami nastąpił skręt wykonany przez pojazd, im dłuższa strzałka, tym dłuższa była droga bez wykonanego skrętu.



Rysunek 3.28: Autonomiczny tor ruchu pojazdu dla pierwszego przypadku przeszkód.



Rysunek 3.29: Autonomiczny tor ruchu pojazdu dla drugiego przypadku przeszkód.

Dla powyższych dwóch przypadków, przedstawione na rys. 3.28 oraz 3.29, które zostały wybrane spośród kilkudziesięciu eksperymentów, sukcesywnie było powtarzane unikanie kolizji z elementami stałymi, jak kartonowe pudełka, ściany, znajdujące się w polu widzenia kamer stereowizyjnych, tj. na dystansie 33 do 70 cm przed pojazdem.

Dla przyjętej metody, dużym ograniczeniem są nieregularne, wąskie elementy, jak np. kable, które będąc na drodze pojazdu, były ignorowane prowadząc do kolizji. Z faktu, że przyjęta metoda jest metodą lokalną, zdarzały się sytuacje, w których pojazd wykonał skręt unikając kolizji ze ścianą, następnie jadąc równolegle do ściany, napotkał przeszkodę i wykonał skręt w kierunku ściany prowadząc do kolizji. Wynika to z faktu znajdowania się przeszkody w zbyt bliskim obszarze, w tym przypadku bliżej niż 33 cm, co powoduje brak uchwycenia przeszkody w obrazie głębi.

Zarejestrowane zostały również ograniczenia sprzętowe wybranych kamer. Elementy takie jak ilość zdjęć przechowywanych w buforze kamer, czas potrzebny na rejestrację jednego zdjęcia, efekt rozmycia zdjęć, podczas gdy kamery są w ruchu, nie pozwalały na ciągłą analizę przestrzeni. Zamiast ciągłej jazdy, pojazd musi się zatrzymać, wykonać zdjęcia, zbadać przestrzeń, wybrać kierunek i dokonać przemieszczenia.

Ponadto, dociążony pojazd przez obudowę komputera pokładowego oraz przenośną ładowarkę, zwiększył średni czasu na wykonanie skrętu o 90 stopni z 9.7 do 12.5 sekundy. Dodatkowe ograniczenia silników, zwiększyły minimalny kąt skrętu do 10 stopni.

Rozdział 4. Zakończenie

Niniejsza praca była zainspirowana procesem zachodzącym w człowieku podczas percepcji otoczenia, tj. wrażeniem wzrokowym i głębi podczas poruszania się. W pracy zostało to przełożone na eksplorację procesu stereowizji, która została połączona z metodą unikania kolizji. Obie te rzeczy zostały użyte w zbudowanym prototypie pojazdu. W głównym zamysle pracy było zbadanie stereowizji jako jedynego źródła informacji o otoczeniu przekazywanych dalej do modułu decyzyjnego pojazdu w celu podjęcia decyzji o dalszym kierunku przemieszczania.

Cele postawione przez autora pracy to zrealizowanie i zbadanie procesu widzenia stereowizyjnego i metody unikania kolizji na taniem komputerze pokładowym prototypu. Oba cele zostały osiągnięte, prowadząc do ciekawych problemów, rozwiązań i wniosków podczas realizacji metod.

Jako „tani komputer pokładowy” został wybrany Raspberry Pi 3 Model B+ [31], w niektórych eksperymentach porównany z komputerem Dell XPS 9560 [33].

Z perspektywy pytań, które zadał sobie autor na samym początku, zostały uzyskane następujące odpowiedzi:

- Uzyskany obraz głębi niedokładnie odzwierciedla otoczenia, natomiast jest wystarczającym źródłem informacji na temat przeszkód znajdujących się przed pojazdem.
- Obrany komputer pokładowy jest w stanie wygenerować obraz głębi oraz zrealizować proces wykrywania przeszkód.
- Obrany komputer pokładowy nie jest w stanie w czasie rzeczywistym analizować obrazu w celu unikania kolizji.

Praca miała trzy główne rozdziały: teoretyczne rozważania, metody oraz eksperymenty.

1. W teoretycznych rozważaniach nakreślono szerszy kontekst pracy, jego praktyczne wykorzystanie oraz potrzebną teorię na rzecz dalszego rozdziału.
2. W rozdziale metodycznym przedstawiono wszystkie metody potrzebne do realizacji procesu widzenia maszynowego oraz metody unikania kolizji. Określono sposoby badania elementów wchodzących w skład procesu stereowizji oraz metody unikania kolizji.
3. W rozdziale eksperymentalnym zostały przedstawione eksperymenty oraz wyniki dla procesu sterowizyjnego i metody unikania kolizji. Dla kalibracji kamer i kalibracji układu stereo, wchodzących w skład procesu stereowizyjnego, zostały zaprezentowane wyniki liczbowe. Dla badania metody generującej obraz macierzy rozbieżności zostały przedstawione uzyskane obrazy podczas modyfikowania parametrów generowania obrazów. Natomiast dla metody unikania kolizji jako wynik eksperymentu wykorzystano zdjęcie początkowe pojazdu z naniesioną na nie drogą, którą pokonał w czasie manewrowania między przeszkodami.

Zastosowane metody badawcze polegały na empirycznym weryfikowaniu wyników poszczególnych etapów. Wizualna weryfikacja okazała się interesującym narzędziem oceny. Dla obrazu głębi narzędzie wygenerowało dwuwymiarowy obraz, który przedstawiał obszary jaśniejsze oraz ciemniejsze odpowiednio dla bliżej, oraz dalej położonych przedmiotów względem kamer. Dla metody generowania kierunku zastosowano trójstopniową weryfikację:

- weryfikowanie decyzji dla pary obrazów,
- weryfikowanie decyzji dla manualnie zmienianego otoczenia przed kamerami,
- weryfikowanie decyzji dla samodzielnie poruszającego się prototypu.

W wyniku badań przedstawionych w niniejszej pracy pokazano, iż stereowizja jest skutecznym sposobem na ocenę otoczenia i uzyskania informacji na temat otaczających przeszkode. Odnosząc się do analizy obrazu głębi na tani komputerze pokładowym, badania wykazały, że jest to możliwe, ale nawet przy zastosowaniu prostej metody lokalnego dopasowania blokowego, działanie algorytmu było powolne. Dodatkowo realizacja prototypu uświadomiła autora o ograniczeniach sprzętowych kamer, które zwiększyły czas uzyskiwania zdjęć dla metody unikania kolizji.

Autor pracy zrealizował wszystkie założenia, które były postawione na początku związane z eksploracją i użyciem stereowizji w prototypie pojazdu autonomicznego. Nie było obiektywnej przyczyny, która uniemożliwiłaby dokonanie któregokolwiek z zamierzonych założeń.

Na podstawie przeprowadzonych badań, zostały wysunięte następujące wnioski:

- Stereowizja jest skutecznym sposobem na dostarczenie informacji potrzebnych do detekcji przeszkode stojących na drodze prototypu pojazdu autonomicznego.
- Przedstawiona metoda unikania kolizji jest efektywna w detekcji przeszkode z obrazu głębi i ustalania alternatywnego kierunku pojazdu.
- Tani komputer pokładowy jest w stanie generować obraz głębi oraz generować decyzję o kierunku pojazdu, aczkolwiek jest to proces powolny.
- Dla rozdzielczości 640x480px optymalne ustawienie liczby rozbieżności to 144 px, które powoduje generowanie obrazu głębi na dystansie powyżej 33 cm od układu kamer stereo.

Dalsze rozważania można przeprowadzić na wielu płaszczyznach:

- Dopracowanie algorytmu dopasowującego pary obrazów, aby uzyskać dokładniejszy obraz głębi.
- Segmentacja obrazu głębi, aby uzyskać dokładniejsze informacje o przeszkode.
- Zaprzegnięcie uczenia maszynowego do generowania macierzy rozbieżności.
- Optymalizacja algorytmu podejmującego decyzję o kolejnym kroku.
- Rozbudowa pojazdu autonomicznego o dodatkowe sensory do uzupełnienia informacji na temat przeszkode.
- Optymalizacja programu realizującego unikanie kolizji pod komputer z procesorem architektury ARM.
- Rozszerzenie rozwiązania o jednoczesne lokalizowanie i mapowanie terenu, w celu zapamiętania przebytej drogi, zamiast polegania tylko na lokalnej informacji o przestrzeni.

Dalszy rozwój autonomii pojazdów przewidywany jest w kierunku lepszego zrozumienia otoczenia, aby jednostka decyzyjna samochodu była w stanie podejmować trafne decyzje. Aktualnie firma Waymo testuje na szeroką skalę autonomię pojazdów na drodze, korzystając tylko z czujników, bez polegania na uczeniu maszynowym, maksymalnie eksplotując i łącząc wcześniej wymienione sensory w sekcji 2. rozdziału 1., aby pojazd właściwie interpretował otoczenie. Ciekawym kierunkiem jest użycie głębokiego uczenia do tworzenia obrazu przestrzeni

wokół pojazdu, przewidywania zachowania się obiektów otaczających pojazd. Głębokie uczenie, działające podobnie do ludzkiego mózgu ma przewagę nad standardowymi rozwiązaniami komputerowymi w kontekście interpretacji scen. To jest bardzo istotny aspekt ze względu na interpretację np. nieczytelnych obszarów dróg i poboczy, nieprzewidywalne zachowanie innych uczestników ruchu. Ponadto, co najmniej przez najbliższą dekadę, nadal istotnym kierunkiem rozwoju zostanie dalsza eksploracja szerszego wspomagania kierowcy, aby go odciążyć od prowadzenia samochodu i tylko w newralgicznych sytuacjach, zapewnić płynne przejście z pasywnego trybu pasażera do aktywnego kierowania pojazdem.

Bibliografia

- [1] A. López, A. Imiya, T. Pajdla, J. Álvarez, *Computer Vision in Vehicle Technology: Land, Sea, and Air*, Wiley, **2017**.
- [2] F. Duchoň, P. Hubinský, J. Hanzel, A. Babinec, M. Tölgessy, *Procedia Engineering* **2012**, 48, Modelling of Mechanical and Mechatronics Systems, 105–114, DOI <https://doi.org/10.1016/j.proeng.2012.09.492>.
- [3] T. M. Gasser, C. Arzt, M. Ayoubi, A. Bartels, L. Bürkle, J. Eier, F. Flemisch, D. Häcker, T. Hesse, W. Huber i in., *Bundesanstalt für Straßenwesen* **2013**.
- [4] L. Fridman, MIT Self-Driving Cars (2018), Youtube, **2018**, https://www.youtube.com/watch?v=_OCjqIgxwHw (term. wiz. 28.01.2018).
- [5] S. Arnoud, Sacha Arnoud, Director of Engineering, Waymo - MIT Self-Driving Cars, Youtube, **2018**, <https://www.youtube.com/watch?v=LSX3qdy0dFg> (term. wiz. 28.01.2018).
- [6] G. Rudolph, U. Voelzke, Three Sensor Types Drive Autonomous Vehicles, Questex LLC., **2019**, <https://www.sensorsmag.com/components/three-sensor-types-drive-autonomous-vehicles> (term. wiz. 28.01.2019).
- [7] A. Carullo, M. Parvis, *IEEE Sensors Journal* **2001**, 1, 143–, DOI [10.1109/JSEN.2001.936931](https://doi.org/10.1109/JSEN.2001.936931).
- [8] M. Barnard, Tesla & Google Disagree About LIDAR – Which Is Right?, Sustainable Enterprises Media, Inc., **2018**, <https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right> (term. wiz. 28.01.2019).
- [9] A. Neal, LiDAR vs. RADAR, Questex LLC. All rights reserved, **2019**, <https://www.sensorsmag.com/components/lidar-vs-radar> (term. wiz. 28.01.2019).
- [10] D. Rzeszotarski, P. Strumiłło, P. Pełczyński, B. Więcek, A. Lorenc, *Elektronika : prace naukowe* **2005**, nr 10, 165–184.
- [11] C. Chen, Y. F. Zheng, *Industrial Electronics IEEE Transactions on* **1995**, 42, 300–306, DOI [10.1109/41.382141](https://doi.org/10.1109/41.382141).
- [12] M. Brown, D. Burschka, G. Hager, *Pattern Analysis and Machine Intelligence IEEE Transactions on* **2003**, 25, 993–1008, DOI [10.1109/TPAMI.2003.1217603](https://doi.org/10.1109/TPAMI.2003.1217603).
- [13] B. Jähne, H. Haussecker, P. Geissler, *Handbook of Computer Vision and Applications: Sensors and imaging*, Academic Press, **1999**.
- [14] A. Kaehler, G. Bradski, *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*, O'Reilly Media, **2016**.
- [15] D. Rzeszotarski, P. Sawicki, B. Ostrowski, *Archiwum Fotogrametrii Kartografii i Tele-detekcji* **2007**, 17, 717–727.
- [16] K. Dawson-Howe, *A practical introduction to computer vision with opencv*, John Wiley & Sons, **2014**.
- [17] R. J. Radke, *Computer Vision for Visual Effects*, Cambridge University Press, **2012**, DOI [10.1017/CBO9781139019682](https://doi.org/10.1017/CBO9781139019682).

- [18] Z. Zhang, *IEEE Transactions on pattern analysis and machine intelligence* **2000**, *22*.
- [19] J. Wang, F. Shi, J. Zhang, Y. Liu, *Pattern Recognition* **2008**, *41*, 607–615.
- [20] R. Szeliski, *Computer vision: algorithms and applications*, Springer Science & Business Media, **2010**.
- [21] J. P. de Villiers, F. W. Leuschner, R. Geldenhuys w, **2008**.
- [22] B. Cyganek, J. Siebert, *An Introduction to 3D Computer Vision Techniques and Algorithms*, Wiley, **2011**.
- [23] B. Cyganek, *Przegląd Spawalnictwa-Welding Technology Review* **2013**, *85*.
- [24] R. I. Hartley, *International Journal of Computer Vision* **1999**, *35*, 115–127.
- [25] J. Park, J. S. Choi w *Recent Advances in Mobile Robotics*, InTech, **2011**.
- [26] I. Kostavelis, L. Nalpantidis, A. Gasteratos w IARP/EURON International Workshop on Robotics for risky interventions and Environmental Surveillance-Maintenance, Sheffield, UK, **2010**.
- [27] A. H. A. Hasan, R. A. Hamzah, M. H. Johar w Computer Technology and Development, 2009. ICCTD'09. International Conference on, *t. 1*, IEEE, **2009**, s. 98–102.
- [28] T. Pire, P. de Cristóforis, M. Nitsche, J. J. Berlles, *IEEE VI RAS Summer School on “Robot Vision and Applications” Santiago Chile* **2012**.
- [29] Baron-4WD Arduino Mobile Robot Platform with Encoder, **2019**, <https://www.dfrobot.com/product-261.html> (term. wiz. 28.01.2019).
- [30] Romeo BLE - Arduino Robot Control Board with Bluetooth 4.0, **2019**, <https://www.dfrobot.com/product-1176.html> (term. wiz. 28.01.2019).
- [31] Raspberry Pi 3 Model B+, **2019**, <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf> (term. wiz. 28.01.2019).
- [32] HD Pro Webcam C920 - Logitech Support, **2019**, https://support.logitech.com/en_us/product/hd-pro-webcam-c920/specs (term. wiz. 28.01.2019).
- [33] XPS 15 9560 High Performance Laptop with InfinityEdge Display, **2019**, <https://www.dell.com/en-us/shop/dell-laptops/xps-15/spd/xps-15-9560-laptop> (term. wiz. 28.01.2019).

Spis rysunków

1.1	Cechy kamery (żółty), radaru (pomarańczowy) i czujnika ultradźwiękowego (szary) ocenione w skali 0 - 5 [8].	7
1.2	Cechy lidaru ocenione w skali 0 - 5 [8].	8
1.3	Przekształcona płaszczyzna modelu kamery otworkowej [14].	9
1.4	Znieksztalcenia obrazu [20].	11
1.5	Przykładowy system stereo w postaci kanonicznej [22].	12
1.6	Relacja dwóch płaszczyzn obrazujących za pomocą macierzy rotacji R oraz wektora translacji T [14].	13
1.7	Układ geometrii epipolarnej [22].	14
1.8	Płaszczyzny obrazowania przed i po rektyfikacji [23].	16
1.9	Odwzorowanie odwrotne na przykładzie punktu znajdującego się w prawym zrektyfikowanym obrazie [14].	17
1.10	Fizyczna odległość Z punktu P [14].	18
1.11	Klasyfikacja trudności przeszkód ze względu na podział sztywne - niesztywne przeszkody [25].	19
1.12	Podział obrazu głębi na trzy okna w celu oceny możliwego kierunku [26].	19
1.13	Ograniczony obszar zainteresowania wyznaczony przez wnętrze między zielonymi poziomymi liniami [27].	20
1.14	Użycie wzniesienia do identyfikacji punktów podłogi oraz punktów zidentyfikowanych jako punkty przeszkód [28].	20
2.1	3W zarejestrowana przestrzeń przed pojazdem (na niebiesko) wraz z jego torem (na zielono).	30
2.2	3W przestrzeń przed pojazdem zawierająca przeszkodę na torze pojazdu (przeszkoda na czerwono).	30
2.3	Wyznaczony tor alternatywny w celu ominięcia przeszkody.	31
2.4	3W przestrzeń zawierająca szeroką przeszkodę typu ściana (przeszkoda na czerwono).	31
2.5	Rzut z góry przestrzeni z przeszkodą w torze głównym przed pojazdem.	33
2.6	Wyznaczenie kąta θ skrętu pojazdu.	33
3.1	Wysokopoziomowy schemat komponentów pojazdu autonomicznego.	35
3.2	Wzór szachownicy kalibrującej użytej do kalibracji kamer [14].	36
3.3	Wizualizacja punktów przecięć znalezionych funkcją <i>findChessboardCornersSB</i>	37
3.4	Zbliżenie na błędnie określony punkt przecięcia przez procedurę <i>cornerSubPix</i>	37
3.5	Błąd reprojekcji punktów w zależności od ilości zastosowanych zdjęć dla metody z określonym punktem odniesienia w rozdzielcości 640x360px.	38
3.6	Błąd reprojekcji punktów w zależności od ilości zastosowanych par zdjęć w rozdzielcości 640x360px.	40
3.7	Para zdjęć szachownicy kalibrującej przed rektyfikacją.	42
3.8	Para zdjęć szachownicy kalibrującej po rektyfikacji.	43

3.9	Para obrazów wykorzystanych do eksperymentów z metodami generowania obrazu rozbieżności.	43
3.10	Wizualizacja macierzy rozbieżności dla metod BM oraz SGBM.	44
3.11	Wpływ szerokości dopasowywania na wynikową macierz rozbieżności.	45
3.12	Wpływ minimalnej rozbieżności na wynikową macierz rozbieżności.	45
3.13	Wpływ rozmiaru bloku na uzyskaną macierz rozbieżności.	45
3.14	Wpływ typu filtrowania wstępного par obrazów na wynikową macierz rozbieżności.	45
3.15	Wpływ wielkości okna plamki na wynikową macierz rozbieżności przy dopuszczalnej zerowej różnicy między rozbieżnościami w obrębie okna plamki.	46
3.16	Wpływ parametru określającego dopuszczalne różnice rozbieżności między kolejnymi pikselami w obrębie plamki o szerokości 1px na wynikową macierz rozbieżności.	46
3.17	Przykładowa para obrazów oraz obraz rozbieżności wykorzystany do weryfikacji odległości przedmiotu od kamery.	47
3.18	Ograniczenia nанiesione na obraz głębi wykorzystane w module decyzyjnym.	49
3.19	Zdjęcie z góry toru pojazdu zawierającego przeszkodę o szerokości 23 cm znajdująca się 50 cm przed pojazdem.	50
3.20	Zdjęcie z góry toru pojazdu zawierającego przeszkodę o szerokości 5 cm znajdująca się 40 cm przed pojazdem.	50
3.21	Zdjęcie z góry toru pojazdu zawierającego przeszkodę o szerokości 1.6 cm znajdująca się 43 cm przed pojazdem.	51
3.22	Zdjęcie z góry toru pojazdu, z ulokowanym oknem umożliwiającym przejazd między przeszkodami.	51
3.23	Wizualizacja obrazu głębi ograniczonej z góry i dolu dla sytuacji, gdy pojazd znajduje się 45 cm przed ścianą z 5-stopniowym skrętem w lewo oraz prawo.	51
3.24	Sytuacja, w której szersza przeszkoda jest umieszczona z prawej strony pojazdu.	53
3.25	Sytuacja, w której szersza przeszkoda jest umieszczona z lewej strony pojazdu.	53
3.26	Sytuacja, w której wąska przeszkoda jest umieszczona z prawej strony pojazdu.	54
3.27	Sytuacja, w której wąska przeszkoda jest umieszczona z lewej strony pojazdu.	54
3.28	Autonomiczny tor ruchu pojazdu dla pierwszego przypadku przeszkód.	55
3.29	Autonomiczny tor ruchu pojazdu dla drugiego przypadku przeszkód.	56
A.1	Interfejs programu do oceny rozbieżności na podstawie wybranych parametrów.	68

Spis tabel

3.1	Suma odległości punktów pomiędzy poszczególnymi metodami.	37
3.2	Błąd reprojekcji dla zadanej metody oraz rozdzielczości.	38
3.3	Obliczone długości rzutowania f_x, f_y dla zadanej metody oraz rozdzielczości.	38
3.4	Środkowa płaszczyzny obrazowania (c_x, c_y) dla zadanej metody oraz rozdzielczości.	39
3.5	Współczynniki zniekształceń radialnych k_1, k_2 dla zadanej metody oraz rozdzielczości.	39
3.6	Współczynniki zniekształceń tangensowych p_1, p_2 dla zadanej metody oraz rozdzielczości.	39
3.7	Błąd reprojekcji stereo dla zastosowanej metody oraz rozdzielczości.	39
3.8	Macierz rotacji kamer, jej odpowiedniki w stopniach dla osi X, Y, Z oraz wektory translacji dla trzech rozdzielczości.	40
3.9	Porównanie ilości wygenerowanych obrazów rozbieżności w ciągu jednej minuty dla zadanej rozdzielczości oraz metody na komputerze Dell XPS [33] oraz Raspberry Pi.	44
3.10	Różnice między faktycznymi odległościami przedmiotu od kamer a dystansem wyznaczonym w obrazie głębi.	47
3.11	Kąty skrętu dla przeszkody o szerokości 23 cm znajdującej się 50 cm przed pojazdem, przedstawionej w rys. 3.19.	50
3.12	Kąty skrętu dla przeszkody o szerokości 5 cm znajdującej się 40 cm przed pojazdem, przedstawionej w rys. 3.20.	50
3.13	Kąty skrętu dla przeszkody o szerokości 1.6 cm znajdującej się 43 cm przed pojazdem, przedstawionej w rys. 3.21.	51
3.14	Kąty skrętu dla okna ulokowanego między przeszkodami, przedstawionego w rys. 3.22.	51
3.15	Decyzje o dalszej drodze pojazdu dla przeszkody o dużej szerokości względem pojazdu na postawie obrazów macierzy rozbieżności przedstawionych w rys. 3.23.	52
3.16	Czas generowania decyzji na komputerze Dell XPS [33] oraz Raspberry Pi.	52

Spis załączników

Program do generowania obrazu rozbieżności	68
--	----

Załączniki

Załącznik A: Program do generowania obrazu rozbieżności



Rysunek A.1: Interfejs programu do oceny rozbieżności na podstawie wybranych parametrów.

W powyższym interfejsie programu, jest widoczna możliwość ustawienia parametrów danej metody wybranej w "Metoda stereo", która dla wartości "0" jest BM, natomiast dla wartości "1" jest "SGBM". Podczas przesuwania poszczególnych pasków, w czasie rzeczywistym jest generowany obraz rozbieżności, który jest przedstawiony w górnej połowie interfejsu programu.