

LECH Piotr¹

WIZYJNE STEROWANIE ROBOTAMI MOBILNYMI PODĄŻAJĄCYMI ZA LIDEREM

W pracy przedstawiono koncepcję zastosowania techniki rzeczywistości rozszerzonej (Augmented Reality) w sterowaniu robotami mobilnymi. Przeprowadzono analizę doboru kamer wideo względem stabilności algorytmu sterowania. Zbadano również wpływ na sterowanie w zależności od zmiany odległości między znacznikiem a kamerą. Badania przeprowadzono w oparciu o testową aplikację z wykorzystaniem NyARToolkit dla realizacji zagadnień związanych z techniką Augmented Reality. Sterowanie robotem zrealizowano w oparciu o symulator robotów mobilnych Simbad.

VISION BASED LEADER-FOLLOWER FORMATION CONTROL FOR MOBILE ROBOTS

The paper presents an expanded concept of Augmented Reality applications for the control of mobile robots. The study was designed to verify the impact of use of different cameras on the stability of algorithms. Indirectly, it was also examined the impact of the distance between the marker and the camera. The Simbad simulator of mobile robots with a NyARToolkit (extended library for Augmented Reality) was used to implement the test application.

1. WSTĘP

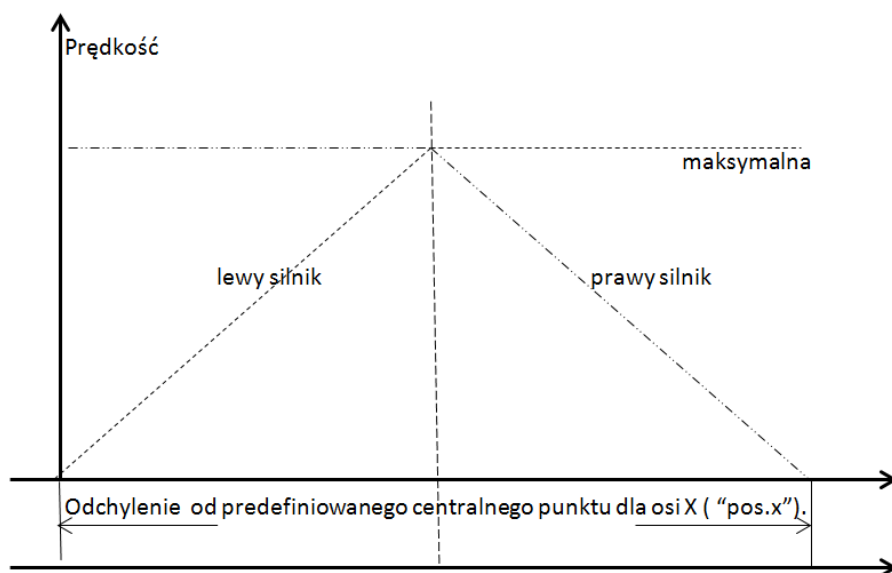
Jedną ze strategii zautomatyzowanego poruszania się robotów mobilnych lub innych autonomicznych urządzeń transportowych (np. wózków z towarem) jest podążanie za liderem. Liderem może być robot poruszający się względem zadanej trajektorii ruchu lub człowiek. Podstawowym zadaniem urządzenia mobilnego jest śledzenie i powtarzanie ruchu lidera [1]. Technika rzeczywistości rozszerzonej (AR Augmented Reality) jest stosowana w studiach filmowych lub telewizji. Pozwala na łączenie świata wirtualnego z rzeczywistym [2]. Algorytmy AR mogą być wykorzystane w sterowaniu robotem mobilnym podążającym za liderem, ze względu na pokrewieństwo realizowanych zadań. Techniki AR opierają się między innymi o algorytmy śledzenia znaczników i transformacji obrazów pozyskanych z kamery wideo. Obliczone parametry ruchu znaczników są wykorzystane do obliczenia pozycji modelu 3D, który ostatecznie zostaje osadzony na

¹Piotr Lech, Katedra Przetwarzania Sygnałów i Inżynierii Multimedialnej, Zachodniopomorski Uniwersytet Technologiczny w Szczecinie ul. 26. Kwietnia 10,71-126 Szczecin E-mail: piotr.lech@zut.edu.pl

obserwowanej klatce rzeczywistego obrazu pochodzącego z kamery wideo. Jednocześnie parametry niezbędne do renderowania obiektu 3D są podstawą do wyznaczenia sterowania robota mobilnego.

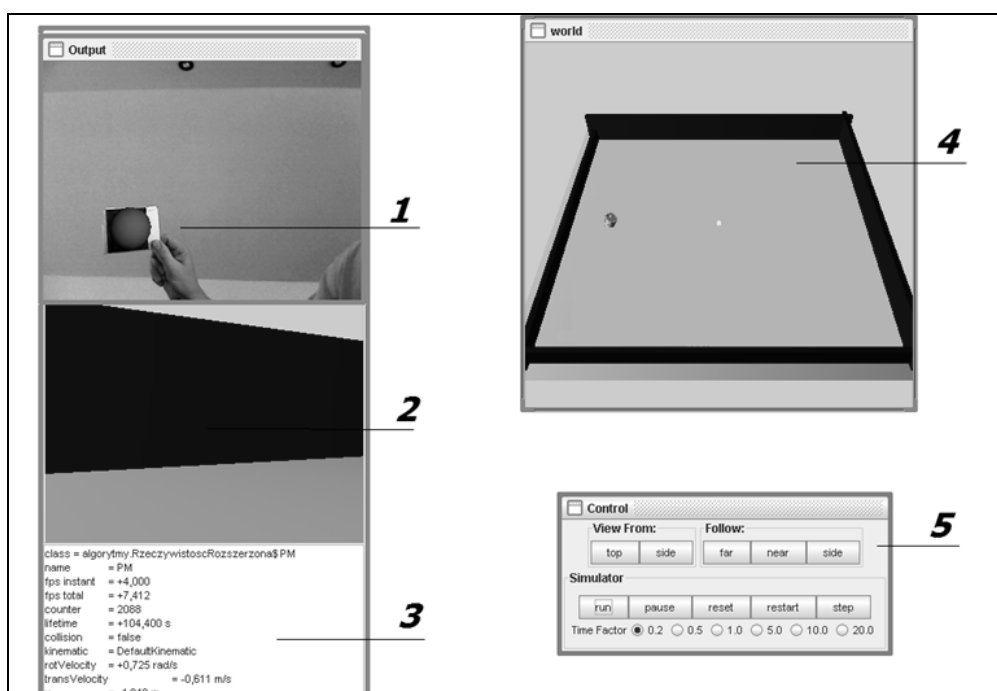
2. KONCEPCJA STEROWANIA

Sterowanie robotem odbywa się za pomocą kwadratowego arkusza z wydrukowaną czarną ramką, która jest śledzona i osadzonym w niej znakiem graficznym, który podlega identyfikacji. Techniki AR pozwalają na śledzenie wielu tak wykonanych znaczników. Robot posiada różnicowe sterowanie napędu (rys. 1). Znacznik umiejscowiony jest na liderze będącym w ruchu przed obiektywem kamery wideo. Zadaniem robota jest powtórzenie tego ruchu, w powiązaniu z odległością między liderem a robotem. Warunkiem niezbędnym do uruchomienia silników jest konieczność wystąpienia zmiany odległości między kamerą a znacznikiem. Wizualizacja zmian ruchu znacznika odbywa się poprzez właściwe pozycjonowanie i operowanie wielkością obiektu 3D. Ruch do przodu realizowany jest wtedy, gdy znacznik jest wciąż na tej samej pozycji centralnej (na kolejnych klatkach obrazu wideo) a obiekt 3D zwiększa swoją wielkość. Obrót w lewo (lub prawo) następuje kiedy znacznik przesuwa się w lewo (lub prawo) ze wstępnie zdefiniowanej pozycji centralnej. Skręt robota zależy od odchylenia znacznika od pozycji środkowej. Na podobnej zasadzie działa robot podążający za linią o sterowaniu zrealizowanym w oparciu o metody wizyjne [3]. W naszym przypadku pozycja znacznika, a w konsekwencji wyrenderowanego obiektu 3D, zastępuje linię.



Rys.1. Koncepcja sterowania różnicowego

W praktyce program testowy identyfikuje przesunięcia znacznika - w celu obliczenia współrzędnych i niezbędnych transformacji do osadzenia obiektu 3D (kula) - na klatce wizualizowanego obrazu (rys. 2). Jednocześnie wyliczona pozycja obiektu 3D, w postaci współrzędnych przekazana jest do procedury sterowania robotem mobilnym.



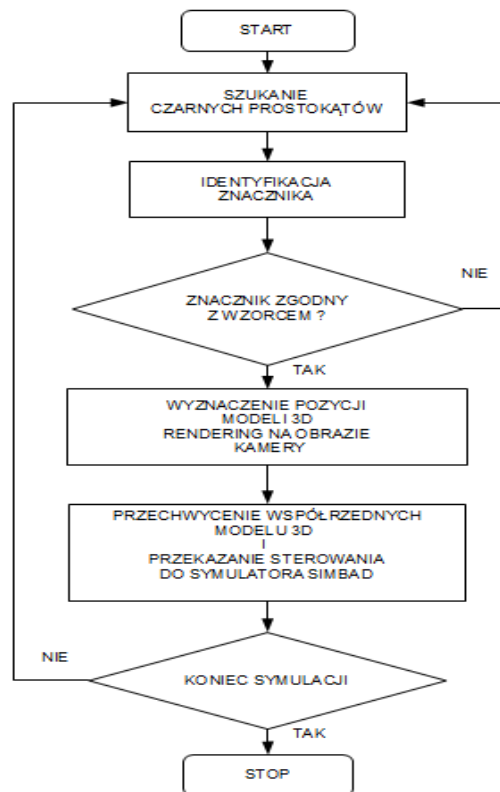
Rys.2. Aplikacja testowa (1-widok z kamery z odnalezionym znacznikiem i nałożonym obiektem 3D, 2-widok z kamery robota, 3-parametry symulacji, 4- wirtualne środowisko robota, 5-panel kontrolny symulatora)

3. WERYFIKACJA PRAKTYCZNA

3.1 Aplikacja testowa

Zadaniem aplikacji testowej (rys. 2) było sprawdzenie przydatności technologii AR w omawianym w pracy zagadnieniu. Program został napisany w języku Java w środowisku NetBeans. Symulator robotów mobilnych Simbad posłużył do realizacji modelu robota wraz z jego otoczeniem. Biblioteka AR pochodzi z pakietu NyARToolkit przystosowanego do realizacji zadań w czasie rzeczywistym, będącego odpowiednikiem biblioteki JARToolkit [4]. Natomiast pakiet Java Media Framework został wykorzystany do akwizycji obrazów z kamery wideo. Simbad [5] jest kompletnym środowiskiem do symulacji robotów mobilnych w języku JAVA z dostępnymi kodami źródłowymi. Symulator tworzy trójwymiarowe otoczenie, z serii klas niezbędnych do realizacji różnego rodzaju obiektów 3D, pomiędzy którymi porusza się robot. W eksperymentach użyto

zdefiniowanego wstępnie modelu robota Kaphera. Schemat blokowy aplikacji przedstawia rysunek 3.



Rys.3. Schemat blokowy aplikacji

NyARToolkit zapewnia:

- prosty kod do kalibracji kamery,
- pozycjonowanie i śledzenie orientacji kamery wideo,
- możliwość zastosowania dowolnych wzorów (identyfikacja) osadzonych w czarne kwadraty,
- kod śledzenia,
- kod wystarczająco szybki do użycia w dowolnej aplikacji AR w czasie rzeczywistym.

Program testowy określa przesunięcie znaczników [6] za pomocą funkcji dostępnych w NyARToolkit, w celu obliczenia współrzędnych i niezbędnych przekształceń do renderowania obiektów 3D. Te same współrzędne są podstawą obliczania parametrów sterowania robota. Kluczowym elementem programu jest procedura onUpdate (rys. 4), która pochodzi z biblioteki NyARToolkit. Procedura ta jest wywoływana za każdym razem,

gdy współrzędne znacznika są zmieniane. Gdy zmiany są zerowe, w stosunku do poprzedniej klatki, nic się nie dzieje. Jednakże, jeśli któraś z tych wartości będzie zmieniona, nowy wynik przekazywany jest do zmiennej „pos”. Zmienna „scale” wykorzystywana jest przy kalibracji.

```
public void onUpdate (boolean i_is_marker_exist, i_transform3d
Transform3D) {

    if (i_transform3d! = null) {

        i_transform3d.get (pos);

        pos.x * = scale.x;

        pos.y * = scale.y;

        pos.z * = scale.z;

    }

}
```

Rys.4. Procedura onUpdate.

3.2 Kalibracja

Przed każdą zmianą kamery wideo, system został skalibrowany [7,8] z wykorzystaniem narzędzi dostarczonych wraz z pakietem JARToolkit. Potencjalne możliwości pomyłki wynikające z błędów programistycznych zostały usunięte. Do automatyzacji badań wykonano urządzenie zapewniające powtarzalność procesu badawczego (rys. 5). Współczynnik „scale” dobierano tak, aby wizualizacja w środowisku Simbad była optymalna. Kalibracja stanowiska badawczego polegała na zdefiniowaniu punktów skrajnych i centralnego dla znacznika, oraz odpowiadających im pozycją w środowisku Simbad.

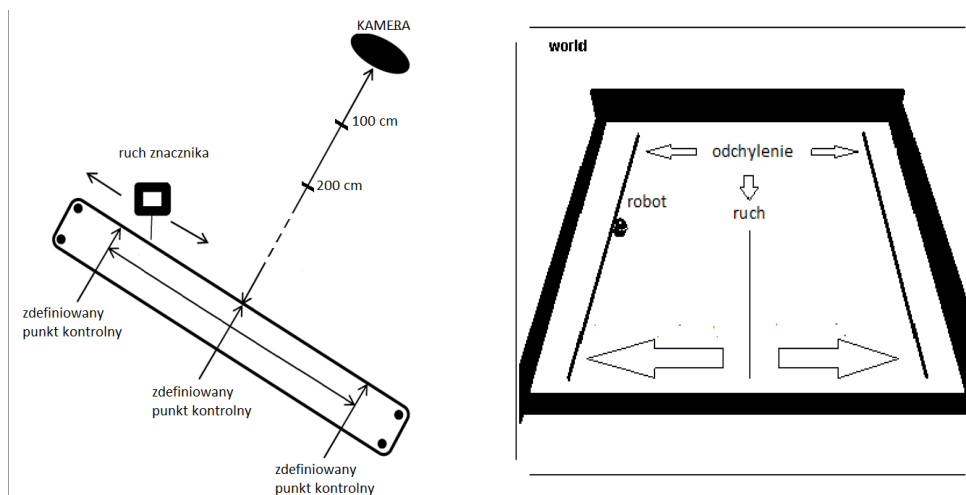
3.3 Dobór kamery

Podczas badań zaobserwowano pewne nieprawidłowości w działaniu robota, wynikające z braku możliwości określenia pozycji lub identyfikacji znacznika. Przyczyną utraty stabilności okazała się niska jakość kamery (tabela I). Głównym źródłem błędów była zła jakość cyfrowych obrazów uzyskanych z kamery [7].

W badaniach wykorzystano trzy kamery:

- 1 - Prosta kamera kolorowa, USB Dual Mode DSC (2770), maksymalna rozdzielczość 320 x 240 pikseli, obiektyw typu pinhole.
- 2 - Kamera Logitech USB Webcam Ka20s, maksymalna rozdzielczość 640 x 480 pikseli, obiektyw z ręczną regulacją ostrości.

3 - Media Tech MT4018 AUTOPIX, maksymalna rozdzielczość 1280 x 1024 pikseli wyposażony w automatyczną regulację ostrości.



Rys.5. Schemat ilustrujący automatyzację badań

W programie korzystano z wstępnie ustalonej rozdzielczości 320x240 pikseli preferowanej w środowisku Simbad. Badania wykazały, że główną przyczyną utraty kontroli nad robotem była utrata ostrości prostokątnego obiektu ze znacznikiem, przez co niemożliwe było zidentyfikowanie znacznika. Najlepsze efekty przyniosło użycie kamery z autofocusem (automatyczną regulacją ostrości) zapewniającym ciągłe utrzymanie ostrości na obserwowanym obiekcie.

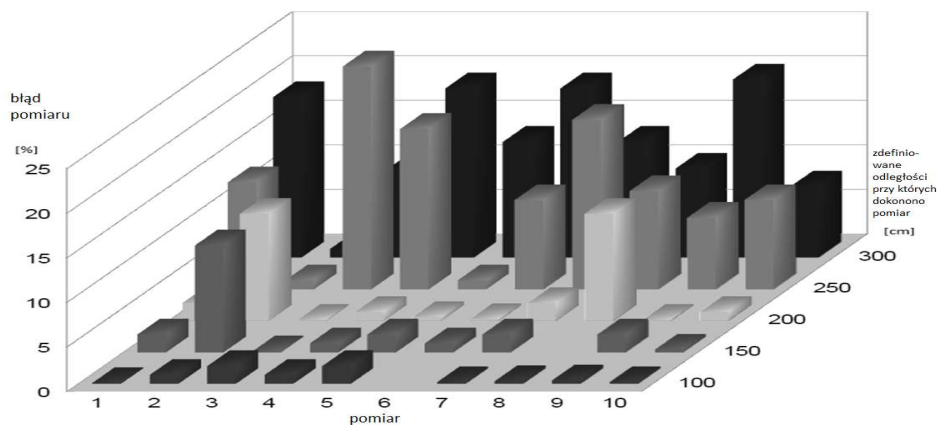
Tab. 1. Parametry interfejsów pomiarowych

Kamera	Utrata stabilności w 100 pomiarach
1. prosta kamera internetowa z obiektywem typu pinhole	60
2. kamera internetowa z obiektywem umożliwiającym ręczną regulację ostrości	45
3. kamera internetowa z automatyczną regulacją ostrości	0

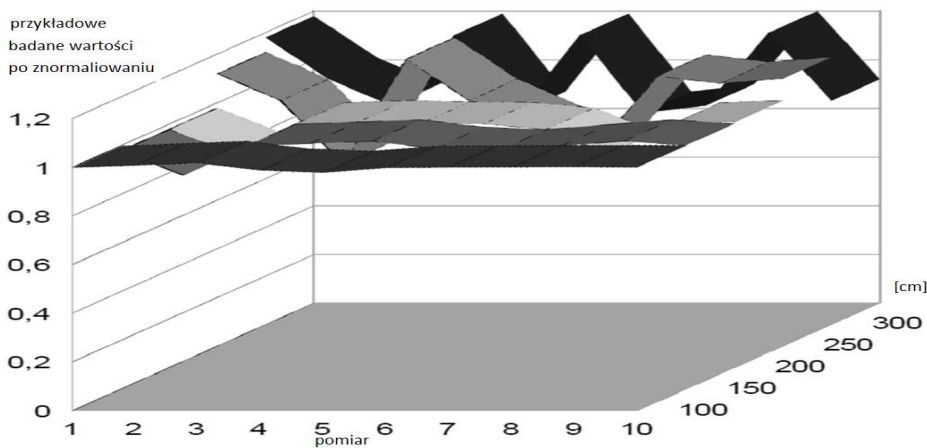
3.4 Testy odwzorowania ruchu

Badanie zostało przeprowadzone w oparciu o zautomatyzowany ruch znacznika (rys. 5). Trzy pozycje zostały zdefiniowane na etapie kalibracji - centralna, skrajna lewa, skrajna prawa - dla środowiska realnego i odpowiadającego mu wirtualnego. W testach

rejestrowano maksymalne odchylenie od centralnej pozycji w środowisku symulatora i przyrównywano je do wcześniej zdefiniowanych.



Rys.6. Błędy odwzorowania osiągnięcia maksymalnego odchylenia od punktu centralnego dla różnych odległości markera od kamery.



Rys.7. Znormalizowane wartości odwzorowania osiągnięcia maksymalnego odchylenia od punktu centralnego dla różnych odległości markera od kamery

Wszystkie wartości liczbowe w kierunku osi X zostały znormalizowane względem zdefiniowanych skrajnych lokalizacji.

Badania przeprowadzono dla określonych odległości bazowych (rys. 5) od kamery. Największe z błędów [9] pozycjonowania robota zaobserwowano na długich dystansach do kamery. Na rysunku 6 i 7 pokazano przykłady znormalizowanych wartości i błędy pozycjonowania robota, przy wykorzystaniu najlepszej z dostępnych kamer.

4. WNIOSKI

Podczas przeprowadzonych testów zauważono i ustalono, że niezbędne do poprawnego sterowania wirtualnym robotem jest pozyskanie ostrego obrazu prostokąta i znacznika. Najlepszym rozwiązaniem było użycie kamery z automatyczną regulacją ostrości. Simbad i biblioteki NyARToolkit - są to bardzo dobre środki do szybkiego prototypowania programów z technikami Augmented Reality. Prezentowany algorytm może zostać wdrożony w systemach, w których robot mobilny ma za zadanie podążać za liderem.

5. BIBLIOGRAFIA

- [1] Cowan, N., Shakernia, O., Vidal, R., & Sastry, S.: „*Vision-based*” follow-the-leader, Proceedings of IEEE/RSJ ICIRS, pp. 1797-1801, Las Vegas, October 2003.
- [2] Green S. A., Billingham M., Chen X., and Chase J. G.: *Human-Robot Collaboration: A Literature Review and Augmented Reality Approach in Design*, International Journal of Advanced Robotic Systems, vol. 5, pp. 1- 18, March, 2008.
- [3] Okarma K., Lech P.: *A fast image analysis technique for the line tracking robots*, Lecture Notes in Artificial Intelligence, vol. 6114 (Part II), Springer Verlag, ICAISC 2010.
- [4] Geiger C., Paelke V., Reimann C., Stoecklein J.R.: *JARToolkit-A Java Binding for ARToolkit*. 1st IEEE int. Workshop on the ARToolkit, Darmstadt, Sep 2002.
- [5] Hugues L., Bredèche N.: *Simbad: an Autonomous Robot Simulation Package for Education and Research*, Simulation of Adaptive Behavior (SAB) (2006). Rome, Italy. 2006.
- [6] Azuma R.T.: *A survey of augmented reality*, Presence: Teleoperators and virtual environments, vol. 6, no. 4, pp. 355-385, 1997.
- [7] Tsai R.: *A versatile camera calibration technique for highaccuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses*, IEEE Journal of Robotics and Automation, pp.323-344, August 1987.
- [8] Brown D.C.: *Close-range camera calibration*, Photogrammetric Engineering, pp. 855-866, 1971.
- [9] Holloway R.: *Registration Errors in Augmented Reality*, Ph.D dissertation. UNC Chapel Hill Department of Computer Science technical report TR95-016, August 1995.