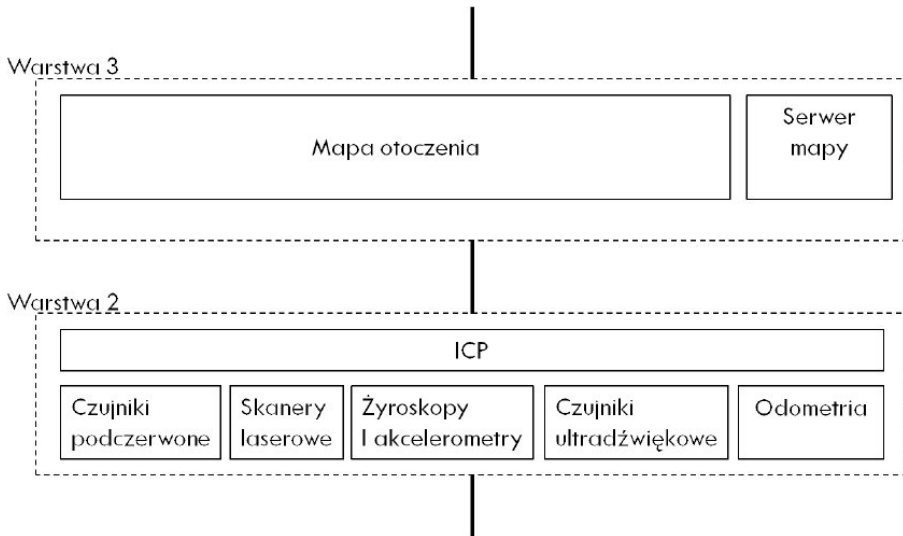


Dominik Sankowski*, Sławomir Jeżewski*, Adam Wulkiewicz*

Uogólniony model dwuwymiarowego czujnika pomiaru odległości dla systemu operacyjnego robota mobilnego

1. System operacyjny robota mobilnego

System operacyjny jest to oprogramowanie zarządzające sprzętem komputerowym, tworzące środowisko do uruchamiania i kontroli zadań użytkownika. Stanowi on swego rodzaju warstwę abstrakcji. System operacyjny proponowany przez nas posiada strukturę warstwową. W kontekście modelu czujnika ważne są dwie warstwy systemu operacyjnego (rys. 1), druga i trzecia.



Rys. 1. Ogólny schemat drugiej i trzeciej warstwy systemu operacyjnego robota mobilnego

* Katedra Informatyki Stosowanej, Politechnika Łódzka

W warstwie drugiej znajdują się mechanizmy obsługi różnego rodzaju czujników odległości, położenia i orientacji. Dane z czujników integrowane są w warstwie trzeciej w celu budowy mapy otoczenia, która może zostać wykorzystana przez wyższe warstwy systemu operacyjnego na przykład przez algorytm wyznaczania ścieżki ruchu robota.

2. Uogólniony model dwuwymiarowego czujnika pomiaru odległości dla systemu operacyjnego robota mobilnego

Dane pomiarowe zbierane są z czujników różnego rodzaju. Czujniki te mają często różne charakterystyki oraz właściwości. Następnie dane te muszą zostać zintegrowane w mapie otoczenia. Jest ona strukturą danych nazywaną siatką zajętości. Posiada ona strukturę siatki i przechowuje prawdopodobieństwa występowania obiektów w przestrzeni. Ponieważ celem jest wykorzystanie czujników o różnej charakterystyce należało opracować matematyczny model czujnika uniwersalnego oraz algorytm integracji danych w przestrzeni percepcyjnej.

2.1. Dwuwymiarowe czujniki odległości stosowane w robotyce

Do najczęściej stosowanych w robotyce dwuwymiarowych czujników odległości należą czujniki ultradźwiękowe, świetlne, w tym podczerwone i skanery laserowe.

Zasada działania czujników ultradźwiękowych (rys. 2) jest podobna do zasady działania radaru. Generują one falę ultradźwiękową i badają czas powrotu echa odbitego od przeszkody. Wykorzystuje się je również w procesach przemysłowych do wykrywania obiektów na liniach produkcyjnych oraz w medycynie. Pomiaru wykonane przez ten czujnik mogą różnić się w zależności od rodzaju powierzchni obiektu, od którego odbiła się fala ultradźwiękowa.



Rys. 2. Zbliżeniowy czujnik ultradźwiękowy firmy SICK

Istnieje wiele rodzajów czujników świetlnych mających różne zastosowanie. Najczęściej stosowanymi w robotyce są zbliżeniowe czujniki podczerwone (rys. 3). Emitują one światło podczerwone i badają jaka część tego światła do nich powróciła.



Rys. 3. Zbliżeniowy czujnik podczerwony firmy Sharp

Dalmierze laserowe są to urządzenia wykorzystujące wiązkę lasera w celu wyznaczenia odległości od obiektu. Odległość może być obliczana na zasadzie zbadania czasu powrotu wygenerowanego impulsu lub wyznaczenia przesunięcia fazowego wiązki powracającej do czujnika. Dwuwymiarowe skanery laserowe (rys. 4) rozszerzają funkcjonalność dalmierzy. Wykonują one pomiary w pewnym zakresie kątów typowo 90° , 180° , 270° czy 360° . Pomiary wykonywane są co pewną stałą wartość kąta. Jest to możliwe poprzez obracanie lustra odbijającego wiązkę lasera w odpowiednim kierunku i wykonywanie pomiarów dla odpowiedniego ustawienia osi optycznej.

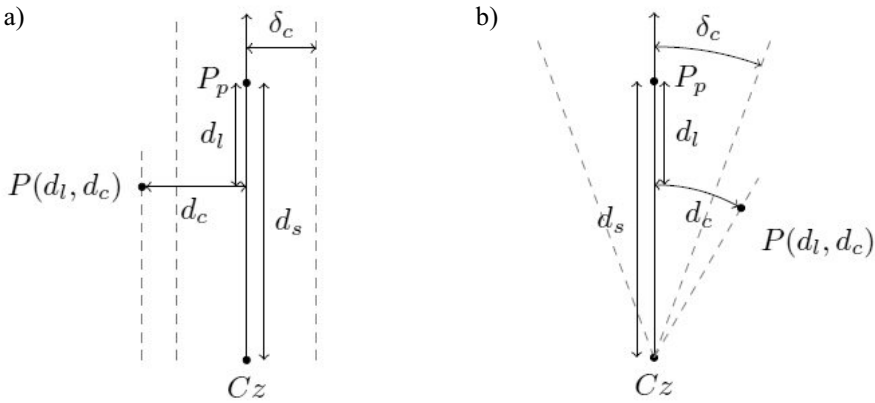


Rys. 4. Dwuwymiarowy skaner laserowy firmy SICK

2.2. Model matematyczny

Model matematyczny czujnika opisuje prawdopodobieństwo występowania obiektu w otoczeniu punktu pomiarowego. Na wartość prawdopodobieństwa ma wpływ rodzaj

czujnika oraz błędy pomiaru, jakimi się on charakteryzuje. Pierwszym rodzajem błędów pomiaru są błędy, które nazwano poprzecznymi. W zależności od rodzaju czujnika mogą one zależeć od kąta odchylenia od punktu pomiarowego lub od odległości od tego punktu. Błąd poprzeczny zależny od kąta jest charakterystyczny dla czujników ultradźwiękowych, gdzie prawdopodobieństwo wystąpienia obiektu w przestrzeni maleje wraz z odchyleniem kątowym od punktu pomiaru. W czujnikach laserowych, błąd poprzeczny ma charakter odległościowy. Drugim rodzajem błędów są błędy podłużne, które mogą być względne lub bezwzględne, czyli odpowiednio zależne lub niezależne od odległości punktu pomiarowego.



Rys. 5. Charakterystyka czujników: a) czujnik charakteryzujący się poprzecznym błędem d_c odległościowym; b) czujnik charakteryzujący się poprzecznym błędem d_c kątowym

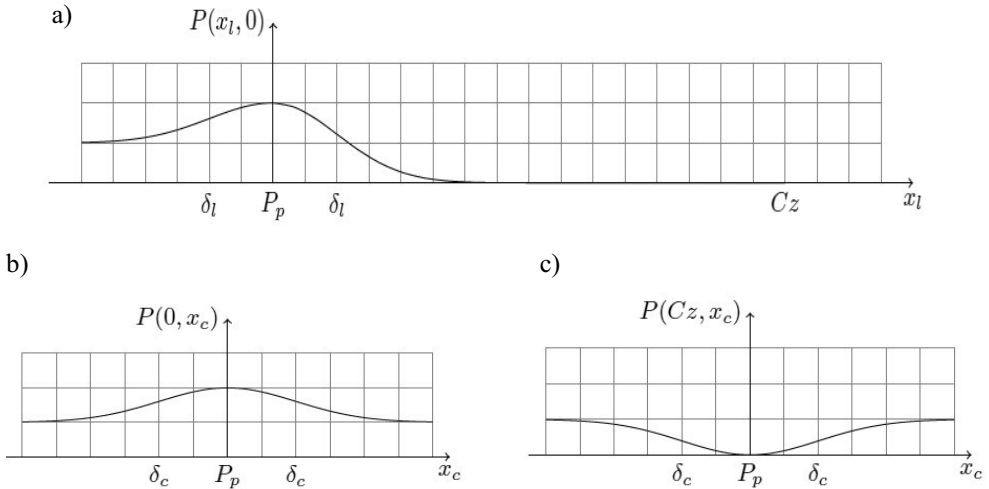
Na rysunku 5 przedstawione są dwa rodzaje czujników charakteryzujące się różnymi błędami poprzecznymi. Odległość punktu pomiarowego P_p od czujnika wynosi d_s . Prawdopodobieństwo wystąpienia obiektu obliczane jest dla punktu $P(d_l, d_c)$. Odległość d_l jest podłużną składową wektora odległości punktu $P(d_l, d_c)$ od punktu pomiarowego P_p . Odległość d_c jest składową odległościową poprzeczną i w zależności od typu czujnika jej jednostką jest jednostka długości albo miara kąta.

$$P(d_l, d_c) = \begin{cases} 0,5 + \left(e^{-\frac{d_l^2}{2\delta_l^2}} - 0,5 \right) e^{-\frac{d_c^2}{2\delta_c^2}} & \text{dla } |P, Cz| < |P_p, Cz| \\ 0,5 + 0,5 e^{-\frac{d_l^2}{2\delta_l^2} - \frac{d_c^2}{2\delta_l^2}} & \text{dla } |P, Cz| \geq |P_p, Cz| \end{cases} \quad (1)$$

gdzie:

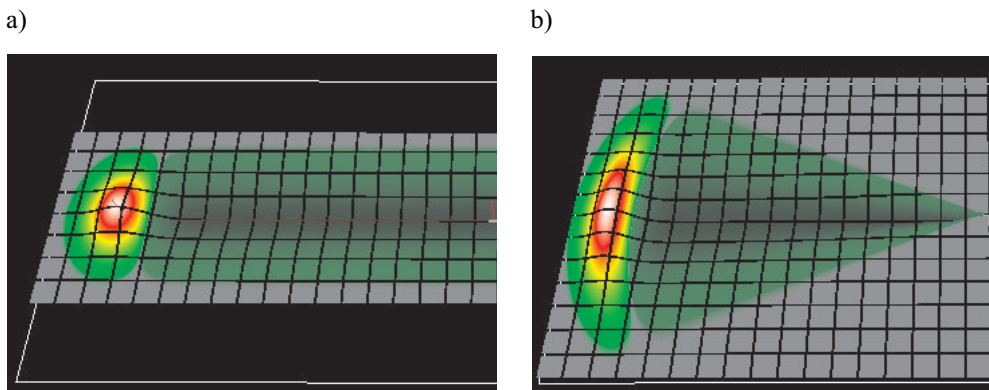
d_l, d_c – długości składowych podłużnej i poprzecznej,
 δ_l, δ_c – odchylenia standardowe podłużne i poprzeczne.

Model matematyczny czujnika przedstawia wzór (1). Po zastosowaniu centralnego twierdzenia granicznego funkcja prawdopodobieństwa może być modelowana za pomocą funkcji Gaussa. Zagadnienie zostało podzielone na dwie części. Część przestrzeni, której punkty znajdują się dalej od czujnika niż punkt pomiarowy, oraz część przestrzeni, której punkty znajdują się bliżej. Co więcej, łatwo można zauważyć, że wzory można rozbić na dwie części składowe, podłużną oraz poprzeczną. Obrazuje to rysunek 6 i rysunek 7.



Rys. 6. Składowe podłużna i poprzeczna. Przesunięcie w osi podłużnej oznaczone jest jako x_l , natomiast w osi poprzecznej jako x_c , przy czym przesunięcie liczone jest od pozycji próbki;

- a) składowa podłużna; b) składowa poprzeczna dla $x_l = 0$ (pozycja próbki);
c) składowa poprzeczna dla $x_l = Cz$ (pozycja czujnika)



Rys. 7. Dwuwymiarowy wykres prawdopodobieństwa punktu $P(d_p, d_c)$:

- a) czujnik charakteryzujący się błędem odległościowym; b) czujnik charakteryzujący się błędem kątowym

3. Algorytm integracji w siatce zajętości

Siatka zajętości charakteryzuje się tym, że przechowuje nie tylko dane opisujące występowanie obiektów w przestrzeni, ale też dane opisujące przestrzeń pustą oraz niezbadaną. W czasie integracji modyfikowany jest obszar siatki, który pokrywa się z pomiarem wykonanym przez czujnik. Prawdopodobieństwa wszystkich punktów siatki w tym obszarze muszą zostać zmodyfikowane według wzoru (2). Więcej informacji można znaleźć w [1].

$$P(p) = \left(1 + \frac{1 - P_{new}(p)}{P_{new}(p)} \frac{1 - P_{old}(p)}{P_{old}(p)} \right)^{-1} \quad (2)$$

gdzie:

- $P(p)$ – zmodyfikowane prawdopodobieństwo punktu p ,
- $P_{new}(p)$ – prawdopodobieństwo wyznaczone na podstawie aktualnego pomiaru,
- $P_{old}(p)$ – znane prawdopodobieństwo występowania obiektu w punkcie p wyznaczone poprzez zintegrowanie wcześniejszych pomiarów.

Poniżej znajduje się opis dwóch algorytmów integracji danych ze skanera laserowego. Pierwszy z nich, algorytm zorientowany na punkty, w którym prawdopodobieństwa punktów w siatce obliczane są w sposób nieuproszczony. Nie są w nim stosowane żadne uproszczenia modelu matematycznego. Algorytm ten stanowi punkt odniesienia. Można go zastosować w celu integracji danych pochodzących z dowolnego wyżej wymienionego czujnika. Również z czujników wykonujących wiele pomiarów jednocześnie. Drugi algorytm, zorientowany na próbki, jest algorytmem uproszczonym. Uproszczony jest model matematyczny, a pozycja czujnika i próbek obliczana jest z dokładnością do odległości punktów siatki zajętości. W tej formie ma on zastosowanie jedynie w przypadku czujników charakteryzujących się poprzecznym błędem odległościowym, którego wartość nie przekracza połowy odległości między punktami siatki. Przykładem takiego czujnika jest skaner laserowy.

3.1. Algorytm zorientowany na punkty

W trakcie działania algorytmu zorientowanego na punkty badane są wszystkie punkty w obszarze pomiaru. Dla danego typu i wartości błędu poprzecznego, dla każdego punktu obliczany jest zakres kątów próbek pomiarowych, których obszary działania zawierają w sobie punkt. Następnie sprawdzana jest każda próbka z tego zakresu. W zależności od rodzajów błędów badane jest wzajemne położenie pozycji próbki oraz punktu, odległościowe lub kątowe (rys. 5) i na tej podstawie obliczane prawdopodobieństwa ze wzoru (1).

Wybierana jest próbka, dla której prawdopodobieństwo występowania obiektu w punkcie siatki jest największe i większe od 0,5. Jeśli takiej próbki nie ma, wybierana jest ta, dla której prawdopodobieństwo jest najmniejsze. Następnie dane mogą być zintegrowane w siatce zajętości według wzoru (2).

3.2. Uproszczony algorytm zorientowany na próbki

W trakcie działania algorytmu zorientowanego na próbki badane są wszystkie próbki. Dla każdej z nich wyznaczane są punkty siatki, które są najbliższe pozycji czujnika oraz pozycji próbki. Następnie algorytm przechodzi po wszystkich punktach pośrednich. W tym celu wykorzystywany jest algorytm Bresenhama [4, 5]. Dla punktów najbliższych pozycji próbki, prawdopodobieństwo ustawiane jest na maksymalne, dla pozostałych punktów prawdopodobieństwo ustawiane jest na minimalne.

3.3. Wyniki

Oba algorytmy były testowane na komputerze klasy PC z procesorem 2 GHz. Pomiaru zostały wykonane w pomieszczeniach Katedry Informatyki Stosowanej (KIS) Politechniki Łódzkiej. Do tego celu wykorzystano dwuwymiarowy skaner laserowy firmy SICK.

Siatka zajętości ma wielkość 2000 na 2000 punktów i zawiera informację o przestrzeni 40 na 40 metrów. Daje to rozdzielczość 50 punktów na metr lub 2 cm na punkt.

Generalnie oba algorytmy dały zbliżone wyniki w postaci siatki zajętości przechowującej dane o otaczających obiektach i niezajętej przestrzeni. Rysunek 8 przedstawia mapę pomieszczeń KIS, w których zbierane były dane pomiarowe. Kolorem białym oznaczona jest pusta przestrzeń, kolorem czarnym wykryte objekty, na przykład ściany. Kolorem szarym oznaczona jest przestrzeń nieodkryta. Oznaczona jest również ścieżka, po której poruszała się robot.

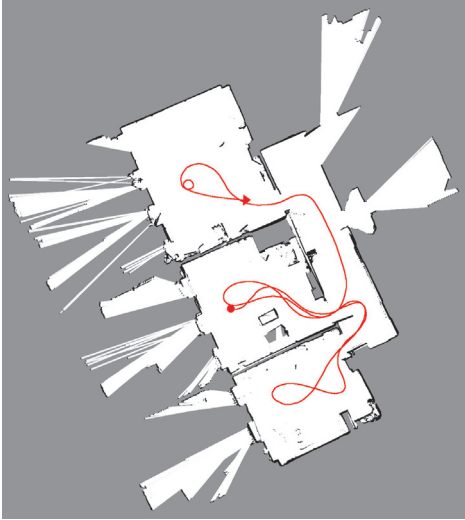
Po lewej stronie obu części rysunku 8 widoczne są okna. Wydaje się iż promienie lasera przez nie przechodzą. W rzeczywistości są to promienie odbite. Dostrzec można odbicie lustrzane części przeciwległych ścian.

Na zbliżeniu mapy (rys. 9 i 10) widoczne są różnice w dokładności mapy. Algorytm zorientowany na próbki generuje większy szum, a dla dużych odległości od czujnika widać poszczególnie promienie. Jest to spowodowane przyjęciem uproszczeń.

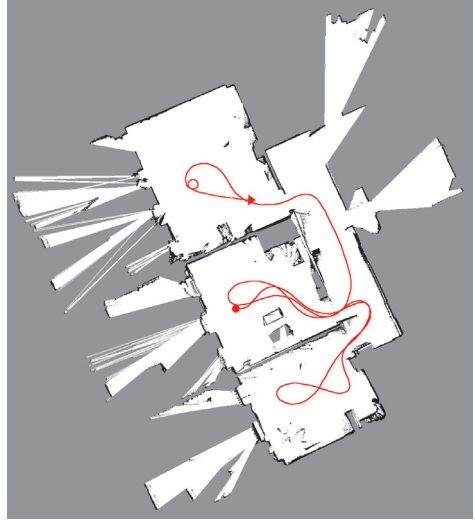
Czasy integracji danych wykonanej za pomocą obu testowanych algorytmów przedstawia rysunek 11. Najmniejsze modyfikowane obszary zawierały w przybliżeniu 100 000 punktów. Odpowiada to średniej długości promienia wynoszącej 1,39 m. W przypadku algorytmu zorientowanego na punkty, modyfikacja takich obszarów trwała średnio 87 ms, w przypadku algorytmu uproszczonego 8 ms. Największe modyfikowane obszary zawierały

w przybliżeniu 562 856 punktów. Odpowiada to średniej długości promienia 3,02 m. W przypadku algorytmu zorientowanego na punkty, modyfikacja takich obszarów trwała średnio 322 ms, w przypadku algorytmu uproszczonego 22 ms.

a)

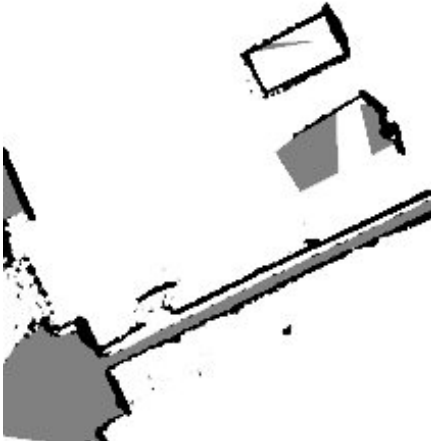


b)



Rys. 8. Wyniki działania algorytmów integracji: a) wynik działania algorytmu zorientowanego na punkty; b) wynik działania algorytmu zorientowanego na próbki

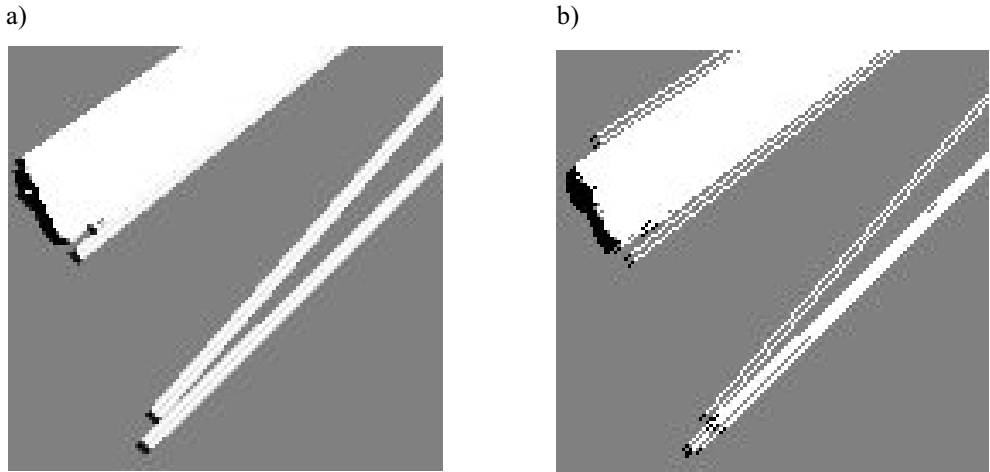
a)



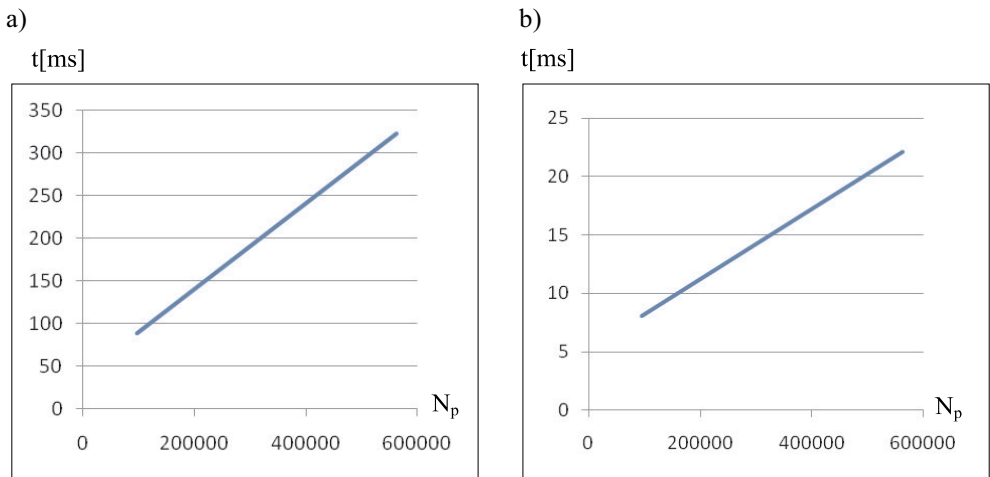
b)



Rys. 9. Wyniki działania algorytmów integracji: a) wynik działania algorytmu zorientowanego na punkty; b) wynik działania algorytmu zorientowanego na próbki



Rys. 10. Wyniki działania algorytmów integracji: a) wynik działania algorytmu zorientowanego na punkty; b) wynik działania algorytmu zorientowanego na próbki



Rys. 11. Wykresy przedstawiające średni czas integracji próbek w zależności od liczby modyfikowanych punktów siatki: a) algorytm zorientowany na punkty; b) uproszczony algorytm zorientowany na promienie

4. Wnioski

Przez zastosowanie uogólnionego modelu czujnika odległości możliwe jest zbudowanie mapy otoczenia robota. Wyniki działania obu algorytmów są zbliżone, jednak algorytm uproszczony jest dziesięciokrotnie szybszy, ale mniej elastyczny.

Nie bez znaczenia jest fakt, że budowana mapa jest dokładna. W praktyce wystarczająca byłaby dokładność rzędu 20 cm. Zmniejszenie dokładności spowodowałoby wzrost prędkości integracji.

Należy nadmienić, iż zastosowano algorytm budowania mapy ze znanymi pozami. W przypadku zastosowania dodatkowego algorytmu lokalizacji czas integracji pomiaru się zwiększy. W takim przypadku mogłoby się okazać, iż integracja danych przez algorytm nieuproszczony trwa zbyt długo.

Literatura

- [1] Siliciano B., Khatib O., Groen F. (ed.), *Robotic Mapping and Exploration*. Springer-Verlag, Berlin, 2009.
- [2] Siliciano B., Khatib O., Groen F. (ed.), *3D Robotic Mapping. The Simultaneous Localization and Mapping Problem with Six Degrees Of Freedom*. Springer-Verlag, Berlin, 2009.
- [3] Siliciano B., Khatib O., Groen F. (ed.), *Spatial Representaion and Reasoning for Robot Mapping. A Shape-Based Approach*. Springer-Verlag, Berlin, 2008.
- [4] Abrash M., *Michael Abrash's Graphics Programming Black Book Special Edition*. The Coriolis Group Inc., Scottsdale, Arizona, 1997, 654–678.
- [5] Glassner A.S. (ed), *Graphics Gems*. Academic Press Inc., USA, 1990, 98–106.