

**Michał Nowicki**  
**Politechnika Poznańska**  
**Wydział Automatyki, Robotyki i Elektrotechniki**

**Załącznik 2**

**Autoreferat przedstawiający opis dorobku i osiągnięć naukowych**

2023

1. **Imię i nazwisko**

Michał Nowicki

2. **Posiadane dyplomy, stopnie naukowe z podaniem nazwy, miejsca i roku ich uzyskania oraz tytuł rozprawy doktorskiej**

2018 r. doktor nauk technicznych

**Dyscyplina naukowa:** automatyka i robotyka

Politechnika Poznańska

Wydział Elektryczny

**Tytuł rozprawy doktorskiej:** Methods for the fusion of quantitative and qualitative information using factor graph optimization for the simultaneous localization and mapping problem

**Promotor:** dr hab. inż. Piotr Skrzypczyński, prof. PP

**Recenzenci:** prof. dr hab. inż. Dariusz Uciński (Uniwersytet Zielonogórski),  
dr hab. inż. Bogdan Kwolek, prof. AGH (Akademii Górniczo-Hutniczej im. Stanisława Staszica)

2014 r. magister

**Kierunek:** automatyka i robotyka

**Specjalność:** robotyka

Politechnika Poznańska

Wydział Elektryczny

**Tytuł pracy dyplomowej:** Development and implementation of a visual odometry algorithm for mobile devices

**Promotor:** dr hab. inż. Piotr Skrzypczyński, prof. PP

2014 r. inżynier

**Kierunek:** informatyka

Politechnika Poznańska

Wydział Informatyki

**Tytuł pracy dyplomowej:** Gesture recognition library for Leap Motion controller

**Promotor:** dr inż. Wojciech Jaśkowski

2013 r. inżynier

**Kierunek:** automatyka i robotyka

**Specjalność:** robotyka

Politechnika Poznańska

Wydział Elektryczny

**Tytuł pracy dyplomowej:** Metody wykrywania i opisu punktów kluczowych na obrazach RGB-D

**Promotor:** dr hab. inż. Piotr Skrzypczyński, prof. PP

### 3. Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych

- Instytut Robotyki i Inteligencji Maszynowej, Politechnika Poznańska, adiunkt od 01.10.2018 r.
- Instytut Automatyki, Robotyki i Inżynierii Informatycznej, Politechnika Poznańska, asystent od 01.10.2014 r. do 30.09.2018 r.

### 4. Omówienie osiągnięć, o których mowa w art. 219 ust. 1 pkt. 2 ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2021 r. poz. 478 z późn. zm.):

#### a) Tytuł osiągnięcia naukowego

Cykl publikacji powiązanych tematycznie pod wspólną nazwą:

#### **Wielosensoryczna lokalizacja robotów mobilnych**

Osiągnięcie habilitacyjne stanowi cykl 8 publikacji przedstawionych poniżej.

#### b) Publikacje lub inne prace wchodzące w skład osiągnięcia naukowego

Indywidualny wkład do każdej publikacji przedstawiony jest w dokumencie zawierającym wykaz osiągnięć naukowych lub artystycznych oraz w indywidualnych oświadczeniach współautorów.

##### [A1] **M.R. Nowicki (100%),**

Spatiotemporal Calibration of Camera and 3D Laser Scanner, IEEE Robotics and Automation Letters, 5(4), s. 6451–6458, 2020

doi: 10.1109/LRA.2020.3014639.

(**IF<sub>2020</sub>: 3.741, 200 pkt MEiN<sup>1</sup>**)

##### [A2] **T. Nowak (60%), M.R. Nowicki (30%), P. Skrzypczyński (10%),**

Vision-based positioning of electric buses for assisted docking to charging stations, International Journal of Applied Mathematics and Computer Science (AMCS), 32(4), s. 583–599, 2022

doi: 10.34768/amcs-2022-0041.

(**IF<sub>2021</sub>: 2.157, 100 pkt MEiN**)

---

<sup>1</sup>Wartość Impact Factor (IF) podana zgodnie z rokiem publikacji lub wg IF dla roku 2021 w przypadku publikacji z lat 2022-2023 (dla tych lat IF jest jeszcze niedostępny) – źródło: Journal Citation Report. Punkty MEiN podano zgodnie z rokiem publikacji.

- [A3] **M.R. Nowicki (100%)**,  
 A data-driven and application-aware approach to sensory system calibration in an autonomous vehicle, *Measurement*, 194, 2022,  
 doi: 10.1016/j.measurement.2022.111002.  
**(IF<sub>2021</sub>: 5.131, 200 pkt MEiN)**
- [A4] **K. Żywanowski (20%), A. Banaszczyk (20%), M.R. Nowicki (60%)**,  
 Comparison of camera-based and 3D LiDAR-based place recognition across weather conditions, 2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV), Shenzhen, China, 2020, s. 886–891,  
 doi: 10.1109/ICARCV50220.2020.9305429.  
**(140 pkt MEiN, CORE A)**
- [A5] **K. Żywanowski (20%), A. Banaszczyk (20%), M.R. Nowicki (55%), J. Komorowski (5%)**,  
 MinkLoc3D-SI: 3D LiDAR Place Recognition With Sparse Convolutions, Spherical Coordinates, and Intensity, *IEEE Robotics and Automation Letters*, 7(2), s. 1079–1086, 2022,  
 doi: 10.1109/LRA.2021.3136863.  
**(IF<sub>2021</sub>: 4.321, 200 pkt MEiN)**
- [A6] **M. Bednarek (34%), M.R. Nowicki (33%), K. Walas (33%)**,  
 HAPTR2: Improved Haptic Transformer for legged robots' terrain classification, *Robotics and Autonomous Systems*, 158, 2022, ISSN 0921-8890,  
 doi: 10.1016/j.robot.2022.104236.  
**(IF<sub>2021</sub>: 3.7, 140 pkt MEiN)**
- [A7] **M. Łysakowski (30%), M.R. Nowicki (35%), R. Buchanan (10%), M. Camurri (10%), M. Fallon (10%), K. Walas (5%)**,  
 Unsupervised Learning of Terrain Representations for Haptic Monte Carlo Localization, 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, Maj 23-27, 2022, s. 4642–4648, 2022,  
 doi: 10.1109/ICRA46639.2022.9812296.  
**(70 pts MEiN, CORE B)**
- [A8] **D. Sójka (30%), M.R. Nowicki (60%), P. Skrzypczyński (10%)**,  
 Learning an Efficient Terrain Representation for Haptic Localization of a Legged Robot, 2023 International Conference on Robotics and Automation (ICRA), London, UK, 29 Maj-2 Czerwiec 2023.  
**(70 pkt MEiN, CORE B)**

Seria składa się z 8 publikacji, wśród których 5 zostało opublikowanych w czasopiśmie indeksowanych w bazie Journal Citation Reports (JCR). Dwie publikacje w zbiorze są pracami autorskimi. Sumaryczny współczynnik Impact Factor publikacji wchodzących w skład osiągnięcia wynosi 19,05. Sumaryczna liczba punktów według wykazu czasopism naukowych opracowanego przez MEiN dla publikacji w cyklu wynosi 1120. Średni procentowy wkład M. Nowickiego w



publikacje wynosi 59,125%.

### c) Omówienie osiągnięć naukowych:

#### 4.1 Motywacja i cel naukowy podjętych badań

Autonomia robotów mobilnych to złożony problem, który wymaga niezawodnego rozwiązania niezależnie od założonego scenariusza. Wyzwanie wynika z tego, że środowiska pracy, roboty, zadania czy dostępna moc obliczeniowa znacznie się różnią w zależności od aplikacji, co wymaga różnych sensorów i algorytmów, aby osiągnąć pożądany cel. Ponieważ problem jest złożony, zwykle dzieli się go na mniejsze zadania związane z percepcją, lokalizacją, planowaniem i sterowaniem, które są rozwiązywane niezależnie, a następnie łączone w kompletny rezultat autonomii. W moich badaniach skupiam się na aspektach percepcji i lokalizacji w autonomii kluczowych dla zrozumienia otaczającego środowiska, co jest podstawą do realizacji dalszych zadań związanych z planowaniem i sterowaniem. Istniejące rozwiązania w tym obszarze opierają się na wielu współpracujących czujnikach, aby zapewnić docelową niezawodność.

Prezentowane osiągnięcie naukowe dotyczy problemu lokalizacji z wykorzystaniem układów wielosensorycznych, począwszy od kalibracji sensorów, aż po sposoby łączenia informacji w celu poprawy wydajności w stosunku do systemów korzystających z pojedynczych sensorów. Mój obszar badawczy określiłem na podstawie ograniczeń zaobserwowanych podczas realizacji pracy doktorskiej. Dotyczyła ona metod fuzji informacji ilościowych i jakościowych z wykorzystaniem optymalizacji grafu ograniczeń dla problemu jednoczesnej lokalizacji i mapowania (ang. *Simultaneous Localization and Mapping* - SLAM).

Z perspektywy praktyki, rozwiązania SLAM są złożonymi systemami, których wynik zależy od historii poprzednich pomiarów. Systemy SLAM oferują dużą dokładność w przypadku skutecznego działania, ale kilka błędnych pomiarów może doprowadzić do problemów z oszacowaniem położenia i mapy w danej sesji pracy. Ta specyfika wynika ze względu na wykorzystanie historii pomiarów do uzyskania najdokładniejszej lokalizacji w systemach SLAM. Ponadto wiele aplikacji robotów mobilnych nie potrzebuje systemów SLAM, ponieważ pewne wcześniejsze informacje (np. mapy) są już dostępne i mogą być wykorzystane do zapewnienia skutecznego rozwiązania lokalizacyjnego. Dlatego w serii prezentowanych artykułów skupiam się na ogólniejszych problemach lokalizacyjnych oraz powiązanych z nimi problemach percepcyjnych, które mogą być rozwiązane za pomocą konfiguracji wielosensorycznej. Artykuły prezentują zastosowanie proponowanych algorytmów w praktycznych scenariuszach, ale te metody mogą zostać zaadaptowane w podobnych aplikacjach, torując drogę do stworzenia rozwiązań lokalizacyjnych dla robotów bez wyraźnej potrzeby stosowania pokładowych systemów SLAM.

Aby czerpać korzyści z układów wielosensorycznych, trzeba pokonać trudności w ich użyciu. Jednym z ograniczeń systemów wielosensorycznych jest potrzeba kalibracji, czyli określenia odpowiedniości pomiędzy pomiarami z sensorów będących częścią systemu. Bez kalibracji otrzymujemy pomiary z sensorów, które nie mogą być połączone w celu stworzenia jednolitego obrazu otaczającego środowiska. Posiadanie skalibrowanych przestrzennie i czasowo

zestawów sensorycznych okazuje się problemem naukowym w momencie, gdy oczekujemy dokładności określonej w milimetrach oraz milisekundach. W momencie podjęcia tego wyzwania, żadne gotowe rozwiązania kalibracji kamera-skaner laserowy (ang. *Light Detection and Ranging* - LiDAR) 3D nie oferowało wymaganej dokładności. Istniejący stan wiedzy w zakresie systemów kalibracyjnych skupiał się na LiDAR-ach 2D, oferując jedynie określenie relacji przestrzennych pomiędzy sensorami, ignorując dodatkowe przesunięcia znaczników czasowych pomiędzy sensorami, które występują w przypadku braku wyzwalaczy sprzętowych. Aby rozwiązać ten problem, stworzyłem algorytm umożliwiający wyznaczenie dokładnych relacji przestrzennych i czasowych pomiędzy tymi sensorami. Rozwiązanie to może być wykorzystane w klasycznych, geometrycznych, wielosensorycznych systemach SLAM.

Wymóg dokładnej kalibracji geometrycznej okazał się krytyczny w problemie lokalizacji w odniesieniu do małego obiektu z dużej odległości. Mój scenariusz testowy dotyczył autobusu elektrycznego dokującego do ładowarki elektrycznej z odległości ponad 30 metrów. Dzięki testom na autobusie zaobserwowaliśmy, że tylko rozwiązanie oparte na kamerze było w stanie wykryć ładowarkę elektryczną, ponieważ był to mały obiekt. Problemem było zapewnienie pożądanej dokładności lokalizacji, aby umożliwić planowanie i sterowanie długim, przegubowym autobusem, ponieważ małe niedokładności ustawień sensorów prowadziły do błędów, który był nie do przyjęcia dla algorytmów planowania ruchu. Rozwiązaniem okazała się kalibracja kompletnego zestawu sensorów w docelowej aplikacji na autobusie, prezentując nowe podejście do kalibracji oparte na wydajności w aplikacji końcowej względem typowych podejść opartych na błędzie kalibracji.

Doświadczenia z kalibracją wielosensoryczną sprawiły, że jednocześnie szukałem również sposobów, które mogłyby doprowadzić do powstania systemów wielosensorycznych bez żmudnej kalibracji. Połączenie sensorów kamera i LiDAR 3D postanowiłem wykorzystać w globalnym problemie rozpoznawania miejsc, gdzie LiDAR 3D wspomaga dodatkowymi informacjami geometrycznymi typowe rozwiązanie oparte na kamerze. Założyłem, że systemy oparte na kamerach charakteryzują się zmniejszoną wydajnością w trudnych warunkach pogodowych, takich jak deszcz, noc lub bezpośrednio światło słoneczne. Z drugiej strony, LiDAR 3D dostarcza danych uzupełniających, które mogą poprawić wydajność systemu globalnej lokalizacji. Analiza tego pomysłu doprowadziła do nowego problemu badawczego: jak powinniśmy reprezentować chmury punktów 3D uzyskane z LiDAR-ów 3D dla głębokich sieci neuronowych?

Problem reprezentacji danych jest kluczowy podczas pracy z sieciami neuronowymi i szczególnie trudny, gdy mamy do czynienia z danymi nieobrazowymi. Niektóre sensory dostarczają stosunkowo mało informacji, które jednak wciąż mogą być istotne jako pewne wskazówki lokalizacyjne krytyczne dla autonomicznej lub półautonomicznej eksploracji z wykorzystaniem robotów mobilnych. Znaczenie tych podejść wzrasta, gdy założymy, że nawet algorytmy oparte na kamerach i LiDAR-ach 3D mogą zawieść w przypadku trudnych warunków środowiskowych, gdzie od autonomicznego robota mobilnego wymagamy nadal skutecznej lokalizacji, aby zapewnić bezpieczne działanie. W swojej pracy naukowej

spotkałem się z tym problemem w przypadku robota kroczącego wyposażonego w czujniki siły i momentu oferujące dane komplementarne, ale dające relatywnie mało informacji do lokalizacji na znanej mapie terenu.

Na podstawie przedstawionych wyzwań cele szczegółowe związane z przedstawionym cyklem publikacji można zdefiniować w następujący sposób:

- Projekt i implementacja dokładnego oprogramowania do kalibracji kamer RGB oraz nowoczesnych obrotowych LiDAR-ów 3D. Oprogramowanie powinno umożliwiać wyznaczenie trójwymiarowej relacji geometrycznej pomiędzy sensorami oraz przesunięcia znacznika czasu w przypadku braku wyzwalań sprzętowego. Dostępność tych informacji jest kluczowa dla możliwości określenia odpowiedniości pomiędzy pomiarami z obu sensorów, które mogą być wykorzystane w nowoczesnych konfiguracjach wielosensorycznych.
- Projekt i implementacja rozwiązania kalibracyjnego, które skupia się na poprawie wydajności zestawu sensorycznego w docelowej aplikacji, zamiast na znalezieniu najdokładniejszej transformacji geometrycznej. Ideą jest pokazanie możliwości zastosowania takiego podejścia w scenariuszu percepcji robota mobilnego podczas rozwiązywania docelowego problemu.
- Projekt i implementacja systemu rozpoznawania miejsc, który może skutecznie określić lokalizację na podstawie obrazów RGB i chmur punktów z LiDAR-ów 3D. Zakładam, że połączenie informacji z obu źródeł może prowadzić do dokładniejszego rozwiązania w porównaniu z wykorzystaniem jednego sensora. Aby rozwiązać ten problem, wymagane jest określenie najbardziej odpowiedniej reprezentacji chmury punktów 3D dla metod głębokiego uczenia.
- Projekt i implementacja rozwiązania do lokalizacji z wykorzystaniem skąpych, ale użytecznych danych sensorycznych, gdy kamera i LiDAR 3D są niedostępne. W tym przykładzie rozważane są roboty kroczące, które mogą postrzegać środowisko za pomocą sygnałów haptycznych. Zakładam, że wcześniejsze doświadczenia podczas przejść przez dany teren mogą posłużyć do lokalizacji robota podczas ponownego pojawienia się na danym obszarze.

Założone cele zostały osiągnięte dzięki zastosowaniu zestawu następujących narzędzi:

- narzędzie do optymalizacji oparte na grafach ograniczeń  $g2o^2$ ,
- rzadkie konwolucje 3D zaimplementowane w *Minkowski engine*<sup>3</sup>,
- błąd trójkowy (ang. *triplet loss*) z wyszukiwaniem trudnych przykładów do nauki sieci neuronowych,
- algorytmy wykrywania obiektów 2D takie jak HRNet oraz inne zaimplementowane w bibliotece *MMPose*<sup>4</sup> library,
- architektura Transformera [70],

---

<sup>2</sup><https://github.com/RainerKuemmerle/g2o>

<sup>3</sup><https://github.com/NVIDIA/MinkowskiEngine>

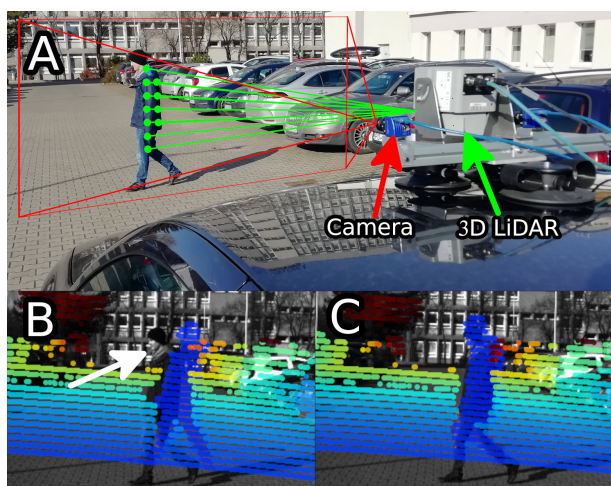
<sup>4</sup><https://github.com/open-mmlab/mmpose>

- lokalizacja cząsteczkowa Monte Carlo.

Cykl publikacji przedstawionych w osiągnięciu naukowym pokazuje drogę do wspomnianych rozwiązań z wykorzystaniem wymienionych narzędzi. Wszystkie wykorzystane dane liczbowe oraz rysunki pochodzą bezpośrednio z publikacji zawartych w osiągnięciu naukowym.

## 4.2 Wstęp

Obecnie układy sensoryczne robotów mobilnych składają się z coraz większej liczby kamer, LiDAR-ów, radarów oraz jednostek pomiaru orientacji (ang. *Inertial Measurement Unit* - IMU), które dostarczają komplementarnych informacji wykorzystywanych przez roboty autonomiczne. Aby wykorzystać różne sensory, konieczne jest przedstawienie pomiaru z jednego sensora w układzie współrzędnych drugiego sensora. W warunkach statycznych potrzebna jest jedynie transformacja przestrzenna. Ale gdy sensory są w ruchu (lub gdy zmienia się środowisko), konieczna jest znajomość zależności czasowych pomiędzy pomiarami z sensorów.



Rysunek 1: Wspólna obserwacja z zestawu sensorycznego kamera-LiDAR 3D zamontowanego na samochodzie (A) oraz punkty z LiDAR-u rzutowane na obraz w dynamicznym scenariuszu, gdy kalibracja przestrzenna (B) lub czasoprzestrzenna (C) została przeprowadzona przy użyciu proponowanych ram. Warto zwrócić uwagę, że brak kalibracji czasowej prowadzi do nieprawidłowej korespondencji pomiarów (biała strzałka).

Kalibracja przestrzenna może być określona za pomocą już istniejących programów do kalibracji lub założona na podstawie rysunków CAD. Kalibracja zależności czasowych jest niekiedy wykonywana za pomocą dodatkowego sprzętu, który realizuje jednoczesne wyzwolenie pobrania zdjęć z kamer i synchronizuje te dane ze znacznikami czasu dla pomiarów chmury punktów pozyskanej z LiDAR-u 3D. Ten dodatkowy sprzęt często nie jest dostępny, a niektóre tańsze wersje sensorów w ogóle nie oferują takiej możliwości. Wtedy

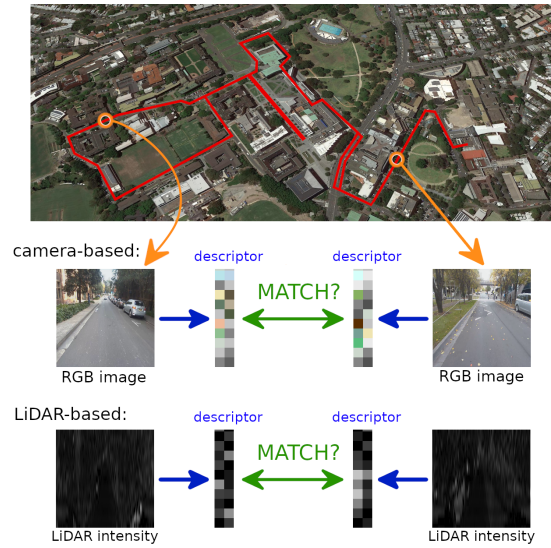
najczęściej pomija się problem kalibracji czasowej lub rozwiązuje go prostymi metodami nie gwarantującymi dokładności mierzonej jako kilka milisekund.

Przykładowy układ wielosensoryczny składający się z kamery i LiDAR-u 3D oraz wymagający kalibracji czasoprzestrzennej przedstawiono na rys. 1A. Rozwiązanie kalibracji przestrzennej daje możliwość określenia zgodności pomiarów z sensorów w warunkach statycznych, ale zawodzi w środowisku dynamicznym (rys. 1B), natomiast kalibracja czasoprzestrzenna daje oczekiwane rezultaty (rys. 1C). Kalibracja czasoprzestrzenna jest koniecznością dla popularnych obecnie, ściśle zintegrowanych rozwiązań odometrii wizyjnej (ang. *Visual Odometry* - VO) lub systemów jednoczesnej lokalizacji i budowy mapy SLAM, które wspólnie i jednocześnie przetwarzają dane z kamer i LiDAR-ów 3D [31].

Para kamera RGB-LiDAR 3D jest najpopularniejszym zagadnieniem badawczym kalibracji zewnętrznej poza typowymi konfiguracjami stereowizyjnymi, na co wskazuje liczba prac dotyczących ich wspólnego wykorzystania [15]. Prac dotyczących tematu kalibracji kamera-IMU jest znacznie mniej, gdzie najbardziej znane rozwiązanie zaimplementowane w *kalibr* [55, 56] opiera się na szybkich obrotach układu sensorycznego przed statycznym znacznikiem kalibracyjnym. Jedną z najnowszych prac [28] dodatkowo poprawia kalibrację kamera-IMU przedstawioną w *kalibr*, gdy IMU nie znajduje się blisko kamery poprzez wykorzystanie dodatkowych ograniczeń. Dla mniej popularnych par sensorów, jak RADAR-kamera czy LiDAR-RADAR, istnieje jedynie kilka prac naukowych [41, 60], a zaprojektowane procedury kalibracyjne są sformułowane w sposób podobny do wcześniej opisanych procedur dla innych par sensorów.

Problemem z każdą kalibracją systemu sensorycznego realizowanego pomiędzy dwoma sensorami jest brak gwarancji dokładności kalibracji kompletnego systemu wielosensorycznego składającego się z większej liczby sensorów, ponieważ błędy kalibracji mogą się kumulować, gdy poszczególne wyniki kalibracji są łączone. Dlatego też pomysł włączenia pomiarów z wszystkich sensorów do wspólnej kalibracji zyskuje na popularności, czego przykładem są podejścia do systemów LiDAR 3D-kamera-RADAR [18, 19, 53] albo LiDAR 2D-kamera-IMU jak w *kalibr* [54]. Jednakże podejście polegające na włączeniu wszystkich sensorów do tej samej procedury kalibracji wymaga wspólnego pola widzenia tych czujników. Taki wymóg ogranicza uniwersalny aspekt proponowanych rozwiązań, ponieważ niektóre konfiguracje sensorów nie mogą być skalibrowane przy użyciu tych podejść. Głównym ograniczeniem jest jednak fakt, że istniejące metody kalibracji skupiają się na znalezieniu najdokładniejszych parametrów, zaniedbując jednocześnie fakt, że ostatecznym celem kalibracji jest poprawa wydajności używanego systemu wielosensorycznego.

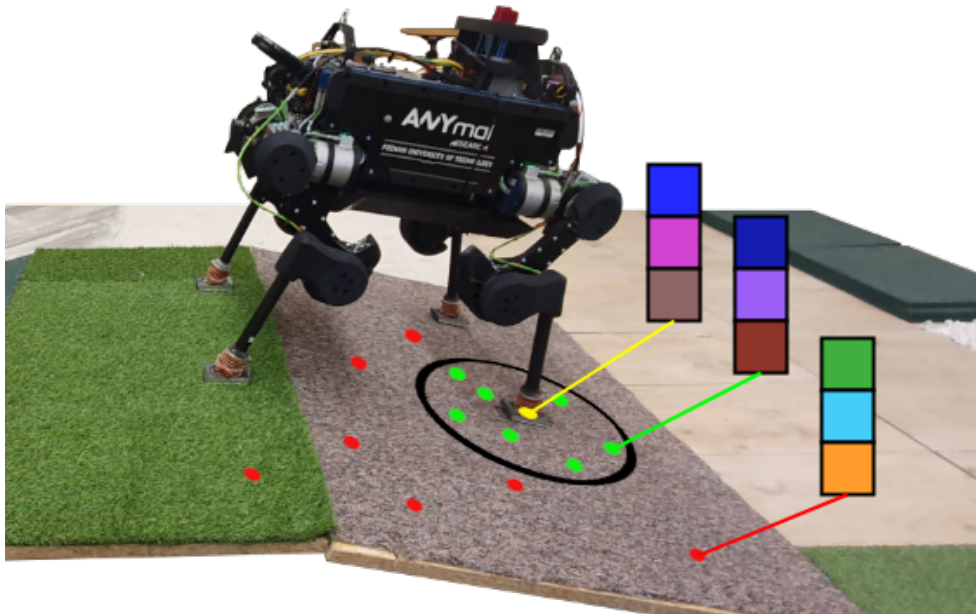
Na szczęście niektóre aplikacje lokalizacyjne nie wymagają dokładnej kalibracji. Jednym z takich zastosowań jest rozpoznawanie miejsc, które jest kluczowym elementem efektywnej lokalizacji robotów w długim czasie, ponieważ redukuje nagromadzony dryf, gdy robot powraca do już znanej lokalizacji [14]. Problem inaczej znany jako zamykanie pętli (ang. *Loop Closure*) może być rozwiązany z wykorzystaniem różnych danych, z których najczęściej używane są GPS i kamera. Systemy wykrywania zamknięcia pętli z użyciem kamer RGB są już stosowane w rzeczywistych scenariuszach [16, 24, 47]. Wydajność tych metod zależy



Rysunek 2: Porównanie rozpoznawania miejsc na podstawie deskryptorów uczonych na obrazach RGB (opartych na kamerze), wartościach natężenia powracającego sygnału dla LiDAR-u 3D oraz połączonych danych wejściowych z obrazów RGB i wartości natężenia sygnału w LiDAR-ze 3D (opartych na fuzji danych kamera-LiDAR 3D) w różnych warunkach pogodowych dla wielu przebiegów na tej samej trajektorii z wykorzystaniem zbioru danych USyD.

od jakości obrazu, która ulega pogorszeniu w nocy, w niekorzystnych warunkach pogodowych lub gdy słońce świeci bezpośrednio w obiektyw, oślepiając kamerę. Pozytywnym jest fakt, że większość samochodów autonomicznych jest wyposażona w inne sensory, takie jak LiDAR-y 3D, oferując wsparcie w tych wymagających scenariuszach. Chmury punktów z LiDAR-ów 3D dostarczają informacji o geometrii otoczenia robota, uzupełniając obrazy RGB z kamer, tworząc kompletny widok środowiska.

Zaprojektowanie efektywnego, opartego na uczeniu się systemu rozpoznawania miejsc na podstawie chmur punktów jest wciąż otwartym problemem. Kluczowym wyzwaniem jest znalezienie najlepszej reprezentacji danych 3D, która może być efektywnie przetwarzana za pomocą sieci neuronowych w celu wydobycia znaczących cech dla skutecznego rozpoznania miejsca. Istniejące prace badały już reprezentacje obrazowe, oparte na woksłach [77], oparte na widokach z lotu ptaka [25, 26, 40] lub reprezentacje w formie nieuporządkowanego zbioru punktów [68, 78]. Ostatnio obserwuje się wzrost liczby podejść opartych na rzadkich konwolucjach 3D [35, 38] i modułach uwagi [2], gdzie oba trendy są też łączone [73]. Wiele z tych metod jest uczonych, ocenianych i porównywanych na zbiorze danych Oxford RobotCar [21, 32, 35, 43, 68, 72, 74], który został zebrany przez połączenie wielu skanów 2D obejmujących odległość 20 m i próbkowanie do 4096 punktów [68]. Te dane nie odpowiadają charakterystyce rzeczywistych danych z LiDAR-ów 3D, gdzie pojedynczy skan obejmuje znacznie większy obszar (nawet do 160-200 metrów średnicy) i ma większą liczbę punktów (do 260k punktów dla Ouster OS1-128).



Rysunek 3: Lokalizacja haptyczna wymaga rozróżnialnej reprezentacji sygnałów z interakcji stopa/teren, aby móc rozróżnić lokalizacje. Zaproponowałem uczenie sieci neuronowej opartej na architekturze transformera z błędem trójkowym w celu zminimalizowania różnicy między reprezentacjami (ang. *embedding*) dla kroków blisko siebie i maksymalizowania tej różnicy dla kroków będących dalej od siebie.

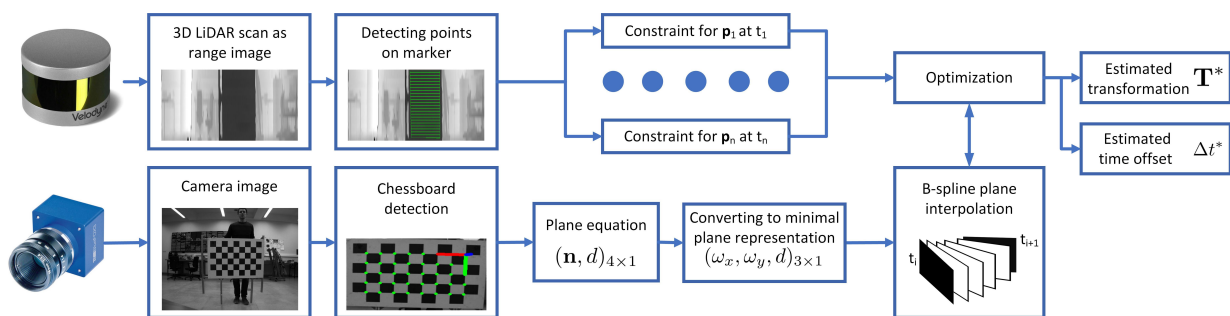
Algorytmy do zastosowań w robotyce mobilnej są zwykle projektowane z myślą o aplikacji, ponieważ scenariusze świata rzeczywistego nakładają dodatkowe ograniczenia na proponowane rozwiązania. Zwiększenie dojrzałości technologii mechaniki robotów kroczących sprawiło, że mogą one znaleźć zastosowanie w wielu aplikacjach przemysłowych. Stąd pojawiła się potrzeba projektowania algorytmów zwiększających ich skuteczną pracę. Zdolność do działania pomimo awarii czujników jest często kluczowym komponentem, zwłaszcza w przypadku długich rutynowych zadań w niebezpiecznych środowiskach, takich jak ciemne i zapyłone kopalnie. W pracy skupiłem się na autonomicznym robocie czteronożnym wykonującym zadania inspekcji (rys. 3), gdzie celem jest niezawodna lokalizacja przy użyciu tylko czujników proprioceptywnych (IMU, stany nóg, czujniki dotyku). Celem podejścia jest umożliwienie ukończenia zadania lub powrót do bezpiecznego miejsca, nawet gdy kamery czy LiDAR-y 3D są niedostępne.

W takim scenariuszu pomiary haptyczne, zwane także taktylnymi, mogą być wykorzystane do określenia klasy terenu [5, 30, 34], co z kolei może być wykorzystane do zlokalizowania nóg robota we wcześniej przygotowanej mapie [12]. Istniejące podejścia do lokalizacji z wykorzystaniem pomiarów haptycznych były jednak mało praktyczne, ponieważ wymagały gęstej mapy środowiska opisanej klasami terenu, co często wymaga zewnętrznych pomiarów referencyjnych terenu. Ponadto klasy terenu są często rozróżniane na podstawie ludzkiej percepcji, a nie sygnałów haptycznych z robota. Celem jest zaproponowanie rozwiązania



lokalizacyjnego, które nie wymaga wcześniejszych danych od człowieka, ale zamiast tego buduje mapę haptyczną w całości z wewnętrznych czujników robota i ma możliwość lokalizacji z użyciem tej mapy. Problem lokalizacji jest ściśle związany z problemem reprezentacji danych dla sygnałów siły/momentu obrotowego pochodzących z czujników stóp robota. Przy odpowiedniej reprezentacji powinno być możliwe wygenerowanie mapy środowiska sygnałów haptycznych, gdy dostępna jest lokalizacja oparta na wizji lub LiDAR-ach 3D. Następnie, gdy robot odwiedza ten sam obszar bez wizji komputerowej, używa zakodowanych sygnałów dotykowych do lokalizacji na podstawie mapy. W ten sposób robot realizuje podejście znane jako scenariusz *teach & repeat* [23].

### 4.3 Kalibracja czasoprzestrzenna kamery i LiDAR-u 3D



Rysunek 4: Ogólne etapy przetwarzania proponowanej kalibracji czasoprzestrzennej kamera-LiDAR 3D.

W pracy [A1] zająłem się problemem wspólnej kalibracji czasowej i przestrzennej systemu sensorycznego składającego się z kamery i LiDAR-u 3D, przedstawionego na rys. 1A. Proponowany system przeznaczony jest do układów sensorycznych z kamerami posiadającymi globalną migawkę (ang. *global shutter*) i LiDAR-ami 3D, które są wspólnie sztywno zamontowane. Zakładam, że kamera jest skalibrowana wewnętrznie, a parametry kamery są stałe podczas kalibracji. Ponadto zakładam, że przesunięcie czasowe znaczników pomiarów pomiędzy sensorami jest nieznane, ale relatywnie małe i w przybliżeniu stałe. Procedura kalibracji polega na ruchu wzorca kalibracyjnego, który powinien być wspólnie i w sposób ciągły obserwowany przez kamerę i przez LiDAR 3D. Po nagraniu dane przetwarzam w sposób przedstawiony już wcześniej na rys. 4. Wzorec kalibracyjny jest wykrywany na obrazach z kamery za pomocą typowego algorytmu detekcji szachownicy z OpenCV. Z chmury punktów 3D, punkty należące do płaszczyzny wzorca kalibracyjnego są wykrywane za pomocą obrazów głębi (ang. *range image*) poprzez ręczne oznaczenie narożników znacznika albo za pomocą półautomatycznej procedury śledzenia z poprzedniej detekcji. Bez kalibracji czasowej, kalibracja przestrzenna może być określona przez optymalizację błędu odległości punktu z chmury punktów 3D do płaszczyzny określonej przez kamerę na podstawie dyskretnej liczby pól kalibracyjnych:

$$\mathbf{T}^* = \arg \min_{\mathbf{T}} \sum_i \sum_j \pi(t_i)^\top \mathbf{T} \mathbf{p}_i, \quad (1)$$



gdzie  $\mathbf{T}^*$  jest szukaną transformacją z układu współrzędnych sensora LiDAR do układu współrzędnych kamery,  $\mathbf{p}_i$  oznacza jednorodne współrzędne  $i$ -tego punktu lasera 3D zmierzonego dla wzorca kalibracyjnego, a  $\boldsymbol{\pi}(t_i)$  jest równaniem płaszczyzny wzoru szachownicy oszacowanym przez kamerę w momencie czasowym  $(t_i + \Delta t)$ . Płaszczyznę  $\boldsymbol{\pi}(t_i)$  dla znacznika czasu  $t_i$  można wyznaczyć poprzez znalezienie równania płaszczyzny dla detekcji na obrazie z kamery o najbliższym znaczniku czasu. W przedstawionym sformułowaniu częstotliwość przechwytywanych danych z obu systemów może być różna, ale zazwyczaj wynosi 10 Hz.

W tej pracy naukowej zajmuję się również kalibracją czasową, co wiąże się z kilkoma wyzwaniem. Dane z obu sensorów nie są rejestrowane w tych samych momentach (ang. *timestamps*), a LiDAR-y 3D stale się obracają, co powoduje zniekształcenia ruchu podobne do tych w kamerach (ang. *rolling-shutter*). Zniekształcenia wynikające z ruchu muszą być kompensowane w celu stworzenia chmury punktów w wybranym momencie czasowym. Taką kompensację realizuje się w ramach rozwiązań SLAM, tak jak w przypadku LiDAR-ów 3D w [75] lub kamer w [45]. Niestety, wykorzystanie SLAM nieuchronnie wprowadza dodatkowe źródła błędów kompensacji wynikające z dokładności samych systemów SLAM. Dlatego zdecydowałem się na niezależne traktowanie każdego punktu z odpowiadającym mu znacznikiem czasu, podobnie jak w przypadku pierwszych kroków odometrii laserowej [75] lub w [9]. W tym bardziej skomplikowanym przypadku, kalibracja czasoprzestrzenna jest określona przez następującą optymalizację:

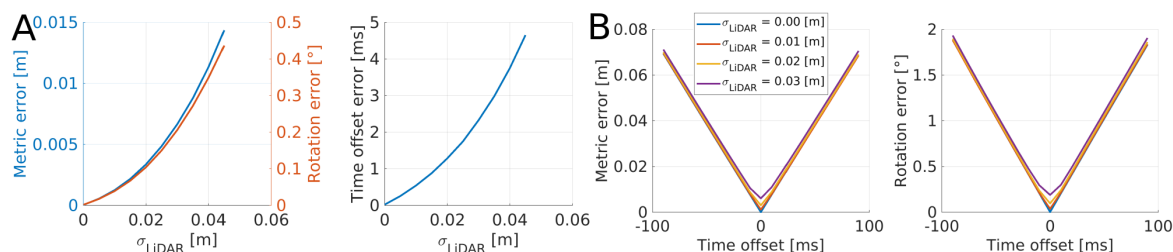
$$\mathbf{T}^*, \Delta t^* = \arg \min_{\mathbf{T}, \Delta t} \sum_i \boldsymbol{\pi}(t_i + \Delta t)^\top \mathbf{T} \mathbf{p}_i, \quad (2)$$

gdzie  $\Delta t^*$  jest szacowaną różnicą czasu pomiędzy znacznikami czasu kamery i LiDAR-u 3D,  $\boldsymbol{\pi}(t_i + \Delta t)$  jest równaniem płaszczyzny wzoru szachownicy określonym przez kamerę w momencie czasowym  $(t_i + \Delta t)$ , a  $t_i$  jest znacznikiem czasu pozyskania  $i$ -tego punktu lasera  $\mathbf{p}_i$  leżącego na płaszczyźnie znacznika kalibracyjnego. Sformułowanie problemu jak w (2) wymaga indywidualnych znaczników czasu dla każdego punktu chmury punktów 3D oraz znajomości równania płaszczyzny dla dowolnego znacznika czasu. W optymalizacji wykorzystano algorytm Levenberga-Marquardta z odporną funkcją kosztu Hubera dostępną w bibliotece g2o [39].

Aby określić równanie płaszczyzny dla dowolnego znacznika czasu, używam następujących etapów przetwarzania danych:

- A. Na każdym zarejestrowanym obrazie z kamery wykrywany jest znacznik kalibracyjny.
- B. Znając parametry wewnętrzne kamery i rzeczywisty rozmiar znacznika szachownicy, wyznaczana jest transformacja pomiędzy układem współrzędnych kamery a układem współrzędnych wzoru kalibracyjnego.
- C. Obliczam równanie płaszczyzny znacznika, które jest reprezentowane przez 4-wymiarowy wektor  $(\mathbf{n}, d)$ , gdzie  $\mathbf{n}$  to znormalizowany wektor normalny płaszczyzny, a  $d$  to odległość od początku układu współrzędnych.
- D. Ze wszystkich obrazów otrzymuję zestaw równań płaszczyzny obserwowanych dla dyskretnych znaczników czasu zarejestrowanych obrazów.

Aby wyznaczyć równania płaszczyzny w momentach czasowych pomiędzy znacznikami czasowymi obrazów, muszę interpolować uzyskane równania płaszczyzny. Reprezentacja 4-wymiarowa nie jest minimalna; dlatego interpolacja może spowodować powstanie równań płaszczyznowych wymagających dalszej renormalizacji. Aby uniknąć tego problemu, zaproponowałem minimalną 3-wymiarową reprezentację płaszczyzny wykorzystującą własności grupy  $SO(3)$  (podążając za ideami z [4]) i odpowiadające jej elementy algebry Liego. Minimalna reprezentacja płaszczyzny umożliwia wyznaczenie równań płaszczyzny w wymaganych momentach czasowych poprzez konwersję do minimalnej reprezentacji, wykonanie interpolacji, a następnie powrót do zwykłego 4-wymiarowego równania płaszczyzny. Do tego zadania próbowałem liniowej interpolacji parametrów, ale brak ciągłych pochodnych jest istotnym problemem, który powoduje, że optymalizacja utyka w lokalnych minimach. Dlatego zaproponowałem wykonanie interpolacji za pomocą sześciennych krzywych B-sklejanych (ang. *B-spline*) dla ciągłych reprezentacji płaszczyzny, które gwarantują ciągłe pierwsze i drugie pochodne minimalnej reprezentacji płaszczyzny [27, 52, 64].



Rysunek 5: Błędy metryczne, rotacyjne i przesunięcia czasowego kalibracji w zależności od: A) odchylenia standardowego pomiaru zasięgu LiDAR-u ( $\sigma_{LiDAR}$ ), B) wybranego przesunięcia czasowego, gdy wykonywana jest tylko kalibracja przestrzenna (brak kalibracji czasowej).

System oceniałem w serii eksperymentów symulacyjnych typu Monte-Carlo. Każdy eksperyment rozpoczyna się od stworzenia ruchu przypominającego rzeczywisty ruch znacznika kalibracyjnego przed zestawem sensorycznym i trwa przez 50 sekund. Przeprowadziłem 1900 losowych eksperymentów (100 różnych trajektorii ruchu znacznika kalibracyjnego dla 19 różnych przesunięć czasowych pomiędzy sensorami). W symulacji zmierzyłem błąd przekształcenia metrycznego, rotacji i przesunięcia czasowego pomiędzy wartościami rzeczywistymi i oszacowanymi (rys. 5A). Dla średniej dokładności Velodyne VLP-16 wynoszącej  $\pm 2$  cm uzyskałem średnie błędy kalibracji 0,12 cm, 0,04° i 0,54 ms, udowadniając, że jest to skuteczne podejście do kalibracji. Oszacowałem również błąd, gdy pomimo przesunięć czasowych stosowana jest tylko transformacja przestrzenna (rys. 5B). Stwierdziłem, że oszacowanie przesunięcia czasowego jest kluczowe, ponieważ średni błąd metryczny transformacji wynoszący 3,07 cm i 0,83° został zmierzony, gdy przesunięcie czasowe było równe 40 ms i wzrastał, gdy przesunięcie czasowe było jeszcze większe. Dowodzi to, że kalibracja czasoprzestrzenna powinna być wybierana zamiast przestrzennej w prawie wszystkich scenariuszach (z wyjątkiem sytuacji, gdy dostępna jest synchronizacja sprzętowa).

Przeprowadziłem również testy z wykorzystaniem rzeczywistych konfiguracji sensorycznych. Konfiguracja eksperymentalna do weryfikacji w świecie rzeczywistym składała się z dwóch kamer MV BlueFox3-2016C-1112 z globalną migawką i obiektywem Basler C125-0418-5M F1.8 f4mm sztywno zamontowanych po obu stronach Velodyne VLP-16, jak pokazano na rys. 6. W tym układzie brakuje sprzętowego wyzwalania, więc kamery były wyzwalane programowo za pomocą naszego własnego sterownika, zapewniając bliskie (ale nieidealne) czasy wyzwalania.



Rysunek 6: Układ eksperymentalny z dwiema kamerami MV BlueFox3-2016C-1112 i Velodyne VLP-16 LiDAR bez wyzwalania sprzętowego (A), który został wykorzystany do mapowania wewnątrz pomieszczeń po zamontowaniu na kasku (B).

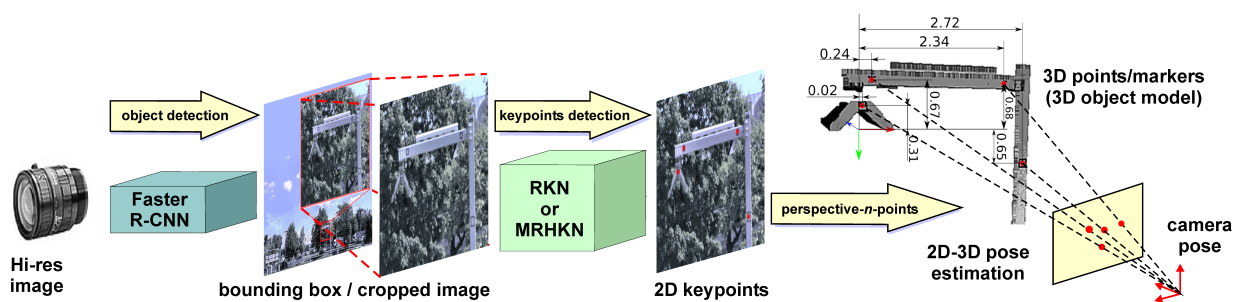
Na początku skalibrowałem obie kamery używając *kalibr* [22, 54], uzyskując transformację z lewej kamery do prawej kamery,  ${}^{\text{right}}\mathbf{K}_{\text{left}}$ , która była prawie identyczna z ręcznie zmierzonym przesunięciem. Uzyskano dokładność  $\pm 0.06$  cm i  $\pm 0.0991^\circ$  wg *kalibr* i dlatego traktuję te pomiary jako referencję do badania swojego podejścia. Przy oryginalnej bazie stereowizyjnej wynoszącej 31 cm, zmierzyłem różnicę translacji przekształcenia  $\mathbf{T}$  jako 0,74 cm i różnicę rotacyjną jako 0,97 stopnia. Dla obu konfiguracji, lewa kamera-LiDAR 3D oraz prawa kamera-LiDAR 3D, uzyskałem przesunięcia czasowe wynoszące odpowiednio  $\Delta t_1 = 3,99$  ms i  $\Delta t_2 = 3,77$  ms. Różnicę między  $\Delta t_1$  i  $\Delta t_2$  można przypisać niedoskonałemu wyzwalaczowi programowemu naszego rozwiązania konfiguracji sensorycznej.

W ramach swoich prac naukowych zaproponowałem nowe oprogramowanie do kalibracji kamery-LiDAR-u 3D. Jest to pierwsze rozwiązanie oparte na markerach, które zapewnia czasoprzestrzenną kalibrację, dzięki nowatorskiej interpolacji krzywych b-sklejanych dla równań płaszczyzn wykorzystujących minimalną reprezentację w algebrze Liego. Rozwiązanie wymaga jedynie powszechnie dostępnego znacznika kalibracyjnego i krótkiej, jednodominutowej sesji kalibracyjnej, aby zapewnić powtarzalne i dokładne wyniki. Ponadto jest ono dostępne jako otwarty pakiet oprogramowania kompatybilny z systemem ROS<sup>5</sup>.

<sup>5</sup><https://github.com/LRMPUT/CameraLidarCalibrator>

#### 4.4 Podejście do kalibracji systemów sensorycznych oparte na danych z uwzględnieniem docelowej aplikacji w problemie wizyjnego pozycjonowania autobusów elektrycznych

Proponowany system ma na celu wspomaganie kierowców autobusów podczas próby dokonania pantografu zamontowanego na autobusie do ładowarki elektrycznej zamontowanej na maszcie w pobliżu drogi. Kompletny system wsparcia kierowców (ang. *Advanced driver-assistance system* - ADAS) składa się z modułu lokalizacyjnego, modułu planowania oraz interfejsu człowiek-maszyna. W [A2] skupiam się na projekcie rozwiązania lokalizacyjnego, natomiast [A3] prezentuje podejście do zwiększenia dokładności tego systemu z użyciem procedury kalibracyjnej.



Rysunek 7: Schemat blokowy potoku przetwarzania obrazu wykorzystany do pozycjonowania autobusów elektrycznych względem stacji ładowania. Model CAD jest wizualizacją chmury punktów 3D uzyskanej ze skanera laserowego 3D SURPHASER 100HSX z zaznaczonymi wymiarami pomiędzy sztucznymi znacznikami.

System ADAS został wyposażony w wysokiej rozdzielczości kamerę 20 MP RGB obserwującą ładowarkę elektryczną z częstotliwością około 4 Hz oraz dokładny system *ang. Differential Global Positioning System* (DGPS) do pomiaru dokładności rozwiązania opartego na kamerze. Cały zestaw sensoryczny został zamontowany za pomocą przyssawek do dachu autobusu. DGPS, zawierający dwie anteny oddzielone od siebie o kilka metrów, został skonfigurowany do pracy w trybie kinematycznym czasu rzeczywistego (ang. *Real Time Kinematic* - RTK) dostarczając położenie i kąt układu autobusu w układzie współrzędnych Ziemi z częstotliwością 8 Hz. Pozycja z DGPS jest zwracana z dokładnością 1 cm + 1 ppm (*parts per million*) CEP (*circular error probable*) [67], co oznacza, że mediana błędu wynosi w naszym przypadku 1,003 cm gdyż stacja RTK znajdowała się 3 km od terenu eksperymentu [67]. Kąt układu autobusu jest określany z globalną dokładnością wynoszącą 0,4 stopnia [67]. Podczas pracy obrazy z kamery RGB, jak również pomiary z DGPS, są zapisywane na tym samym komputerze. DGPS zapewnia wystarczającą dokładność lokalizacji dla potrzeb proponowanego systemu ADAS, jednak partner komercyjny w projekcie NCBR (firma Solaris Bus & Coach) uznał DGPS za zbyt niepraktyczny do zastosowania w świecie rzeczywistym ze względu na zmniejszoną dokładność w pobliżu wysokich budynków i ciągłą potrzebę zewnętrznych korekt RTK [48].

Przetwarzanie systemu lokalizacji ADAS (rys. 7) rozpoczyna się od rejestracji obrazu z kamery RGB. Następnie obraz jest przetwarzany przez moduł głębokiego uczenia generujący cztery współrzędne na obrazie odpowiadające punktom charakterystycznym należącym do wybranych lokalizacji zaznaczonych na maszcie i samej ładowarce. Lokalizacje tych punktów charakterystycznych zostały wybrane tak, aby były wizualnie wyraźne i łatwe do powtarzalnego wykrycia, a jednocześnie łatwe do oznaczenia na potrzeby uczenia modułu uczenia głębokiego. Dla współrzędnych obrazu usuwam następnie zniekształcenia i wykorzystuję je do uzyskania estymacji pozycji 3D kamery względem głowy ładowarki elektrycznej poprzez rozwiązanie problemu *perspective-n-point* (PnP) [76]. Problem PnP określa pozycję 3D kamery na podstawie położenia  $n$  punktów wykrytych na obrazie oraz odpowiadającego im położenia 3D tych punktów w modelu. Spośród wielu dostępnych algorytmów rozwiązywania problemu PnP wybrałem algorytm iteracyjny, który gwarantuje dokładne wyniki, gdy znana jest przybliżona pozycja początkowa [42].

W przedstawionym przypadku, dla  $i$ -tej detekcji, cztery punkty wykryte na obrazie ( $\mathbf{p}_i$ ) są porównywane do punktów referencyjnych w modelu 3D ( $\mathbf{q}$ ), aby określić pozycję ładowarki w układzie kamery  $\mathbf{T}_i$  za pomocą iteracyjnego algorytmu solvePnP dostępnego w OpenCV [10]:

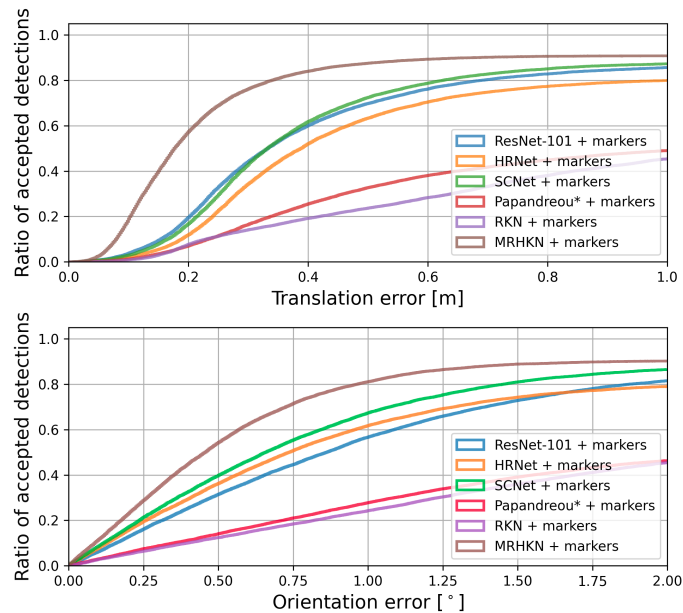
$$\mathbf{T}_i = \arg \min_{\mathbf{T}} d_i(\mathbf{T}, \mathbf{p}_i, \mathbf{q}), \quad (3)$$

gdzie  $d_i(\cdot)$  odległością zdefiniowaną jako:

$$d_i(\mathbf{T}, \mathbf{p}_i, \mathbf{q}) = \sum_k^4 \left( \mathbf{p}_i^k - \Pi(\mathbf{T}\mathbf{q}^k) \right). \quad (4)$$

$\mathbf{p}_i^k$  określa współrzędne obrazowe  $k$ -tego punktu,  $\Pi(\cdot)$  jest funkcją projekcji określającą koordynaty obrazowe na podstawie położenia punktów 3D oraz parametrów wewnętrznych kamery, a  $\mathbf{q}^k$  jest położeniem 3D  $k$ -tego punktu referencyjnego. Dla bezpieczeństwa sprawdzam jeszcze, czy ostateczny błąd po optymalizacji  $d_i(\mathbf{T}, \mathbf{p}_i, \mathbf{q})$  nie przekracza założonego poziomu błędów  $d_{\text{RMSE}}$ . Błąd może przekroczyć próg, jeśli wykryte punkty ładowarki należą do całkowicie błędnej lub niedokładnej detekcji i w takich przypadkach odrzucam analizowaną detekcję.

W [A2] zbadaliśmy dwa podejścia do wykonywania detekcji punktów kluczowych: rozwiązanie oparte na Faster R-CNN nazwane *Regression Keypoint Network* (RKN) oraz rozwiązanie wykorzystujące mapy ciepła do wykrywania punktów kluczowych nazwane *Max Resolution Heatmap Keypoint Network* (MRHKN). Ich szczegółowy opis można znaleźć w artykule [A2]. Na rysunku 8 przedstawiono wydajność podejść RKN i MRHKN w porównaniu z wydajnością modelu state-of-the-art z [51] oraz rozwiązań dostępnych w bibliotece MMPose. Z kształtu krzywych błędów wynika, że metoda MRHKN znacznie dokładniej określa zarówno pozycję, jak i orientację kamery w stosunku do pozostałych metod. Dla naszej metody RKN oraz metody Papandreou rozkład błędów jest w przybliżeniu liniowy i gorszy niż dla ocenianych modeli ResNet101, HRNet, SCNet oraz naszej metody MRHKN. Zakładamy, że gorsze wyniki sieci state-of-the-art w porównaniu z naszym rozwiązaniem MRHKN są spowodowane inną architekturą głowicy przetwarzania



Rysunek 8: Skumulowane funkcje rozkładu błędu translacji 2D (góra) i błędu orientacji (dół) dla proponowanych podejść RKN i MRHKN w porównaniu z podejściami zaimplementowanymi w ramach biblioteki MMPose i sieci zaproponowanej przez Papandreou potwierdzają wyższą dokładność podejścia MRHKN.

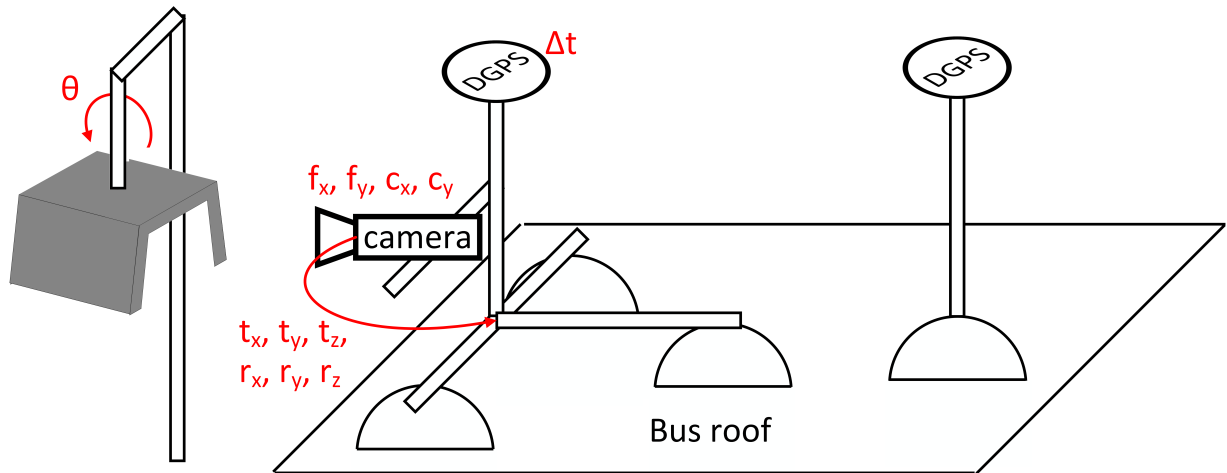
odpowiedzialnej za estymację punktów kluczowych, która zawiera mniejszą liczbę warstw konwolucyjnych. Wynik HRNet może być również gorszy ze względu na stosunkowo małą liczbę map cech, które zwracają początkowe warstwy sieci. Wszystkie metody zaimplementowane w MMPose (HRNet, SCNet i ResNet101) wypadają dość podobnie, przy czym najlepszy błąd translacji uzyskano dla ResNet101, ale mniejsze błędy orientacji można zaobserwować dla HRNet i SCNet.

Uzyskanie prezentowanych wyników z [A2] było możliwe dzięki nowatorskiemu podejściu do kalibracji, które ma na celu zwiększenie dokładności przetwarzania opartego na kamerze tak, aby odpowiadała ona dokładności systemu uzyskanej za pomocą DGPS w idealnych warunkach. W tym układzie dążę do uzyskania systemu lokalizacji opartego na kamerze z dokładnością DGPS, jednocześnie działającego z pożądaną dokładnością nawet w warunkach, gdy DGPS nie jest dostępny.

Jakość działania systemu i samej kalibracji zależy od doboru parametrów kalibracyjnych. W zaproponowanym układzie sensorycznym wyznaczyłem następujące parametry tworzące zbiór parametrów kalibracyjnych  $\mathbf{P}$ :

- metryczne umiejscowienie kamery w podstawie przyssawki użytej do montażu  $(t_x, t_y, t_z)$ ,
- obrót kamery w stosunku do podstawy montażowej  $(r_x, r_y, r_z)$ ,
- ogniskowe kamery  $(f_x, f_y)$ ,
- środek optyczny kamery  $(c_x, c_y)$ ,





Rysunek 9: Parametry kalibracyjne wpływają na model pomiarowy kamery podczas manewrów dokowania. W kalibracji uwzględniam pozę kamery  $(t_x, t_y, t_z, r_x, r_y, r_z)$ , wewnętrzne parametry kamery  $(f_x, f_y, c_x, c_y)$ , kąt odchylenia daszku ładowniarki elektrycznej  $(\theta)$  oraz przesunięcie czasowe pomiędzy pomiarami kamery i DGPS  $(\Delta t)$ .

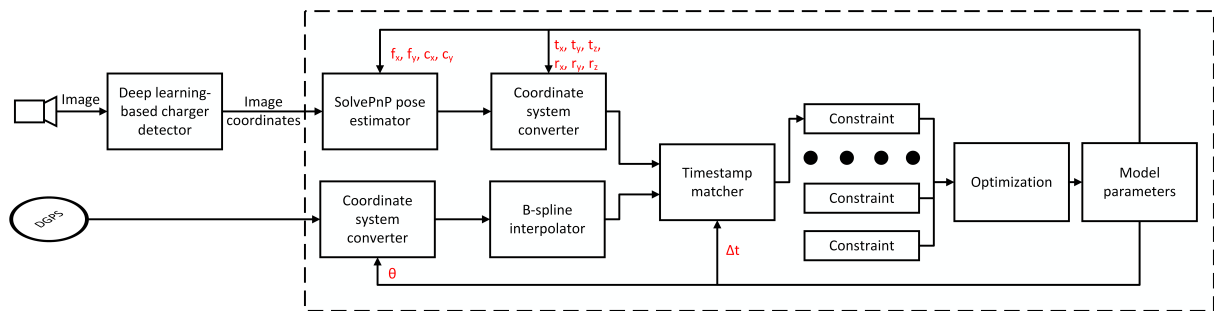
- korekta kąta odchylenia daszka ładowniarki elektrycznej  $(\theta)$ ,
- przesunięcie znaczników czasu między pomiarami uzyskanymi dla obrazów z kamery i pomiarów DGPS  $(\Delta t)$ .

Wartości tych parametrów zostały dobrane w trakcie procedury kalibracji i są wizualnie przedstawione na rys. 9. Zestaw parametrów kalibracyjnych obejmuje najbardziej oczywiste źródła niedokładności w końcowych pomiarach lokalizacyjnych począwszy od pozycji  $(t_x, t_y, t_z)$ , orientacji  $(r_x, r_y, r_z)$  oraz parametrów wewnętrznych  $(f_x, f_y, c_x, c_y)$  kamery. W optymalizacji uwzględniłem również kąt odchylenia dachu ładowniarki elektrycznej  $(\theta)$ , ponieważ kąt daszka ładowniarki nie mógł być dokładnie zmierzony w modelu 3D ze względu na niewielką długość ładowniarki (ok. 1 m). Ponadto obrazy z kamery i pomiary DGPS mają niezależne źródła oznaczenia znacznikami czasu. Dlatego, aby osiągnąć większą skuteczność, zdecydowałem się uwzględnić w kalibracji przesunięcie znacznika czasu  $(\Delta t)$  między obrazami z kamery RGB i pomiarami DGPS.

Jeśli nie zaznaczono inaczej, szukam wartości wszystkich wymienionych parametrów kalibracyjnych, nawet jeśli niektóre z nich można dokładnie oszacować przed kalibracją. Na przykład położenie kamery na płaszczyźnie dachu  $(t_x, t_y)$  można było łatwo i dość dokładnie zmierzyć za pomocą miary. Dodatkowo wyróżniam kalibrację, która poszukuje wartości parametrów bez ręcznie zmierzonych wartości  $t_x$  i  $t_y$ , nazywając ją zredukowanym (RED) zestawem parametrów kalibracji.

Aby poprawić estymację pozy autobusu, zweryfikowałem trzy podejścia do kalibracji. Pierwsze z nich, nazwane kalibracją klasyczną (ang. *classical calibration* - CC), wyko-

rzystuje publicznie dostępne oprogramowania do kalibracji oraz model CAD do określenia relacji geometrycznych pomiędzy pomiarami z kamery a systemem autobusu. Metoda ta została uznana za referencyjną dla pozostałych dwóch proponowanych podejść: kalibracji wspomaganiej przez człowieka (ang. *human-assisted calibration* - HAC) i automatycznej kalibracji (ang. *automatic calibration* - AC), które są oparte na optymalizacji w celu zminimalizowania różnicy pomiędzy proponowanym systemem opartym na kamerze i pomiarach referencyjnych DGPS. W drugiej metodzie HAC zbieram zestaw statycznych i dynamicznych pomiarów referencyjnych przy użyciu kamery i DGPS z różnych pozycji w stosunku do ładowarki. Informacja o lokalizacji punktów kluczowych jest następnie wykorzystywana w procedurze optymalizacji w celu określenia parametrów kalibracji. W trzeciej metodzie AC zbieram dane podczas symulowanych manewrów dokowania, a lokalizacja punktów ładowarki jest automatycznie wyznaczana przez proponowaną sieć neuronową. Podobnie jak w przypadku HAC, optymalizuję parametry kalibracji, starając się zminimalizować błąd pomiędzy systemem opartym na kamerze i DGPS. Ostatecznie, podejście AC może wykorzystać o rząd wielkości więcej pomiarów w porównaniu z HAC, ponieważ nie wymaga wkładu człowieka.



Rysunek 10: Etapy przetwarzania systemu kalibracji z poszukiwanymi parametrami kalibracji zaznaczone kolorem czerwonym. Estymacja pozycji na podstawie obrazów RGB w systemie ADAS jest rozszerzona o pomiary DGPS traktowane jako pomiar referencyjny w celu sformułowania problemu optymalizacyjnego. Prostokąt narysowany przerywanymi liniami zawiera część przetwarzania zależną od parametrów kalibracji wyznaczonych w wyniku optymalizacji.

Kalibracja jest sformułowana jako gradientowa procedura optymalizacji Levenberga-Marquardta, która iteracyjnie minimalizuje różnicę pomiędzy pozami estymowanymi ze współrzędnych obrazu ( $c_i$ ) i DGPS ( $g_j$ ) traktowanymi jako referencyjne za pomocą potoku przetwarzania przedstawionego na rys. 10. W tym podejściu detektor punktów kluczowych na obrazie RGB oparty na głębokim uczeniu znajduje się poza zakresem pętli kalibracyjnej, ponieważ wartości estymowanych parametrów kalibracyjnych nie mają wpływu na otrzymane współrzędne obrazu ( $c_i$ ). W związku z tym pochodna tego modułu nie jest potrzebna do optymalizacji. W przypadku pozostałych modułów przedstawionych na rys. 10 muszą posiadać możliwość określenia pochodnej wyjścia modułu względem



informacji wejściowej. Przykładowo obliczam pochodną poży w procedurze iteracyjnej estymacji poży względem wewnętrznych parametrów kamery. Te pochodne są potrzebne do obliczenia zmian wartości parametrów kalibracyjnych, które zmniejszają wartość optymalizowanej funkcji kosztu.

Celem optymalizacji jest wyznaczenie wartości parametrów kalibracyjnych ( $\mathbf{P}$ ):

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} \sum_i^m h(f(c_i, g_j, \mathbf{P})), \quad (5)$$

gdzie  $m$  to zbiór obserwacji dla danej kalibracji,  $f(\cdot)$  to wybrana funkcja błędu, a  $h(\cdot)$  jest odporną funkcją kosztu Cauchy'ego, która odpowiada za zmniejszenie wpływu błędnych detekcji. Istnieją trzy definicje funkcji błędu  $f(\cdot)$ , które przebadalem:  $f_{3D}$  oblicza różnicę dla pól 3D w przestrzeni 3D,  $f_{2D}$  jest różnicą dla pól 3D w przestrzeni 2D, a  $f_{\Pi}$  jest błędem reprojekcji:

$$f_{3D}(c_i, g_j, \mathbf{P}) = e_{3D}(\mathbf{T}_{c_i}^{-1} \mathbf{T}_{g_j}), \quad (6)$$

$$f_{2D}(c_i, g_j, \mathbf{P}) = e_{2D}(\mathbf{T}_{c_i}^{-1} \mathbf{T}_{g_j}), \quad (7)$$

$$f_{\Pi}(c_i, g_j, \mathbf{P}) = \sum_k^4 (\mathbf{p}_i^k - \Pi(\mathbf{T}_{g_j} \mathbf{q}^k)), \quad (8)$$

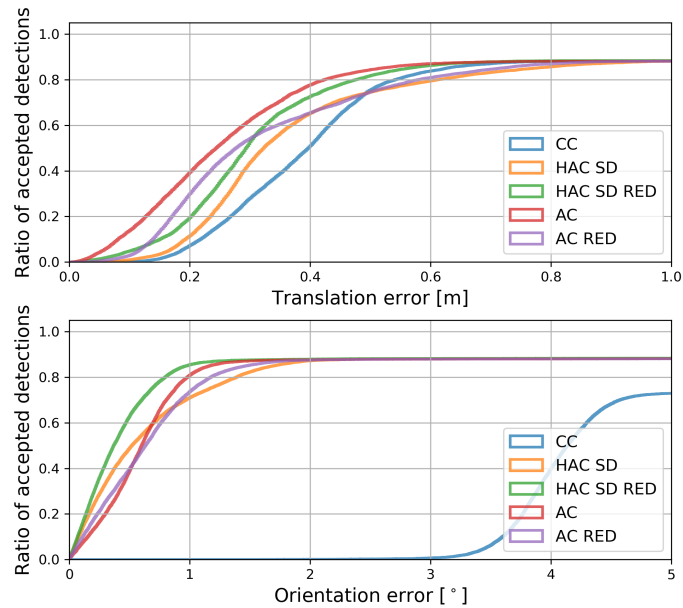
gdzie  $\mathbf{T}_{c_i}$  jest oszacowaniem pozycji 3D na podstawie współrzędnych obrazu ( $c_i$ ),  $\mathbf{T}_{g_j}$  jest oszacowaniem pozycji 3D z DGPS ( $g_j$ ), a błąd reprojekcji  $f_{\Pi}$  mierzy odległość na obrazie między wykrytymi punktami charakterystycznymi ładowarki ( $\mathbf{p}_i^k$ ) oraz ich położeniem na podstawie pomiaru DGPS rzutowanego na płaszczyznę obrazu ( $\Pi(\mathbf{T}_{g_j} \mathbf{q}^k)$ ) wyrażoną w pikselach. Błędy poży 3D  $e_{3D}(\cdot)$  i 2D  $e_{2D}(\cdot)$  są zdefiniowane w następujący sposób:

$$e_{3D}(\mathbf{T}) = \sqrt{t_x^2 + t_y^2 + t_z^2 + r_x^2 + r_y^2 + r_z^2}, \quad (9)$$

$$e_{2D}(\mathbf{T}) = \sqrt{t_x^2 + t_y^2 + r_z^2}, \quad (10)$$

gdzie przekształcenie  $\mathbf{T}$  składa się z trzech metrycznych komponentów translacji ( $t_x, t_y, t_z$ ) oraz trzech komponentów rotacji wyrażonych w algebrze Liego ( $r_x, r_y, r_z$ ) [17]. Błąd na płaszczyźnie 2D jest określony przez dwie składowe translacyjne ( $t_x, t_y$ ) i jeden obrót  $r_z$  (ang. *heading*). Jeśli nie zaznaczono inaczej, wszystkie procedury kalibracji oparte na optymalizacji wykorzystują błąd poży 3D jako domyślną funkcję błędu ( $f_{3D}$ ).

Zaproponowana autokalibracja (AC) została oceniona na dokładnie tym samym zestawie danych testowych, co klasyczna kalibracja (CC) i kalibracja wspomagana przez człowieka (HAC). Z funkcji skumulowanego rozkładu błędów (rys. 11) wynika, że zarówno kalibracja wspomagana przez człowieka (HAC), jak i autokalibracja (AC), uzyskują większą dokładność niż klasyczne podejście do kalibracji (CC). Wydajność systemu po autokalibracji (AC) przy optymalizacji wszystkich wymienionych parametrów kalibracyjnych osiągnęła medianę błędu metrycznego równą 0,22 m i medianę błędu kąтового równą 0,55°. W przeciwieństwie do kalibracji wspomaganej przez człowieka (HAC), zmniejszenie liczby parametrów optymalizacyjnych (RED) spowodowało wzrost błędów medianowych do wartości



Rysunek 11: Skumulowany wykres błędów dla kalibracji klasycznej (CC), kalibracji wspomaganej przez człowieka (HAC) na zbiorze danych *staticdynamic* (SD) oraz autokalibracji (AC). Oceniałem również konfiguracje optymalizujące tylko podzbiór całego zestawu parametrów (RED).

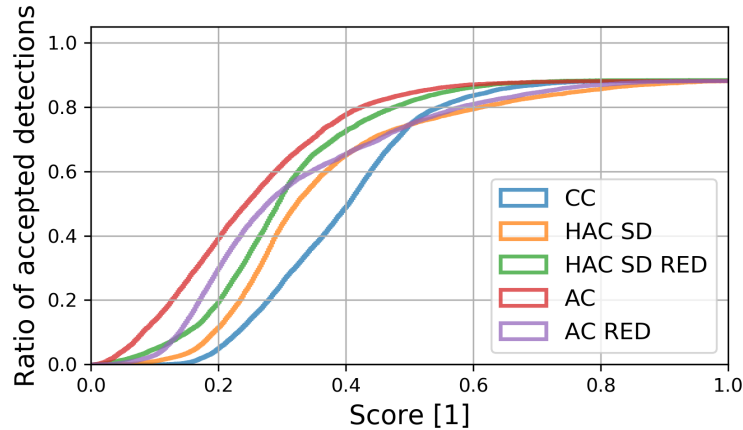
0,25 m i  $0,56^\circ$ . Uważam, że wynika to z faktu, że autokalibracja (AC) jest wykonywana na większym zbiorze pomiarów kalibracyjnych, prowadząc do lepiej ograniczonego problemu optymalizacyjnego, umożliwiając skuteczne wyznaczenie wartości większej liczby parametrów. Autokalibracja w obu konfiguracjach (z podzbiorem lub ze wszystkimi parametrami) skutkuje mniejszą medianą błędu metrycznego (o co najmniej 0,06 m) niż w przypadku konfiguracji wspomaganej przez człowieka (HAC), przy jednoczesnym odnotowaniu nieznacznie zwiększonego błędu kąowego (do około  $0,2^\circ$ ).

Z perspektywy aplikacji chciałem uszeregować proponowane konfiguracje kalibracyjne za pomocą jednej miary ujmującej zarówno dokładność metryczną, jak i rotacyjną. W związku z tym postanowiłem wprowadzić miarę  $s$  rozwiązania zdefiniowaną jako:

$$s = t [m] + r [rad], \quad (11)$$

gdzie  $t$  jest składową translacyjną błędu, a  $r$  jest składową rotacyjną błędu. Wynik  $s$  jest bezwymiarową wielkością miary, która łączy składowe translacyjne i rotacyjne. Aby dać wyobrażenie, 10 cm błędu translacji odpowiada temu samemu wkładowi do miary  $s$  co  $5,7^\circ$  błędu rotacji. Zdefiniowałem również medianę wyniku  $s_M$  jako medianę wszystkich uzyskanych miar  $s$  na testowym zbiorze danych. Wprowadzona miara jest typowym sposobem uwzględniania błędów metrycznych i rotacyjnych w funkcjach kosztu w zadaniach optymalizacyjnych, gdy łączone są składowe translacyjne i rotacyjne [39] oraz sposobem, w jaki było to robione w procedurach kalibracyjnych opartych na optymalizacji. Funkcja rozkła-

du kumulatywnego wprowadzonej punktacji  $s$  dla analizowanych pomiarów przedstawiona jest na rys. 12.



Rysunek 12: Skumulowana funkcja rozkładu wprowadzonej miary  $s$  łączącej błędy translacji i orientacji do bezpośredniego porównania wydajności kalibracji klasycznej (CC), kalibracji wspomagananej przez człowieka (HAC) i autokalibracji (AC).

Z przedstawionych wyników widać, że system po autokalibracji (AC) uzyskuje dokładniejszą lokalizację niż wersja kalibrowana przy wsparciu człowieka (HAC). Gdy weźmiemy pod uwagę medianę wartości miary  $s_M$ , to najlepszy wynik wynoszący 0,22 uzyskuje autokalibracja (AC), po której następuje wartość 0,28 dla kalibracji wspomagananej przez człowieka (HAC) oraz wynik 0,38 dla kalibracji klasycznej (CC). W [A3] sprawdziłem również doświadczalnie wpływ wybranego typu błędu optymalizacji, strategię wyboru parametrów do optymalizacji oraz odporność uzyskanych parametrów kalibracji na wybór progów w systemie kalibracji.

Przeprowadzone eksperymenty bezpośrednio dowodzą, że mimo wszelkich starań, kalibracja oparta na pomiarach fizycznych wspomaganym plikami CAD (CC) prowadzi do znacznego błędu. Ten błąd wynika z błędów rotacji, gdyż dokładne zmierzenie parametrów obrotów jest zazwyczaj problematyczne. Kalibracja wspomaganą przez człowieka (HAC) lub autokalibracja (AC) wyraźnie przewyższa kalibrację klasyczną. Z eksperymentów wynika, że zdolność do przetwarzania o rząd wielkości większej ilości danych w podejściu autokalibracji (AC) jest istotniejsza niż dokładniejsze anotacje człowieka na mniejszym zbiorze kalibracyjnym. Dokładne badania pokazały także, że użycie w optymalizacji miary błędu obejmującej największą liczbę stopni swobody, takiej jak błąd pozy 3D, prowadzi do najdokładniejszej kalibracji, nawet jeśli ostatecznie dokładność systemu mierzona jest z użyciem innych miar. Co więcej, możliwe jest uwzględnienie większej liczby parametrów w modelu kalibracyjnym (więcej stopni swobody), ponieważ na podstawie obserwacji można sformułować więcej ograniczeń dotyczących wartości parametrów kalibracyjnych systemu.

Potraktowanie problemu kalibracji w zadaniu docelowej aplikacji okazuje się kluczowe dla uzyskania jak najdokładniejszych wyników z kompletnego systemu lokalizacji. Wierzę, że

nowatorskie spojrzenie na kalibrację jako część kompletnego systemu docelowej aplikacji może zrewolucjonizować sposób przeprowadzania kalibracji systemów wielosensorycznych. Oczywistą wadą jest to, że uzyskana procedura nie jest uniwersalna dla wszystkich zastosowań, a zaproponowana idea może nie mieć zastosowania do systemów składających się z nieróżniczkowalnych modułów, gdzie nie ma możliwości propagacji błędów.

#### 4.5 Rozpoznawanie miejsc na podstawie wyglądu przy użyciu kamery i LiDAR-u 3D

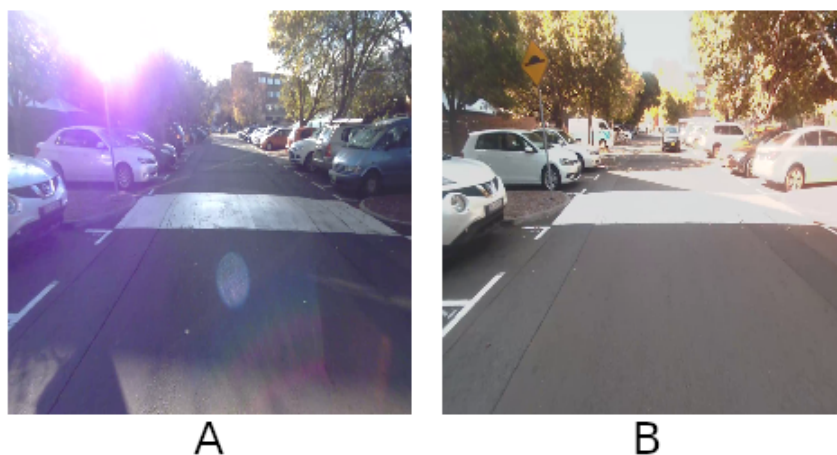
W [A4] skupiłem się na porównaniu systemów rozpoznawania miejsc opartych na kamerach, LiDAR-ach oraz proponowanym połączeniu kamery z LiDAR-em 3D w warunkach rzeczywistych. Celem tych metod jest skuteczne określenie położenia na podstawie pojedynczego zdjęcia lub jednej chmury punktów 3D na terenie na określonym i znanym obszarze, ale bez wykorzystywania informacji o wcześniejszej lokalizacji. Eksperymenty przeprowadzono na zbiorze danych USyD, uzyskując wyniki w różnych warunkach pogodowych obserwowanych dla każdego tygodnia w 6 kategoriach: słonecznie (**S**), pochmurno (**C**), słonecznie/pochmurno (**S/C**), po deszczu (**AR**), o zachodzie słońca (**SS**) i bardzo pochmurno (**VC**). Takie grupowanie umożliwia sprawdzenie, czy i jak warunki pogodowe wpływają na działanie systemu rozpoznawania miejsc.

W proponowanym podejściu, na wejście sieci neuronowej podawane są dane sensoryczne, gdzie na wyjściu otrzymujemy liczbowy opis danej lokalizacji zwany także deskryptorem (ang. *descriptor*). Podczas oceny działania metody porównujemy deskryptor lokalizacji uzyskany dla aktualnie analizowanego miejsca względem wszystkich deskryptorów lokalizacji dostępnych w referencyjnej bazie miejsc. Jeśli najbardziej podobna lokalizacja na podstawie podobieństwa deskryptorów znajdowała się w granicach  $\pm 10$  metrów od lokalizacji zmierzonej z GPS, znaleziona lokalizacja była uważana za poprawną. W pozostałych przypadkach lokalizacja testowa była oznaczana jako nieprawidłowa. Próg  $\pm 10$  metrów został wybrany eksperymentalnie jako wystarczająca dokładność rozwiązania opartego na wyglądzie miejsc, która powinna być zbieżna z rzeczywistą lokalizacją metryczną, jeśli zastosujemy znane podejście do lokalizacji opartej na geometrii, takie jak ICP [62].

Tabela 1: Procentowa dokładność rozpoznawania w referencyjnych (kolumny) i testowych (wiersze) warunkach pogodowych dla rozpoznawania miejsc za pomocą kamery. Zwracam uwagę na niższą wydajność w porównaniu ze średnią podczas testowania w słonecznych warunkach.

Kamera	S	C	S/C	AR	SS	VC	Średnia
<b>S</b>	80,32	83,20	85,39	84,61	83,33	83,03	82,08
<b>C</b>	83,26	85,55	86,68	87,27	87,74	86,99	84,78
<b>S/C</b>	86,00	87,24	87,73	89,99	88,06	88,46	86,98
<b>AR</b>	83,67	85,74	89,02	90,14	89,87	86,71	85,53
<b>SS</b>	79,50	84,39	83,37	89,77	90,63	83,28	82,39
<b>VC</b>	83,15	84,78	88,24	88,25	86,41	-	84,68
<b>Średnia</b>	81,82	84,37	86,15	86,47	85,66	84,72	<b>83,49</b>

Na podstawie wszystkich testów zmierzono, że dokładność rozpoznawania miejsc za pomocą kamery jest równa 83,49%, co oznacza, że dla 16,51% system określił lokalizację znajdującą się dalej niż 10 metrów od lokalizacji prawdziwej. Uzyskany wynik na podstawie danych z kamery służy nam jako punkt odniesienia do dalszych badań. Dokładna wydajność w zależności od zmiennych warunków pogodowych została przedstawiona w tab. 1. Najgorsza wydajność, oznaczona czerwonym kolorem tła, została zaobserwowana dla warunków słonecznych i w przypadku rejestracji danych podczas zachodu słońca. Bezpośrednie promienie słoneczne są czynnikiem, który może negatywnie wpłynąć na proces akwizycji obrazu prowadząc do przeświecienia, które drastycznie zmienia percepcję lokalizacji, tak jak to przedstawiono na rys. 13. W takich przypadkach nie ma prostego sposobu na poprawę jakości obrazów z wybranej kamery. W praktyce najprostszym sposobem radzenia sobie z takimi sytuacjami jest użycie innej kamery, która mogłaby być skierowana do tyłu lub oparcie się na innym typie sensora, np. LiDAR.



Rysunek 13: Wizualne porównanie zdjęć tej samej lokalizacji wykonanych podczas zachodu słońca (A) i w słonecznych warunkach (B). Wizualna różnica na zdjęciach stanowi wyzwanie dla systemów rozpoznawania miejsc za pomocą kamery ze względu na bezpośrednie światło słoneczne.

Podobnie jak w przypadku rozpoznawania miejsc za pomocą kamery, zaproponowałem wersję działającą na intensywnościach powracającego sygnału w LiDAR-ze 3D przedstawioną jako obraz. Dokładność rozpoznawania w tym przypadku wyniosła 81,11%, co jest wskaźnikiem niższym niż dokładność rozpoznawania uzyskana dla rozwiązania opartego na kamerze. Uważam, że należało się tego spodziewać, ponieważ Velodyne VLP-16 LiDAR użyty w zbiorze danych USyD ma tylko 16 niezależnych, poziomych linii skanowania, które muszą być znacznie przeskalowane, aby dopasować się do oczekiwanego rozmiaru wejściowego sieci. W przypadku kamery oryginalny obraz zawiera więcej pikseli i jego rozmiar musi być zmniejszony. Bardziej szczegółowe wyniki w różnych warunkach pogodowych przedstawiono w tab. 2.

W większości przypadków rozwiązanie oparte na chmurach punktów 3D działa podobnie,

Tabela 2: Dokładność rozpoznawania w procentach na podstawie referencyjnych (kolumny) i testowych (wiersze) warunków pogodowych dla rozpoznawania miejsc na podstawie LiDAR-u. Zwracam uwagę, że ogólna wydajność jest podobna, z wyjątkiem warunków po deszczu (AR)

LiDAR 3D	S	C	S/C	AR	SS	VC	Średnia
S	81,21	81,27	82,05	81,05	80,56	81,47	81,25
C	81,22	78,58	80,29	78,69	79,15	79,64	80,24
S/C	82,87	82,71	82,99	82,00	81,50	84,16	82,71
AR	80,39	77,80	79,66	83,31	84,81	78,03	80,24
SS	81,02	79,90	77,87	86,30	89,58	82,58	81,39
VC	82,12	83,32	82,35	80,35	80,49	-	82,09
Średnia	81,29	80,55	81,26	81,15	81,10	81,05	81,11

wykazując pewną odporność na warunki pogodowe, ze spadkiem odporności, gdy akwizycja danych została przeprowadzona po deszczu (czerwone tło). Warunki po deszczu są problematyczne ze względu na dodatkowe krople deszczu zwiększające liczbę brakujących pomiarów w danych LiDAR 3D. Z drugiej strony system rozpoznawania miejsc oparty na LiDAR jest odporny na zmiany warunków oświetleniowych, skutecznie pracując w warunkach słonecznych i przy zachodzie słońca (zielone tło).



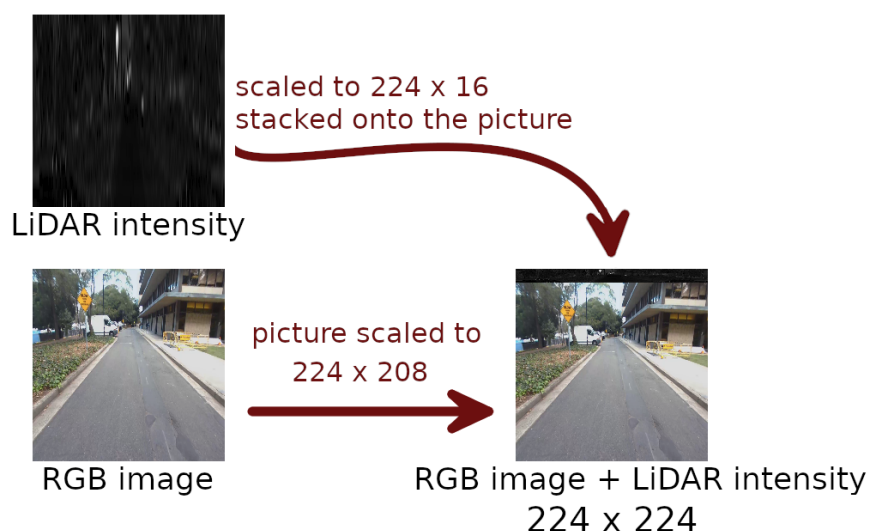
Rysunek 14: Przykładowa lokalizacja widoczna w różnych momentach: rozwiązanie oparte na LiDAR działa pewniej w przypadku większej struktury i intensywnego słońca (A), podobna wydajność wersji opartej na kamerze i LiDAR w typowych warunkach (B) oraz wersja oparta na kamerze przewyższająca LiDAR w idealnych warunkach oświetleniowych (C).

Przyjrzenie się bliżej ujawnia, że wydajność rozwiązania opartego na LiDAR jest bardziej zależna od geometrii sceny niż od jej wyglądu. Taki przypadek jest widoczny na rys. 14, kiedy brak plakatu na ścianie powoduje błędne rozpoznanie miejsca przez system oparty na



kamerze, ale wersja oparta na LiDAR 3D radzi sobie z tym przypadkiem. Mimo to system rozpoznawania miejsc oparty na LiDAR-ach 3D wypada ogólnie gorzej niż rozwiązanie oparte na kamerze.

Na podstawie tych wyników zaproponowałem system rozpoznawania miejsc oparty na parze kamera-LiDAR 3D, który powstał poprzez połączenie obrazu intensywności LiDAR-u z obrazem z kamery w celu utworzenia sztucznego obrazu wejściowego do sieci. Proces tworzenia sztucznego obrazu został przedstawiony na rys. 15. W tym sztucznym obrazie pierwsze 16 rzędów zawiera zmierzoną informację o intensywności LiDAR, natomiast pozostałe 208 rzędów zawiera przeskalowany obraz RGB. Podobnie jak w przypadku poprzednich wersji, przygotowaliśmy nową bazę danych uczących, nauczyliśmy, a następnie zweryfikowaliśmy wydajność sieci.



Rysunek 15: Wizualna reprezentacja obrazu opartego na kamerze-LiDAR, która powstaje przez połączenie zmierzonych intensywności LiDAR-u z przeskalowanym obrazem RGB w celu utworzenia sztucznego obrazu. Sztuczny obraz ma taki sam rozmiar jak dane wejściowe w rozwiązaniach opartych na kamerze i LiDAR.

System rozpoznawania miejsc oparty na kamerze-LiDAR osiągnął w sekwencji testowej poprawność rozpoznania na poziomie 86,91%, przewyższając wyniki uzyskane na podstawie pojedynczych sensorów. Dokładna wydajność w analizowanych warunkach pogodowych została zmierzona i przedstawiona w tab. 3.

System rozpoznawania miejsc oparty na kamerze-LiDAR osiąga najlepsze wyniki we wszystkich analizowanych warunkach pogodowych w porównaniu z rozwiązaniami opartymi na kamerze i LiDAR. W porównaniu z systemem rozpoznawania miejsc opartym na kamerze, prezentowana wersja osiąga najlepsze wyniki w warunkach zachodu słońca (wzrost wskaźnika rozpoznawania o 5,47 punktów procentowych) oraz w warunkach słonecznych (wzrost wskaźnika rozpoznawania o 3,56 punktów procentowych). Dowodzi

Tabela 3: Dokładność rozpoznawania w procentach na podstawie referencyjnych (kolumny) i testowych (wiersze) warunków pogodowych dla rozpoznawania miejsc opartego na kamerze i LiDAR-ze. Zwracam uwagę na wzrost wydajności w porównaniu z rozwiązaniami opartymi na samej kamerze lub samym LiDAR-ze.

<b>Kamera LiDAR 3D</b>	<b>S</b>	<b>C</b>	<b>S/C</b>	<b>AR</b>	<b>SS</b>	<b>VC</b>	<b>Średnia</b>
<b>S</b>	83,86	86,74	88,25	87,64	87,70	88,54	85,61
<b>C</b>	86,31	87,75	90,08	90,14	90,49	89,93	87,67
<b>S/C</b>	88,05	90,32	89,81	92,92	90,16	92,08	89,38
<b>AR</b>	87,27	89,59	91,36	92,94	92,55	91,52	88,99
<b>SS</b>	85,52	90,01	88,41	92,55	94,79	90,24	87,86
<b>VC</b>	87,08	89,19	90,95	90,94	92,33	-	88,58
<b>Średnia</b>	85,29	87,81	89,14	89,42	89,36	89,56	<b>86,91</b>

to, że dodatkowe dane o intensywności z LiDAR-u sprawiają, że rozpoznawanie miejsc jest bardziej odporne na bezpośrednie działanie promieni słonecznych.

W [A4] przedstawiłem porównanie systemów rozpoznawania miejsc opartych na kamerach, LiDAR-ach oraz proponowane połączenie systemów rozpoznawania miejsc opartych na kamerach i LiDAR-ach w różnych warunkach pogodowych na tych samych trajektoriach z wykorzystaniem publicznie dostępnego zbioru danych USyD. Przedstawiona ocena eksperymentu dowodzi, że wielosensoryczne rozpoznawanie miejsc powinno być rozważane w scenariuszach rzeczywistych, ponieważ zapewnia bardziej skuteczne rozwiązanie niż wykorzystanie tylko jednego sensora. Zauważyłem, że każda modalność powinna mieć inny potok przetwarzania bardziej dostosowany do danych wejściowych z każdego czujnika. Dlatego w [A5] bezpośrednio skupiłem się na określeniu najbardziej odpowiedniej reprezentacji chmury punktów 3D z LiDAR-u, która przyczyniłaby się do poprawy systemu rozpoznawania miejsc. Oparłem swój pomysł na architekturze sieci MinkLoc3D [35], która opiera się na lokalnej sieci ekstrakcji cech wykorzystującej rzadkie konwolucje 3D i warstwę GeM (*generalized-mean pooling layer*) [37]. W tym podejściu wejściowa chmura punktów  $(x_i, y_i, z_i)$  jest kwantyzowana do skończonej liczby prostopadłościanów zwanych wokselami z pojedynczym punktem 3D zlokalizowanym w każdym prostopadłościanie. Przetworzona informacja jest następnie wykorzystywana do utworzenia pojedynczego rzadkiego tensora skwantowanych punktów  $\hat{\mathbf{C}}$  i związanego z nim cechy  $\mathbf{F}$  [36]:

$$\hat{\mathbf{C}} = \begin{bmatrix} \hat{x}_1 & \hat{y}_1 & \hat{z}_1 \\ & \vdots & \\ \hat{x}_N & \hat{y}_N & \hat{z}_N \end{bmatrix}, \mathbf{F} = \begin{bmatrix} f_1^T \\ \vdots \\ f_N^T \end{bmatrix}. \quad (12)$$

W przypadku MinkLoc3D jest to podejście czysto geometryczne, w którym każda powiązana cecha jest równa jednej wartości ( $f_i = [1]$ ) wskazując na istnienie punktu 3D.

W [A3] proponuję MinkLoc3D-S wykorzystujący sferyczną reprezentację współrzędnych



3D punktów, która jest naturalną reprezentacją pomiarów LiDAR-ów 3D. Zmiana ta została zaproponowana, ponieważ chmury punktów 3D uzyskane z LiDAR-ów 3D mają różną gęstość, z większą ilością punktów bliżej środka skanera i zwykle mniejszym zakresem wartości w kierunku pionowym. Dlatego regularna siatka współrzędnych kartezjańskich nie jest dobrze dostosowana do rozmieszczenia punktów 3D położonych dalej od skanera. W miarę odsuwania się od skanera odległości między punktami naturalnie rosną, co utrudnia ekstrakcję cech wysokopoziomowych. MinkLoc3D-S korzysta z rzadkich konwolucji, ale używa alternatywnej reprezentacji sferycznej. W proponowanym podejściu każdy punkt 3D  $(x, y, z)$  z wysokością reprezentowaną przez składową  $z$  jest konwertowany na odpowiadającą mu reprezentację sferyczną  $(r, \theta, \phi)$ :

$$r = \sqrt{x^2 + y^2 + z^2}, \quad (13)$$

$$\theta = \text{atan2}(y, x), \quad (14)$$

$$\phi = \text{atan2}(z, \sqrt{x^2 + y^2}), \quad (15)$$

gdzie  $r$  to odległość pomiędzy punktem 3D a skanerem,  $\theta$  to poziomy kąt skanowania, natomiast  $\phi$  to pionowy kąt skanowania. W efekcie objętość prostopadłościanu (woksela) zwiększa się dla punktów 3D położonych dalej od środka LiDAR-u 3D.

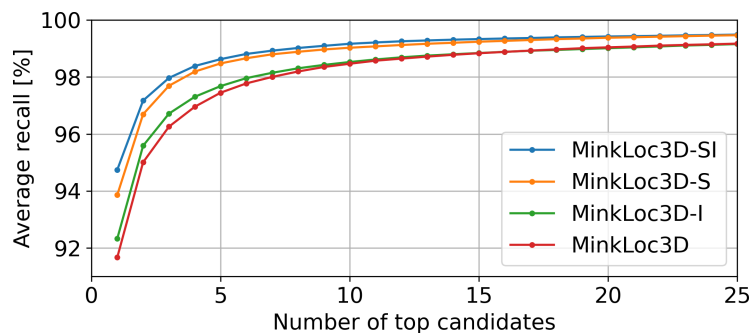
Oryginalny MinkLoc3D wykorzystuje tylko geometrię punktów 3D do rozpoznania miejsca. Ale jak stwierdzono w [3, 13] i w oparciu o nasze poprzednie wyniki w [A4], intensywność zwróconego sygnału dla LiDAR-ów 3D jest cenna i może być wykorzystana do skonstruowania lokalnego deskryptora miejsca. W związku z tym proponuję włączenie informacji o intensywności LiDAR-u do rzadkich konwolucji w MinkLoc3D-I. Intensywność  $s_i$  dla  $i$ -tego punktu 3D jest wypełniana w części mapy cech w procesie przetwarzania  $f_i = [s_i]$ , co nie zwiększa wymiarowości konwolucji. Surowe wartości intensywności są normalizowane do przedziału 0–1. Gdy wiele punktów mieści się w tym samym prostopadłościanie kwantyzacji, podczas uczenia losowo wybieram jedną z wartości intensywności z prostopadłościanu, aby zapewnić odpowiednią skuteczność. Mimo to, uśredniam wartości intensywności podczas wnioskowania, aby uzyskać powtarzalne wyniki. MinkLoc3D-SI łączy oba ulepszenia: reprezentację sferyczną i wykorzystanie intensywności pomiarów z LiDAR-u.

Wydańność systemu rozpoznawania miejsc mierzona jest za pomocą średniej czułości  $AR@X$ . Przyjmuje się, że miejsce dla  $AR@X$  jest poprawnie rozpoznane, jeśli dopasowanie z bazy danych  $X$  najbardziej podobnych miejsc zawiera co najmniej jedno miejsce w odległości  $c = 10$  m od miejsca zapytania. Wśród różnych wartości  $X$ , wartość na poziomie 1 ( $AR@1$ ) jest najważniejsza dla zastosowań robotycznych (przykładowo SLAM), ponieważ mierzy jak często pierwsze dopasowanie z bazy danych pasuje do lokalizacji zapytania. W scenariuszach SLAM włączenie błędnego rozpoznania może ostatecznie całkowicie zepsuć działanie rozwiązania lokalizacyjnego. Podaję również miarę  $AR@1\%$ , która zakłada, że miejsce jest poprawnie rozpoznane jeśli przynajmniej jedno miejsce znajduje się w odległości  $c = 10$  m od miejsca zapytania wśród  $Y$  najbardziej podobnych miejsc, gdzie  $Y$  równy jest wartości 1% z liczby możliwych miejsc w bazie danych.

Uzyskane wyniki numeryczne dla proponowanych modyfikacji zebrano w tab. 4 z wizualną reprezentacją miary  $AR@X$  dla  $X$  od 1 do 25 przedstawioną na rys. 16.

Tabela 4: Wyniki uzyskane na zbiorze danych USyd. Wykorzystując reprezentację sferyczną (MinkLoc3D-S), z uwzględnieniem intensywności (MinkLoc3D-I), jak również oba ulepszenia (MinkLoc3D-SI), uzyskuje się wyniki lepsze od rozwiązań bazowych, zwłaszcza dla miary  $AR@1$ .

USyd dataset	Źródło wyników	$AR@1\%$	$AR@1$
MinkLoc3D [35]	własne uruchomienie	98,1	91,7
Scan Context [25]	własne uruchomienie	88,7	86,0
MinkLoc3D-I	własne uruchomienie	98,2	92,3
MinkLoc3D-S	własne uruchomienie	98,8	93,9
MinkLoc3D-SI	własne uruchomienie	<b>99,0</b>	<b>94,7</b>



Rysunek 16: Wizualizacja miary  $AR@X$  dla różnych wartości  $X$  dla MinkLoc3D i proponowanych systemów. MinkLoc3D-SI uzyskuje najlepsze wyniki, co jest widoczne dla  $AR@1$ . Scan Context nie jest pokazany ze względu na znacznie gorsze wyniki.

MinkLoc3D-I poprawił wyniki względem podstawowego MinkLoc3D, uzyskując  $AR@1\%$  równe 98,2% oraz  $AR@1$  równe 92,3%, co potwierdza, że intensywność jest komplementarna do informacji o geometrycznej lokalizacji punktów 3D. MinkLoc3D-S ze sferyczną reprezentacją współrzędnych punktów 3D również poprawia wyniki do poziomu  $AR@1\%$  równego 98,8% i  $AR@1$  równego 93,9%. Reprezentacja sferyczna w tym przypadku jest lepiej przystosowana do uchwycenia relacji międzypunktowych dla punktów położonych dalej od skanera, co poprawia zdolność modelu do tworzenia wysokopoziomowych cech. Najbardziej znaczące wyniki dla obu analizowanych rozwiązań widoczne są dla  $AR@1$ , ponieważ  $AR@1\%$  jest już stosunkowo wysoki, a możliwości poprawy są niewielkie. Połączenie obu pomysłów w MinkLoc3D-SI uzyskuje najlepszy  $AR@1$  (94,7) i najlepszy  $AR@1\%$  (99,0) wśród ocenianych rozwiązań.

W [A5] zbadalem również wydajność, gdy rozwiązanie uczone na jednym zbiorze danych (Oxford RobotCar Intensity) zostanie sprawdzone na innym zbiorze danych (KITTI dataset), aby zmierzyć zdolność do generalizacji na niewidziane dane. W tym zadaniu generali-

Tabela 5: Wyniki generalizacji na zbiorze danych KITTI. MinkLoc3D-SI przewyższa pozostałe rozwiązania. Oxford RC oznacza Oxford RobotCar, natomiast Oxford RCI oznacza Oxford RobotCar Intensity.

KITTI dataset	Zbiór uczący	Źródło wyników	$AR@1\%$	$AR@1$
PointNetVLAD [68]	Oxford RC	[50]	72,4	–
LPD-Net [43]	Oxford RC	[50]	74,6	–
MinkLoc++ (3D) [38]	Oxford RC	[38]	72,6	–
Scan Context [25]	-	własne uruchomienie	75,0	71,4
MinkLoc3D [35]	USyd	własne uruchomienie	73,8	69,1
MinkLoc3D-SI (our)	Oxford RCI	własne uruchomienie	<b>81,0</b>	72,6
MinkLoc3D-SI (our)	USyd	własne uruchomienie	<b>81,0</b>	<b>78,6</b>

zacji, MinkLoc3D-SI przewyższa inne rozwiązania, niezależnie od tego, czy konkurencyjne podejścia zostały uczone na USyd czy też na Oxford RobotCar Intensity (tab. 5). Wybrana reprezentacja sferyczna z informacją o intensywności w MinkLoc3D-SI lepiej pasuje do chmur punktów 3D zarejestrowanych z pojedynczego skanu, co ma miejsce w przypadku zbioru danych KITTI.

Zaproponowany MinkLoc3D-SI został również zweryfikowany w różnych zmiennych warunkach pogodowych w sposób analogiczny do rozwiązań z [A4]. Uzyskane wyniki sugerują, że wydajność proponowanego MinkLoc3D-SI jest w większości niezależna od warunków pogodowych (tab. 6). MinkLoc3D-SI daje poprawę  $AR@1$  o 1,7 do 3,5 punktów procentowych w stosunku do MinkLoc3D dla wszystkich scenariuszy pogodowych, udowadniając, że reprezentacja sferyczna i informacja o intensywności są istotne we wszystkich przypadkach. W porównaniu z metodami w [A4], MinkLoc3D-SI przewyższa rozwiązania oparte na kamerach, LiDAR-ach oraz połączone kamery-LiDAR-y przedstawione w poprzedniej pracy.

Uzyskane wyniki sugerują, że współrzędne sferyczne z intensywnością dla punktów 3D są obiecującą modyfikacją do przetwarzania chmur punktów z obrotowego LiDAR-a 3D, a tym samym mogą być zastosowane w innych rozwiązaniach z architekturą rzadkich konwolucji 3D lub do innych zastosowań. Moja praca dowodzi również, że architektura sieci powinna odpowiadać fizycznym właściwościom pomiarów, aby uzyskać najlepsze rezultaty. W porównaniu z systemami rozpoznawania miejsc opartymi na kamerach, podejścia oparte na chmurach punktów są odporne na zmiany oświetlenia, uzupełniając informacje pochodzące z obrazów RGB.

Tabela 6: Wydajność MinkLoc3D-SI mierzona za pomocą  $AR@1$  w różnych warunkach pogodowych: słonecznie (S), pochmurno (C), słonecznie/pochmurno (S/C), po i w lekkim deszczu (AR), zachód słońca (SS), bardzo pochmurno (VC). **Imp.** przedstawia poprawę o punkt procentowy w stosunku do MinkLoc3D w tych samych warunkach pogodowych.

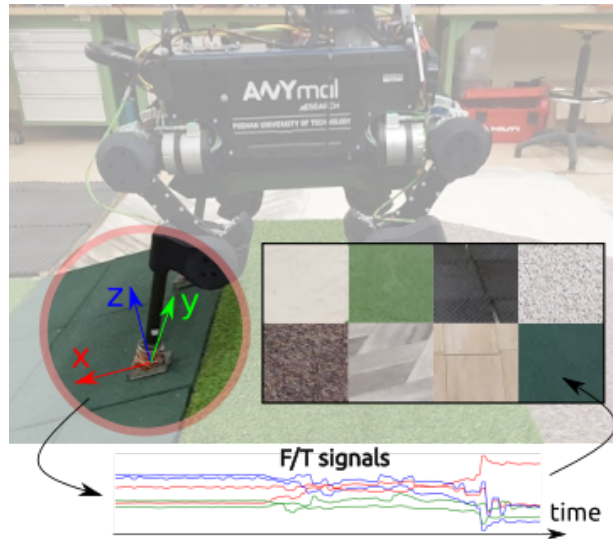
Ref. Q.	S	C	S/C	AR	SS	VC	Średnia	Imp.
S	95.1	93.1	96.0	95.0	93.4	96.0	94.6	+3.1
C	95.2	92.8	96.0	94.5	92.7	94.7	94.5	+2.6
S/C	96.5	94.7	96.6	95.6	94.9	96.4	95.9	+2.7
AR	95.3	93.1	95.9	94.8	94.3	95.5	94.8	+2.8
SS	94.0	91.8	94.6	95.3	94.9	96.2	93.9	+3.5
VC	95.7	93.0	95.6	94.7	94.8	–	94.9	+1.7
Średnia	95.2	93.1	95.9	94.9	93.6	95.7	94.7	+2.9

#### 4.6 Lokalizacja haptyczna robota kroczącego

Roboty kroczące są specjalnym rodzajem robotów mobilnych, które mają dostęp do dodatkowego źródła informacji - pomiarów haptycznych ze stóp podczas pokonywania terenu. Zazwyczaj roboty te wykorzystują kamery oraz LiDAR-y 3D, ale haptyka może być dodatkowym źródłem informacji w przypadku awarii głównych sensorów lub w trudnych środowiskach. Przykładem takiego środowiska są kopalnie, kiedy klasyczne podejścia mogą zawieść ze względu na trudne warunki m.in. z powodu zapylenia. Wyniki prac [A6,A7,A8] zostały uzyskane na robocie ANYMAL chodzącym nieprzerwanie po różnych rodzajach terenu w świecie rzeczywistym bez dodatkowych ruchów eksploracyjnych podczas testów. Wykorzystana mapa z ośmioma różnymi typami terenu została przedstawiona na rys. 17.

Podczas sesji chodzenia robot ANYMAL wyposażony był w sensoryczne stopy [69], które składają się z płaskich powierzchni kontaktowych z zakresem ruchu  $50^\circ$  dla osi *pitch* i  $30^\circ$  dla osi *roll*. Robot chodził po płaskiej powierzchni i rampie wymagającej dostosowania się stóp do rodzaju i kształtu terenu. Czujniki siły/momentu (F/T) umieszczone wewnątrz stóp zostały wykonane na zamówienie i mogą wyczuwać do 1000 N w kierunku Z (wzdłuż nogi robota), 400 N w kierunkach na powierzchni ziemi oraz do 10 Nm momentu obrotowego w każdej osi z częstotliwością 400 Hz. Sygnały F/T ograniczyliśmy do 160 próbek zarejestrowanych w chwili kontaktu. W eksperymentach wykorzystano osiem różnych terenów: dywan, sztuczną trawę, gumę, piasek, piankę, skały, płytki ceramiczne i PCV. Można zaobserwować, że stopa adaptacyjnie nachyla się w różny sposób w zależności od rodzaju terenu, jego właściwości i kształtu. Aby zapewnić stabilność ruchu, robot poruszał się z użyciem statycznie stabilnego chodu z tylko jedną nogą w fazie ruchu w danym momencie. Po fazie ruchu wszystkie stopy są stawiane na ziemi. Następnie faza ruchu rozpoczyna się od nowa z inną nogą.

Biorąc pod uwagę robota czteronożnego z 12 aktywnymi stopniami swobody (ang. *degree of freedom* - DoF), chcemy zlokalizować bazę robota względem wcześniejszej mapy, biorąc



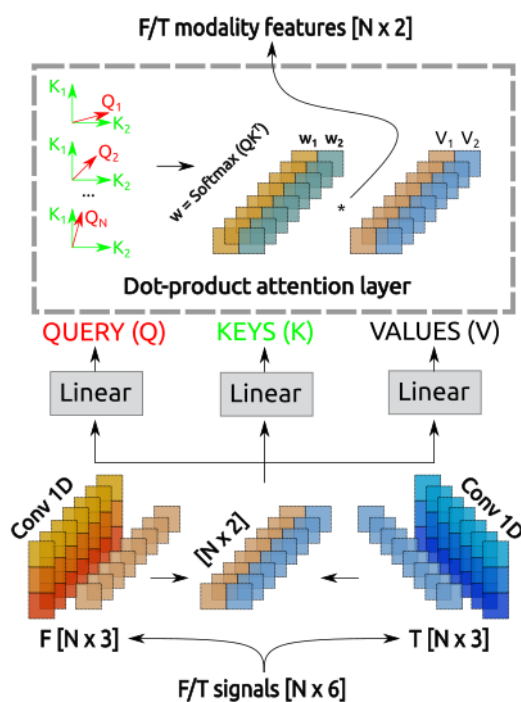
Rysunek 17: Robot ANYMAL zbierający haptyczne sygnały siły/momentu obrotowego (F/T) za pomocą swoich stóp podczas chodzenia po różnych terenach. Klasyfikacja terenu jest kluczowa dla adaptacji chodu w celu zapewnienia stabilności robota i musi być wykonana w krótkim czasie na jednostce obliczeniowej o ograniczonych zasobach. Dlatego proponujemy oparte na transformerze metody HAPTR i HAPTR2, aby spełnić te wymagania.

pod uwagę źródło odometrii oraz sygnały siły/momentu obrotowego (F/T) mierzone w stopach po kontakcie z podłożem. Dodatkowo zakłada się, że wcześniejsza mapa została zbudowana przy użyciu systemu lokalizacji pozy robota z małym dryfem lub bez dryfu błędu, na przykład przy użyciu LiDAR-u 3D w czasie, gdy jest minimalne zapylenie. Do naszych eksperymentów używamy robota ANYMAL B300 [33] wyposażonego w wersję sensorycznych stóp [69] (bez IMU) oraz estymator odometrii nożnej TSIF [8]. Ponieważ możliwe oszacowanie lokalizacji zależy od kształtu mapy, rozkład prawdopodobieństwa estymowanej pozy robota może nie być gaussowski.

Jednym ze sposobów wydobywania informacji z sygnałów haptycznych jest wykonanie klasyfikacji terenu, aby móc wykonać działanie lub wykorzystać identyfikację terenu do lokalizacji [11]. Klasyfikacja terenu jest dobrze zbadanym tematem, gdzie większość rozwiązań skupia się na sygnałach siły i momentu obrotowego z czujników zamontowanych na stopach robota [1, 5, 6, 34]. Metody te skupiają się na uzyskaniu jak najlepszych wyników mierzonych jako dokładność na zarejestrowanym zbiorze danych. W [A6] uważamy, że dokładność jest tylko częścią całego obrazu. Projektowana metoda musi nie tylko osiągnąć zadowalającą klasyfikację terenu, ale musi umożliwiać szybką inferencję, aby zapewnić dynamiczne działanie robota krocącego. Ponadto musi być wykonana przy ograniczonych zasobach przetwarzania, ponieważ autonomiczny robot kroczący posiada wiele modułów działających jednocześnie, aby osiągnąć pożądaną odporność na wyzwania świata rzeczywistego. Aby spełnić te wymagania, w [A6] zaproponowaliśmy HAPTR i HAPTR2, które

są wydajnymi metodami głębokiego uczenia opartymi na architekturze transformera. W celu określenia, czy proponowane rozwiązanie dobrze nadaje się do rzeczywistego wdrożenia, zaproponowałem nowe metody oceny ukierunkowane na wcześniej pomijane aspekty, takie jak czas wnioskowania i odporność.

HAPTR2 wykorzystuje mechanizm uwagi (ang. *attention*) zamiast konwolucji do klasyfikacji terenu. Podobnie do Vision Transformer (ViT) używanego w [20], HAPTR2 używa uczącej się warstwy projekcji liniowej do mapowania sygnału 6D (np. siły i momentu obrotowego w trzech osiach) na sekwencję o tej samej długości w 16-wymiarowej przestrzeni cech. Każdą próbkę tej sekwencji określam jako tzw. patch. Następnie do każdego patcha dodawane jest kodowanie pozycyjne (ang. *positional encoding* - PE) w celu zachowania informacji o położeniu i przekazywane jest to do warstwy Transformer Encoder Layer. Na koniec obliczamy średnią każdego wektora, aby zmniejszyć wymiarowość przed ostateczną klasyfikacją za pomocą warstwy feed-forward.



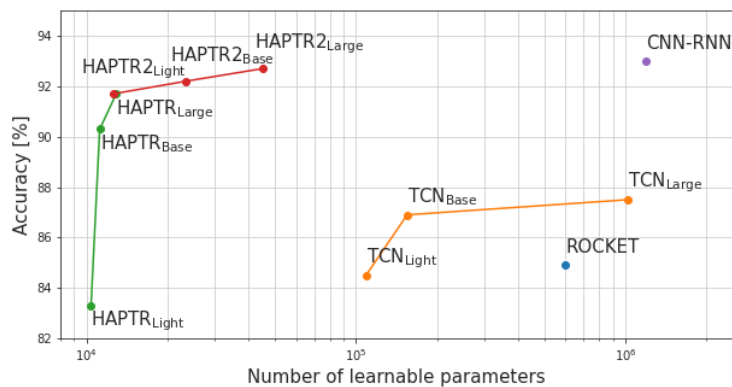
Rysunek 18: Wizualne wyjaśnienie warstwy Modality Attention Layer użytej w naszych eksperymentach. Użyliśmy jej do zwiększenia skutecznego działania modelu HAPTR.

Główną nowością proponowanego rozwiązania jest warstwa *Modality Attention Layer* (MAL) oparta na mechanizmie Multi-Head Attention wprowadzonym w [70]. Najpierw, wejściowe szeregi czasowe są dzielone według modalności (tj. sygnały siły i momentu obrotowego mierzone na stopach robota są przetwarzane oddzielnie) i przekazywane do warstw konwolucyjnych 1D. Te z kolei przekształcają wielowymiarowe sygnały w ich

spłaszczone reprezentacje modalności o tej samej długości co wejścia. Uczone warstwy liniowe przetwarzają i kształtują każdą reprezentację modalności na tzw. zapytania (Q), klucze (K) i wartości (V) dla warstwy uwagi iloczynu punktowego. Zapytań jest tyle, ile próbek wejściowych. Każda próbka jest ważona pomiędzy wszystkimi modalnościami i skalowana przez współczynnik  $(1/\sqrt{d_k})$ , gdzie  $d_k$  jest wymiarowością zwielokrotnionych zapytań i kluczy. Aktywacja warstwy MAL opisana jest równaniem:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (16)$$

Klucze składają się z macierzy o rozmiarze  $d_k \times d_k$ , gdzie wiersze reprezentują modalność  $d$ -tą z  $d = 1, 2, \dots, d_k$ . Dlatego im bliżej zapytania znajduje się odpowiadający mu klucz, tym większa waga jest związana z tą modalnością. Na koniec na skalowanym iloczynie skalarnym stosuje się warstwę softmax, aby uzyskać rozkład prawdopodobieństwa. Rysunek 18 przedstawia zasadę działania MAL.

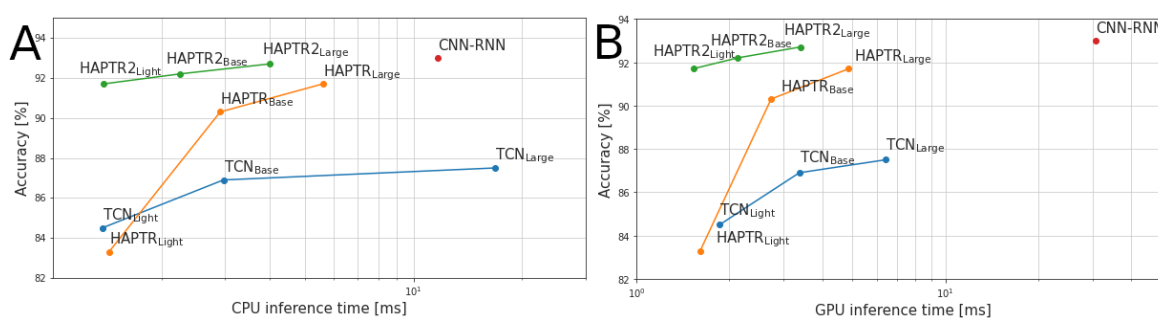


Rysunek 19: Dokładność rozpoznawania klas terenu w zależności od liczby parametrów sieci neuronowej ujawnia skuteczność zastosowanej metody. Zauważmy, że oś parametrów jest w skali logarytmicznej. W tym kontekście HAPTR2 znacząco przewyższa inne algorytmy.

Najlepsza wydajność na wybranym zbiorze danych jest zwykle uzyskiwana przy użyciu najbardziej rozbudowanych sieci, które często są zbyt duże, aby je uczyć lub wykorzystać w jakimkolwiek praktycznym scenariuszu, przez co są traktowane jako rozwiązanie przeznaczone dla danego zbioru danych. W związku z tym zaproponowałem pomiar dokładności sieci w funkcji liczby możliwych do nauczania parametrów, co zostało przedstawione na rys. 19. Zauważyłem, że sieć o najlepszej dokładności (CNN-RNN) nie pasowałaby do ograniczonych zasobów obliczeniowych konfiguracji robota mobilnego. Proponowany HAPTR2 potrzebuje o rząd wielkości mniej parametrów sieci neuronowej, aby osiągnąć wydajność podobną do CNN-RNN. W związku z tym, zdecydowanie opowiadam się za stosowaniem rodziny rozwiązań opartych na transformerze, kiedy tylko wydajność metod klasyfikacji jest kluczowym wymaganiem.



Liczba możliwych do nauczenia się parametrów określa zdolność do wdrażania sieci w środowiskach o ograniczonych zasobach. Mimo to, właściwe porównanie sieci może być trudne, ponieważ zależy od wybranej konfiguracji sprzętowej. Rozmiar sieci ma również wpływ na czas uczenia, który możemy uznać za nieistotny, gdyż sieci były uczone offline. Pozostałym aspektem wybranego rozmiaru modelu jest czas wnioskowania. Czas wnioskowania określa realną przydatność proponowanego rozwiązania, ponieważ posiadanie wydłużonego czasu przetwarzania może być nie do zaakceptowania z perspektywy kontroli, która została przeprowadzona w oparciu o wynik sieci neuronowej. Dlatego też zebraliśmy czasy przetwarzania na CPU (Intel i7-9750H @ 2.600GHz) oraz GPU (NVIDIA GeForce GTX 1660 Ti Mobile) dla wszystkich rozważanych metod, aby określić czy prezentowane rozwiązania mogłyby być wykorzystane na prawdziwym robocie.



Rysunek 20: Dokładność modeli głębokiego uczenia w funkcji średniego czasu wnioskowania na CPU (po lewej) i GPU (po prawej). Zwracam uwagę, że oś czasu wnioskowania jest w skali logarytmicznej.

Spśród testowanych rozwiązań (rys. 20), najkrótszy czas wnioskowania CPU został odnotowany dla HAPTR2 i spełniał praktyczny wymóg uzyskania wyników w ciągu 10 ms, aby zdążyć zareagować na czas w przypadku konieczności zmiany typu chodu lub adaptacji postury robota. W tym porównaniu rozwiązanie CNN-RNN nie spełniło naszych kryteriów, przekraczając ustalony próg czasu wnioskowania, przez co nie nadaje się do pracy w czasie rzeczywistym. Co zaskakujące, w większości przypadków nie zaobserwowaliśmy przyspieszenia w przypadku wykorzystania GPU. W naszych eksperymentach zdefiniowaliśmy czas wnioskowania jako całkowity czas potrzebny na obliczenia i powiązane zadania. Zawiera on czas przesyłania danych pomiędzy pamięcią główną a pamięcią GPU, a ten narzut jest obecny we wszystkich pomiarach czasu wnioskowania uwzględnionych w porównaniu. Po raz kolejny HAPTR2 okazał się najbardziej wydajną metodą w oparciu o dokładność i czas wnioskowania, gdy dostępne jest GPU.

Roboty mobilne działają w różnych warunkach, które mogą wpływać na pomiary sensoryczne. W większości przypadków działanie w nowym środowisku prowadzi do uzyskania gorszych wyników niż w przypadku pracy na danych zbliżonych do danych uczących. Celem jest zaprojektowanie systemu, który może działać skutecznie także w nieprzewidzianych warunkach. Sprawdziliśmy działanie naszej sieci w dwóch symulowanych scenariuszach.



W pierwszym zmieniliśmy ładunek robota, co wpłynęło na rozkład sił i momentów obrotowych. W drugim symulowaliśmy awarię czujników, która może wystąpić, gdy robot mobilny przemierza trudny teren. Naśladowaliśmy te przypadki poprzez dodanie określonego rodzaju szumu do już znormalizowanego wejścia pomiarowego, które zostało określone jako mające zerową średnią i jednostkowe odchylenie standardowe. Zauważyliśmy znaczną poprawę skuteczności modelu dla obu modalności wejściowych, osiągając ponad 10-procentową poprawę dokładności dla najwyższych poziomów szumu, gdy użyto MAL. Ponadto, HAPTR jest bardziej odporny na zmiany w pomiarach siły, udowadniając, że pomiary momentu obrotowego mogą mieć większy wpływ na końcową wydajność sieci.

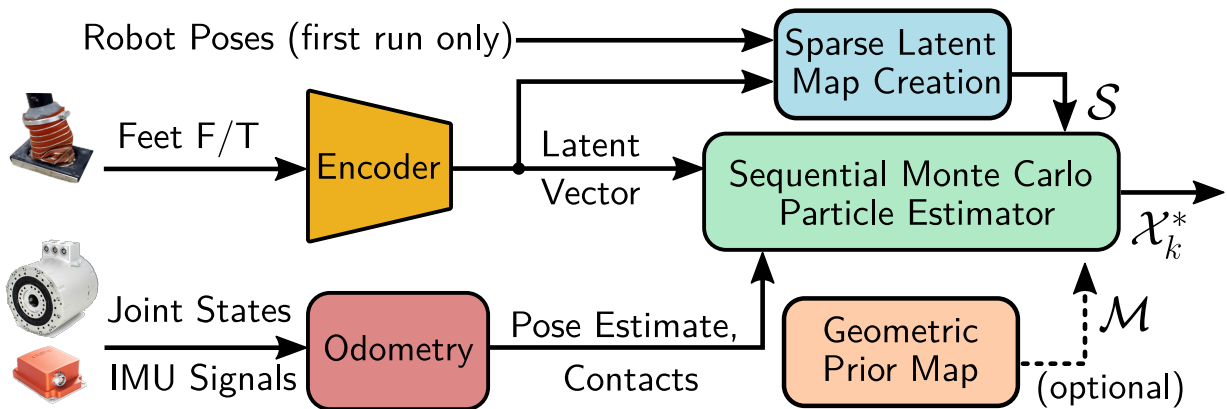
Algorytmy klasyfikacji terenu dla robotów kroczących są w większości nieporównywalne ze względu na różne rodzaje wykorzystywanego terenu oraz różne platformy sprzętowe. Zmienia się to jednak dzięki pojawieniu się publicznych zbiorów danych, które ułatwiają obiektywne porównanie metod. Postanowiliśmy porównać nasze wyniki z najnowszą metodą RNNs+FCL [1]. W tym artykule autorzy oceniali metodę RNNs+FCL na zbiorze danych PUTany oraz na swoim zbiorze danych QCAT (który został udostępniony publicznie). Aby zapewnić właściwe porównanie, oceniliśmy naszą metodę HAPTR2<sub>Light</sub> przy użyciu tej samej procedury oceny na obu tych zbiorach danych, stosując takie same kroki walidacji krzyżowej z tymi samymi podziałami danych. Uzyskane wyniki zostały przedstawione w tabeli. 7. Nasz HAPTR2<sub>Light</sub> przewyższył RNNs+FCL na obu zbiorach danych z marginesem dokładności 0,64% dla zbioru PUTany i 0,73% dla zbioru QCAT.

	Dataset	Średnia [%]	SD [%]	Min [%]	Max [%]
HAPTR2 <sub>Light</sub>	PUTany	<b>93,85</b>	0,82	92,68	95,29
RNNs+FCL [1]		93,20	0,89	92,06	95,39
HAPTR2 <sub>Light</sub>	QCAT	<b>97,33</b>	1,21	95,49	98,96
RNNs+FCL [1]		96,60	0,89	95,49	98,61

Tabela 7: Dokładność klasyfikacji proponowanego HAPTR2<sub>Light</sub> i najnowocześniejszych RNNs+FCL mierzona na zbiorach danych QCAT i PUTany przy 10-krotnej walidacji krzyżowej, z podaniem średniej, odchylenia standardowego (SD) oraz wartości min i max dla każdego podzbioru. Najlepsze wyniki są pogrubione.

Pomimo rosnącej dokładności metod klasyfikacji terenu, informacja o klasyfikacji terenu może nie być najlepszym sposobem wykorzystania informacji haptycznej w systemie lokalizacji. Klasy terenu są zazwyczaj wybierane na podstawie ludzkiej percepcji terenu, która może nie odzwierciedlać różnic we właściwościach haptycznych. Co więcej, zwykle nie możemy założyć, że na danym obszarze jednego typu terenu uzyskamy jednorodną i powtarzalną odpowiedź taktylną. Dlatego w [A7] skupiliśmy się na znalezieniu sposobu wykorzystania informacji haptycznej bez ograniczania się do listy predefiniowanych klas.

Ogólne podejście do proponowanego przez mnie systemu SMC (ang. *Sequential Monte Carlo*) opartego na nienadzorowanej reprezentacji haptycznej przedstawiono na rys. 21. Na początku enkoder określa reprezentację terenu dla każdego kroku. Obliczone kodowa-

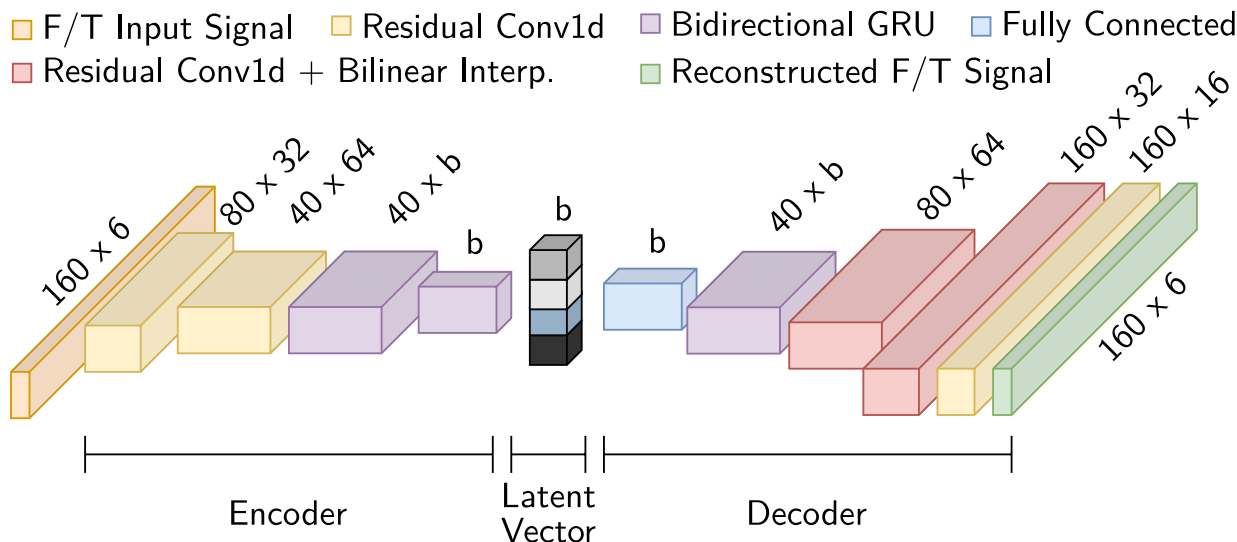


Rysunek 21: Schemat systemu proponowanej przez nas lokalizacji haptycznej Monte Carlo. Odometria pochodząca z robota jest łączona z wektorem reprezentacji z enkodera przy użyciu rzadkiej mapy reprezentacji zbudowanej przed działaniem systemu lokalizacji. Opcjonalnie, dla zwiększenia dokładności, może być użyta również mapa geometryczna.

nie każdego kroku jest łączone z dokładnym źródłem póż robota (z systemu SLAM) w celu zbudowania rzadkiej, haptycznej mapy terenu. Po zbudowaniu mapy lokalizacja jest wykonywana za pomocą predefiniowanej liczby cząstek, w regularnych odstępach czasu podczas ruchu. W każdym nowym kroku estymator SMC aktualizuje stan swoich cząstek wykorzystując odometrię z robota, aktualne sygnały dotykowe oraz opcjonalnie geometrię terenu.

Ponieważ rozproszona mapa reprezentacji terenu zawiera kroki z poprzednich przejść po tej samej trasie, rozmiar mapy naturalnie rośnie liniowo wraz z długością trajektorii. Zakładając systematyczną kontrolę na odcinku 1 km z krokami stawianymi co 10 cm, przybliżony rozmiar mapy dla nienadzorowanych, nieskompresowanych sygnałów może przekroczyć 200 MB. Dlatego poszukujemy kodowania, które wygeneruje zredukowany opis odpowiedzi haptycznej o rozmiarze  $b$  zwany wektorem reprezentacji (ang. *latent vector*). Osiągnięcie tego celu umożliwiłoby zmniejszenie rozmiaru mapy przy zachowaniu właściwości oryginalnych sygnałów, a także zwiększenie generalizacji dla terenów podobnych do wcześniej zmierzonych.

Do kodowania zaproponowaliśmy architekturę zwaną Haptic AutoEncoder (HAE), składającą się z trzech typów bloków: konwolucyjnych, rekurencyjnych i w pełni połączonych, jak pokazano na rys.22. Ta kombinacja bloków, zainspirowana przez nadzorowaną sieć klasyfikacji terenu z pracy [12], wydobywa wysokopoziomową reprezentację cech sygnałów szeregu czasowego i może przetwarzać dane o zmiennej długości. Model HAE wykorzystuje dwie warstwy *residual* z konwolucjami 1D w module CNN oraz dwie dwukierunkowe warstwy z dwoma komórkami GRU w komponencie rekurencyjnym kodera. Część konwolucyjna kodera jest odpowiedzialna za ekstrakcję cech sygnałów i redukcję wymiaru przestrzennego sekwencji przy użyciu stride'u 2. Następnie, w przeciwieństwie do prostego konwolucyjnego



Rysunek 22: Do generowania wektora reprezentacji o wymiarze  $b$  wykorzystano schemat architektury Haptic AutoEncoder (HAE). Głównymi blokami sieci są warstwy konwulcyjne (CNN), dwukierunkowe warstwy GRU (Bidir GRU) oraz warstwa w pełni połączona (FC).

AutoEncodera, dwukierunkowe warstwy GRU są wykorzystywane do uchwycenia globalnych i lokalnych zależności pomiędzy cechami. Na koniec blok RNN produkuje wektor reprezentacji o wymiarze  $b$ .

Porównaliśmy naszą metodę z podejściami przedstawionymi w [12]: Haptic Localization (HL) wykonywanym z geometrią otoczenia wykorzystującym gęstą, znaną mapę geometryczną (HL-G), z nadzorowanymi klasyfikacjami terenu (HL-C) lub z obydwoma tymi źródłami informacji (HL-GC). Pokazujemy również wyniki lokalizacji z proprioceptywnej odometrii robota zwanej TSIF [8]. Dokładność lokalizacji uzyskana w ciągu trzech prób wykorzystanych w [12] jest przedstawiona w tab. 8 jako bezwzględny błąd położenia (ang. *Absolute Pose Error* - APE).

Wyniki te potwierdzają, że przy zastosowaniu nienadzorowanych sygnałów haptycznych z informacją geometryczną (HL-GU), dokładność jest większa niż w przypadku metod wykorzystujących tylko geometrię (HL-G). W porównaniu z nadzorowaną metodą klasyfikacji terenu z geometrią (HL-GC), wykorzystanie nienadzorowanych sygnałów F/T (HL-GU) daje porównywalne wyniki, osiągając największą dokładność dla próby 3. Pokazuje to, że nienadzorowane sygnały F/T mogą uchwycić różnice pomiędzy sygnałami uzyskanymi dla różnych klas.

W niektórych przypadkach stworzenie wcześniejszej mapy geometrycznej zawierającej informację w każdym punkcie mapy może być niewykonalne. Używając informacji z tych samych przejść robota, co w poprzedniej sekcji, zakładam, że tylko rzadka mapa repre-

Absolute Pose Error (APE) $\mu$ [m]				
Sekwencja	TSIF [8, 12]	HL-G [12]	HL-GC [12]	HL-GU
1	0,64	0,23	<b>0,14</b>	0,15
2	1,28	0,25	<b>0,11</b>	0,18
3	0,72	0,21	0,18	<b>0,13</b>

Tabela 8: Porównanie wydajności lokalizacji haptycznej w przypadku wykorzystania wcześniejszej geometrii mapy. Wykorzystanie nienadzorowanych sygnałów F/T (HL-GU) daje porównywalną poprawę lokalizacji, co zastosowanie nadzorowanej klasyfikacji terenu (HL-GC).

zentacji sygnałów haptycznych jest dostępna dla celów lokalizacji. Wyniki uzyskane bez sygnałów F/T (TSIF), z nadzorowaną klasyfikacją terenu (HL-C) oraz bez nadzoru, z kompresją sygnałów haptycznych (HL-U) zostały przedstawione w tab. 9.

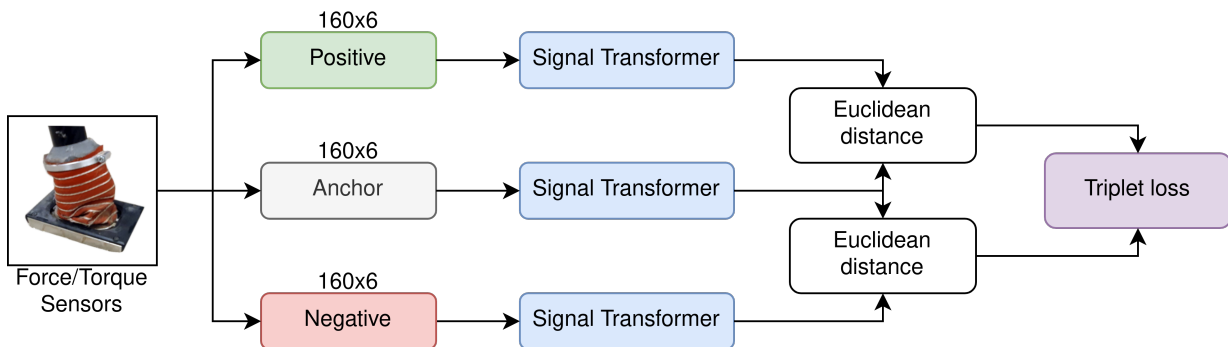
Absolute Pose Error (APE) $\mu$ [m]			
Sekwencja	TSIF [8, 12]	HL-C [12]	HL-U
1	0,64	0,63	<b>0,47</b>
2	1,28	0,73	<b>0,57</b>
3	0,72	0,61	<b>0,5</b>

Tabela 9: Porównanie dokładności lokalizacji uzyskanej za pomocą haptycznej lokalizacji SMC wykorzystującej wewnętrzne sensory ANYmal (TSIF) w połączeniu z klasyfikacją terenu (HL-C) lub z proponowaną, nienadzorowaną reprezentacją terenu (HL-U).

Metoda oparta na nienadzorowanych sygnałach haptycznych (HL-U) przewyższa metodę wykorzystującą nadzorowaną klasyfikację terenu (HL-C) we wszystkich rozpatrywanych próbach, nie wymagając jednocześnie wcześniejszej, gęstej mapy otoczenia. Uzyskana dokładność lokalizacji jest również wyraźnie lepsza w porównaniu z domyślnym estymatorem stanu TSIF. Zastosowanie rzadkiej mapy reprezentacji informacji haptycznej umożliwia lepszą skalowalność, np. dla tuneli podziemnych.

Głównym problemem metody zaproponowanej w [A7] było to, że nie miałem kontroli nad nauką nienadzorowanej reprezentacji informacji haptycznej. Dlatego w [A8] zaproponowałem sieć do określenia reprezentacji, która koduje istotne cechy lokalizacyjne z surowego wejścia haptycznego. Sieć nazywa się Signal Transformer i bazuje na oryginalnej architekturze Transformera z [70]. 6-wymiarowe wejście czujników (z 3-osioowych czujników siły i 3-osioowych czujników momentu) dla 160 kolejnych pomiarów jest przekształcane w 16-wymiarową przestrzeń cech z w pełni połączoną warstwą. Stosujemy normalizację warstw i wzbogacamy sekwencję o możliwe do nauczenia się kodowanie pozycyjne podobne do [A6]. Rozszerzone dane są przekazywane do enkodera referencyjnej architektury transformera

z [70], przy  $h = 2$  głowach uwagi, wymiarowości modelu  $d_m = 16$  i rozmiarze wewnętrznej warstwy feed-forward  $d_{ff} = 8$ . Zastosowanie uśredniania (ang. *average pooling*) zmniejsza rozmiar wyjścia z enkodera. Ostateczna reprezentacja generowana jest poprzez użycie *batch normalization* i podanie znormalizowanych danych do gęstej warstwy feed-forward z funkcją aktywacji ReLU. Ostatnia warstwa ma liczbę neuronów równą długości reprezentacji, która domyślnie jest ustawiona na 256, tak jak w [A7].



Rysunek 23: Sieć do nauki reprezentacji sygnałów jest uczona przy użyciu trójki próbek danych, aby osiągnąć pożądane podobieństwo reprezentacji dla wybranej próbki (ang. *anchor*) i pozytywnej próbki (ang. *positive sample*), przy jednoczesnym zwiększeniu różnicy dla wybranej próbki i negatywnej próbki (ang. *negative sample*).

Główną różnicą podejścia z [A8] względem [A7] jest to, że sieć jest uczona z błędem trójkowym (ang. *triplet loss*) przedstawionym na rys. 23. Proces nauki minimalizuje różnicę między reprezentacjami dla kroków, które są blisko siebie, podczas gdy maksymalizuje różnicę między reprezentacjami, jeśli kroki zostały zarejestrowane daleko od siebie. Podejście uczące z potrójną stratą jest nowatorskie w uczeniu lokalnych reprezentacji, ponieważ wcześniej było z powodzeniem stosowane w globalnych problemach lokalizacyjnych, takich jak przedstawiono w [A4] i [A5]. W celu zwiększenia efektywności stosujemy technikę *online triplet mining*, aby zmniejszyć liczbę nieaktywnych trójek podczas nauki oraz poprawić zbieżność i szybkość nauki. Tworzymy minipróbki losowo, przy czym każdy przykład jest traktowany jako wybrana próbka (ang. *anchor*) podczas uczenia. Dla każdej próbki uczącej potrzebujemy informacji o powiązanej lokalizacji. Pozytywne przykłady dla wybranej próbki to te, które są bliżej wybranej próbki niż zdefiniowany stały próg odległości  $d_{thr}$ . W związku z tym, negatywne przykłady były próbkowane dalej niż  $d_{thr}$ :

$$\begin{cases} d(\mathbf{s}_{b_a}, \mathbf{s}_{b_i}) > d_{thr} \rightarrow b_i \in N_a \\ d(\mathbf{s}_{b_a}, \mathbf{s}_{b_i}) \leq d_{thr} \rightarrow b_i \in P_a, \end{cases} \quad (17)$$

gdzie  $d(\mathbf{s}_{b_a}, \mathbf{s}_{b_i})$  to odległość euklidesowa między pozycją kroku wybranej próbki  $\mathbf{s}_{b_a}$  a pozycją kroku  $\mathbf{s}_{b_i}$  dla  $i$ -tej próbki danych  $b_i$ .  $P_a$  i  $N_a$  oznaczają zbiór pozytywnych i negatywnych kroków dla wybranej próbki  $a$ . Proóg odległości  $d_{thr}$  jest parametrem, który można

regulować. Użyliśmy  $d_{\text{thr}} = 25$  cm, ponieważ zapewniał on najlepszą dokładność lokalizacji. Podczas uczenia pozytywne i negatywne przykłady zależą ściśle od przestrzennych zależności pomiędzy pozycjami kroków bez użycia żadnych etykiet klas terenu.

Zainspirowany przez [29], w ramach nauki używamy strategię nauki znaną jako *Batch All triplet loss*, bez specjalnego podejścia do wyboru próbek w minizbiorach uczących i obliczamy go jako:

$$\mathcal{L} = \sum_{a=1}^B \sum_{p=1}^{|P_a|} \sum_{n=1}^{|N_a|} [d(f(b_a), f(b_p)) - d(f(b_a), f(b_n)) + m]_+, \quad (18)$$

gdzie  $B$  jest rozmiarem minizbioru,  $m$  oznacza margines, a  $|\cdot|$  kardynalność zbiorów. Funkcja odległości  $d(\cdot)$  implementuje odległość euklidesową. Średnia strata trójek jest obliczana z uwzględnieniem tylko tych trójek, które mają niezerowy błąd, jak w [29]. Proces uczenia został przeprowadzony przy użyciu optymalizatora AdamW [44]. Szybkość uczenia była wykładniczo malejąca z wartością początkową  $5 \times 10^{-4}$ . Początkowa wartość rozkładu wag była równa  $2 \times 10^{-4}$  i była zmniejszana z rozkładem cosinusowym. Rozmiar minizbioru został ustawiony na 128. Uczenie trwało przez 200 epok.

Sekwencja	TSIF [8]	HL-C [12]	HL-U [A7]	HL-T
	$t_{2D}$	$t_{2D}$	$t_{2D}$	$t_{2D}$
1	0,34	0,39	0,17	<b>0,07</b>
2	0,92	0,22	0,14	<b>0,06</b>
3	0,51	0,29	0,18	<b>0,08</b>

Tabela 10: Porównanie błędów bezwzględnych 2D (APE, w [m]) dla lokalizacji z wykorzystaniem tylko informacji haptycznej. Nowa metoda HL-T osiągnęła najniższy błąd na wszystkich sekwencjach.

Nowa metoda z nienadzorowanymi reprezentacjami informacji haptycznej HL-T została porównana z wcześniej uzyskanymi wynikami w [A7]. Podsumowanie przedstawionych wyników można znaleźć w tabeli 10. Najgorsze wyniki uzyskuje TSIF, estymator odometrii nogi, który jest zarówno rozwiązaniem podstawowym, jak i składnikiem metody lokalizacji w pozostałych systemach. Wśród porównywanych rozwiązań proponowany HL-T przewyższa poprzednie podejścia z dużym marginesem, zmniejszając wartości błędu APE o prawie 50%. Sądźmy, że wynika to z faktu, że nasza metoda nie jest zawężona do ograniczonej liczby klas dyskretnych jak HL-C. Natomiast HL-U może nauczyć się wewnętrznej reprezentacji sygnałów haptycznych, która wspiera uczenie reprezentacji mogących rozróżnić poszczególne odpowiedzi haptyczne dla wybranych miejsc w mapie. Co ważne, błąd lokalizacji nie przekroczył 10 cm pomimo braku innych modalności sensorycznych, co może być wystarczające do kontynuowania pracy autonomicznego robota, pomimo awarii sensorów wizyjnych lub nagłej zmiany warunków otoczenia.

Rozważmy jeszcze scenariusz, w którym dokładna mapa 2,5D środowiska jest dostępna do celów lokalizacji. Dla takich scenariuszy rozważamy HL-G [11] wykorzystujący gęstą mapę

wysokości terenu do korekcji pozy, HL-GC [12] wykorzystujący zarówno geometrię, jak i klasyfikację terenu, oraz HL-GU [A7], który wykorzystuje geometrię i nienadzorowaną haptyczną reprezentację. W tych eksperymentach nasza metoda HL-GT jest skonfigurowana tak, aby używać gęstej mapy wysokości i rzadkiej mapy reprezentacji informacji haptycznej. Wyniki estymatora odometrii nogi TSIF [8] zostały pominięte, ponieważ zostało już udowodnione, że HL-G, HL-GC i HL-GU przewyższają go w tych próbach. Wyniki dla obu rodzajów błędów ( $\mathbf{t}_{2D}$ ,  $\mathbf{t}_{3D}$ ) zostały przedstawione w tab. 11.

Sekwencja	HL-G [12]		HL-GC [12]		HL-GU [A7]		HL-GT	
	$\mathbf{t}_{3D}$	$\mathbf{t}_{2D}$	$\mathbf{t}_{3D}$	$\mathbf{t}_{2D}$	$\mathbf{t}_{3D}$	$\mathbf{t}_{2D}$	$\mathbf{t}_{3D}$	$\mathbf{t}_{2D}$
1	0,23	0,23	0,14	0,12	0,15	0,09	<b>0,09</b>	<b>0,08</b>
2	0,25	0,20	<b>0,11</b>	0,11	0,18	0,12	<b>0,11</b>	<b>0,10</b>
3	0,21	0,18	0,18	0,17	0,13	0,13	<b>0,09</b>	<b>0,09</b>

Tabela 11: Porównanie błędu bezwzględnego (APE, w [m]) 3D i 2D dla rozwiązań lokalizacyjnych wykorzystujących zarówno wcześniejszą gęstą mapę geometryczną, jak i haptyczne rozwiązania rozpoznawania terenu. Proponowany HL-GT zapewnia najlepsze wyniki przy użyciu zarówno metryki błędu  $\mathbf{t}_{3D}$  jak i  $\mathbf{t}_{2D}$ .

Uzyskane wyniki dla wszystkich rozważanych metod pokazują, że błędy  $\mathbf{t}_{3D}$  i  $\mathbf{t}_{2D}$  prawie się pokrywają, dowodząc, że nie ma znaczącego dryfu w kierunku wysokości ze względu na dostępność gęstej mapy wysokości. Proponowany HL-GT uzyskuje największą dokładność lokalizacji, co sugeruje, że reprezentacja uczona ze stratą potrójną bardziej nadaje się do rozróżniania lokalizacji terenu niż klasyfikacja terenu (HL-GC) lub reprezentacja terenu poprzez kompresję sygnału (HL-GU). Ponadto kodowanie sygnałów haptycznych w HL-GT jest komplementarne do gęstej mapy wysokościowej, ponieważ metoda ta poprawia wydajność podejścia geometrycznego (HL-G).

Jedną z zalet proponowanego rozwiązania jest możliwość wykorzystania informacji o wysokości, nawet jeśli do uczenia dostępna była tylko informacja o lokalizacji. To usprawnienie znacząco wpływa na praktyczność rozwiązania, ponieważ do wykorzystania danych o wysokościach nie jest wymagany LiDAR klasy geodezyjnej. Dlatego zdecydowaliśmy się na porównanie trzech rozwiązań: HL-T, które wykorzystuje do lokalizacji wyłącznie sygnały haptyczne, HL-GT, które wykorzystuje gęstą mapę geometryczną i sygnały haptyczne oraz HL-ST, które wykorzystuje rzadką mapę geometryczną i rzadką mapę reprezentacji informacji haptycznej. Wyniki zostały przedstawione w tab. 12.

Wyniki pokazują, że podejście HL-T zapewnia najlepsze wyniki w 2D. Jednak błąd 3D rośnie z czasem w związku z ogólnym dryfem odometrii nogi, co czyni go niepraktycznym dla jakiegokolwiek autonomicznej operacji. Z drugiej strony, HL-GT zapewnia najdokładniejszą lokalizację 3D dzięki gęstej mapie geometrycznej. Proponowany HL-ST jest dobrym kompromisem pomiędzy tymi podejściami, ponieważ błędy 2D i 3D są porównywalne z HL-T i HL-GT, przy jednoczesnym wykorzystaniu haptyki i jedynie lokalizacji podczas etapu



Sekwencja	HL-T		HL-GT		HL-ST	
	$t_{3D}$	$t_{2D}$	$t_{3D}$	$t_{2D}$	$t_{3D}$	$t_{2D}$
1	0,51	<b>0,07</b>	<b>0,09</b>	0,08	<b>0,09</b>	0,09
2	0,77	<b>0,06</b>	<b>0,11</b>	0,10	<b>0,11</b>	0,11
3	0,44	<b>0,08</b>	<b>0,09</b>	0,09	0,10	0,09

Tabela 12: Porównanie bezwzględnego błędu pozycyjnego (APE, w [m]) w 3D i 2D dla lokalizacji bez geometrii (HL-T), z gęstą mapą geometryczną (HL-GT) i z rzadką mapą geometryczną (HL-ST). HL-ST działa podobnie do HL-GT w 2D i 3D bez żmudnej fazy mapowania.

zbierania mapy. Uważam, że HL-ST jest zatem unikatowym rozwiązaniem, które może wspierać autonomię robotów koczujących w trudnych, rzeczywistych zastosowaniach.

#### 4.7 Rozszerzenia i uogólnienia proponowanych metod

W tym rozdziale prezentuję moje publikacje ściśle związane z problemami dotyczącymi osiągnięcia naukowego, które ukazały się po doktoracie. Prezentowane publikacje nie są częścią osiągnięcia naukowego, ale prezentują szersze spojrzenie na zaproponowane rozwiązania.

Problem dokowania autobusów przedstawiony jako problem lokalizacyjny był badany w większej liczbie moich prac naukowych. W [48] zajmowaliśmy się wytłumaczeniem, w jaki sposób nasza sieć neuronowa określa lokalizację ładowarki elektrycznej, dlatego skupiliśmy się na odpowiednim wygenerowaniu map ciepła wskazujących na uwagę sieci. Takie podejście zapewnia zbadanie niepożądanych, fałszywie pozytywnych detekcji, umożliwiając powiększenie zbioru danych uczących o próbki, które powinny poprawić wydajność systemu.

Możliwość praktycznego wykorzystania systemu dokującego, również z DGPS, została opisana w [46]. System został przedstawiony jako kompletne rozwiązanie zawierające zarówno moduły percepcji, planowania, jak i sterowania, które zostały zintegrowane w celu zaofiarowania rozwiązania ADAS. Przedstawiona praca prezentuje wyniki 24 najazdów podczas dokowania w terenie, co potwierdza, że proponowane rozwiązanie może być stosowane w rzeczywistych scenariuszach.

Początkowe prace dotyczące rozwiązania lokalizacyjnego z robotami koczującymi zostały przedstawione w [12]. Praca ta wykorzystuje klasyfikację terenu do określenia haptycznej pozycji robota z użyciem podejścia Monte Carlo. Zaobserwowaliśmy kilka ograniczeń, ponieważ wymaga ona wyraźnej granicy między obszarami o określonej klasie terenu, a klasy są określane na podstawie ludzkiej percepcji. Niemniej jednak, była to pierwsza praca potwierdzająca, że lokalizacja haptyczna z wykorzystaniem klasyfikacji terenu jest możliwa.

Architektura oparta na transformerach dla klasyfikacji terenu została po raz pierwszy przedstawiona w [7]. Podejście to pokazało, że przetwarzanie sygnałów w predefiniowanym

oknie może być efektywnie wykonane bez użycia RNN, oferując jednocześnie szybki czas wnioskowania. Praca ta stanowiła bazę do dalszych ulepszeń przedstawionych w [A6].

Pomysł połączenia danych z kamer RGB i kamer termowizyjnych do wykrywania pieszych był badany w [59]. Chcieliśmy określić, w jaki sposób obrazy z obu czujników powinny być łączone, aby dostarczyć rozwiązanie oparte na YOLO, które wykonuje bardziej efektywne detekcje w porównaniu z danymi wejściowymi z jednego czujnika. Najlepsze wyniki uzyskaliśmy przy fuzji informacji na poziomie cech - w środku przetwarzania sieci, przed generowaniem ostatecznych wykryć obiektów. W rezultacie przeanalizowaliśmy również, czy proponowany system zapewnia wykrycia wystarczające do celów autonomicznej jazdy zarówno w dzień, jak i w nocy.

#### 4.8 Podsumowanie i możliwe dalsze prace

W podsumowaniu przedstawiono główne wyniki uzyskane w prezentowanym osiągnięciu naukowym. Nowe metody rozszerzają istniejące rozwiązania state-of-the-art w zakresie lokalizacji za pomocą systemów wielosensorycznych. Mój główny wkład w obecny stan wiedzy w zakresie robotyki można zatem podsumować jako:

- Projekt i implementacja oprogramowania do dokładnej kalibracji czasoprzestrzennej kamer RGB i nowoczesnych obrotowych LiDAR-ów 3D [A1].
- Projekt i implementacja rozwiązania kalibracyjnego, które skupia się na poprawie wydajności zestawu sensorycznego w aplikacji docelowej zamiast na znalezieniu najdokładniejszej transformacji geometrycznej [A3]. Ten pomysł kalibracji doprowadził do zaprojektowania dokładnego rozwiązania estymacji pozy dla scenariusza dokowania autobusów [A2].
- Projekt i implementacja systemu rozpoznawania miejsc, który może skutecznie określić lokalizację na podstawie obrazów RGB i chmur punktów z LiDAR-u 3D [A4]. Zostało to następnie rozszerzone o propozycję nowej reprezentacji sieci neuronowej dla chmur punktów z LiDARów 3D [A5].
- Efektywna architektura do wielosensorycznej klasyfikacji terenu z użyciem transformatorów z potencjałem do zastosowania w lokalizacji [A6].
- Projekt i implementacja wydajnego i praktycznego rozwiązania do lokalizacji robota kroczącego z wykorzystaniem rzadkich danych haptycznych bez jawnej listy klas terenu [A7]. Dalsze ulepszenia prowadzące do uzyskania pierwszej uczonej i nienadzorowanej reprezentacji dla celów lokalizacji haptycznej w [A8].

Przedstawione prace tworzą jednolity cykl publikacji, gdyż każda praca bazuje na doświadczeniach z poprzednich podejść, nawet jeśli aplikacja jest inna. Rozwiązanie kalibracji czasoprzestrzennej [A1] było kluczowym elementem przy projektowaniu podejścia do kalibracji dla wybranej aplikacji [A3], które zostało z powodzeniem wykorzystane w [A2]. Z drugiej strony, żmudna praca nad kalibracją systemów kamera-LiDAR zrodziła zainteresowanie rozwiązaniami bezkalibracyjnymi do rozpoznawania miejsc w [A4] i [A5]. Oryginalny pomysł na klasyfikację terenu w [A6] okazał się słabym wyborem dla lokalizacji haptycznej, która została zmodyfikowana do wykorzystania sygnałów nienadzorowanych dla celów lo-

kalizacji w [A7]. Spowodowało to kluczową zmianę w podejściu do przetwarzania sygnałów haptycznych, która została jeszcze bardziej dopracowana w [A8] z wykorzystaniem błędu trójkowego używanego wcześniej do globalnej lokalizacji w [A4] i [A5].

Zaproponowane metody są praktycznymi rozwiązaniami lokalizacyjnymi lub ich elementami, które mogą być wykorzystane w autonomicznych robotach mobilnych. Kalibracja jest już niezbędnym krokiem przed działaniem w środowisku aplikacji, natomiast dokowanie za pomocą kamery było już prezentowane w takim środowisku. Pozostałe rozwiązania (rozpoznawanie miejsca i lokalizacja haptyczna) mogą wspierać komponenty rozmieszczone w świecie rzeczywistym jako mechanizmy awaryjne. Niezależnie od przedstawionych zastosowań, zaproponowane metody są uniwersalne, pokazując jak łączyć dane z systemów wielosensorycznych przy jednoczesnym zapewnieniu pożądanej dokładności.

Nadal jednak istnieje możliwość wprowadzenia dalszych usprawnień. Proponowane rozwiązanie kalibracji mogłoby zostać rozszerzone, aby umożliwić autokalibrację podczas pracy, sprawiając, że system decyduje kiedy, a następnie dokonuje ponownej kalibracji w celu zmniejszenia błędu podczas ostatecznego rozmieszczenia. Zaproponowana w [A3] kalibracja systemu mogłaby zostać rozszerzona o obsługę różnych niepewności pochodzących z sensorów, np. poprzez rozszerzenie [A2] o niepewność wykrytych punktów kluczowych. System rozpoznawania miejsc przedstawiony w [A4] i [A5] mógłby zostać rozszerzony, aby pokazać, jak połączyć klasyczne potoki przetwarzania obrazu z proponowanymi rzadkimi konwolucjami 3D dla chmur punktów 3D. Jedną z możliwości byłoby przetestowanie wielu schematów fuzji, tak jak zostało to przebadane w mojej innej pracy [A4] i [A5], przy jednoczesnym zapewnieniu odpowiednich modułów uwagi jak w [A6]. Pomysły lokalizacji haptycznej dla robotów kroczących mogą być również rozszerzone na scenariusze zewnętrzne, które powinny stanowić nowe wyzwanie ze względu na brak wyraźnych granic pomiędzy klasami i znacznie większe rozmiary map w porównaniu z mapami wewnątrz budynków.

##### **5. Informacja o wykazywaniu się istotną aktywnością naukową albo artystyczną realizowaną w więcej niż jednej uczelni, instytucji naukowej lub instytucji kultury, w szczególności zagranicznej.**

Swoje badania naukowe rozpocząłem jeszcze w czasie studiów inżynierskich, po których w 2013 r. odbyłem swój pierwszy, 3-miesięczny staż naukowy w Forschungszentrum Informatik (FZI) w Karlsruhe w Niemczech, korzystając z możliwości staży w ramach programu ERASMUS. Moim celem było opracowanie nowego systemu reakcyjnego (zachowań) dla robota kroczącego LAURON V umożliwiającego pokonywanie stromych pochyłości. Cel udało się zrealizować, a wynikiem współpracy było opublikowanie dwóch artykułów naukowych [57, 58], które uzyskały uznanie na arenie międzynarodowej i są sumarycznie cytowane 161 razy według Google Scholar.

W roku 2014 odwiedziłem Uniwersytet w Birmingham w Wielkiej Brytanii w celu zaprezentowania swoich prac badawczych w ramach doktoratu. W lipcu 2015 byłem uczestnikiem szkoły letniej “Summer School 2015: Aerospace Information Technology” na Uniwersytecie w Würzburgu w Niemczech, gdzie poruszana była kwestia tworzenia rozwiązań dla zastosowań w kosmosie.

W czasie finalnych prac nad rozprawą doktorską w 2017 r. w ramach programu ETIU-DA zrealizowałem 3-miesięczny staż naukowy na Uniwersytecie w Saragossie w Hiszpanii w grupie prof. Juana Tardósa. Wybrałem tę grupę ze względu na fakt, że jej członkowie byli twórcami najlepszego w tamtym czasie wizyjnego systemu jednocześnie lokalizacji i budowy mapy (SLAM) znanego jako ORB-SLAM2. W ramach swoich badań pracowałem nad możliwością wykorzystania błędu fotometrycznego w systemie SLAM opartym na cechach (ORB-SLAM2). W ramach realizacji prac udało się zaimplementować oczekiwane rozwiązanie, ale wynik naukowy nie potwierdził przypuszczeń o poprawie dokładności systemu i nie doprowadziło to do powstania publikacji.

Okres pandemii był czasem o utrudnionych warunkach współpracy międzynarodowej. Niemniej, w latach 2020–2021 realizowałem projekt “THING subTerranean Haptic INvestiGator” w ramach Horyzontu 2020 Research and Innovation, numer projektu 780883, we współpracy z Uniwersytetem w Oxfordzie, Eidgenössische Technische Hochschule (ETH) w Zurychu, Uniwersytetem w Edynburgu, Uniwersytetem w Pizie i KGHM CUPRUM. Wspólny projekt umożliwił utrzymanie kontaktów międzynarodowych i zaowocował dwoma wspólnymi pracami z Uniwersytetem w Oxford w czasopiśmie *Autonomous Robots* [12] oraz na konferencji ICRA [A7]. W ramach tego projektu realizowaliśmy także wspólne testy z KGHM CUPRUM, których wynikiem była wspólna publikacja podsumowująca te działania [63].

Okres popandemiczny umożliwił powrót do ściślejszej współpracy międzynarodowej. W maju 2022 r. odbyłem staż naukowy na Swiss Federal Institute of Technology at Lausanne (EPFL) w Lozannie w Szwajcarii w grupie prof. Auke Ijspeerta zajmującego się robotami inspirowanymi biologią, gdzie opracowałem system percepcji do budowy mapy wysokości. Staż jest podtrzymaniem współpracy przy tworzeniu wspólnych wniosków europejskich w ramach konkursów Research and Innovation Action – aktualnie trzecia wersja naszej wizji na współpracę w ramach projektu “Amphibious Locomotion with Learned Control and Biologically Inspired Underwater Sensing” jest na liście rezerwowej, gdzie Politechnika Poznańska jest liderem konsorcjum łączącego takie ośrodki jak Czech Technical University (CTU) z Czech, Technische Universität Darmstadt (TUDA) z Niemiec, Scuola Superiore Sant’Anna (SSSA) z Włoch, EPFL ze Szwajcarii, ARTES4.0 z Włoch, Parco Regionale Migliarino, San Rossore, Massaciuccoli (PRMSNM) z Włoch, Openbaar Lichaam Gezamenlijke Brandweer (GBRAND) z Holandii, Veolia Energia Poznan S.A. oraz MAB Robotics Sp. z o.o. z Polski.

W kwietniu 2023 r. zaprezentowałem swoje badania na Uniwersytecie w Cambridge w Wielkiej Brytanii, otwierając pole do współpracy nad technologiami rozwijanymi na potrzeby AR/VR, będącymi przedmiotem realizowanego projektu badawczego finansowanego przez firmę Meta Platforms Inc.

Ostatecznie, w ramach swojego dorobku mogę wyodrębnić następujące publikacje naukowe powstałe w ramach współpracy międzynarodowej oraz z innymi ośrodkami

w Polsce:

1. z Uniwersytetem w Oxfordzie, Wielka Brytania [12][A7],
2. z Politechniką Warszawską, Polska [A5],
3. z KGHM CUPRUM, Polska [63],
4. z Politechniką w Graz, Austria [71],
5. z FZI w Karlsruhe, Niemcy [57, 58],
6. z CTU w Pradze, Czechy [49].

#### **6. Informacja o osiągnięciach dydaktycznych, organizacyjnych oraz popularyzujących naukę lub sztukę.**

Poza działalnością naukową prowadzę również aktywnie działalność popularyzatorską i dydaktyczną. W 2015 roku brałem udział w projekcie “Robotycy – współcześni czarodzieje” finansowany przez Fundację na Rzecz Nauki Polskiej w ramach programu “eNgage”, którego celem była popularyzacja robotyki wśród uczniów szkół średnich. W roku 2016 byłem kierownikiem projektu “Ujarmij magię swojego smartfona” finansowanego przez Fundację na Rzecz Nauki Polskiej także w ramach programu “eNgage”, którego celem było przeprowadzenie serii zajęć z możliwości przetwarzania i wykorzystania sensorów urządzeń mobilnych. W roku 2022 brałem udział w projekcie AI-Tech, którego celem było przygotowanie materiałów dydaktycznych do zajęć z metod sztucznej inteligencji w robotyce.

Oprócz realizacji projektów dydaktycznych wielokrotnie przedstawiałem różne aspekty naukowe w sposób popularyzatorski:

1. 2019, VIII LO w Poznaniu, Konstrukcja oraz implementacja autonomicznego drona,
2. 2017, studenci kierunku Automatyka i Robotyka, Jednoczesna lokalizacja i budowa mapy środowiska dla robotów oraz ludzi,
3. 2015, VIII LO w Poznaniu, Nowe metody wspomaganie nawigacji osób wykorzystujące zasoby urządzeń mobilnych typu tablet lub smartfon,
4. 2015, koło naukowe CybAiR, Nowe metody wspomaganie nawigacji osób wykorzystujące zasoby urządzeń mobilnych typu tablet lub smartfon,
5. 2015, Poznański Festiwal Nauki i Sztuki, Nowe metody wspomaganie nawigacji osób wykorzystujące zasoby urządzeń mobilnych typu tablet lub smartfon,
6. 2014, Konferencja “Zawód naukowiec”, Prezentacja robota TAPAS,
7. 2014, Poznański Festiwal Nauki i Sztuki, Robot TAPAS - jak skonstruować autonomicznego robota?

Do tej pory byłem promotorem 11 magistrów oraz 15 inżynierów w ramach 8 prac magisterskich oraz 7 prac inżynierskich. W 2021 praca zrealizowana przez mojego

dyploanta (Kamil Roszyk) uzyskała I miejsce w konkursie Młodzi Innowacyjni organizowanym przez Przemysłowy Instytut Automatyki i Pomiarów (PIAP). W 2021 r. praca zrealizowana przez promowanych przez mnie magistrantów (Kamil Żywanowski i Adam Banaszczyk) uzyskała III Nagrodę na najlepszą pracę magisterską w konkursie zorganizowanym przez SEP Poznań. Wynikiem mojej współpracy ze studentami w ramach prac dyplomowych są 2 artykuły w czasopismach (RAL, MDPI Sensors) oraz 2 artykuły na konferencjach (ICARCV, ICRA). Pełnię również rolę promotora pomocniczego w trzech otwartych przewodach doktorskich (Tomasz Nowak, Krzysztof Ćwian i Damian Sójka). Prowadzę zajęcia dydaktyczne na kierunkach Automatyka i Robotyka, Informatyka oraz Artificial Intelligence na Politechnice Poznańskiej, które pokrywają się z moimi badaniami:

1. Robotics I (laboratoria i część wykładu, studia 1 stopnia, język angielski, Artificial Intelligence),
2. Nowoczesne Sensory w Robotyce (wykład, studia 2 stopnia, Automatyka i Robotyka, specjalność Roboty i Systemy Autonomiczne),
3. Metody Sztucznej Inteligencji w Robotyce (laboratoria, studia 2 stopnia, Informatyka).

Wcześniej prowadziłem takie zajęcia jak:

1. Informatyka (laboratoria, studia 1 stopnia, Automatyka i Robotyka),
2. Information Engineering (laboratoria, studia 1 stopnia, język angielski, Automatyka i Robotyka),
3. Podstawy Techniki Mikroprocesorowych (laboratoria, studia 1 stopnia, Automatyka i Robotyka),
4. Autonomiczne Roboty Mobilne (laboratoria, studia 2 stopnia, Automatyka i Robotyka, specjalność Roboty i Systemy Autonomiczne),
5. Algebra Liniowa (laboratoria, studia 1 stopnia, Matematyka).

## **7. Inne informacje, ważne z jego punktu widzenia, dotyczące jego kariery zawodowej**

Mój dorobek po uzyskaniu stopnia doktora nauk technicznych obejmuje 23 artykuły. W czasopismach z listy Journal Citation Report zostało opublikowanych 12 artykułów w takich czasopismach jak: IEEE Robotics and Automation Letters (x2), Measurement, Robotics and Autonomous Systems (RAS) (x2), Autonomous Robots, IEEE Vehicular Technology Magazine, International Journal of Applied Mathematics and Computer Science (AMCS) i MDPI Sensors (x4). Pozostałe 11 artykułów zostało opublikowane na konferencjach naukowych, takich jak: International Conference on Robotics and Automation (ICRA) (x2), International Conference on Control, Automation, Robotics and Vision (ICARCV) (x2), IEEE Intelligent Vehicles Symposium

(IV), Application of Computers and Operations Research in the Minerals Industries (APCOM) oraz innych.

Baza Web of Science Core Collection (WoS) raportuje liczbę cytowań 565 (w tym 473 bez autocytowań); H-index równy 13. Baza Scopus raportuje liczbę cytowań wynoszącą 671 razy (w tym 489 bez autocytowań oraz bez cytowań współautorów); H-index równy 14. Baza Google Scholar raportuje liczbę cytowań wynoszącą 999 razy; H-index równy 17. Sumaryczna liczba punktów Ministerstwa Nauki i Szkolnictwa Wyższego dla publikacji po doktoracie wynosi 1620 za czasopisma oraz 730 za konferencje. Sumaryczny Impact Factor dla wszystkich publikacji po doktoracie w czasopismach z listy JCR zgodnie z rokiem opublikowania wynosi 52,983 (łącznie 61,791).

Aktywnie uczestniczę w międzynarodowych konferencjach naukowych, m.in.: IEEE International Conference on Robotics and Automation (ICRA) lub European Conference on Mobile Robot (ECMR). W 2023 r. pełniłem rolę redaktora (Associate Editor) na konferencji 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2023) (140 pkt MEiN, CORE A). Recenzowałem 13 zgłoszonych publikacji naukowych na takich konferencjach jak:

1. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS, CORE A),
2. IEEE International Conference on Robotics and Automation (ICRA, CORE B),
3. IEEE Intelligent Vehicles Symposium (IV, CORE B),
4. International Workshop on Robot Motion and Control (ROMOCO),
5. Krajowa Konferencja Robotyki (KKR),
6. IEEE International Conference on Multisensor Fusion and Integration (MFI),
7. European Conference on Mobile Robots (ECMR),
8. International Conference on Control, Automation, Robotics and Vision (ICARCV, CORE A),
9. Polskie Porozumienie na Rzecz Rozwoju Sztucznej Inteligencji (PP-RAI).

W roku 2022 byłem edytorem sesji specjalnej “Best Practice in Simultaneous Localization and Mapping (SLAM)” w MDPI Sensors wraz z prof. Giorgio Grisettim (Sapienza University of Rome) oraz dr. Marco Camurrim (Free University of Bozen-Bolzano). Recenzowałem także 49 publikacji naukowych w czasopismach (malejąco według punktów MEiN oraz następnie IF):

1. ACM Computing Surveys (IF: 14.324, 200 pkt MEiN),
2. IEEE Internet of Things Journal (IF: 9.936, 200 pkt MEiN),
3. Measurement (IF: 5.131, 200 pkt MEiN),
4. Automation in Construction (IF: 11.45, 140 pkt MEiN),



5. Journal of Cleaner Production (IF: 9.297, 140 pkt MEiN),
6. Neurocomputing (IF: 5.719, 140 pkt MEiN),
7. IEEE-ASME Transactions on Mechatronics (IF: 5.303, 140 pkt MEiN),
8. Robotics and Autonomous Systems (IF: 4.67, 140 pkt MEiN),
9. IEEE Sensors (IF: 4.325, 100 pkt MEiN),
10. MDPI Sensors (IF: 3.847, 100 pkt MEiN),
11. IEEE Signal Processing Letters (IF: 3.67, 100 pkt MEiN),
12. IEEE Access (IF: 3.367, 100 pkt MEiN),
13. Digital Singal Processing (IF: 3.15, 100 pkt MEiN),
14. MDPI Applied Sciences (IF: 2.838, 100 pkt MEiN),
15. Bulletin of the Polish Academy of Sciences-Technical Sciences (IF: 1.515, 100 pkt MEiN),
16. IET Image Processing (IF: 2.20, 70 pkt MEiN),
17. Industrial Robot (IF: 1.95, 70 pkt MEiN),
18. MDPI Robotics (20 pkt MEiN).

Do tej pory brałem udział w trzech projektach europejskich jako wykonawca:

1. 01.2020 – 08.2021, Wykonawca, Politechnika Poznańska  
Projekt „THING subTerranean Haptic INvestiGator” w ramach Horyzontu 2020 Research and Innovation, numer projektu 780883, we współpracy z Uniwersytetem w Oxfordzie, ETH Zurich, Uniwersytetem w Edynburgu, Uniwersytetem w Pizie i KGHM CUPRUM.
2. 05.2019 – 12.2019, Wykonawca, Mandala Robotics  
Opracowanie systemu wizyjnej percepcji dla głowicy sensorycznej w ramach programu European SMEs Robotics Applications, numer projektu 780265, Wniosek nr 234 – MANDEYE wspierany w ramach programu ESMERA FOCE.
3. 09 – 13.05.2022, Staż naukowy, Swiss Federal Institute of Technology at Lausanne (EPFL), Biorobotics Laboratory (BioRob), Lozanna, Szwajcaria, Opracowanie systemu percepcji do budowy mapy wysokości dla robotów inspirowanych biologią w grupie Auke Ijspeerta w ramach programu TERRINET.

Byłem kierownikiem czterech projektów:

1. 02.2022 – 01.2023, Politechnika Poznańska  
Projekt „Nowatorski system lokalizacji wózków samojezdnych AGV z wykorzystaniem skanerów laserowych 3D” w ramach programu NCBR TANGO, TANGO-V-A/0036/2021-00.

2. 02.2016 – 09.2019, Politechnika Poznańska  
Projekt „Nowa metoda fuzji danych ilościowych i jakościowych wykorzystująca optymalizację grafu ograniczeń w problemie jednoczesnej lokalizacji i budowy mapy” finansowany przez Narodowe Centrum Nauki (NCN) w ramach programu PRELUDIUM, 2015/17/N/ST6/01228.
3. 07.2013 – 03.2016, Politechnika Poznańska  
Projekt „Nowe metody wspomaganie nawigacji osób wykorzystujące zasoby urządzeń mobilnych typu tablet lub smartfon” finansowany przez Ministerstwo Nauki i Szkolnictwa Wyższego w ramach programu Diamentowy Grant, DI2012 004142.
4. 09.2013 – 11.2014, Politechnika Poznańska  
Projekt „Projekt autonomicznego robota mobilnego do zadań poszukiwania i przynoszenia przedmiotów na potrzeby międzynarodowego konkursu Robots Intellect” finansowany przez Ministerstwo Nauki i Szkolnictwa Wyższego w ramach programu Generacja Przyszłości, WS-1282/HF/2013.

Byłem również wykonawcą w wielu projektach:

1. 03.2023 – aktualnie, Kierownik zespołu programistów, Politechnika Poznańska  
Badania technologii AR/VR w ramach projektu ARPresence ufundowanego przez Meta Platforms. Inc. w Centrum Sztucznej Inteligencji i Cyberbezpieczeństwa.
2. 01.2020 – 08.2021, Wykonawca, Politechnika Poznańska  
Projekt „THING subTerranean Haptic INvestiGator” w ramach Horyzontu 2020 Research and Innovation, numer projektu 780883, we współpracy z Uniwersytetem w Oxfordzie, ETH Zurich, Uniwersytetem w Edynburgu, Uniwersytetem w Pizie i KGHM CUPRUM.
3. 04.2018 – 06.2021, Wykonawca, Politechnika Poznańska z Solaris Bus & Coach  
Projekt „Zaawansowany system wsparcia precyzyjnych manewrów dla kierowców autobusów miejskich jednosegmentowych i przegubowych” w ramach Programu Operacyjnego Inteligentny Rozwój 2014-2020 współfinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego, POIR.04.01.02-00-0081/17-01 we współpracy z Solaris Bus & Coach.
4. 05.2019 – 12.2019, Wykonawca, Mandala Robotics  
Opracowanie systemu wizyjnej percepcji dla głowicy sensorycznej w ramach programu European SMEs Robotics Applications, numer projektu 780265, Wniosek nr 234 – MANDEYE wspierany w ramach programu ESMERA FOCE.
5. 02.2017 – 12.2019, Wykonawca, Wojskowe Zakłady Lotnicze Nr 2 S.A.  
Opracowanie systemu percepcji i autonomii dla latającego drona w ramach projektu „Opracowanie systemu awioniki pokładowej wielowirnikowej platformy latającej z podsystemem nawigacji wizyjnej” realizowanego w ramach programu sektorowego NCBR INNOLOT, podwykonawstwo na Politechnice Poznańskiej.

6. 07.2017 – 07.2018, Specjalista ds. sensorów, EMPEKO S.A.,  
Projekt „Wytworzenie i przetestowanie prototypu urządzenia do automatycznego monitorowania wpływu elektrowni wiatrowych na nietoperze i ptaki” w ramach programu NCBR 1.2. Sektorowe programy B+R.
7. 01.2018 – 03.2018, Wykonawca, Politechnika Poznańska,  
Projekt „Opracowanie i implementacja nowych metod lokalizacji, budowy mapy oraz planowania ruchu z użyciem czujników RGB-D w zrobotyzowanych systemach elastycznej produkcji” w ramach programu NCBR LIDER, LIDER/33/0176/L-8/16/NCBR/20.
8. 02.2014 – 05.2016, Wykonawca, Politechnika Poznańska  
Projekt „Nowa metoda jednoczesnej samolokalizacji i budowy mapy na podstawie danych RGB-D umożliwiająca precyzyjną nawigację robota autonomicznego w złożonym środowisku” finansowany przez Narodowe Centrum Nauki (NCN) w ramach programu OPUS, 2013/09/B/ST7/01583.

Odbyłem w swojej karierze trzy staże zagraniczne:

- 1) Staż naukowy, 05.2022, Swiss Federal Institute of Technology at Lausanne (EPFL), Lozanna, Szwajcaria.  
Opracowanie systemu percepcji do budowy mapy wysokości dla robotów inspirowanych biologią w grupie Auke Ijspeerta.
- 2) Staż naukowy, 06.2017 – 10.2017, University of Zaragoza, Saragossa, Hiszpania.  
Opracowanie systemu SLAM opartego na cechach (ORB-SLAM2) z wykorzystaniem błędu fotometrycznego pod opieką prof. Juana D. Tardósa.
- 3) Staż naukowy, 07.2013 – 09.2013, Forschungszentrum Informatik (FZI), Karlsruhe, Germany.  
Projekt i implementacja nowego systemu reakcyjnego (zachowań) dla robota kroczącego LAURON V umożliwiającego pokonywanie stromych pochyłości.

Najważniejsze uzyskane stypendia i nagrody przedstawiono poniżej:

- 1) 2022 – 2026, Stypendium dla wybitnych młodych naukowców, Ministerstwo Nauki i Szkolnictwa Wyższego.
- 2) 2021 – 2022, Stypendium FNP START dla 100 wybitnych młodych naukowców,
- 3) 2019, Nagroda I miejsca za najlepszą rozprawę doktorską ze sztucznej inteligencji Polskiego Stowarzyszenia Sztucznej Inteligencji (PSSI),
- 4) 2019, Stypendium dla młodych badaczy poznańskiego środowiska naukowego,
- 5) 2019, Nagroda Miasta Poznania za wyróżniającą się pracę doktorską,
- 6) 2019, Nagroda II miejsca za rozprawę doktorską w konkursie Młodzi Innowacyjni,

- 7) 2016 – 2018, Narodowe Centrum Nauki, program ETIUDA, stypendium na dokończenie rozprawy doktorskiej oraz staż u prof. Juana D. Tardósa,
- 8) 2017 – 2018, Stypendium Ministra Nauki i Szkolnictwa Wyższego dla doktorantów
- 9) 2015 – 2022, Nagroda Rektora Politechniki Poznańskiej za osiągnięcia naukowe (corocznie)
- 10) 2014, Medal dla wyróżniającego się absolwenta przyznawany corocznie 5 najlepszym absolwentom PP,
- 11) 2014, Wyróżnienie za pracę magisterską przez Polskie Towarzystwo Informatyczne,
- 12) 2014, Uzyskanie II nagrody za pracę magisterską w Konkursie organizowanym przez Zarząd Oddziału Poznańskiego Stowarzyszenia Elektryków Polskich oraz Wydział Elektryczny Politechniki Poznańskiej,
- 13) 2014, Nagroda dla najlepszego studenta nauk inżynierskich i technicznych w Polsce w konkursie “Studencki Nobel”,
- 14) 2011 – 2013, Stypendium Ministra Nauki i Szkolnictwa Wyższego dla studentów (3 razy),
- 15) 2012 – 2013, Stypendium Marszałka Województwa Wielkopolskiego (2 razy).

Zestawienie liczbowe osiągnięć znajduje się w tabeli 13.

Tabela 13: Zestawienie osiągnięć

Opublikowane prace naukowe przed doktoratem	41
Opublikowane prace naukowe po doktoracie	23
Opublikowane prace naukowe w czasopismach z listy JCR przed doktoratem	6
Opublikowane prace naukowe w czasopismach z listy JCR po doktoracie	12
Sumaryczny IF opublikowanych publikacji przed doktoratem według listy JCR	8,808
Sumaryczny IF opublikowanych publikacji po doktoracie według listy JCR	52,983
Staże w zagranicznych ośrodkach badawczych	3
Recenzowanie publikacji w czasopismach międzynarodowych	49
Całkowita liczba recenzji prac naukowych	62
Kierowanie projektami badawczymi	4
Udział w projektach badawczych w roli innej niż kierownik	7
Udział w międzynarodowych konferencjach naukowych	15
Promotor prac inżynierskich (liczba osób)	7 (15)
Promotor prac magisterskich (liczba osób)	8 (11)
Promotor pomocniczy w rozprawach doktorskich	3
Liczba cytowań publikacji według bazy WoS (bez autocytowań)	565 (473)
Liczba cytowań publikacji wg. bazy Scopus (bez autocytowań i bez cytowań współautorów)	671 (489)
Liczba cytowań publikacji według bazy Google Scholar (bez autocytowań)	999 (855)
Indeks Hirscha według bazy WoS	13
Indeks Hirscha według bazy Scopus	14
Indeks Hirscha według bazy Google Scholar	17

## Literatura

- [1] A. Ahmadi, T. Nygaard, N. Kottege, D. Howard, and N. Hudson. Semi-Supervised Gated Recurrent Neural Networks for Robotic Terrain Classification. *IEEE Robotics and Automation Letters*, 6(2), s. 1848–1855, 2021.
- [2] T. Barros, L. Garrote, R. Pereira, C. Premevida, and U.J. Nunes. AttDLNet: Attention-based DL Network for 3D LiDAR Place Recognition. *arXiv preprint arXiv:2106.09637*, 2021.
- [3] I.A. Barsan, S. Wang, A. Pokrovsky, and R. Urtasun. Learning to Localize Using a LiDAR Intensity Map. W *2nd Conference on Robot Learning (CoRL)*, s. 605–616, 2018.
- [4] A. Bartoli. On the non-linear optimization of projective motion using minimal parameters'. W *European Conference on Computer Vision (ECCV)*, Kopenhaga, s. 340–354, 2002.

- [5] J. Bednarek, M. Bednarek, P. Kicki, and K. Walas. Robotic Touch: Classification of Materials for Manipulation and Walking. W *IEEE International Conference on Soft Robotics (RoboSoft)*, s. 527–533, 2019.
- [6] J. Bednarek, M. Bednarek, L. Wellhausen, M. Hutter, and K. Walas. What am I touching? Learning to classify terrain via haptic sensing. W *IEEE International Conference on Robotics and Automation (ICRA)*, s. 7187–7193, Maj 2019.
- [7] M. Bednarek, M. Łysakowski, J. Bednarek, M.R. Nowicki, and K. Walas. Fast Haptic Terrain Classification for Legged Robots Using Transformer. W *2021 European Conference on Mobile Robots (ECMR)*, Bonn, Niemcy, s. 1–7, 2021.
- [8] M. Bloesch, M. Burri, H. Sommer, R. Siegwart, and M. Hutter. The Two-State Implicit Filter Recursive Estimation for Mobile Robots. *IEEE Robotics and Automation Letters*, 3(1), s. 573–580, Styczeń 2018.
- [9] M. Bosse, R. Zlot, and P. Flick. Zebedee, Design of a spring-mounted 3D range sensor with application to mobile mapping, *IEEE Transactions on Robotics*, vol. 28, no. 5, s. 1104–1119, 2012.
- [10] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 25(11), s. 120–123, 2000.
- [11] R. Buchanan, M. Camurri and M. Fallon. Haptic Sequential Monte Carlo Localization for Quadrupedal Locomotion in Vision-Denied Scenarios. W *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, s. 3657-3663.
- [12] R. Buchanan, J. Bednarek, M. Camurri, M.R. Nowicki, K. Walas, and M. Fallon. Navigating by touch: haptic Monte Carlo localization via geometric sensing and terrain classification. *Autonomous Robots*, 45, s. 843–857, 2021.
- [13] J. Guo, P. Borges, C. Park, and A. Gawel. Local Descriptor for Robust Place Recognition Using LiDAR Intensity. *IEEE Robotics and Automation Letters*, 4(2), s. 1470–1477, 2019.
- [14] C. Cadena, L. Carlone, H. Carrillo, et al.. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. W *IEEE Transactions on Robotics*, vol. 32, no. 6, s. 1309–1332, 2016.
- [15] Y. Cui, R. Chen, W. Chu, et al.. Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review. *IEEE Transactions on Intelligent Transportation Systems*, s. 1–18, 2021.
- [16] M. Cummins, and P. Newman. Accelerating FAB-MAP with Concentration Inequalities. *IEEE Transactions on Robotics*, vol. 26, no. 6, s. 1042–1050, 2010.

- [17] J. Deray, J. Solà, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics. *J. Open Source Softw.*, 5(46), 1371, 2020.
- [18] J. Domhof, J. Kooij, and D. Gavrilu. An Extrinsic Calibration Tool for Radar, Camera and Lidar. W *2019 International Conference on Robotics and Automation (ICRA)*, s. 8107–8113, 2019.
- [19] J. Domhof, J. Kooij, and D. Gavrilu. A Joint Extrinsic Calibration Tool for Radar, Camera and Lidar. *IEEE Transactions on Intelligent Vehicles*, 2021.
- [20] A. Dosovitskiy, et al., An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. W *9th International Conference on Learning Representations, ICLR 2021*, 2021.
- [21] J. Du, R. Wang, and D. Cremers. DH3D: Deep Hierarchical 3D Descriptors for Robust Large-Scale 6DoF Relocalization. W *European Conf. on Computer Vision*, s. 744–762. Springer, 2020.
- [22] P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. W *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, s. 1280–1286, 2013.
- [23] P. Furgale, and T.D. Barfoot. Visual Teach and Repeat for Long-Range Rover Autonomy. *Journal of Field Robotics*, 27(5), s. 534–560, 2010.
- [24] D. Gálvez-López, and J.D. Tardós. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics*, vol. 28, no. 5, s. 1188–1197, 2012.
- [25] K. Giseop, and K. Ayoung. Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map. W *2018 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*, s. 4802–4809, 2018.
- [26] K. Giseop, C. Sunwook, and K. Ayoung. Scan Context++: Structural Place Recognition Robust to Rotation and Lateral Variations in Urban Environments. *IEEE Transactions on Robotics*, vol. 38, no. 3, s. 1856–1874, 2022.
- [27] M.J. Kim, M.S. Kim, and S. Shin. A general construction scheme for unit quaternion curves with simple high order derivatives. W *ACM Special Interest Group on Computer GRAPHics and Interactive Techniques Conference (SIGGRAPH)*, Los Angeles, s. 369–376, 1995.
- [28] B. Fu, F. Han, Y. Wang, Y. Jiao, X. Ding, Q. Tan, L. Chen, M. Wang, and R. Xiong. High-Precision Multicamera-Assisted Camera-IMU Calibration: Theory and Method. W *IEEE Transactions on Instrumentation and Measurement*, vol. 70, s. 1–17, 2021.
- [29] A. Hermans, L. Beyer, and B. Leibe. In Defense of the Triplet Loss for Person Re-Identification. arXiv preprint arXiv:1703.07737, 2017.



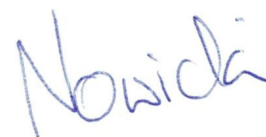
- [30] M.H. Hoepflinger, C.D. Remy, M. Hutter, and R. Siegwart. Haptic Terrain Classification on Natural Terrains for Legged Robots. W *International Conference on Climbing and Walking Robots (CLAWAR)*, s. 785–792, 2010.
- [31] K. Huang, J. Xiao, and C. Stachniss, Accurate Direct Visual-Laser Odometry with Explicit Occlusion Handling and Plane Detection W *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, s. 1295–1301, 2019.
- [32] L. Hui, M. Cheng, J. Xie, and J. Yang. Efficient 3D Point Cloud Feature Learning for Large-Scale Place Recognition. In *IEEE Transactions on Image Processing*, vol. 31, s. 1258–1270, 2022.
- [33] M. Hutter, et al., Anymal - a highly mobile and dynamic quadrupedal robot. W *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, s. 38–44, 2016.
- [34] H. Kolvenbach, C. Bärtschi, L. Wellhausen, R. Grandia, and M. Hutter. Haptic Inspection of Planetary Soils With Legged Robots. *IEEE Robotics and Automation Letters*, 4(2), s. 1626–1632, 2019.
- [35] J. Komorowski. MinkLoc3D: Point Cloud Based Large-Scale Place Recognition. W *Proc. of the IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, s. 1790–1799, 2021.
- [36] C. Choy, J. Gwak, S. Savarese, 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks, W *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, s. 3075–3084, 2019.
- [37] F. Radenović, G. Tolias, O. Chum, Fine-Tuning CNN Image Retrieval with No Human Annotation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (7), s. 1655–1668, 2019.
- [38] J. Komorowski, M. Wysoczańska, and T. Trzcíński. MinkLoc++: Lidar and Monocular Image Fusion for Place Recognition. W *2021 Inter. Joint Conf. on Neural Networks (IJCNN)*, s. 1–8, 2021.
- [39] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, g2o: A general framework for graph optimization, W *IEEE International Conference on Robotics and Automation*, Shanghai, s. 3607–3613, 2011.
- [40] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. PointPillars: Fast Encoders for Object Detection From Point Clouds. W *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, s. 12689–12697, Czerwiec 2019.

- [41] C.L. Lee, Y.H. Hsueh, C.C. Wang, and W.C. Lin. Extrinsic and Temporal Calibration of Automotive Radar and 3D LiDAR. W *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, s. 9976–9983, 2020.
- [42] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An Accurate  $O(n)$  Solution to the PnP Problem. *Int. J. Comput. Vision*, 81(2), s. 155–166, Luty 2009.
- [43] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y.-H. Liu. LPD-Net: 3D Point Cloud Learning for Large-Scale Place Recognition and Environment Analysis. W *Proc. of the IEEE Inter. Conf. on Computer Vision*, s. 2831–2840, 2019.
- [44] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. W *7th International Conference on Learning Representations, ICLR*, 2019, New Orleans, LA, USA, Maj 6-9, 2019.
- [45] S. Lovegrove, A. Patron-Perez, and G. Sibley, Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras, W *British Machine Vision Conference (BMVC)*, 2013.
- [46] M.M. Michalek, T. Gawron, M. Nowicki, P. Skrzypczyński, Precise Docking at Charging Stations for Large-Capacity Vehicles: An Advanced Driver-Assistance System for Drivers of Electric Urban Buses. *IEEE Vehicular Technology Magazine*, 16(3), s. 57–65, 2021.
- [47] P. Neubert, N. Sünderhauf, and P. Protzel, Appearance change prediction for long-term navigation across seasons. W *2013 European Conference on Mobile Robots (ECMR)*, s. 198–203, Barcelona, 2013.
- [48] T. Nowak, M.R. Nowicki, K. Ówian, and P. Skrzypczyński, How to Improve Object Detection in a Driver Assistance System Applying Explainable Deep Learning. W *2019 IEEE Intelligent Vehicles Symposium (IV)*, Paris, Francja, s. 226–231, 2019.
- [49] M. Nowicki, D. Belter, A. Kostusiak, P. Čížek, J. Faigl, P. Skrzypczyński, An Experimental Study On Feature-Based SLAM For Multi-Legged Robots With RGB-D Sensors. *Industrial Robot: An International Journal*, 44(4), s. 428–441, 2017.
- [50] Y. Pan, X. Xu, W. Li, Y. Wang, and R. Xiong. CORAL: Colored structural representation for bi-modal place recognition. W *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, s. 2084–2091, 2021.
- [51] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. W *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, s. 3711–3719, 2017.

- [52] A. Patron-Perez, S. Lovegrove, and G. Sibley. A spline-based trajectory representation for sensor fusion and rolling shutter cameras. W *International Journal of Computer Vision*, vol. 113, no. 3, s. 208–219, 2015.
- [53] J. Peršić, I. Marković, and I. Petrović. Extrinsic 6DoF calibration of a radar–LiDAR–camera system enhanced by radar cross section estimates evaluation. *Robotics and Autonomous Systems*, 114, s. 217–230, 2019.
- [54] J. Rehder, R. Siegwart, and P. Furgale, A general approach to spatiotemporal calibration in multisensor systems, W *IEEE Transactions on Robotics*, vol. 32, no. 2, 2016, s. 383–398.
- [55] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart. Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes. W *2016 IEEE International Conference on Robotics and Automation (ICRA)*, s. 4304–4311, 2016.
- [56] J. Rehder and R. Siegwart. Camera/IMU Calibration Revisited. *IEEE Sensors Journal*, 17(11), s. 3257–3268, 2017.
- [57] A. Roennau, G. Heppner, M. Nowicki, J.M. Zoellner, R. Dillmann. Reactive posture behaviors for stable legged locomotion over steep inclines and large obstacles. W *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, n, s. 4888–4894, Wrzesień 2014.
- [58] A. Roennau, G. Heppner, M. Nowicki, R. Dillmann. LAURON V: A versatile six-legged walking robot with advanced maneuverability. W *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* s. 82–87, Lipiec 2014.
- [59] K. Roszyk, M.R. Nowicki, P. Skrzypczyński. Adopting the YOLOv4 Architecture for Low-Latency Multispectral Pedestrian Detection in Autonomous Driving. *Sensors*, 22, 1082, 2022.
- [60] C. Schöllner, M. Schnettler, A. Krämmer, G. Hinz, M. Bakovic, M. Güzet, and A. Knoll. Targetless Rotational Auto-Calibration of Radar and Camera for Intelligent Transportation Systems. W *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, s. 3934–3941, 2019.
- [61] E. Schubert, J. Sander, M. Ester, H.P. Kriegel, and X. Xu. DBSCAN Revisited: Why and how you should (still) use DBSCAN. *ACM Trans. Database Syst.*, 42(3), 2017.
- [62] A. Segal, D. Hhnel, and S. Thrun. Generalized-ICP, W *Robotics: Science and Systems*, vol. 2, no. 4, s. 435, 2009.
- [63] P. Stefaniak, A. Skoczylas, M. Nowicki, K. Walas, M. Stachowiak. Semi-autonomous inspection of a belt conveyor in an industrial environment with the use of a legged robot. W *APCOM (Applications of Computers and Operations Research in the Minerals Industries)*, Johannesburg, Republika Południowej Afryki, 2021.

- [64] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers Efficient derivative computation for cumulative B-splines on Lie groups. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, s. 11145–11153, 2020.
- [65] Q. Sun, H. Liu, J. He, Z. Fan, and X. Du. DAGC: Employing Dual Attention and Graph Convolution for Point Cloud Based Place Recognition. W *Proc. of the 2020 Inter. Conf. on Multimedia Retrieval, ICMR '20*, s. 224–232, New York, NY, USA, 2020. Association for Computing Machinery.
- [66] M. Tan and Q.V. Le. Efficientnetv2: Smaller models and faster training. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 Lipiec 2021, Virtual Event*, vol. 139 of *Proceedings of Machine Learning Research*, s. 10096–10106. PMLR, 2021.
- [67] u-blox. ZED-F9P, u-blox F9 high precision GNSS module, 2020.
- [68] M.A. Uy and G.H. Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. W *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, s. 4470–4479, 2018.
- [69] G. Valsecchi, R. Grandia, and M. Hutter. Quadrupedal locomotion on uneven terrain with sensorized feet. *IEEE Robotics and Automation Letters*, 5(2), s. 1548–1555, 2020.
- [70] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin. Attention is All you Need. W *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017.
- [71] K. Walas, M. Nowicki, D. Ferstl, P. Skrzypczyński. Depth Data Fusion for Simultaneous Localization and Mapping – RGB-DD SLAM. W *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, s. 9–14, Kongresshaus Baden-Baden, Niemcy, 2016.
- [72] Y. Xia, Y. Xu, S. Li, R. Wang, J. Du, D. Cremers, and U. Stilla. SOE-Net: A Self-Attention and Orientation Encoding Network for Point Cloud based Place Recognition. W *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s. 11348–11357, Czerwiec 2021.
- [73] T.X. Xu, Y.C. Guo, Y.K. Lai, and S.H. Zhang. TransLoc3D: Point Cloud based Large-scale Place Recognition using Adaptive Receptive Fields. *arXiv preprint arXiv:2105.11605*, 2021.
- [74] W. Zhang and C. Xiao. PCAN: 3D Attention Map Learning Using Contextual Information for Point Cloud Based Retrieval. W *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Czerwiec 2019.

- [75] J. Zhang, and S. Singh. LOAM: Lidar odometry and mapping in real-time. W *Robotics: Science and Systems Conference (RSS)*, Pittsburgh, PA, 2014.
- [76] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. W *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, s. 2344–2351, Grudzień 2013.
- [77] Y. Zhou and O. Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. W *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, s. 4490–4499, 2018.
- [78] Z. Zhou, C. Zhao, D. Adolfsson, S. Su, Y. Gao, T. Duckett, and L. Sun. NDT-transformer: Large-scale Dd point cloud localisation using the normal distribution transform representation. W *2021 IEEE International Conference on Robotics and Automation (ICRA)*, s. 5654–5660, 2021.



.....  
Michał Nowicki