

Dominik Belter
Politechnika Poznańska
Wydział Elektryczny

Załącznik 2

Autoreferat przedstawiający opis dorobku i osiągnięć naukowych

25 kwietnia 2019

1. **Imię i nazwisko**

Dominik Belter

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

2012 r. Doktor nauk technicznych

Dyscyplina naukowa: automatyka i robotyka

Politechnika Poznańska

Wydział Elektryczny

Tytuł rozprawy doktorskiej: Gait control of the six-legged robot on a rough terrain using computational intelligence learning and optimization methods (Sterowanie kroczeniem robota sześcionożnego po nierównym terenie z wykorzystaniem uczenia i optymalizacji metodami inteligencji obliczeniowej)

Promotor: dr hab. inż. Piotr Skrzypczyński

Recenzenci: prof. dr hab. inż. Józef Korbicz (Uniwersytet Zielonogórski), prof. dr hab. inż. Andrzej Królikowski (Politechnika Poznańska)

2007 r. Magister inżynier

Kierunek: automatyka i robotyka

Specjalność: robotyka

Politechnika Poznańska

Wydział Elektryczny

Tytuł pracy dyplomowej: Projekt i wykonanie robota kroczącego z pokładowym systemem wizyjnym

Promotor: dr hab. inż. Andrzej Kasiński

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

- Instytut Automatyki, Robotyki i Inżynierii Informatycznej Politechniki Poznańskiej, adiunkt od 01.10.2012 r.
- School of Computer Science, University of Birmingham, postdoctoral researcher w okresach: od 01.02.2013 r. do 18.03.2013 r., od 01.06.2013 r. do 30.09.2013 r., od 07.09.2015 r. do 29.02.2016 r.
- Instytut Automatyki, Robotyki i Inżynierii Informatycznej Politechniki Poznańskiej, asystent od 01.10.2007 r. do 30.09.2012 r.

4. Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. 2017 r. poz. 1789):

a) **Tytuł osiągnięcia naukowego**

Cykl publikacji powiązanych tematycznie pod wspólną nazwą:

Percepcja, planowanie i sterowanie ruchem robota kroczącego poruszającego się po nierównym terenie

Osiągnięcie habilitacyjne stanowi cykl 12 publikacji przedstawionych poniżej.

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

[A1] **D. Belter (100%)**

Optimization-based Approach for Motion Planning of a Robot Walking on Rough Terrain

Journal of Automation, Mobile Robotics & Intelligent Systems, Vol. 7, No. 4, s. 34–41, 2013

DOI: 10.14313/JAMRIS_4-2013/35

(MNiSW lista B: 12 pkt.¹)

[A2] **D. Belter (60%), P. Łabęcki (20%), P. Fankhauser (10%), R. Siegwart (10%)**

RGB-D Terrain Perception and Dense Mapping for Legged Robots

International Journal of Applied Mathematics and Computer Science, vol. 26(1), s. 81–97, 2016

DOI: 10.1515/amcs-2016-0006

(MNiSW lista A: 25 pkt., IF₂₀₁₆=1.420)

[A3] **D. Belter (75%), P. Skrzypczyński (25%)**

Precise Self-Localization of a Walking Robot on Rough Terrain Using Parallel Tracking and Mapping

Industrial Robot: An International Journal, Vol. 40, No. 3, s. 229–237, 2013

DOI: 10.1108/01439911311309924

(MNiSW lista A: 20 pkt., IF₂₀₁₃=0.622)

[A4] **D. Belter (60%), P. Łabęcki (10%), P. Skrzypczyński (30%)**

Adaptive Motion Planning for Autonomous Rough Terrain Traversal with a Walking Robot

Journal of Field Robotics, vol. 33(3), s. 337–370, 2016

¹Wartość Impact Factor (IF) podana zgodnie z rokiem publikacji lub wg IF dla roku 2017 w przypadku publikacji z lat 2018-2019 (z tych lat IF jest jeszcze niedostępny) – źródło: Journal Citation Report. Punkty MNiSW podano zgodnie z rokiem publikacji. Procentowy wkład habilitanta w powstanie poszczególnych prac udokumentowano oświadczeniami współautorów (załącznik nr 5), natomiast w załączniku nr 4 habilitant wyszczególnia swój najważniejszy wkład merytoryczny prac.

DOI: 10.1002/rob.21610
(MNiSW lista A: 40 pkt., IF₂₀₁₆=4.882)

- [A5] **D. Belter (100%)**
Efficient Reactive Behavior for Six-legged Walking on Rough Terrain with Proprioceptive Sensing
Human-Centric Robotics, World Scientific, M.E. Silva et al. (Eds.), Singapore, s. 357–366, 2017
10.1142/9789813231047_0044
(MNiSW: 0 pkt., konferencja z 2017 roku nie jest jeszcze zaindeksowana w Web of Science)
- [A6] **D. Belter (60%), M. Nowicki (10%), P. Skrzypczyński (30%)**
Modeling Spatial Uncertainty of Point Features in Feature-based RGB-D SLAM
Machine Vision and Applications, vol. 29(5), s. 827–844, 2018
10.1007/s00138-018-0936-9
(MNiSW lista A: 25 pkt., IF₂₀₁₇=1.306)
- [A7] **M.R. Nowicki (17%), D. Belter (17%), A. Kostusiak (14%), P. Cížek (16%), J. Faigl (16%), P. Skrzypczyński (20%)**
An Experimental Study on Feature-based SLAM for Multi-Legged Robots with RGB-D Sensors
Industrial Robot: An International Journal, Vol. 44, No. 4, s. 406–415, 2017
10.1108/IR-11-2016-0340
(MNiSW lista A: 20 pkt., IF₂₀₁₇=1.205)
- [A8] **D. Belter (85%), M. R. Nowicki (15%)**
Optimization-based Legged Odometry and Sensor Fusion for Legged Robot Continuous Localization
Robotics and Autonomous Systems, Vol. 111, s. 110–124, 2019
10.1016/j.robot.2018.10.013
(MNiSW lista A: 30 pkt., IF₂₀₁₇=2.638)
- [A9] **D. Belter (86%), K. Piaskowski (7%), R. Staszak (7%)**
Keyframe-based Local Normal Distribution Transform Occupancy Maps for Environment Mapping
IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), s. 706–712, 2018
10.1109/ETFA.2018.8502517
(MNiSW: 15 pkt., konferencja indeksowana cyklicznie w Web of Science)
- [A10] **D. Belter (65%), J. Wietrzykowski (20%), P. Skrzypczyński (15%)**
Employing Natural Terrain Semantics in Motion Planning for a Multi-Legged Robot
Journal of Intelligent & Robotic Systems, Vol. 93(3–4), s. 723–743, 2019

10.1007/s10846-018-0865-x

(MNiSW lista A: 25 pkt., IF₂₀₁₇=1.583)

[A11] M. Wasielica (20%), D. Belter (80%)

RRT-based Motion Planning and Balance Controller for a Biped Robot
Advances in Cooperative Robotics, World Scientific, M.O. Tokhi and G.S.
Virk (Eds.), Singapore, s. 404–411, 2016

10.1142/9789813149137_0048

(MNiSW: 15 pkt., konferencja indeksowana cyklicznie w Web of Science)

[A12] D. Belter (55%), J. Bednarek (25%), H.-C. Lin (7.5%), G. Xin (7.5%), M. Mistry (5%)

Single-shot Foothold Selection and Constraint Evaluation for Quadruped Locomotion

IEEE International Conference on Robotics and Automation, s. 7441–7447, 2019

(MNiSW: 0 pkt., konferencja z 2019 roku nie jest jeszcze zaindeksowana w Web of Science)

Na jednotematyczny cykl składa się 12 publikacji, wśród których 7 zostało opublikowanych w czasopismach indeksowanych w bazie Journal Citation Reports (JCR). Dwie publikacje w zbiorze to prace autorskie. Sumaryczny współczynnik wpływu czasopism (Impact Factor) publikacji wchodzących w skład osiągnięcia wynosi 13,656. Sumaryczna liczba punktów zgodnie z wykazem czasopism naukowych przygotowanym przez Ministerstwo Nauki i Szkolnictwa Wyższego (MNiSW) za publikacje z listy JCR oraz wszystkie publikacje w cyklu – odpowiednio 185 oraz 227. Uśredniony, procentowy udział habilitanta w publikacjach wynosi 70,25%.

- c) **Omówienie celu naukowego ww. prac i osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania**

4.1 Motywacja i cel naukowy podjętych badań

Tematyka przedstawionego cyklu publikacji, będącego podstawą wniosku habilitacyjnego, dotyczy percepcji, planowania i sterowania ruchem robota kroczonego poruszającego się po nierównym terenie. Zagadnienia badawcze podjęte w ramach cyklu wynikają bezpośrednio z wcześniejszych wyników naukowych i ograniczeń metod zaproponowanych w doktoracie. Przedmiotem rozprawy doktorskiej było sterowanie kroczeniem robota sześcionożnego z wykorzystaniem metod inteligencji obliczeniowej. Wynikiem był sterownik ruchu umożliwiający generowanie chodu na podstawie modelu kinematycznego robota. Powstał również system uczenia chodu, który umożliwia syntezę kroczenia robota sześcionożnego w symulatorze z wykorzystaniem algorytmów ewolucyjnych, a następnie przeniesienie uzyskanych wyników na rzeczywistego robota. Prace te doprowadziły do powstania metody planowania ruchu, która umożliwia określenie ścieżki poruszania się robota po nierównym terenie.

Opracowane rozwiązanie ma jednak istotne wady, które wykazano w późniejszych pracach. Wcześniejsze prace zakładają, że robot wykorzystuje mapę terenu zbudowaną przez niezależny system. Założono również, że pozycja robota jest znana podczas wykonywania ruchu. Do uzyskania wyników na nierównym terenie wykorzystano system budowy mapy z użyciem pochylonego skanera laserowego. Do integracji skanów konieczna jest znajomość pozycji robota podczas wykonywania pomiaru. Ze względu na brak takiego systemu wykorzystano pozycję wynikającą z rozkazów generowanych przez planer ruchu. To powoduje, że zarówno estymowany ruch robota, jak i wynikająca z niego mapa były niedokładne. Konsekwencją niedoskonałego systemu percepcji robota jest niska skuteczność algorytmu planowania ruchu w rzeczywistych warunkach. Wnioskiem wynikającym z pierwszych eksperymentów na nierównym terenie jest konieczność opracowania nowego systemu lokalizacji i współpracującego z nim systemu budowy mapy. Tylko dokładny i działający niezawodnie system percepcji umożliwi skuteczne planowanie i poruszanie się robota w nieznanym *a priori* i nieuporządkowanym środowisku.

W chwili rozpoczęcia prac nad systemem percepcji robota kroczonego za najskuteczniejszy wizyjny system lokalizacji uchodził MonoSLAM opracowany przez Andrew Davisona z zespołem [2]. Jest to system, który na podstawie cech wizyjnych wykrytych na obrazie monochromatycznym określa ruch kamery w przestrzeni 3D. W tym czasie pojawiły się niedrogie kamery RGB-D, czyli czujniki, które poza obrazem kolorowym zwracają również obraz głębi. Zostały one bardzo szybko wykorzystane do opracowania znacznie bardziej niezawodnych systemów lokalizacji jak np. RGB-D SLAM [4]. Nadal jednak rozwijane były systemy monokularowe. Przykładem jest ORB SLAM [15], który wykorzystuje cechy punktowe ORB i minimalizację błędu reprojekcji do estymacji ruchu kamery, oraz LSD SLAM [5], który zamiast cech punktowych bezpośrednio minimalizuje błąd fotometryczny na obrazie. Odzyskanie skali, czyli informacji metrycznej, ze wspomnianych systemów lokalizacji (MonoSLAM, ORB-SLAM i LSD-SLAM) jest trudne, jednak bezpośrednio wykorzystanie czujników RGB-D ułatwia to zadanie. Przykładem metody, która skutecznie lokalizuje roboty mobilne w nieznanym środowisku i zwraca informację metryczną jest metoda ORB-SLAM2 [14].

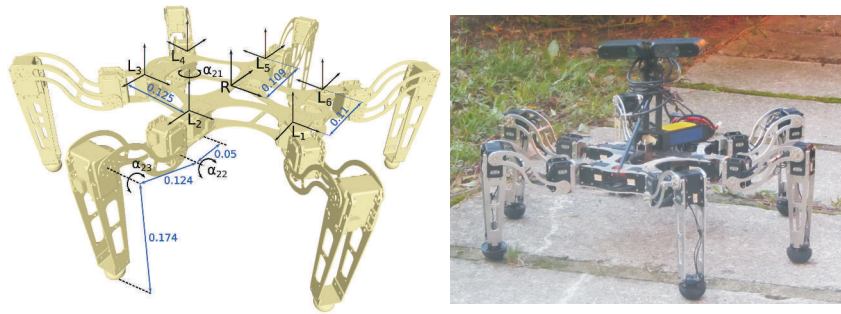
Do modelowania otoczenia robota kroczonego wybrano mapę wysokościową. Jest to bardzo

wydajna reprezentacja umożliwiająca efektywne uzyskanie informacji o kształcie terenu podczas planowania ruchu. W trakcie rozwijania metod percepcji dla robota krocącego pojawiły się nowe metody modelowania otoczenia. Przykładem jest metoda OctoMap [8], która wykorzystuje podział przestrzeni na sześciany do modelowania przeszkód w otoczeniu robota. Również algorytm NDT-OM (ang. *Normal Distributions Transform Occupancy Maps*) wykorzystuje podział przestrzeni na sześcienne komórki (voxele), jednak poza informacją o zajętości komórki przechowywana jest również elipsoida, która reprezentuje rozproszenie pomiarów wewnątrz voxela [17]. W trakcie prac nad syntezą systemu percepcji robota wykorzystywałem podobne mechanizmy do lepszej lokalizacji i modelowania otoczenia robota lub bezpośrednio korzystałem z metody NDT-OM jako elementu składowego nowego systemu modelującego otoczenie [A8].

Głównym celem badań było opracowanie i walidacja eksperymentalna nowych metod lokalizacji, budowy mapy i planowania ruchu robota krocącego. Główny nacisk położony został na percepcję robota. To możliwość postrzegania przez robota obiektów wokół siebie i możliwość modelowania ich kształtu i znaczenia (rozumiana tutaj jako klasyfikacja terenu) jest kluczowa do skutecznego planowania ruchu i poruszania się robota w naturalnym środowisku. Jednocześnie lokalizowanie robota, który ze względu na właściwości terenu ulega częstym poślizgom, jest kluczowe do modelowania terenu, a później wykonania ruchu. Dlatego również planer ruchu powinien wykorzystywać właściwości systemu percepcji, żeby skutecznie i bezpiecznie przeprowadzić robota z punktu początkowego do docelowego.

Cele szczegółowe związane z cyklem publikacji można zdefiniować następująco:

- Opracowanie i implementacja precyzyjnego i skutecznego systemu lokalizacji robota krocącego umożliwiającego określenie pozycji robota (położenia i orientacji) w naturalnym środowisku. Skuteczność systemu lokalizacji wiąże się z możliwością działania systemu w różnych warunkach zewnętrznych. System taki powinien być odporny na zmienne warunki oświetleniowe, brak danych o cechach wizyjnych, brak danych o głębi sceny, czyli sytuacje, które występują bardzo często w rzeczywistych warunkach pracy robota krocącego pracującego na zewnątrz budynków.
- Opracowanie i implementacja metody modelowania otoczenia z wykorzystaniem danych z kamery RGB-D i informacji o pozycji robota. System budowy mapy powinien wykorzystywać właściwości systemu lokalizacji do dokładnego reprezentowania otoczenia oraz powinien umożliwiać konwersję reprezentacji wewnętrznej mapy na reprezentację wykorzystywaną do planowania ruchu robota (mapę wysokościową).
- Opracowanie i implementacja metody planowania ruchu, która będzie wykorzystywała informacje zawarte w mapie otoczenia robota. Metoda ta powinna dopasowywać sposób planowania ruchu do przewidywanej dokładności danych zawartych w mapie. W przypadku dostępnych danych semantycznych robot powinien dostosowywać ścieżkę do rodzaju terenu oraz modyfikować sposób kroczenia. Podczas wykonywania ruchu robot powinien dostosować swoje działanie w zależności od dostępnych danych sensorycznych. W przypadku braku informacji z systemu wizyjnego i informacji o mapie terenu, robot powinien poruszać się na podstawie danych z sensorów proprioceptywnych (jednostki inercyjnej, enkoderów, czujników kontaktu).



Rysunek 1: Struktura kinematyczna robota Messor II oraz robot wyposażony w kamerę RGB-D Asus Xtion Pro

Postawione cele udało się osiągnąć dzięki wykorzystaniu zbioru następujących narzędzi:

- optymalizacji typu *black box*, np. Particle Swarm Optimization (PSO) [10] i Covariance Matrix Adaptation-Evolutionary Strategy (CMA-ES) [9],
- optymalizacji gradientowej [11],
- optymalizacji grafów ograniczeń [13],
- modelowaniu rozkładu prawdopodobieństwa za pomocą metody Kernel Density Estimation (KDE) i wnioskowaniu probabilistycznemu [7],
- oraz standardowym metodom używanym w robotyce, jak na przykład Filtr Kalmana, analiza głównych składowych (ang. *Principal Component Analysis*, PCA), mieszaniny Gaussowskie (ang. *Gaussian Mixture*, GM), algorytm Umeyamy [18].

Przedstawiony w autoreferacie cykl publikacji przedstawia ścieżkę dojścia do wymienionych rozwiązań z użyciem wspomnianych narzędzi.

4.2 Wprowadzenie do tematyki badań

Przedmiotem badań są wielonożne roboty kroczące. Większość badań eksperymentalnych wykonano na robocie sześcionożnym Messor II przedstawionym na rysunku 1. Każda z nóg robota wyposażona jest w trzy serwomechanizmy umożliwiające zadawanie pozycji w węźle kinematycznym. Kinematyka nogi oparta jest o strukturę antropomorficzną. Sterownik robota oparty został o układ PandaBoard ES z zainstalowanym systemem Linux. Umożliwia on uruchomienie głównego sterownika kinematycznego na pokładzie robota. Większość obliczeń związanych z budowaniem mapy i lokalizacją robota dokonywana jest na komputerze zewnętrznym.

Pokładowy system sensoryczny robota składa się z zestawu czujników proprioceptywnych i zewnętrznych (kamery RGB-D). Do czujników wewnętrznych mierzących stan robota zaliczamy enkodery umieszczone w serwomechanizmach, które zwracają informację o kącie obrotu, jednostkę inercyjną AHRS (ang. *Attitude and Heading Reference System*) Xsens MTi oraz czujniki kontaktu stóp z podłożem.

Jako czujniki otoczenia wykorzystane zostały różne czujniki RGB-D. Większość doświadczeń wykonana została z czujnikiem Asus Xtion. Jest to czujnik wyposażony w kamerę RGB, kamerę działającą w zakresie podczerwieni oraz oświetlacz na podczerwień. Oświe-

tlacz wyświetla losowy ale stały wzór, który jest następnie dopasowywany na obrazie z kamery na podczerwień i obliczana jest odległość do przeszkód. Część eksperymentów została wykonana również z użyciem pasywnej kamery stereo, czujnika Kinect v2 oraz kamery Intel RealSense D435. W przypadku kamery stereo dopasowywanie punktów wykonywane jest na obrazach kolorowych, bez użycia oświetlacza na podczerwień. Czujnik Kinect v2 pośrednio mierzy czas przelotu światła pomiędzy kamerą a przeszkodami i jest najbardziej dokładny z wymienionych czujników. Kamera Intel RealSense jest połączeniem kamery z oświetleniem strukturalnym z pasywną kamerą stereo. Dlatego czujnik ten zwraca dane o głębi wewnątrz budynku jak i na zewnątrz przy pełnym oświetleniu słonecznym. Czujnik ten został wykorzystany głównie podczas eksperymentów na zewnątrz budynku. Algorytmy i metody prezentowane w autoreferacie działają z każdym z tych sensorów, ponieważ metody wykorzystują obraz kolorowy RGB oraz obraz głębi, zwracany przez każdy z tych czujników.

Punktem wyjścia do prowadzonych badań jest algorytm planowania ruchu oparty o planer RRT (ang. *Rapidly-exploring Random Trees*). W pracy [A1] problem planowania ruchu w 24-wymiarowej przestrzeni poszukiwań określony został jako sekwencja formalnie zdefiniowanych problemów optymalizacji. W pierwszej kolejności robot wybiera punkty podparcia, które są określane na podstawie rastrowej mapy wysokościowej [1]. Podczas wyboru punktów podparcia orientacja korpusu jest prostopadła względem wektora grawitacji. Następnie wyznaczane są kąty roll_R , pitch_R korpusu robota oraz jego wysokość w układzie mapy:

$$\underset{\mathbf{p}_R}{\operatorname{argmax}} d_l^{\text{KM}}(\mathbf{q}^{24\text{D}}), \mathbf{q}^{24\text{D}} \in C_{\text{free}}, C_{\text{stab}}, C_{\text{work}} \quad (1)$$

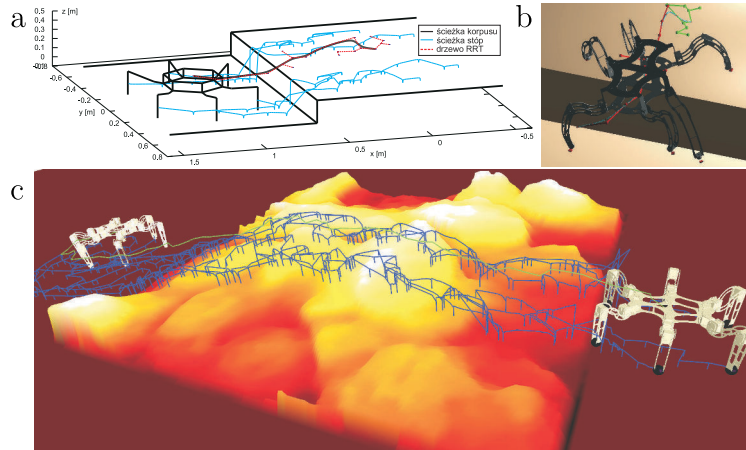
gdzie: $\mathbf{p}_R = [\text{roll}_R, \text{pitch}_R, z]$, $d_l^{\text{KM}}(\mathbf{q}^{24\text{D}})$ jest marginesem kinematycznym robota dla danej konfiguracji robota \mathbf{q} , C_{free} jest przestrzenią wolną od kolizji, C_{work} jest przestrzenią roboczą robota, C_{stab} jest statycznie stabilną konfiguracją robota. Problem optymalizacji (1) jest rozwiązywany za pomocą metody PSO i zajmuje średnio 100 ms.

Ponieważ wstępna ścieżka ruchu stóp w trakcie przenoszenia nogi nad przeszkodami jest definiowana w układzie mapy, może dochodzić do sytuacji, w której zadana pozycja stopy jest nieosiągalna. Aby zapobiec takim sytuacjom, iteracyjnie znajdowana jest najbliższa pozycja stopy wewnątrz przestrzeni roboczej nogi. Następnie z użyciem optymalizacji gradientowej znajdowana jest konfiguracja nogi, która spełnia następujące wymagania:

$$d_l^{\text{KM}}(\mathbf{q}_l) > d_{\text{safe}}^{\text{KM}} \in C_{\text{free}}, \quad (2)$$

gdzie d_l^{KM} jest marginesem kinematycznym nogi dla konfiguracji \mathbf{q}_l , a d_{safe} jest ustaloną wartością bezpieczną marginesu kinematycznego nogi.

Ścieżka ruchu robota uzyskana z algorytmu RRT jest wynikiem złożenia ruchów prostoliniowych. W wyniku tego robot porusza się ruchem, który nie jest płynny. Dlatego w [A1] zaproponowałem procedurę wygładzania ścieżki ruchu robota. Celem procedury wygładzania ścieżki jest minimalizacja funkcji kosztu F_{sm} . Argumentem funkcji F_{sm} są kolejne po-



Rysunek 2: Ścieżka ruchu robota zaplanowana podczas pokonywania stopnia (a), wizualizacja stanu robota podczas wykonywania zadanej ścieżki (b) oraz ścieżka ruchu zaplanowana na makiecie nierównego terenu

zycje korpusu robota \mathbf{p} określone przez zaproponowaną metodę planowania ruchu i punkty wygładzonej ścieżki \mathbf{p}' :

$$\arg \min_{\mathbf{p}'} F_{\text{sm}}(\mathbf{p}') = \sum_{i=0}^n \left\{ \frac{\alpha}{2} (p_i - p'_i)^2 + \frac{\beta}{2} (p'_i - p'_{i-1})^2 + \frac{\beta}{2} (p'_i - p'_{i+1})^2 \right\}. \quad (3)$$

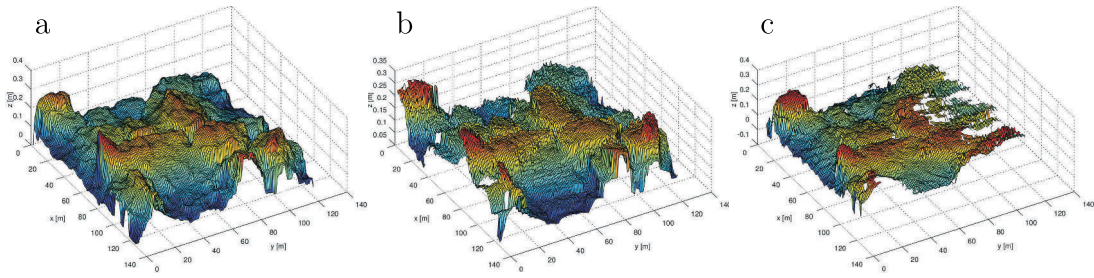
Parametry α i β określają stopień wygładzenia ścieżki.

Przykładowe wyniki uzyskane z planera ruchu przedstawione zostały na rysunku 2. Algorytm ten pozwala na efektywne planowanie ruchu robota na nierównym terenie przy założeniu, że znany jest precyzyjny model otoczenia. W pracy [A1] nie przedstawiono wyników eksperymentalnych na rzeczywistym robocie. Wynika to z braku wystarczająco precyzyjnego systemu budowy mapy i lokalizacji robota. Wstępne eksperymenty na robocie pokazały jednak konieczność opracowania takich systemów, aby możliwe było uruchomienie planowania ruchu na rzeczywistym robocie.

4.3 System budowy mapy

W pracy [A2] zaadoptowany został system budowy mapy opracowany wcześniej dla dalmierza laserowego 2D [1]. System ten opiera się o zestaw filtrów Kalmana dla mapy voxelowej. Filtry Kalmana estymują wysokość terenu dla każdego voxela. Następnie wysokość estymowana w najwyższym voxelu znajdującym się nad każdą komórką w płaszczyźnie poziomej xy jest wykorzystywana do estymacji odpowiadających komórek mapy wysokościowej.

W ramach pracy [A2] porównałem dwa rodzaje czujników RGB-D: Asus Xtion i kamerę stereo. Przykładowe wyniki porównania znajdują się na rysunku 3. Na rysunku 3a pokazana została mapa referencyjna terenu zeskanowanego przez dwa sensory. Na rysunku 3b i rysunku 3c pokazano odpowiednio mapę uzyskaną z kamery Asus Xtion i kamery stereo. Porównanie ilościowe przedstawione w pracy [A2] pokazuje większą dokładność czujnika Asus Xtion. Przeprowadziłem również dodatkowe eksperymenty systemu budowy mapy dla



Rysunek 3: Porównanie map wysokościowych uzyskanych z różnych czujników RGB-D: mapa referencyjna (a), mapa uzyskana z czujnika Asus Xtion (b) i mapa uzyskana z kamery stereo (c)

kamery umieszczonej na robocie krocącym bez systemu lokalizacji, na statywie przemieszczanym o znaną odległość oraz na ramieniu przemysłowym KUKA, żeby określić wpływ błędu lokalizacji kamery na dokładność budowy mapy.

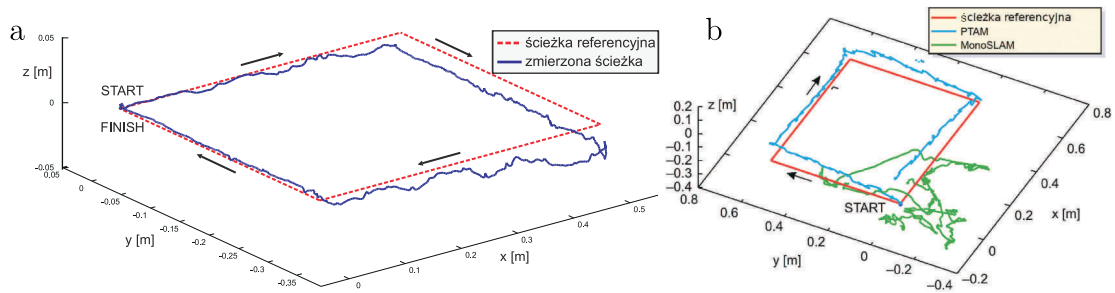
Dodatkowo system budowy mapy został porównany z innymi metodami z literatury: NDT-OM [17] i OctoMap [8]. Ponieważ porównywane systemy budują mapy 3D, zostały one zmodyfikowane, tak aby możliwa była konwersja do mapy wysokościowej. Uzyskane wyniki pokazują wyższą dokładność mapy wysokościowej utworzonej za pomocą proponowanej metody w porównaniu z metodami NDT-OM i OctoMap. Przeprowadzono również porównanie z metodą budowy mapy wysokościowej opracowanej dla robota StarLETH [6]. Eksperymenty na robocie StarLETH zostały wykonane przez zespół z Autonomous Systems Lab w Zurychu. Tutaj jednak obie metody uzyskały zbliżone wyniki.

Wstępne eksperymenty przeprowadzone na robocie krocącym pokazują również, że możliwe jest uzyskanie mapy wystarczającej do precyzyjnego planowania ruchu robota krocącego. Konieczne jest jednak precyzyjne estymowanie pozycji robota w trakcie wykonywania ruchu.

4.4 System lokalizacji robota krocącego wykorzystujący algorytm PTAM

W pracy [A3] podjąłem się zagadnienia implementacji i testowania dwóch wybranych algorytmów wizyjnej lokalizacji. Przetestowany został algorytm MonoSLAM [2] oraz PTAM [12]. MonoSLAM jest systemem monokularowym, który wykorzystuje lokalne cechy wizyjne do estymowania ruchu kamery. Do śledzenia cech i ruchu kamery zastosowany został rozszerzony filtr Kalmana (ang. *Extended Kalman Filter*). Ze względu na ograniczenia wynikające z rozmiaru wektora stanu filtru Kalmana śledzona jest niewielka liczba cech. Odmierna strategia została wykorzystana w algorytmie PTAM. System ten wykorzystuje optymalizację typu *bundle adjustment* do estymowania pozycji cech w przestrzeni 3D i ruchu kamery. Żeby możliwe było działanie algorytmu w czasie rzeczywistym, PTAM operuje na ramkach kluczowych (ang. *keyframes*) zawierających zestaw cech punktowych. Implementacja oparta jest na dwóch wątkach, z których jeden zajmuje się śledzeniem cech, a drugi optymalizacją *bundle adjustment* na ramkach kluczowych.

W ramach prac związanych z systemem lokalizacji wyznaczyłem teoretyczną wymaganą



Rysunek 4: Porównanie ścieżki referencyjnej i ścieżki ruchu kamery uzyskanej z systemu PTAM podczas eksperymentu z robotem KUKA (a) oraz porównanie algorytmów PTAM i MonoSLAM na robocie krocącym (b)

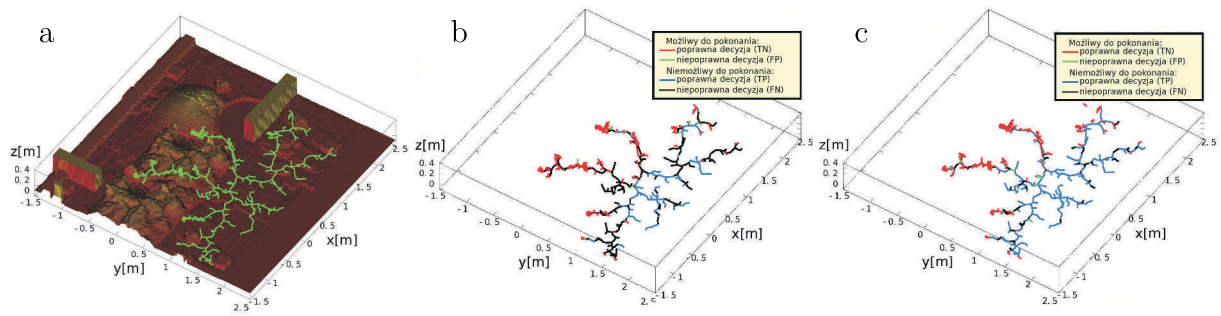
dokładność systemu lokalizacji, potrzebną do poprawnego poruszania się robota po nierównym terenie. W tym celu zakłócałem pozycję robota szumem o znanym rozkładzie. Podczas gdy błąd lokalizacji jest mniejszy niż 2 cm, skuteczność sterownika ruchu pozostaje na poziomie 90%. Później skuteczność gwałtownie spada i robot nie jest w stanie poprawnie realizować zadanego ruchu z wykorzystaniem sprzężenia zwrotnego od systemu lokalizacji.

Następnie w pracy [A3] porównałem dwa wybrane systemy lokalizacji. Kamera RGB została umieszczona na ramieniu robota KUKA KR200, który dostarcza informację o rzeczywistym położeniu kamery. Eksperyment pokazał, że system PTAM jest w stanie lokalizować robota z wybraną precyzją. Porównanie ścieżki referencyjnej i ścieżki uzyskanej z systemu PTAM znajduje się na rysunku 4a. Wyniki te zostały również potwierdzone podczas eksperymentów na rzeczywistym robocie krocącym. System MonoSLAM bardzo szybko tracił możliwość śledzenia ruchu kamery. Wynika to z małej liczby cech, które są śledzone przez ten system. Porównanie obu metod podczas eksperymentu na robocie krocącym przedstawione zostało na rysunku 4b.

Warto również wspomnieć, że wybrane systemy są systemami monokularowymi i nie zwracają informacji metrycznej. Żeby odzyskać skalę otrzymanej trajektorii, zaproponowana została odpowiednia procedura kalibracyjna wykorzystująca zdefiniowane ruchy korpusu robota krocącego. Dodatkowym ograniczeniem systemu PTAM jest możliwość pracy tylko w niewielkim obszarze. PTAM posiada mechanizmy, które umożliwiają jego pracę w większym obszarze roboczym. Niestety mechanizmy te często zawodzą i wtedy robot traci informację o swoim położeniu.

4.5 Adaptacyjny system planowania ruchu

Opracowanie działających systemów budowy mapy i lokalizacji dla robota krocącego umożliwiło przeprowadzenie szerokiej walidacji eksperymentalnej algorytmu planowania ruchu. W pracy [A4] rozwinięty został algorytm guidedRRT, który wykorzystuje dwuetapowe planowanie ruchu. Planowanie ruchu dostosowane jest do systemu percepcji robota krocącego. Ponieważ obiekty znajdujące się dalej od robota reprezentowane są w mapie ze znacznie mniejszą dokładnością niż obiekty, które znajdują się blisko robota, w pierwszym



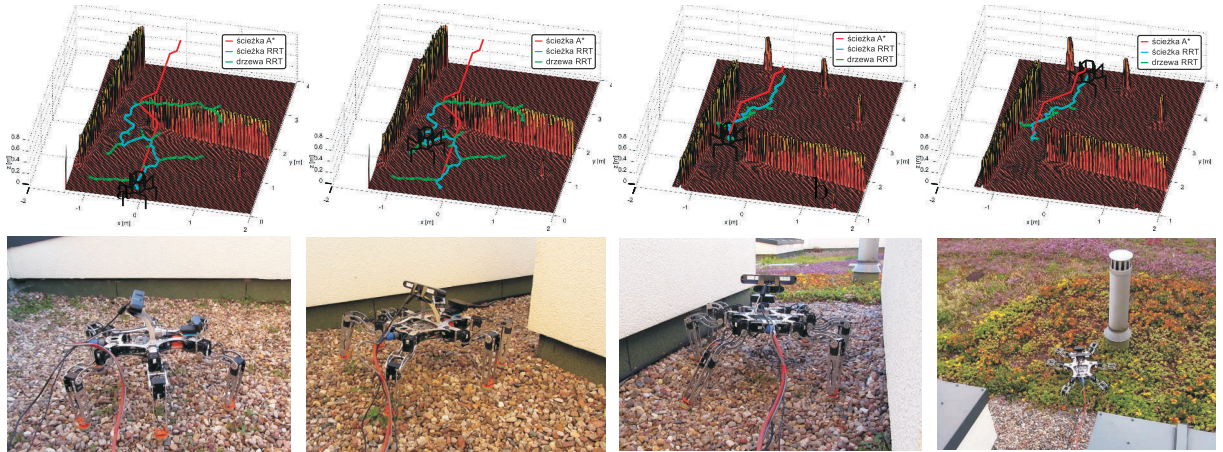
Rysunek 5: Drzewo RRT (a) użyte do eksploracji i gromadzenia danych uczących system oceny kosztu pokonywania przeszkód oraz wyniki działania modułu oceny kosztu pokonywania przeszkód za pomocą ręcznie strojonej heurystyki (b) i za pomocą proponowanego systemu (c). Kolory krawędzi w drzewie oznaczają wynik oceny kosztu pokonywania przeszkód

etapie użyty został zgrubny planer (A^*), który planuje ścieżkę ruchu do odległego celu. Następnie planowany jest precyzyjny ruch robota z użyciem zaproponowanego wcześniej planera RRT pomiędzy aktualną pozycją, a pozycją na ścieżce wyznaczonej przez zgrubny planer. Następnie robot wykonuje wyznaczoną ścieżkę ruchu i w trakcie ruchu gromadzi nowe informacje o otoczeniu. Procedura planowania i wykonywania ruchu jest powtarzana, aż robot osiągnie pozycję zadaną.

Również reprezentacja otoczenia jest dostosowana do właściwości systemu percepcji robota. Robot buduje dwie mapy. Pierwsza z nich obejmuje większy obszar, jednak jest mniej dokładna. Wielkość komórki została ustalona na 100 mm. Wartość ta wynika bezpośrednio z właściwości geometrycznych systemu percepcji robota. W przypadku mniejszych komórek wiele z nich jest pustych i nie jest aktualizowanych przez pomiary z czujnika głębi. Druga mapa obejmuje obszar znajdujący się blisko robota. Rozmiar rastra to 15 mm. Mapa ta służy do precyzyjnego planowania ruchu robota kroczącego. Środki obu map pokrywają się z układem robota i poruszają się wraz z nim.

W pracy [A4] zaproponowałem adaptacyjny system uczenia kosztu pokonania danego fragmentu terenu z użyciem mapy o zmniejszonej rozdzielczości. We wcześniejszych pracach koszt pokonania terenu na podstawie mapy rastrowej był określany za pomocą ręcznie dobranych parametrów. Koszt pokonania terenu jest następnie wykorzystywany w planerze A^* działającym na mapie o zmniejszonej rozdzielczości. Planer jednak często błędnie szacuje możliwość przejścia fragmentów terenu i unika fragmentów, które mogą zostać przez niego pokonane, lub eksploruje obszary, które zawierają przeszkody niemożliwe do przejścia przez robota.

Aby rozwiązać problem błędnego szacowania przez planer kosztu pokonania przeszkód z użyciem mniej precyzyjnej mapy, w pracy [A4] zaproponowałem system automatycznego uczenia kosztu pokonywania przeszkód na mapie o zmniejszonej rozdzielczości. Metoda opiera się na idei wykorzystania planera RRT planującego ruch robota z użyciem mapy precyzyjnej jako informacji referencyjnej do uczenia predyktora kosztu pokonania fragmen-



Rysunek 6: Kolejne sekwencje działania algorytmu planowania ruchu robota kroczącego z proponowanym systemem oceny kosztu pokonywania przeszkód oraz robot podczas wykonywania zaplanowanej ścieżki

tu terenu na mapie o mniejszej rozdzielczości. Uczenie dokonywane jest na mapie o dużej rozdzielczości. Na mapie wejściowej uruchamiany jest algorytm RRT określający precyzyjny ruch robota. Każde dodanie gałęzi do drzewa wiąże się z planowaniem pojedynczego kroku. W przypadku sukcesu lub porażki informacja ta jest zapisywana jako próbka ucząca. Wraz z informacją, czy możliwe jest pokonanie przez robota danego fragmentu terenu, zapisywane są trzy współczynniki określające właściwości terenu o sztucznie zmniejszonej rozdzielczości: odległość Euklidesowa pomiędzy węzłami (długość kroku c_1), wariancja sferyczna (c_2), oraz informacja, czy model kinematyczny pozwala na tranzycję pomiędzy dwoma węzłami drzewa (c_3). W trakcie budowania drzewa zbierane są kolejne próbki do uczenia systemu. Przykładowe drzewo eksplorujące mapę terenu pokazane zostało na rysunku 5a.

Do oceny kosztu przejścia wykorzystany został algorytm Kernel Density Estimation. Metoda ta została użyta do znalezienia rozkładu $p(p_t | c_1, c_2, c_3)$ określającego prawdopodobieństwo tego, że tranzycja pomiędzy dwoma węzłami jest możliwa przy danych współczynnikach określających właściwości geometryczne terenu $\mathbf{c} = [c_1, c_2, c_3]^T$. Do wyznaczenia rozkładu wspólnego $p(p_t, c_1, c_2, c_3)$ i rozkładu brzegowego $p(c_1, c_2, c_3)$ wykorzystane zostały następujące zależności:

$$p(p_t, c_1, c_2, c_3) = \frac{1}{n} \sum_{i=1}^n K_{h_2}(\mathbf{c} - \mathbf{c}_i) K_{h_1}(p_t - p_{ti}), \quad (4)$$

$$p(c_1, c_2, c_3) = \frac{1}{n} \sum_{i=1}^n K_{h_2}(\mathbf{c} - \mathbf{c}_i), \quad (5)$$

gdzie \mathbf{c}_i i p_{ti} są próbkami treningowymi parametrów wejściowych \mathbf{c} i wyjścia p_t . K_{h_1} i K_{h_2} są funkcjami Gaussa, oraz n oznacza liczbę próbek. Gęstość rozkładu warunkowego

obliczona jest za pomocą równania:

$$p(p_t|\mathbf{c}) = \frac{p(p_t, \mathbf{c})}{p(\mathbf{c})}. \quad (6)$$

Aby znaleźć maksymalne prawdopodobieństwo i wartość p_t , która odpowiada wektorowi danych wejściowych \mathbf{c} , rozwiązywany jest następujący problem optymalizacji

$$p_t = \arg \max_{p_t} p(p_t|\mathbf{c}), \quad (7)$$

z użyciem algorytmu Mean-shift. Wyjście p_t z KDE jest wartością ciągłą w przedziale od 0 do 1, które jest następnie dyskretyzowane. Wartości z przedziału od 0 do 0.5 określają teren niemożliwy do pokonania przez robota, pozostałe wartości związane są z terenem możliwym do przejścia.

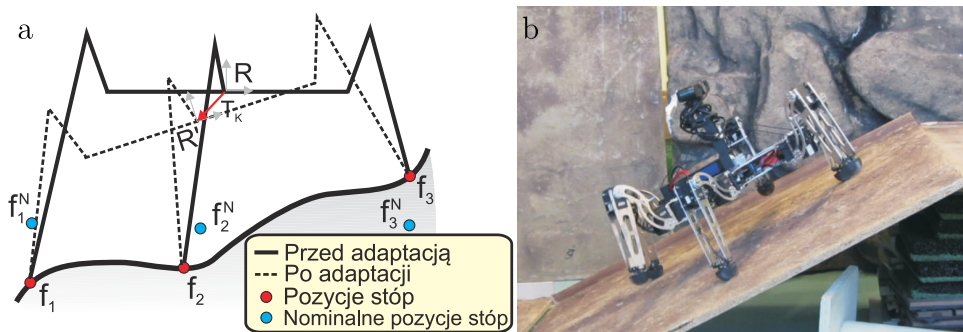
Wyniki na danych testowych pokazują, że dzięki zastosowanej metodzie liczba poprawnych decyzji wzrosła z 51.8% do 70.1%. Przykładowe wyniki oceny kosztu pokonywania terenu przed i po uczeniu pokazane zostały odpowiednio na rysunku 5b i rysunku 5c.

Opracowany system został wykorzystany do planowania ruchu szescionożnego robota kroczącego podczas eksperymentów w laboratorium i w naturalnym środowisku zewnętrznym. Przykładowe wyniki zostały pokazane na rysunku 6. Na kolejnych rysunkach widać ścieżki zaplanowane przez zgrubny planer A^* oraz precyzyjną ścieżkę ruchu zaplanowaną przy użyciu algorytmu RRT. Dzięki zaproponowanej metodzie robot jest w stanie autonomicznie znaleźć i wykonać zaplanowaną ścieżkę z punktu startowego do docelowego.

4.6 Sterowanie reakcyjne na podstawie danych z sensorów wewnętrznych

Mimo wykorzystania wizyjnych systemów budowy mapy i lokalizacji robota może dochodzić do sytuacji, w których robot nie ma informacji o modelu otoczenia. Takie sytuacje są powszechne, gdy robot porusza się w wysokiej trawie i obraz z kamery nie dostarcza informacji o kształcie terenu lub interesujący fragment otoczenia jest przysłonięty przez wysokie przeszkody. Dlatego w pracy [A5] opracowany został system adaptacyjnego poruszania się robota po nierównym terenie z wykorzystaniem wyłącznie informacji enteroceptywnej.

Robot Messor II nie ma możliwości sterowania siłowego. Sterowanie ruchem robota odbywa się wyłącznie poprzez zadawanie pozycji stóp w przestrzeni 3D. W wyniku sterowania pozycyjnego na nieznanym terenie dochodzi do sytuacji, które mogą prowadzić do utraty równowagi. Pozycja zadana stopy podczas fazy podparcia powinna znajdować się dokładnie na podłożu. W przypadku niedokładności w wykonaniu ruchu robot może odpychać się od podłoża. Natomiast jeżeli ruch zostanie zatrzymany przed osiągnięciem powierzchni terenu, robot nie znajdzie stabilnego podparcia. Dlatego do robota zostały dodane specjalnie zaprojektowane stopy, które w sposób pasywny dopasowują się do kształtu terenu, i dzięki wykorzystaniu sprężyn są w stanie skompensować niedokładności systemu sterowania i percepcji robota. W stopie zamontowano również przełączniki, które niezawodnie wykrywają kontakt stopy z podłożem i pozwalają na zatrzymanie ruchu nogi w odpowiednim momencie.



Rysunek 7: Strategia adaptacji postury robota na podstawie pozycji stóp na nierównym podłożu (a) oraz robot podczas wspinaczki po rampie o nachyleniu 25°

Proponowana w pracy [A5] metoda wykorzystuje podatne stopy do wykrycia kontaktu z podłożem. Robot poruszając się w zadanym kierunku (ruch zadany jest przez operatora), przenosi naprzemiennie trzy nogi. Opuszczając nogi, poszukuje stabilnego podparcia. Następnie modyfikowana jest postura robota, tak żeby efektywnie dopasowywać się do lokalnego nachylenia terenu. W [A5] zaproponowałem metodę, która nie wymaga bezpośredniej estymacji średniego nachylenia terenu, jak to jest najczęściej robione na robotach kroczących [16]. Zaproponowałem natomiast wykorzystanie metody Umeyamy [18] do określenia postury robota.

Wejściem do algorytmu korygującego posturę robota na nierównym terenie są dwa zbiory punktów $P = \{f_1, \dots, f_6\}$ i $Q = \{f_1^N, \dots, f_6^N\}$. Punkty f_1, \dots, f_6 są pozycjami stóp na nierównym terenie. Punkty f_1^N, \dots, f_6^N są pozycjami nominalnymi stóp, w których robot ma największy zakres ruchu. Oznaczenia nóg zostały pokazane na rysunku 7a. Stosując metodę Umeyamy, znajduję przekształcenie \mathbf{T}_k , które koryguje aktualną pozycję i orientację robota. Obliczona postura jest kompromisem pomiędzy aktualnym ustawieniem nóg, a pozycją posiadającą maksymalny margines kinematyczny. Dzięki temu robot autonomicznie dostosowuje orientację korpusu do nachylenia terenu oraz zachowuje duże możliwości ruchowe.

Dodatkowo robot stosuje metodę korygującą posturę robota w celu zwiększenia stabilności. Metoda ta wykorzystuje informację o położeniu środka ciężkości robota. Znalezione przekształcenie stabilizujące \mathbf{T}_s jest dodawane do wcześniej wyznaczonego przekształcenia korygującego posturę robota. Ponieważ obie metody posiadają rozwiązanie analityczne, obliczenie korekty postury robota odbywa się bardzo szybko. Obliczenia dokonywane są on-line w trakcie ruchu, co jest bardzo ważne dla zachowania równowagi podczas pokonywania przeszkód.

Zaproponowana metoda została przetestowana na makiecie nierównego terenu oraz na pochylni o znanym nachyleniu. Dzięki temu możliwe jest porównanie zaproponowanej strategii do innych metod z literatury. Dzięki zaproponowanej metodzie robot jest w stanie wejść na pochylnię o nachyleniu 25° . Dla porównania robot Lauron V może wejść na pochylnię o maksymalnym nachyleniu 21° . Robot Messor II podczas wykonywania ruchu na pochylni pokazany został na rysunku 7b.

4.7 Dedykowany system lokalizacji robota kroczącego

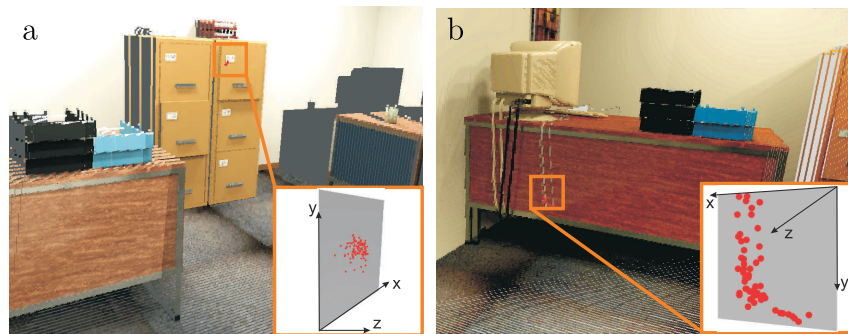
Podczas wybranych eksperymentów na zewnątrz budynków przedstawionych w pracy [A4] użyty został algorytm PTAM. Jego zastosowanie jest ograniczone ze względu na możliwość działania tylko w ograniczonym obszarze roboczym. Podczas dłuższych eksperymentów system ten przestaje śledzić cechy i nie jest w stanie dodawać nowych ramek kluczowych. Dlatego konieczne stało się opracowanie systemu lokalizacji, który umożliwi śledzenie trajektorii robota podczas dłuższej pracy autonomicznej. W wyniku tej motywacji powstał system lokalizacji wizyjnej PUT SLAM, który opracowany został w Instytucie Automatyki, Robotyki i Inżynierii Informatycznej Politechniki Poznańskiej. W tym projekcie byłem odpowiedzialny za optymalizację grafową, będącą elementem kluczowym podsystemu estymującego trajektorię ruchu kamery. W odróżnieniu od filtra Kalmana, który estymuje tylko aktualną wartość zmiennej na podstawie aktualnych pomiarów, optymalizacja grafowa pozwala na optymalizację całej trajektorii na podstawie historii wszystkich pomiarów. Przy zachowaniu odpowiednich ograniczeń podczas tworzenia grafu optymalizacja odbywa się w czasie rzeczywistym.

Opracowany system SLAM wykorzystuje cechy punktowe na obrazie RGB, które rozszerzone o pomiar głębi tworzą cechy punktowe w przestrzeni 3D. „Front-end” systemu SLAM zajmuje się przetwarzaniem obrazów kolorowych i głębi w celu detekcji cech najlepszych do określenia ruchu kamery. Następnie śledzone cechy są przekazywane do grafu w celu wyznaczenia trajektorii kamery. Cechy punktowe 3D \mathbf{f}_j i pozycje kamery \mathbf{c}_i są reprezentowane przez wierzchołki grafu. Krawędzie łączące wierzchołki w grafie reprezentują pomiary (ograniczenia). Pomiary \mathbf{m}_{ij} pomiędzy i -tą pozycją kamery i j -tą cechą są zdefiniowane przez krawędź 3-D $\mathbf{t}_{ij} \in \mathbb{R}^3$. Przekształcenie pomiędzy dwiema pozycjami kamery i oraz k zmierzonymi za pomocą odometrii wizyjnej są reprezentowane przez krawędzie $\mathbf{o}_{ik} \in \mathbf{SE}(3)$, gdzie $\mathbf{SE}(3)$ reprezentuje przekształcenie bryły sztywnej (translację i rotację) dla sześciu stopni swobody. W celu znalezienia optymalnej sekwencji pozycji kamery $\mathbf{c}_1, \dots, \mathbf{c}_n$ i pozycji wszystkich cech $\mathbf{f}_1, \dots, \mathbf{f}_m$ minimalizowana jest następująca funkcja kosztu:

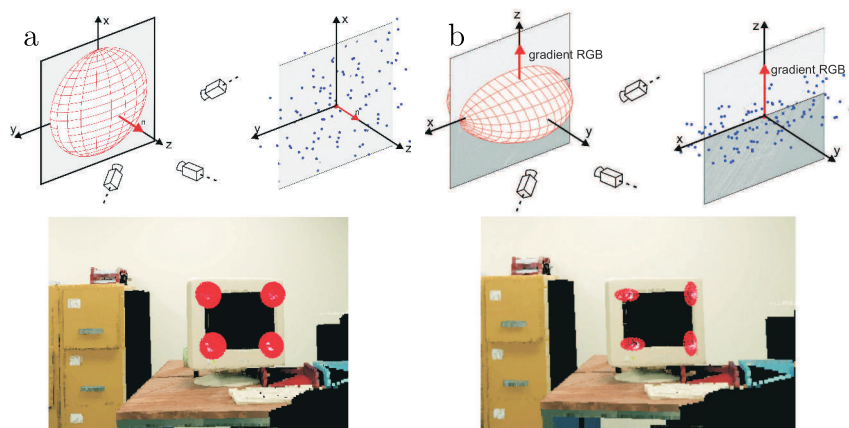
$$\operatorname{argmin}_{\mathbf{f}, \mathbf{c}} F = \sum_{i=1}^n \sum_{j=1}^m \mathbf{e}(\mathbf{c}_i, \mathbf{f}_j, \mathbf{m}_{ij})^T \boldsymbol{\Omega}_{ij}^t \mathbf{e}(\mathbf{c}_i, \mathbf{f}_j, \mathbf{m}_{ij}), \quad (8)$$

gdzie $\mathbf{e}(\mathbf{c}_i, \mathbf{f}_j, \mathbf{m}_{ij})$ jest funkcją błędu obliczoną dla estymowanej i zmierzonej pozycji wierzchołka. Pomiary \mathbf{m}_{ij} są przekształceniem 3-D \mathbf{t}_{ij} (translacja) dla krawędzi pomiędzy pozycjami kamery a cechami lub przekształceniem $\mathbf{SE}(3)$ \mathbf{o}_{ij} dla krawędzi pomiędzy pozycjami kamery. Do optymalizacji grafu (minimalizacji kosztu zdefiniowanego w równaniu (8)) wykorzystana została biblioteka g²o [13].

Dokładność pomiarów pomiędzy pozycją kamery a cechami 3D w grafie jest reprezentowana przez macierz informacyjną $\boldsymbol{\Omega}_{i,j}^t$. Macierz informacyjna może zostać uzyskana poprzez odwrócenie macierzy kowariancji pomiaru. Głównym celem pracy [A6] było odpowiednie modelowanie niepewności pomiaru położenia cech punktowych w przestrzeni 3D. Niepewność pomiaru pomiędzy pozycjami kamery $\boldsymbol{\Omega}_{i,j}^o$ nie jest uwzględniana w tej metodzie (podczas optymalizacji przyjmuje wartości równe macierzy jednostkowej). Przeprowadzona w pracy



Rysunek 8: Rozkład cech 3D uzyskany dzięki analizie pomiarów z systemu SLAM przy założeniu znajomości rzeczywistej trajektorii kamery: cechy wykryte na narożniku (a) i krawędzi (b) na obrazie RGB



Rysunek 9: Zaproponowane modele niepewności pomiarów: model oparty o wektor normalny do lokalnej powierzchni (a) oraz model oparty o gradient RGB (b). Na rysunku pokazano elipsoidę reprezentującą macierz kowariancji, przykładowe rozmieszczenie pomiarów cechy w 3D oraz wizualizację elipsoidy dla rzeczywistych cech 3D

[A6] analiza symulacyjna pokazała, że modelując niepewność pomiaru i uwzględniając ją podczas obliczania odległości Mahalanobisa w równaniu (8), można znacząco poprawić dokładność estymowanej trajektorii kamery. W pracy [A6] zaproponowałem również system nazwany „reverse SLAM tool”. System ten umożliwia analizę rozkładu cech na podstawie znanej trajektorii kamery i obrazów wejściowych. Przykładowy rozkład cech 3D przedstawiony został na rysunku 8. Na rysunku 8a cechy rozmieszczone są na powierzchni obiektu (szafki). W sytuacji przedstawionej na rysunku 8b można zauważyć, że cechy przesuwają się wzdłuż krawędzi przewodu.

Analiza rozkładu cech w przestrzeni 3D na podstawie wyników z narzędzia „reverse SLAM tool” umożliwiła zaproponowanie nowych modeli niepewności pomiaru cech z czujnika RGB-D. W pracy [A6] zaproponowałem dwa modele. Pierwszy z nich wykorzystuje informację o tym, że cechy rozmieszczone są na powierzchniach obiektów. Model ten oparty

jest o kierunek określony przez wektor normalny \mathbf{n} do powierzchni i został pokazany na rysunku 9a. Żeby określić macierz kowariancji \mathbf{C}_n , obliczana jest macierz rotacji \mathbf{R} , tak aby oś z układu R pokrywała się z wektorem normalnym n . Oś x i y wybierane są dowolnie, tak aby razem z osią z tworzyły prawoskrętny układ współrzędnych. Macierz kowariancji obliczana jest następująco:

$$\mathbf{C}_n = \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{R}^{-1}, \mathbf{S} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix}, \quad (9)$$

gdzie \mathbf{S} jest macierzą parametrów skalujących elipsoidę niepewności. Zmieniający jest tylko współczynnik S_z , który przyjmuje wartości z przedziału od 0 do 1. Pozostałe wartości na diagonalu (S_x, S_y) mają wartość 1. Dzięki temu elipsoida związana z macierzą kowariancji przyjmuje kształt spłaszczonych dysków (rys. 9a).

Drugi zaproponowany model wykorzystuje zjawisko rozproszenia pomiarów wzdłuż krawędzi RGB. Procedura wyznaczenia macierzy kowariancji jest analogiczna do procedury związanej z wyznaczeniem modelu opartego o wektor normalny (9). Tym razem oś z pokrywa się z gradientem RGB wyznaczonym na obrazie kolorowym, natomiast oś x wzdłuż krawędzi RGB. Oś y jest skierowana tak, aby tworzony był prawoskrętny układ współrzędnych. Wizualizacja elipsoidy związanej z macierzą kowariancji, uzyskaną na podstawie gradientu RGB, przedstawiona została na rysunku 9b.

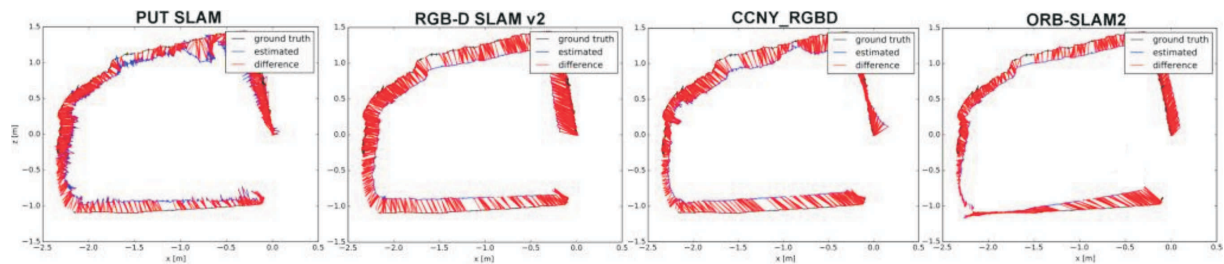
Zaproponowane modele zostały zaimplementowane w systemie PUT SLAM i przetestowane na zbiorach danych ICL-NUIM i TUM RGB-D. Wyniki pokazują, że dzięki zastosowaniu modeli niepewności można zwiększyć dokładność systemu SLAM nawet o 70% [A6].

4.8 Weryfikacja eksperymentalna na sześcionożnym robocie krocącym

System lokalizacji PUT SLAM posiada cechy, które powodują, że może on zostać wykorzystany na robotach krocących. Ponieważ wykorzystywana jest zarówno informacja z obrazu kolorowego, jak i z obrazów głębi, wszystkie cechy w grafie posiadają zainicjalizowaną pozycję 3D. Dzięki temu optymalizacja grafu jest bardziej stabilna. Również wymagana jest mniejsza liczba cech do śledzenia ruchu kamery niż w przypadku wykorzystania np. systemu ORB-SLAM2 [14], który minimalizuje błąd reprojektacji. Bezpośrednie wykorzystanie danych o głębi powoduje jednak, że błąd pomiaru głębi przenosi się na estymowany ruch kamery. W efekcie w sytuacji, w której obserwowanych jest wiele cech i ruch kamery jest płynny, lepszą jakość śledzenia uzyskuje się stosując np. ORB-SLAM2, który nie wykorzystuje bezpośrednio do lokalizacji danych o głębi.

W pracy [A7] przeprowadzono eksperymentalną weryfikację systemu PUT SLAM oraz porównano jego działanie z innymi systemami wizyjnej lokalizacji (RGB-D SLAM v2 [4], CCNY_RGBD [3] i ORB-SLAM2 [14]). Eksperymenty zostały przeprowadzone na robotach Messor II oraz PhantomX przez zespół z CTU w Pradze. Podczas eksperymentów rzeczywista trajektoria robota została zarejestrowana za pomocą referencyjnego systemu wizyjnego (rys. 10).

Dodatkowo w pracy [A7] zaproponowałem nową metodę zarządzania cechami w grafie.



Rysunek 10: Wyniki porównania systemu jednoczesnej budowy mapy i lokalizacji PUT SLAM z algorytmami RGB-D SLAM v2 [4], CCNY_RGBD [3] i ORB-SLAM2 [14] na sześcionożnym robocie kroczącym

Metoda polega na cyklicznym przeglądaniu grafu w poszukiwaniu cech, które są słabo powiązane z resztą grafu za pomocą pomiarów. Takie cechy wraz z połączeniami są usuwane z grafu, żeby zapobiec niekontrolowanemu rozrostowi grafu i znacznemu spowolnieniu działania systemu. Z grafu nie są usuwane cechy, które są powiązane z pozycjami kamery, z których obserwowana była bardzo mała liczba cech. Pozwala to zachować ciągłość śledzenia ruchu robota w sytuacjach, w których widoczna jest niewielka liczba cech.

4.9 Odometria nożna i kompletny system lokalizacji robota kroczącego

Podczas pracy rzeczywistego robota kroczącego występują sytuacje, w których niemożliwe jest estymowanie ruchu kamery. Bardzo często rozmycie obrazu z kamery, które towarzyszy gwałtownym ruchom robota podczas kroczenia, utrudnia śledzenie cech wizyjnych. Problem ten może zostać rozwiązany przez zastosowanie szybszej kamery z mechanizmem „global shutter”. Jednak w sytuacji niewystarczającego oświetlenia, zbyt bliskiego podejścia do przeszkody i przysłaniania kamery przez obiekty (wysoka trawa, gałęzie), poruszania się w jednolitym korytarzu, lokalizacja wizyjna staje się niemożliwa. Dlatego w pracy [A8] zaproponowałem wykorzystanie wszystkich dostępnych na pokładzie robota źródeł informacji do lokalizacji robota kroczącego.

W pierwszej kolejności przeprowadzone zostały eksperymenty pokazujące wpływ ciągłej lokalizacji na otrzymywaną mapę rastrową otaczającego terenu. W eksperymentach na robocie porównano różne systemy lokalizacji wizyjnej. Uzyskane wyniki potwierdzają wnioski z pracy [A8]. Pokazano dodatkowo wpływ braku informacji o pozycji robota na mapę rastrową otoczenia robota. Ze względu na brak tej informacji mapa rastrowa nie może być aktualizowana i w efekcie robot traci informację o wielu przeszkodach. Również mniejsza liczba pomiarów, która jest zintegrowana w mapie, powoduje, że jest ona mniej dokładna od mapy uzyskanej z systemu, który przez cały eksperyment dostarcza informację o pozycji robota i kamery.

W [A8] zaproponowałem system lokalizacji robota kroczącego, który wykorzystuje wszystkie sensory dostępne na pokładzie robota, tj. enkodery w serwomechanizmach, czujniki kontaktu umieszczone w stopach, jednostkę AHRS oraz kamerę RGB-D. W pierwszej kolejności zaproponowałem algorytm obliczający ruch robota na podstawie pomiarów z enkoderów i informacji z czujników kontaktu. Następnie estymowany ruch robota jest korygowany na

podstawie danych z jednostki AHRS. Uzyskane przemieszczenie jest wykorzystane w fazie predykcji filtru Kalmana i integrowane z danymi z sensorów.

Opracowany system umożliwia śledzenie ruchu robota niezależnie od warunków oświetleniowych i systemu wizyjnego. Jest jednak wrażliwy na kumulujący się błąd pomiaru (dryft). Dlatego zaproponowałem zintegrowanie danych z odometrii nożnej z danymi z systemu wizyjnego. Ponieważ główną rolą systemu wizyjnego ma być kompensacja dryftu, wybrano ORB-SLAM2, który w przeciwieństwie do systemu PUT SLAM posiada wydajne narzędzia do relokalizacji kamery.

Do estymowania ruchu robota na podstawie danych z enkoderów i czujników kontaktu wykorzystany został model kinematyczny robota. Niemożliwe jest jednak bezpośrednie wykorzystanie tego modelu do obliczenia ruchu. Podczas kroczenia statycznie stabilnego robota sześcionożnego przynajmniej trzy nogi mają kontakt z podłożem. Ogólnie ruch robota \mathbf{M} może zostać znaleziony poprzez rozwiązanie układu równań dla nóg mających kontakt z podłożem na początku i na końcu ruchu (pozostałe nogi nie są brane pod uwagę, ponieważ nie powodują ruchu postępowego robota):

$$\begin{cases} \mathbf{M} \cdot {}^R\mathbf{L}_1 \cdot fk(\mathbf{c}_1^{t+1}) = {}^R\mathbf{L}_1 \cdot fk(\mathbf{c}_1^t) \\ \dots \\ \mathbf{M} \cdot {}^R\mathbf{L}_6 \cdot fk(\mathbf{c}_6^{t+1}) = {}^R\mathbf{L}_6 \cdot fk(\mathbf{c}_6^t) \end{cases} \quad (10)$$

gdzie \mathbf{c}_l^t jest aktualną konfiguracją nogi o numerze l , \mathbf{c}_l^{t+1} jest konfiguracją nogi l w chwili $t+1$, $fk()$ jest rozwiązaniem zadania kinematyki prostej, a ${}^R\mathbf{L}_l$ jest transformacją z układu robota do układu nogi l .

W praktyce niemożliwe jest analityczne rozwiązanie układu równań (10), ponieważ pomiary kątów w węzłach \mathbf{c}_l są zaszumione i każde równanie zwraca inny ruch korpusu \mathbf{M} . Dlatego zamiast rozwiązywania układu równań (10) zaproponowany został system optymalizacji, który znajduje ruch robota \mathbf{M} minimalizujący błąd pomiędzy zmierzoną konfiguracją robota i konfiguracją wynikającą z estymowanego ruchu. Problem optymalizacji zdefiniowany został następująco:

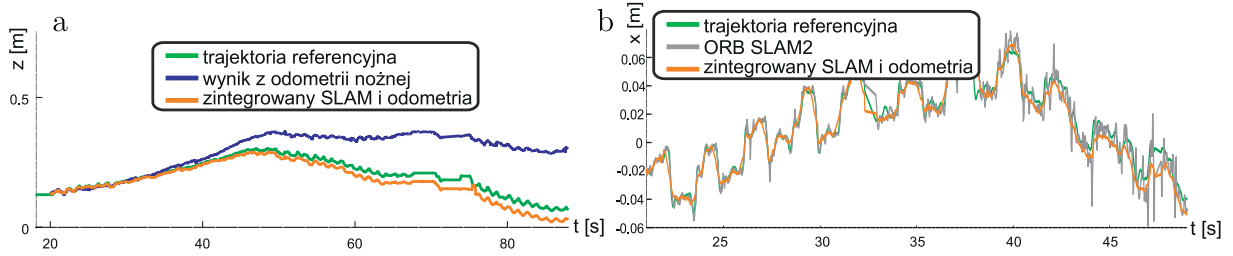
$$\arg \min_{\mathbf{M}} \sum_{l=1}^6 e_l^2, \quad (11)$$

gdzie l jest numerem nogi, e_l różnicą pomiędzy zmierzoną konfiguracją nogi \mathbf{c}_1^{t+1} a konfiguracją, która generuje ruch robota \mathbf{M} :

$$e_l = \|\mathbf{c}_l^{t+1} - ik((\mathbf{M} \cdot {}^R\mathbf{L}_l)^{-1} \cdot {}^R\mathbf{L}_l \cdot fk(\mathbf{c}_l^t))\|_1. \quad (12)$$

gdzie $ik()$ jest rozwiązaniem zadania odwrotnego dla kinematyki nogi.

Do znalezienia minimum funkcji (12) wykorzystane zostały dwie metody CMA-ES i PSO. Metoda PSO potrafi szybciej znaleźć rozwiązanie pod warunkiem, że rozwiązanie początkowe znajduje się blisko rozwiązania optymalnego. Dlatego w celu znalezienia rozwiązania początkowego zaproponowałem algorytm Umeyamy [18]. Wejściem do algorytmu Umeyamy są dwa zbiory punktów $L^t = \{\mathbf{L}_1^t, \dots, \mathbf{L}_N^t\}$ i $L^{t+1} = \{\mathbf{L}_1^{t+1}, \dots, \mathbf{L}_N^{t+1}\}$, które są związane



Rysunek 11: Trajektorie uzyskane podczas wchodzenia i schodzenia robota z rampy: wysokość korpusu robota (a) oraz ruchy stabilizujące robota w płaszczyźnie prostopadłej do kierunku ruchu (b)

z pozycją N nóg w fazie podparcia przed i po zakończeniu ruchu. Metoda oblicza optymalną macierz rotacji \mathbf{R} i wektor translacji \mathbf{t} , która nakłada na siebie dwa zbiory punktów L^t i L^{t+1} :

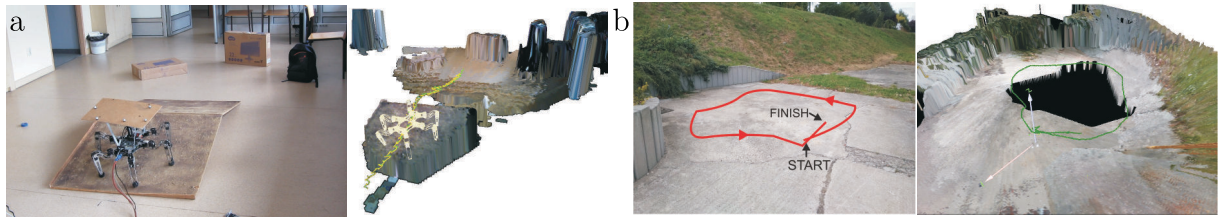
$$e_L^2 = \frac{1}{N} \sum_{l=1}^N \|\mathbf{L}_l^t - (\mathbf{R} \cdot \mathbf{L}_l^{t+1} + \mathbf{t})\|^2, \quad (13)$$

gdzie \mathbf{L}_l^t i \mathbf{L}_l^{t+1} są odpowiadającymi sobie parami punktów. Metoda ta jest metodą przybliżoną znalezienia przekształcenia \mathbf{M} , pozwala jednak znacząco skrócić czas optymalizacji. Dzięki temu algorytm pozwala w czasie rzeczywistym estymować stan robota.

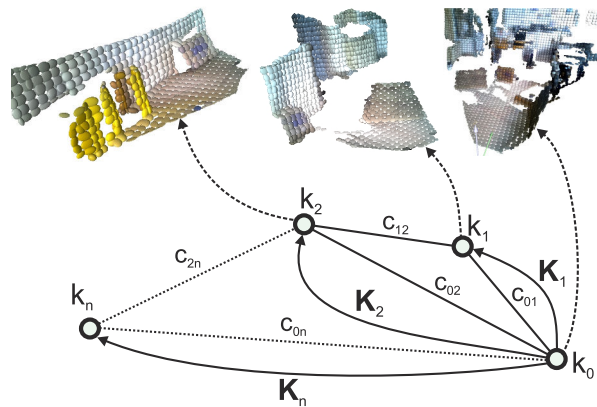
W pracy [A8] zaproponowałem również metodę integracji pomiarów z systemu odometrii nożnej i wizyjnego systemu SLAM. W tym celu użyty został graf pozycji, który jest optymalizowany w celu wyznaczenia optymalnej trajektorii ruchu robota. Pomiarzy z odometrii dodawane są do grafu, tak aby tworzyły łańcuch przekształceń. Ponieważ na robocie używany jest globalny system lokalizacji, pomiary z systemu SLAM tworzą połączenia pomiędzy układem globalnym a układami, w których dokonywany jest pomiar. Dzięki temu pomiary z odometrii wpływają na lokalny kształt estymowanej trajektorii, a pomiary z systemu SLAM są w stanie korygować dryft tej trajektorii.

Proponowany system lokalizacji został przetestowany na robocie krocącym Messor II w eksperymentach w pomieszczeniu i z referencyjnym systemem lokalizacji oraz w eksperymentach na zewnątrz budynku. Pokazano, że cały system jest w stanie estymować ruch robota mimo braku śledzenia z systemu wizyjnego spowodowanego szybkimi ruchami robota. Na rysunku 11a przedstawione zostały trajektorie z odometrii i trajektoria uzyskana z proponowanego systemu lokalizacji podczas wchodzenia i schodzenia robota z rampy. Można zauważyć, jak trajektoria z odometrii jest korygowana przez system wizyjny, zachowując przy tym swój lokalny kształt. Na rysunku 11b przedstawione zostały ruchy poziome stabilizujące robota. Można zauważyć, że ruchy robota są lepiej rejestrowane przez proponowany system lokalizacji niż przez system ORB-SLAM2.

Na rysunku 12 przedstawiono uzyskane trajektorie podczas eksperymentu na rampie i na zewnątrz budynku. Na rysunku 12 pokazano również uzyskane mapy wysokościowe. Komórki mapy rastrowej zawierają dodatkowo informację o kolorze przeszkód, co ułatwia



Rysunek 12: Trajektorie ruchu robota oraz model otoczenia uzyskane podczas eksperymentu na rampie (a) oraz na zewnątrz budynku (b)



Rysunek 13: Struktura grafu reprezentującego model otoczenia zbudowanego z map lokalnych NDT-OM. Rysunek zaczerpnięty z [A9]

porównanie rzeczywistego otoczenia robota i uzyskanego modelu terenu. Dzięki systemowi lokalizacji możliwe jest wierne odwzorowanie rzeczywistego otoczenia robota.

4.10 Modelowanie otoczenia z użyciem grafu map lokalnych

System lokalizacji robota oparty jest o graf pozycji. W sytuacji, w której robot wraca do wcześniej obserwowanego miejsca, możliwe jest zamknięcie pętli (ang. *Loop Closure*) i korekcja całej trajektorii robota. Takich właściwości nie ma globalna mapa otoczenia. Pomiary są dodawane do mapy w trakcie poruszania się robota. W przypadku korekty pozycji, z których dokonywane były pomiary, nie ma możliwości korekcji mapy globalnej. Doświadczenia z grafowymi systemami lokalizacji pozwoliły mi zaproponować nowy system budowy mapy, który pozbawiony jest wspomnianej wady i umożliwia korekcję mapy globalnej po zamknięciu pętli.

Struktura proponowanego modelu otoczenia przedstawiona została na rysunku 13. Podczas poruszania się robota tworzony jest graf pozycji. Każdy węzeł grafu związany jest z ramką kluczową (ang. *keyframe*). Na każdym obrazie RGB znajdowane są cechy punktowe, które są następnie porównywane do ostatniej ramki kluczowej. Jeżeli stopień podobieństwa (liczba dopasowanych cech punktowych z aktualnej ramki i ostatniej ramki kluczowej) spadnie poniżej ustalonego progu, tworzona jest nowa ramka kluczowa.

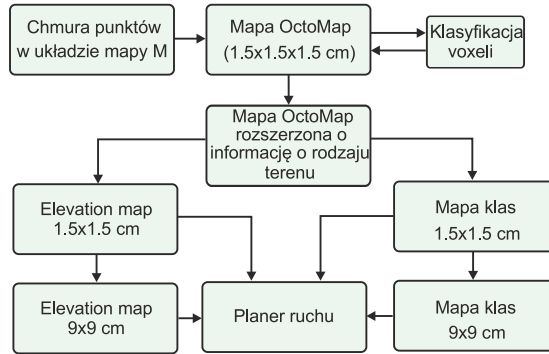
Do każdej ramki kluczowej w grafie przypisana jest lokalna mapa NDT-OM [17]. Wybrany został ten model otoczenia, ponieważ pozwala na dokładniejsze reprezentowanie kształtu przeszkód niż OctoMap [8] i konwersję do innych reprezentacji otoczenia (OctoMap, mapa wysokościowa). Proponowana architektura jest jednak ogólna i można wybrać dowolną metodę modelowania otoczenia. Krawędzie w grafie reprezentują przekształcenia SE3 pomiędzy ramkami kluczowymi. Dzięki temu możliwe jest sprowadzenie map lokalnych do wspólnej mapy globalnej. Możliwa jest również zmiana wzajemnego położenia map lokalnych po zamknięciu pętli i korekcja dryftu nie tylko dla pozycji kamery, ale również dla pomiarów zintegrowanych w mapie globalnej. Wyniki szczegółowe zostały przedstawione w [A9].

Dodatkowo w pracy [A9] zaproponowałem nową metodę filtrowania chmury punktów z kamery RGB-D. Metoda filtrowania wykorzystuje analizę PCA do znalezienia płaskich struktur na obrazie głębi. Dla wszystkich elipsoid związanych z płaskimi fragmentami otoczenia generowane są dodatkowe punkty 3D w chmurze punktów zgodnie z rozkładem danym przez macierz kowariancji wyznaczoną za pomocą metody PCA. Metoda ta pozwala na znacznie lepsze reprezentowanie płaszczyzn (ścian), których wektor normalny jest prostopadły do osi optycznej kamery i ze względu na usytuowanie względem kamery nie posiadają wystarczającej liczby punktów pomiarowych. W pracy [A9] pokazano również sposób konwersji uzyskanej reprezentacji na globalną mapę OctoMap.

4.11 Planowanie ruchu z wykorzystaniem informacji semantycznej

Metoda planowania ruchu przedstawiona w [A4] wykorzystuje wyłącznie informację o kształcie geometrycznym przeszkód. Takie podejście sprawdza się na sztywnym podłożu. W praktyce robot porusza się również w środowisku, w którym występują rośliny, np. trawa, krzaki, oraz podłoże o różnych właściwościach. Informacja geometryczna otrzymywana z sensorów RGB-D odzwierciedla kształt geometryczny roślin, jednak kształt ten nie zapewnia stabilnego podparcia dla robota. Również poruszanie się po trawie jest bardziej ryzykowne niż poruszanie się po chodniku lub asfalcie. Z tego powodu robot poza informacją geometryczną powinien również wykorzystywać informację o rodzaju podłoża podczas planowania swojej ścieżki. Dlatego w pracy [A10] zaproponowałem metodę planowania ruchu, która uwzględnia również informację o rodzaju terenu podczas planowania ruchu robota kroczącego.

Aby możliwe było planowanie ruchu z informacją o rodzaju terenu, konieczne jest zawarcie tej informacji w modelu otoczenia. W [A10] zaproponowałem model otoczenia, który jednocześnie przechowuje informację o rodzaju terenu, ale również ułatwia jego klasyfikację. Reprezentacja terenu wraz z przepływem danych od sensora RGB-D (chmura punktów) została przedstawiona na rysunku 14. Podstawowy model otoczenia przechowywany jest w mapie 3D opartej o strukturę drzewa ósemkowego (octree). Podczas aktualizacji voxeli, aktualizowany jest również ich średni kolor. W odróżnieniu od standardowych podejść w robotyce, które wykorzystują obrazy do klasyfikacji, zaproponowałem klasyfikację voxeli. Uzyskana mapa Octree jest następnie rzutowana na płaszczyznę 2D w celu uzyskania map wysokościowych i odpowiadających im map zawierających wynik klasyfikacji. Mapy ra-



Rysunek 14: Model otoczenia wykorzystany podczas planowania ruchu z użyciem informacji o rodzaju terenu

strowe są tworzone w dwóch rozdzielczościach, tak żeby mogły zostać użyte przez planer A^* i planner oparty o metodę RRT.

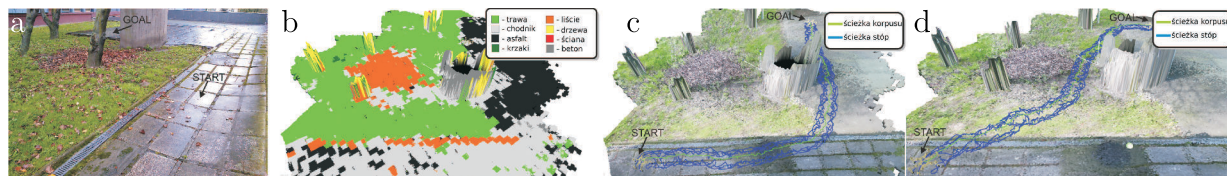
W podejściu do planowania ruchu zaproponowanym w [A10] wykorzystano założenie, że w jednej komórce może znajdować się kilka obiektów różnych klas. Bardzo rzadko klasyfikator określa prawdopodobieństwo wystąpienia danej klasy w komórce na 1.0. Powszechne są również sytuacje, w których w jednym obszarze i odpowiadającej mu komórce mapy znajduje się kilka obiektów różnych klas. Przykładem może być rzadka trawa lub trawa wyrastająca z popękanego chodnika. Dlatego w komórce mapy poza identyfikatorem klasy, której obecność jest najbardziej prawdopodobna, przechowywane są również prawdopodobieństwa wystąpienia instancji pozostałych klas.

Podczas planowania ruchu bezpośrednio wykorzystana jest informacja o rodzaju terenu zapisana w komórce mapy rastrowej. Planowanie ruchu z uwzględnieniem rodzaju terenu realizowane jest w planerze A^* algorytmu guidedRRT. Do określenia kosztu przejścia pomiędzy sąsiednimi komórkami mapy zaproponowane zostało równanie:

$$c_{\text{final}} = k_1 \cdot (d - \omega) + k_2 \cdot \sum_{i=0}^5 (w_i \cdot P(c_i)), \quad (14)$$

gdzie d jest odległością pomiędzy sąsiednimi komórkami, ω jest wariancją sferyczną określającą stopień nierówności terenu, $P(c_i)$ jest prawdopodobieństwem wystąpienia i -tej klasy w komórce docelowej, w_i jest kosztem przypisaniem do danej klasy, a k_1 i k_2 są stałymi współczynnikami określającymi wpływ parametrów geometrycznych i semantycznych na uzyskaną ścieżkę ruchu. Wagi w_i zostały ręcznie przypisane do klas rozpoznawanych przez robota. Jeżeli szacujemy, że trawa jest bardziej ryzykownym terenem dla robota niż chodnik, powinna ona mieć wyższą wagę. W efekcie robot będzie unikał trawy, a preferował chodnik.

Przykładowe wyniki działania planera pokazane zostały na rysunku 15. Środowisko zostało dobrane tak, aby zawierało różne rodzaje terenu i obiektów. Na rysunku 15b przedstawiony został wynik klasyfikacji, natomiast na rysunku 15 – wynik planowania ruchu z uwzględnieniem tej informacji. Ponieważ koszt związany z poruszaniem się po chodniku i asfalcie



Rysunek 15: Eksperyment dotyczący planowania ruchu z wykorzystaniem informacji semantycznej: miejsce eksperymentu (a), wynik klasyfikacji terenu (b), planowanie z uwzględnieniem rodzaju terenu (c), planowanie najkrótszej ścieżki (d)

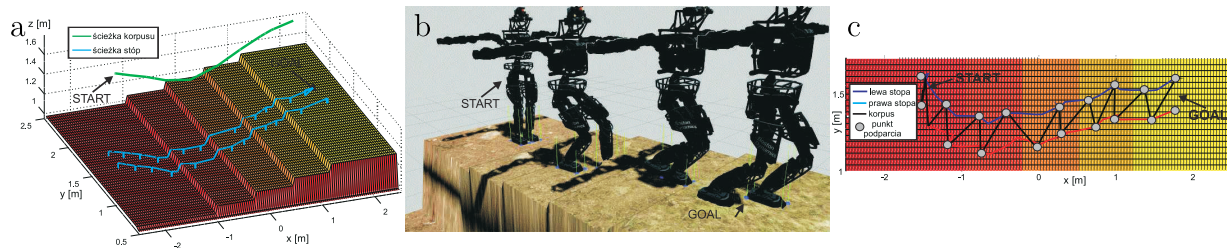
został ustalony jako niski, a koszt poruszania się po trawie i liściach jako wysoki, robot wybiera dłuższą ścieżkę, ale omijającą trawę i liście. W przypadku planowania ruchu robota kroczącego bez uwzględnienia informacji o rodzaju terenu (rys. 15d) robot znajduje najkrótszą ścieżkę pomiędzy punktem startowym a docelowym. W pracy [A10] pokazano również możliwość modyfikacji trybu kroczenia w zależności od rodzaju terenu. Poruszając się po trawie, robot wybiera chód pięciopodporowy, który jest chodem bezpieczniejszym. Natomiast planując swój chód na chodniku lub asfalcie, robot przełącza się na chód trójpodporowy, który jest chodem znacznie szybszym niż chód pięciopodporowy.

4.12 Rozszerzenia i uogólnienia zaproponowanych metod

W ramach prac nad metodami planowania ruchu podjąłem się również uogólnienia metody i implementacji planera ruchu na innych robotach kroczących. Ponieważ w instytucie IARiII dysponujemy wyłącznie robotami szczęściożnymi, prace były weryfikowane w symulatorze Gazebo lub we współpracy z ośrodkiem zagranicznym. W pracy [11] metoda planowania ruchu wykorzystująca algorytm RRT została przeniesiona na robota humoidalnego Atlas. Ponieważ metoda planowania ruchu wykorzystująca algorytm RRT jest modułowa, wystarczyła wymiana czterech modułów do użycia tego samego algorytmu na robocie dwunożnym.

W pierwszej kolejności zmodyfikowany został algorytm wyboru punktów podparcia. Robot Atlas, w przeciwieństwie do robota Messor II, ma płaskie stopy. Dlatego niemożliwe jest użycie metody zaprojektowanej dla sferycznej stopy robota Messor. Nowa metoda oceny jakości punktów podparcia ocenia obszar wokół punktu nominalnego danego przez planer RRT. Dla potencjalnych pozycji stopy obliczane są trzy współczynniki: odległość od punktu nominalnego, wariancja wysokości terenu pod powierzchnią stopy oraz kąt pomiędzy średnim wektorem normalnym do powierzchni a wektorem grawitacji. Na tej podstawie obliczany jest koszt potencjalnych punktów podparcia i spośród nich wybierany jest najlepszy.

Kolejnym modułem dopasowanym do robota jest moduł optymalizujący posturę robota. Ponieważ procedura planowania ruchu zakłada, że robot podczas wykonywania ruchu będzie stabilizowany przez sterownik reakcyjny, optymalizacja postury robota Atlas została znacząco uproszczona. Oblicza ona wysokość biodra nad terenem, która zapewnia minimalną odległość pomiędzy korpusem a terenem oraz znajduje się w przestrzeni roboczej. Ze względu na różnice w budowie robotów Atlas i Messor II zmodyfikowana została również



Rysunek 16: Planowanie robota humanoidalnego Atlas na schodach: zaplanowana ścieżka ruchu (a), symulowany robot w środowisku Gazebo podczas wykonywania ruchu (b) oraz wykonana ścieżka ruchu (c)

metoda sprawdzająca kolizje i przestrzeń roboczą.

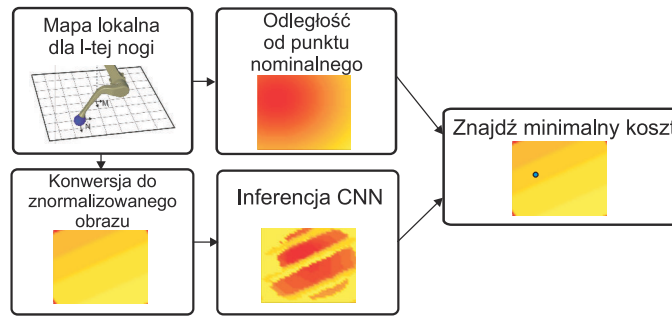
Przykładowe wyniki działania planera ruchu dla robota Atlas przedstawione zostały na rysunku 16. Zadaniem robota było znalezienie ścieżki ruchu umożliwiającej wejście na schody. Następnie robot realizował zadaną ścieżkę w środowisku symulacyjnym Gazebo (rys. 16b). Zaplanowana ścieżka różni się od ścieżki wykonanej przez robota, ponieważ sterownik reakcyjny dodatkowo balansuje korpusem w celu ustabilizowania robota.

Drugim przykładem rozszerzenia wcześniej opracowanych metod jest nowa metoda wyboru punktów podparcia zaimplementowana dla robota czteronożnego ANYmal. Wcześniej opracowana i używana w planerze ruchu metoda wyboru punktu podparcia działała sekwencyjnie. Dla każdego potencjalnego punktu podparcia przewidywany jest poślizg, który bezpośrednio określa jakość punktu podparcia. Następnie sprawdzane są ograniczenia kinematyczne i kolizje. Procedura sprawdzania wszystkich potencjalnych punktów podparcia jest iteracyjna i przez to czasochłonna.

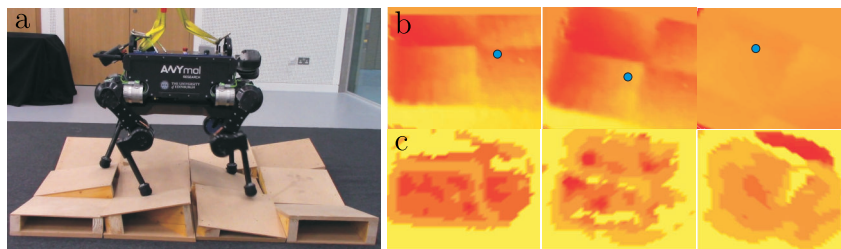
W pracy [A12] zaproponowałem nową metodę wyboru punktów podparcia dla robotów kroczących, która wykorzystuje właściwości konwolucyjnych sieci neuronowych. Konwolucyjne sieci neuronowe przyjmują na wejściu obraz i dzięki przetwarzaniu na GPU są w stanie bardzo szybko wyprodukować obraz wyjściowy. Obrazem wyjściowym w tym przypadku jest odpowiednio przygotowana mapa rastrowa, natomiast na wyjściu sieć generuje obraz, w którym każdy piksel określa jakość punktów podparcia w mapie (obrazie) wejściowej. Ponadto sieć neuronowa jest w stanie nie tylko oceniać geometrię mapy wejściowej, ale również uwzględnić ograniczenia kinematyczne robota oraz kolizje.

Procedura oceny punktów podparcia przedstawiona została na rysunku 17. W pierwszym kroku z mapy globalnej wycinany jest fragment wokół planowanej pozycji pierwszego ogniwa nogi. Następnie mapa rastrowa jest normalizowana i podawana na wejście sieci neuronowej, która ocenia jakość potencjalnych punktów podparcia i sprawdza ograniczenia kinematyczne. Niezależnie obliczany jest koszt związany z odległością od nominalnego punktu podparcia (punktu, który wybrałby robot gdyby teren był płaski). Informacja z sieci neuronowej i odległość od nominalnej pozycji stopy jest uwzględniana podczas wyznaczania optymalnego punktu podparcia.

Wynik eksperymentu na rzeczywistym robocie zaprezentowany został na rysunku 18. Eksperymenty na robocie ANYmal zostały przeprowadzone przez zespół z Uniwersytetu w



Rysunek 17: Procedura wyboru punktów podparcia dla robota ANYmal



Rysunek 18: Robot ANYmal wybierający punkty podparcia na makiecie nierównego terenu: robot na makiecie (a), lokalna mapa wysokościowa z wybranym punktem podparcia (b) oraz wyjście z sieci neuronowej (c). Ciemniejszy kolor oznacza lepszą ocenę punktu podparcia

Edynburgu. Pokazane zostały przykładowe mapy wysokościowe, na których zaznaczono wybrany punkt podparcia. Na rysunku 18 przedstawiono odpowiadające mapom wejściowym obrazy wygenerowane przez sieć neuronową. Można zaobserwować, że sieć lepiej ocenia punkty znajdujące się na płaszczyznach, natomiast odrzuca punkty znajdujące się na krawędziach. Na obrazach widać również zarys przestrzeni roboczej nogi. Punkty znajdujące się poza tym obszarem mają bardzo niską ocenę.

4.13 Podsumowanie oraz kierunki dalszych prac

W zamieszczonym opisie przedstawiłem główne wyniki moich badań naukowych zawartych w monotematycznym cyklu publikacji. Mają one charakter nowych metod, które poszerzają istniejące metody *state of the art* w zakresie percepcji, planowania i sterowania ruchem robota krocącego. Metody te mieszczą się w nurcie badań podstawowych w dyscyplinie naukowej automatyka i robotyka. Swoje prace skoncentrowałem na trzech głównych zagadnieniach podstawowych w robotyce: lokalizacji, budowie mapy oraz planowaniu ruchu. W większości zaproponowane metody zostały przetestowane na rzeczywistym robocie z użyciem danych z rzeczywistych sensorów, co znacząco skraca drogę do wykorzystania praktycznego zaproponowanych metod.

Poniżej zamieszczone zostały moje główne osiągnięcia związane z cyklem publikacji, które wnoszą wartość dodaną w rozwój dyscypliny automatyka i robotyka:

- Metoda planowania ruchu dla sześcionożnych robotów krocących oparta o algorytm

RRT, w którym problem eksploracji wielowymiarowej przestrzeni poszukiwań został podzielony na sekwencję problemów optymalizacji [A1].

- Adaptacja metody budowy mapy [A2] oraz systemu lokalizacji wizyjnej [A3] do działania na robocie krocącym oraz szeroka weryfikacja eksperymentalna.
- Metoda adaptacyjnego planowania ruchu robota krocącego, która wykorzystuje metody probabilistyczne do oceny przejścia przez robota wybranych fragmentów terenu [A4]. Metoda jest dostosowana do właściwości systemu percepcji robota.
- Metoda reakcyjnego sterowania robotem krocącym, która pozwala adaptować posturę robota podczas poruszania się po nierównym terenie na podstawie danych z sensorów enteroceptywnych [A5].
- Metoda modelowania niepewności pomiaru w systemie RGB-D SLAM, która poprawia wyniki dotyczące estymacji trajektorii robota za pomocą optymalizacji grafowej [A6] oraz szeroka weryfikacja eksperymentalna na robocie krocącym [A7].
- Metoda odometrii nożnej dla robotów krocących wykorzystująca dane z sensorów wewnętrznych robota i optymalizację w czasie rzeczywistym oraz kompletny system lokalizacji robota integrujący dane z odometrii nożnej i systemu wizyjnego SLAM za pomocą optymalizacji grafowej [A8].
- Nowa metoda budowania mapy otoczenia robota wykorzystująca zbiór map lokalnych połączonych ze sobą za pomocą grafu pozycji [A9].
- Metoda planowania ruchu wykorzystująca informację semantyczną do określenia ścieżki ruchu robota [A10].
- Metoda precyzyjnego planowania ruchu dla robota humanoidalnego [A11].
- Metoda jednoczesnego wyboru punktów podparcia i ewaluacji ograniczeń kinematycznych dla robota czteronożnego [A12].

Opracowane metody posiadają duży potencjał praktyczny. Mogą zostać wykorzystane w autonomicznych robotach krocących używanych w misjach poszukiwawczych i ratowniczych, czyli wszędzie tam, gdzie wysłanie człowieka niesie ze sobą ryzyko narażenia życia lub zdrowia. Opracowane metody umożliwiają zwiększenie autonomii robotów krocących. Zdalne sterowanie takim robotem jest utrudnione. Dzięki opracowanym metodom roboty mogą samodzielnie poruszać się w nieuporządkowanym środowisku naturalnym. Zadaniem człowieka jest tylko zdefiniowanie punktu docelowego. Robot podejmuje decyzję, gdzie postawić stopy i jak przemieszczać korpus, żeby dotrzeć do zadanej pozycji. System lokalizacji i budowy mapy znacząco ułatwia to zadanie.

5. Omówienie pozostałych osiągnięć naukowo-badawczych.

Do pozostałych osiągnięć naukowo-badawczych, które nie weszły w skład osiągnięcia wymienionego w pkt. 4b zaliczam wybrane publikacje związane z realizacją następujących prac badawczych:

a) **Detekcja i modelowanie właściwości obiektów manipulowanych**

Podczas stażu podoktorskiego na Uniwersytecie w Birmingham pracowałem nad przewidywaniem przez roboty ruchu manipulowanych obiektów [B1]. Robot na podstawie ruchu końcówki roboczej miał za zadanie przewidzieć zachowanie obiektów o różnym kształcie podczas kontaktu za pomocą jednego palca lub podczas chwycenia obiektu trzema palcami. W tym celu opracowana została metoda wykorzystująca wnioskowanie probabilistyczne. Dodatkowo zaproponowałem moduł, który za pomocą optymalizacji uwzględnia ograniczenia wynikające z kolizji, które są trudne do zamodelowania za pomocą wykorzystanych metod probabilistycznych. Tematykę związaną z robotami manipulacyjnymi kontynuuję w ramach projektu Lider, którego jestem kierownikiem. Jednym z wyników projektu jest praca [B2] dotycząca detekcji obiektów na obrazie RGB z użyciem konwolucyjnych sieci neuronowych.

b) **Projekt i wykonanie sterowników pokładowych robotów kroczących**

W ramach prac konstrukcyjnych związanych z robotami kroczącymi [B3,B4] zajmowałem się integracją sterowników pokładowych robotów kroczących. W ramach tych prac projektowałem i programowałem elektronikę pokładową. Jestem również głównym autorem oprogramowania dla robota Messor II. Oprogramowanie jest podzielone na wiele modułów. Część modułów uruchomiona jest na sterowniku pokładowym robota, a pozostałe na komputerze zewnętrznym. W pracy [B5] natomiast kierowałem pracami dotyczącymi projektowania konstrukcji podatnej nogi robota czterońskiego, w którym elementy sprężyste pomagają wykonać dynamiczne manewry, np. skoki.

c) **Modelowanie i symulacja robotów kroczących**

Przedmiotem moich prac badawczych było również modelowanie robotów kroczących z użyciem tzw. silników do gier. Są to biblioteki programowe, które umożliwiają iteracyjne modelowanie obiektów dynamicznych. Za ich pomocą możliwe jest tworzenie brył sztywnych, łączenie ich za pomocą złączy obrotowych i przesuwnych. W ten sposób zostały zasymulowane wszystkie roboty kroczące w laboratorium IARiII oraz robot StarLETH z ETH w Zurychu [B6]. Kierowałem później pracami nad symulatorem opartym o bibliotekę PhysX, który został sprametryzowany, tak żeby możliwe było wczytywanie dowolnych robotów [B7].

d) **Dodatkowe moduły sterujące dla robotów kroczących**

W pracy [B8] zaproponowałem strategię pozwalającą na dostosowanie trybu kroczenia do nierówności terenu. Wraz ze wzrostem nierówności terenu robot zmienia tryb kroczenia na bardziej bezpieczny chód tzw. „free gait”. Dzięki temu robot jest w stanie nie tylko bezpieczniej pokonywać nierówności, ale również radzi sobie z większymi przeszkodami. W pracy [B9] zaproponowana została metoda sterowania ruchem robota za pomocą gestów. Czujnik Kinect rozpoznaje

ruchy operatora, do których przypisane są rozkazy ruchowe robota. Na podstawie rozpoznanych gestów możliwe jest wykonywanie zdalnych operacji, włącznie z manipulacją obiektami.

e) **Kalibracja systemu sensorycznego robota oraz rozwój systemu budowy mapy**

Przed użyciem systemu sensorycznego na robocie do budowy modelu otoczenia konieczna jest kalibracja zewnętrzna czujników (kamer i dalmierzy laserowych). W pracy [B10] zaproponowane zostało wykorzystanie algorytmu PSO i optymalizacji metodą Levenberg-Marquardta do kalibracji kamery stereo i skanera laserowego 2D. Błąd został zdefiniowany na podstawie odległości punktów mierzonych przez oba sensory od płaszczyzny. Do kalibracji sensorów nie jest potrzebny specjalny marker, wystarczy fragment płaskiej podłogi lub ściana.

f) **Rozwój systemu SLAM**

W pracach [B11] i [B12] przedstawiono przykładowe prace związane z rozwojem systemu SLAM RGB-D. Opracowany system jest ogólny i umożliwia lokalizowanie dowolnego robota. W pracy [B11] przedstawiono porównanie różnych konfiguracji systemu opartego o graf pozycji. Przedstawiono też strategię zarządzania pozycjami w grafie i odrzucaniem błędnych pomiarów, które mogą być spowodowane np. niepoprawnym dopasowywaniem cech. W pracy [B12] przedstawiona została pełna architektura systemu PUT SLAM, który do lokalizacji robota wykorzystuje graf pozycji i cech.

- [B1] D. Belter, M. Kopicki, S. Zurek, J. Wyatt, Kinematically Optimised Predictions of Object Motion, IEEE/RSJ 2014 International Conference on Intelligent Robots and Systems, Chicago, USA, s. 4422–4427, 2014 (**konferencja CORE, kategoria A**)
- [B2] K. Piaskowski, D. Belter, Fast Object Detector based on Convolutional Neural Networks, In: Image Processing, Computer Vision, Pattern Recognition, and Graphics, Barneva R.P. et al. (Eds.), 6th International Symposium CompIMAGE'18 – Computational Modeling of Objects Presented in Images: Fundamentals, Methods, and Applications, s. 173–185, 2019 (**Best paper award**)
- [B3] D. Belter, K. Walas, A Compact Walking Robot - Flexible Research and Development Platform, Recent Advances in Automation, Robotics and Measuring Techniques, vol. 267, R. Szewczyk, C. Zieliński, M. Kaliczynska (Eds.), s. 343–352, 2014
- [B4] D. Belter, P. Skrzypczyński, K. Walas, D. Wlodkowic, Affordable Multi-Legged Robots for Research and STEM Education: A Case Study of Design and Technological Aspects, Recent Advances in Automation, Robotics and Measuring Techniques, vol. 351, R. Szewczyk, C. Zieliński, M. Kaliczynska (Eds.), s. 23–34, 2015
- [B5] M. Zieliński, D. Belter, Mechanical Design and Control of Compliant Leg for a Quadruped Robot, In: Szewczyk R., Zieliński C., Kaliczynska M. (eds) Automation 2018, Advances in Intelligent Systems and Computing, vol. 743, s. 500–509, 2018

- [B6] D. Belter, P. Skrzypczyński, K. Walas, P. Fankhauser, C. Gehring, M. Hutter, M. Hoepflinger, R. Siegwart, Dynamic simulation of legged robots using a physics engine, *Mobile Service Robotics*, World Scientific, K. Kozłowski et al. (Eds.), Singapore, s. 567–574, 2014
- [B7] M. Spis, A. Matecki, P. Maik, A. Kurzawa, M. Kopicki, D. Belter, Optimized and Reconfigurable Environment for Simulation of Legged Robots, In: Szewczyk R., Zieliński C., Kaliczyńska M. (eds) *Automation 2017, Advances in Intelligent Systems and Computing*, vol. 550, s. 290–299, 2017
- [B8] D. Belter, Gait Modification Strategy for a Six-legged Robot Walking on Rough Terrain, *Adaptive Mobile Robotics*, World Scientific, A. Azad et al. (Eds.), Singapore, s. 367–374, 2012
- [B9] W. Cieślak, S. Rodykow, D. Belter, Teleoperation of a Six-legged Walking Robot Using a Hand Tracking Interface, *Human-Centric Robotics*, World Scientific, M.E. Silva et al. (Eds.), Singapore, s. 527–536, 2017
- [B10] P. Łabęcki, D. Belter, Sensory system calibration method for a walking robot, *Journal of Automation, Mobile Robotics & Intelligent Systems*, Vol. 7, No. 2, s. 39–45, 2013
- [B11] D. Belter, M. Nowicki, P. Skrzypczyński, On the Performance of Pose-based RGB-D Visual Navigation Systems, *Computer Vision - ACCV 2014, part II, Lecture Notes on Computer Science 9004*, Springer, D. Cremers et al. (Eds.), London, s. 407–423, 2015 (**konferencja CORE, kategoria B**)
- [B12] D. Belter, M. Nowicki, P. Skrzypczyński, Accurate Map-Based RGB-D SLAM for Mobile Robots, *Robot 2015: Second Iberian Robotics Conference*, Vol. 418 of the series *Advances in Intelligent Systems and Computing*, s. 533–545, 2015

6. Podsumowanie

Mój dorobek po uzyskaniu stopnia doktora nauk technicznych obejmuje 42 artykuły (w tym 4 samodzielne). W czasopiśmie z listy Journal Citation Report zostało opublikowanych 7 artykułów. Natomiast 4 prace opublikowano w czasopiśmie specjalistycznych z zakresu automatyki i robotyki nieposiadających współczynnika wpływu (Impact Factor), m.in. JAMRIS (*Journal of Automation, Mobile Robotics & Intelligent Systems*), *Pomiary Automatyka Robotyka*. Kolejne 4 prace zostały opublikowane w materiałach konferencji z listy CORE, w tym dwie z kategorii A (*IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*) i dwie z kategorii B (*IEEE International Conference on Robotics and Automation (ICRA)*, *Asian Conference on Computer Vision (ACCV)*). Pozostałe publikacje zostały opublikowane w materiałach renomowanych konferencji międzynarodowych, m.in.: *Climbing and Walking Robots (CLAWAR)*, *International Conference on Robotics and Biomimetics (ROBIO)*, *IEEE International Symposium on Safety Security and Rescue Robotics (SSRR)*, *Iberian Robotics Conference (Robot)*, oraz odbywających się w Polsce konferencji międzynarodowych jak *Computational Modeling of Objects Presented in Images (CompIMAGE)* oraz *Automation*.

Baza Web of Science Core Collection (WoS) indeksuje 38 publikacji cytowanych 237 razy (w tym 130 bez autocytowań); H-index równy 9. Ponadto baza zawiera 49 publikacji nieindeksowanych w bazie (w zakładce „Cited Reference Search”) cytowanych 96 razy (w tym 53 cytowań bez autocytowań). Sumarycznie wynikiem przeszukania bazy Web of Science Core Collection jest 333 cytowań (w tym 183 bez autocytowań).

Baza Scopus indeksuje 41 publikacji cytowanych 323 razy (w tym 197 bez autocytowań); H-index równy 11. Dodatkowo 36 publikacji nieindeksowanych w bazie Scopus (pobrane z pola References artykułów cytujących – secondary documents) jest cytowanych 77 razy (w tym 48 cytowań bez autocytowań). Sumarycznie wynikiem przeszukania bazy Scopus jest 400 cytowań (w tym 245 bez autocytowań).

Sumaryczna liczba punktów Ministerstwa Nauki i Szkolnictwa Wyższego dla publikacji po doktoracie wynosi 559 (661 łącznie) (stan z dn. 12.04.2019 r.). Sumaryczny Impact Factor dla wszystkich publikacji po doktoracie w czasopismach z listy JCR zgodnie z rokiem opublikowania wynosi 13,656 (łącznie 17.181).

Aktywnie uczestniczę w międzynarodowych konferencjach naukowych. Od 2012 roku pełnię funkcję członka zespołu redakcyjnego czasopisma z listy JCR International Journal of Advanced Robotic Systems wydawanego obecnie przez wydawnictwo Sage. Od 2013 roku jestem w komitecie programowym konferencji European Conference on Mobile Robots, a od 2015 roku w komitecie programowym konferencji Climbing and Walking Robots. Jestem zapraszany do przygotowywania recenzji artykułów do czasopism indeksowanych w bazie JCR, np.: IEEE Transaction on Robotics, Robotics and Autonomous Systems, IEEE Transactions on Systems, Man, and Cybernetics: Systems, IEEE Robotics and Automation Letters, Industrial Robot: an International Journal, Journal of Intelligent and Robotic Systems, International Journal of Applied Mathematics and Computer Science, International Journal of Advanced Robotic Systems, Journal of Experimental & Theoretical Artificial Intelligence, IEEE Access oraz czasopism spoza bazy, np. Foundations of Computing and Decision Sciences.

Cyklicznie recenzuję prace zgłaszane na konferencje naukowe, takie jak: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE International Conference in Robotics and Automation, IFAC World Congress, Robot Motion and Control, European Control Conference, Climbing and Walking Robots, IEEE International Conference on Automation Science and Engineering, International Conference on Methods and Models in Automation and Robotics, Student Conference on Planning in Artificial Intelligence and Robotics.

Do tej pory brałem udział w trzech projektach europejskich. Dwa projekty: GeRT – Generalizing Robot Manipulation Tasks i PacMan – Probabilistic and Compositional Representations for Object Manipulation, w których byłem wykonawcą na Uniwersytecie w Birmingham, były finansowane z 7 Programu Ramowego (FP7). Obecnie jestem wykonawcą w projekcie THING – subTerranean Haptic INvestiGator, który jest finansowany z programu Horyzont 2020.

Po doktoracie kierowałem dwoma projektami naukowymi. Pierwszy z nich, pt. „No-

wa metoda planowania ruchu robota kroczącego i jej weryfikacja eksperymentalna”, był finansowany z Narodowego Centrum Nauki w ramach programu Preludium. Drugi projekt, którego jestem obecnie kierownikiem, pt. „Opracowanie i implementacja nowych metod lokalizacji, budowy mapy oraz planowania ruchu z użyciem czujników RGB-D w zrobotyzowanych systemach elastycznej produkcji”, jest finansowany przez NCBR w ramach programu Lider. Byłem również opiekunem naukowym grupy studentów realizujących projekt w programie *Generacja Przyszłości*, pt. „Projekt autonomicznego robota mobilnego do zadań poszukiwania/przynoszenia przedmiotów na potrzeby międzynarodowego konkursu Robots Intellect”. Byłem również wykonawcą w finansowanym przez NCN projekcie Opus, pt. „Nowa metoda jednoczesnej samolokalizacji i budowy mapy na podstawie danych RGB-D umożliwiająca precyzyjną nawigację robota autonomicznego w złożonym środowisku”. Przez 3 lata kierowałem pracami badawczymi w ramach projektów finansowanych z dotacji celowej na prowadzenie badań naukowych lub prac rozwojowych oraz zadań z nimi związanych, służących rozwojowi młodych naukowców (lata 2013-2016). W 2017 roku pełniłem również funkcję doradcy naukowego w projekcie „Adaptive Informative Path Planning in Autonomous Data Collection in Dynamic Unstructured Environments” realizowanym na CTU w Pradze.

Poza działalnością naukową prowadzę aktywnie działalność popularyzatorską i dydaktyczną. W 2012 roku uczestniczyłem w projekcie „Era Inżyniera. Rozbudowa potencjału rozwojowego Politechniki Poznańskiej” współfinansowanym z Europejskiego Funduszu Społecznego, z którego uzyskałem finansowanie na wygłoszenie referatu podczas konferencji IROS 2012. W sierpniu 2017 roku na zaproszenie oraz w ramach programu Erasmus+ prowadziłem wykłady i warsztaty dla studentów i doktorantów w Czech Technical University w Pradze, pt. „Summer School on Autonomous Navigation of Hexapod Crawling Robots”. W semestrze 2017/2018 prowadziłem dla studentów koła naukowego CybAiR kurs Robot Operating System. Aktywnie biorę udział w imprezach popularnonaukowych, podczas których wygłaszam prezentacje lub organizuję pokazy robotów, m.in.: EU Robotics Week 2016, festiwal robotyki CybairBot, prezentacja w ramach projektu „naukowcy - współcześni czarodzieje”, konwent fantastyki „Pyrkon”, Festiwal Nauki i Sztuki, seminarium Komisji Automatyki i Informatyki oddziału PAN w Poznaniu oraz spotkanie stowarzyszenia na rzecz rozwoju Sztucznej Inteligencji PSSI.

Do tej pory wypromowałem 24 magistrów, 26 inżynierów. W 2017 roku praca zrealizowana przez wypromowanych magistrantów, pt. „Konfigurowalny symulator robota kroczącego”, zdobyła wyróżnienie w konkursie Młodzi Innowacyjni 2017 organizowanym przez PIAP. Wynikiem mojej współpracy ze studentami jest 6 artykułów opublikowanych w czasopiśmie i materiałach konferencji międzynarodowych. Pełnię również funkcję promotora pomocniczego w trzech otwartych przewodach doktorskich. Latem 2018 roku organizowałem praktyki studenckie dla studenta z Hiszpanii w ramach programu IAESTE. W latach 2012-2013 byłem opiekunem koła naukowego CybAiR. Cyklicznie recenzuję również prace magisterskie na Uniwersytecie CTU

Prague. Prowadzę zajęcia dydaktyczne na kierunku *automatyka i robotyka* oraz *informatyka* na Politechnice Poznańskiej, które tematycznie wpisują się moje prace badawcze i obejmują:

- podstawy technik mikroprocesorowych (studia I stopnia)
- narzędzia i oprogramowanie dla systemów robotycznych (studia I stopnia)
- laboratorium problemowe (studia II stopnia)
- roboty autonomiczne (studia II stopnia)

Prowadzę również wykład w języku angielskim pt. „tools and software for robotic systems” na kierunku Automatic Control and Robotics oraz indywidualne zajęcia ze studentami z Hiszpanii w ramach programu Erasmus.

Odbyłem trzy staże podoktorskie na Uniwersytecie w Birmingham w latach 2013–2016, na którym spędziłem łącznie 12 miesięcy. W 2014 roku odbyłem również staż w firmie Airoptic sp.z o.o., gdzie realizowałem projekt pt. „Projekt i wykonanie systemu automatycznej oceny zachowania muszek z gatunku *Drosophila melanogaster* do testowania nowych leków przeciwdziałających degradacji układu nerwowego”.

Do tej pory otrzymałem 6 nagród przyznanych przez Rektora Politechniki Poznańskiej za działalność naukową (lata 2013–2018). Otrzymałem również nagrodę Polskiego Stowarzyszenia sztucznej Inteligencji (PSSI) w 2013 roku za najlepszą rozprawę doktorską ze sztucznej inteligencji w 2012 roku. Moje prace współautorskie zostały nagrodzone podczas konferencji międzynarodowych CompImage 2018 (Best Paper Award) oraz konferencji CLAWAR 2013 (Industrial Robot Innovation Award - Highly Commended paper). Zestawienie osiągnięć znajduje się w tabeli 1. Wybrane, najważniejsze osiągnięcia i aktywności po uzyskaniu stopnia doktora nauk technicznych wyszczególniono i scharakteryzowano w załączniku nr 4.

Tabela 1: Zestawienie osiągnięć

Opublikowane prace naukowe po doktoracie	42
Opublikowane prace naukowe przed doktoratem	23
Opublikowane prace naukowe w czasopismach z listy JCR przed doktoratem	3
Opublikowane prace naukowe w czasopismach z listy JCR po doktoracie	7
Udział w zespołach redakcyjnych czasopism z listy JCR	1
Udział w komitetach organizacyjnych konferencji międzynarodowych	14
Staże w zagranicznych ośrodkach badawczych	4
Recenzowanie publikacji w czasopismach międzynarodowych	20
Całkowita liczba recenzji prac naukowych	112
Kierowanie projektami badawczymi	5
Udział w projektach w roli innej niż kierownik	7
Udział w konferencjach naukowych międzynarodowych	16
Promotor prac inżynierskich (liczba osób)	20 (26)
Promotor prac magisterskich (liczba osób)	17 (24)
Promotor pomocniczy w rozprawach doktorskich	3
Sumaryczny IF opublikowanych publikacji po doktoracie według listy JCR	13.656
Sumaryczny IF wszystkich opublikowanych publikacji według listy JCR	17.181
Liczba cytowań publikacji według bazy WoS (bez autocytowań)	237 (130)
Indeks Hirscha według bazy WoS	9

Literatura

- [1] D. Belter, P. Skrzypczyński, Rough terrain mapping and classification for foothold selection in a walking robot, *Journal of Field Robotics*, vol. 28(4), s. 497–528, 2011
- [2] A.J. Davison, I. Reid, N. Molton, O. Stasse, MonoSLAM: real-time single camera SLAM, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 29(6), s. 1052–1067 (2007)
- [3] I. Dryanovski, R. Valenti, J. Xiao, Fast visual odometry and mapping from RGB-D data, *IEEE International Conference on Robotics and Automation*, Karlsruhe, s. 5704–5711, 2013
- [4] F. Endres, J. Hess, J. Sturm, D. Cremers, W. Burgard, 3D Mapping with an RGB-D Camera, *IEEE Transactions on Robotics*, s. 177–187, 2014
- [5] J. Engel, T. Schöps, D. Cremers, LSD-SLAM: Large-Scale direct monocular SLAM, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), *Computer Vision – ECCV 2014*, Springer, Cham, 2014, s. 834–849 (2014)

- [6] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, R. Siegwart, Robot-centric elevation mapping with uncertainty estimates, International Conference on Climbing and Walking Robots (CLAWAR), Poznan, Poland, s. 433–440, 2014
- [7] E.B. Hansen, Nonparametric conditional density estimation, Neural Computation, Vol. 21(2), s. 533–559, 2004
- [8] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, OctoMap: An efficient probabilistic 3D mapping framework based on octrees, Autonomous Robots, Vol. 34(3), s. 189–206, 2013
- [9] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evolutionary Computation, Vol. 15(1), s. 1–28, 2007
- [10] J. Kennedy, R. Eberhart, Particle swarm optimization, IEEE International Conference on Neural Networks, Perth, Australia, s. 1942–1948, 1995
- [11] C.T. Kelley, Iterative methods for optimization, SIAM, Philadelphia, 1999
- [12] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, Proc. International Symposium on Mixed and Augmented Reality, Nara, Japan, s. 225–234, 2007
- [13] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, g2o: A General Framework for Graph Optimization, IEEE International Conference on Robotics and Automation, s. 3607–3613, 2011
- [14] R. Mur-Artal, J.D. Tardós, ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras, IEEE Transaction Robotics, Vol. 33(5), s. 1255–1262, 2017
- [15] R. Mur-Artal, J.M.M. Montiel, J.D. Tardós, ORB-SLAM: A versatile and accurate monocular SLAM system, IEEE Transactions Robotics, Vol. 31(5), s. 1147–1163, 2015
- [16] A. Roennau, G. Heppner, M. Nowicki, J.M. Zoellner, R. Dillmann, Reactive posture behaviors for stable legged locomotion over steep inclines and large obstacles, IEEE/RSJ 2014 International Conference on Intelligent Robots and Systems, s. 4888–4894, 2014
- [17] J. Saarinen, H. Andreasson, T. Stoyanov, A.J. Lilienthal, 3D normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments, International Journal of Robotics Research, Vol. 32(14), pp. 1627–1644, 2013
- [18] S. Umeyama, Least-squares estimation of transformation parameters between two point patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.13(4), pp. 80–92, 1991

Belter

.....
Dominik Belter