

dr Barbara Siemiątkowska
IPPT PAN, Wydział Mechatroniki PW
dr Jacek Szklarski
Instytut Podstawowych Problemów Techniki
mgr inż. Jan Syrczyński
Wydział Mechatroniki PW
mgr inż. Piotr Węclewski
Wydział Mechatroniki PW
dr Michał Gnatowski
Instytut Podstawowych Problemów Techniki

WYKORZYSTANIE FILTRÓW CZĄSTECZKOWYCH W PROCESIE LOKALIZACJI ROBOTA MOBILNEGO

W pracy omówiono metody określenia zmian położenia robota mobilnego w środowisku 3D. Przedstawiono dwie implementacje filtra cząsteczkowego, których celem jest przyspieszenie obliczeń. Omówiono metodę, w której zmiana położenia i orientacji odbywa się oddzielnie oraz metodę wykorzystującą procesory graficzne w algorytmie lokalizacji

THE APPLICATION OF PARTICLE FILTERS IN MOBILE ROBOT LOCALIZATION

In the article the overview of localization methods is presented. Two modifications of particle filter algorithm are described. In the first approach the position and orientation of the mobile robot are determined separately. In second method parallel processing units are used. Both methods allow us to speedup the process of localization.

1. WSTĘP

Robotykę można postrzegać jako naukę traktującą o percepcji i modyfikacji świata rzeczywistego przez urządzenia sterowane komputerem. Z zagadnieniem tym związana jest niepewność informacji o aktualnym stanie świata. Niepewność ta jest spowodowana nieprzewidywalną zmiennością środowiska oraz niedokładnością sensorów robota. Obraz z kamery, dalmierza laserowego, czy przetwornika zliczającego kąt obrotu kół – są to wszystkie informacje obciążone jakimś błędem. W robotyce mobilnej jednym z kluczowych pytań na które robot musi sobie „odpowiedzieć” brzmi: „gdzie jestem?” [1, 2, 3, 4, 5].

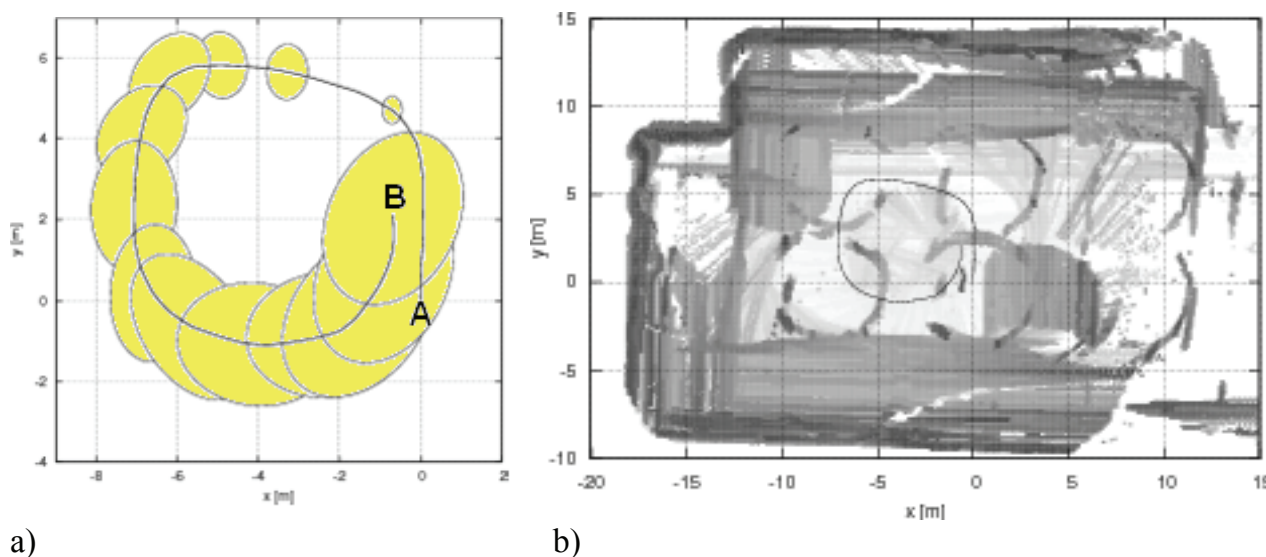
Metody lokalizacji dzielimy w zależności od konkretnego zastosowania. Możemy wyróżnić:

- Lokalizację lokalną – robot zna swoją pozycję początkową z dobrą dokładnością i przemieszcza się lokalnie, tzn. o stosunkowo mały odcinek drogi, a informacje o przemieszczeniu pobiera np. z odometrii.
- Lokalizację globalną – początkowa pozycja jest nieznana i robot na podstawie pomiarów sam musi określić swoje położenie. Gdy już to zrobi, np. poprzez zaobserwowanie jakiegoś znacznika, może tą informację wykorzystywać przy kolejnych ocenach pozycji.
- Problem porwanego robota – podobnie jak lokalizacja globalna, ale tu robot musi liczyć się z tym, że w dowolnej chwili zostanie przeniesiony w inne miejsce. Jest to znacznie trudniejszy problem od zwykłej lokalizacji globalnej.

Metody lokalizacji zależą także od rodzaju środowiska. Możemy wyróżnić:

- Środowisko typu wnętrze pomieszczenia, w którym robot przemieszcza się po płaszczyźnie, a układ ścian określa główne kierunki rozmieszczenia przeszkód.
- Środowisko zewnętrzne, zwykle o dużych nierównościach terenu i jednocześnie z brakiem wyróżnionych kierunków głównych.
- Środowisko statyczne – w którym jedynym poruszającym się obiektem jest robot mobilny, a reszta otoczenia nie zmienia się.
- Środowisko dynamiczne – otoczenie robota zmienia się (na dodatek niezależnie od akcji robota). Oczywiście wszystkie typowe środowiska, gdzie obecni są ludzie (jak np. wnętrze budynku czy autostrada) są dynamiczne.

Z punktu widzenia lokalizacji, pomiary dokonywane przez robota mobilnego możemy podzielić na dwie kategorie: względne i bezwzględne. Do pierwszych zaliczyć możemy enkodery, a do drugiej grupy GPS.



Rys. 1. Mapa rastrowa otoczenia robota, do której wykorzystano pomiary z dalmierzy laserowych oraz pozycje, wyznaczona bez korekcji za pomocą obserwacji

Określanie pozycji metodą pozycji zliczonej to najstarszy sposób lokalizacji. Polega on na wyznaczeniu bieżącej pozycji na podstawie ostatnio obserwowanej pozycji oraz przesunięcia wyznaczonego za pomocą znanej prędkości, kierunku ruchu i minionego czasu. Lokalizacja zliczeniowa (na podstawie odometrii) jest możliwa jedynie w sytuacji, w której robot przemieszcza się na stosunkowo krótkim odcinku.

Na rys. 1a przedstawiono przykładową trasę przebytą przez robota. Punkt A wskazuje położenie początkowe, punkt B położenie końcowe. Kolorem żółtym zaznaczono elipsę niepewności położenia robota. Niepewność wzrasta wraz z długością trasy przebytej przez robota. Na rys. 1b przedstawiono mapę rastrową otoczenia utworzoną w sytuacji, gdy informacja o otoczeniu jest określana jedynie na podstawie odometrii. Wyraźnie widać, że mapa jest nieprzydatna.

W 1960 powstała metoda filtru Kalmana [4], która przez dziesięciolecia dominowała w dziedzinie filtrów adaptacyjnych. W robotyce mobilnej najczęściej jest stosowany do określenia położenia robota (wektora stanu) na podstawie obserwacji. W algorytmie tym zakłada się, że rozkład szumu pomiarowego ma charakter gaussowski. W filtrze Kalmana estymacja stanu procesu jest liniową kombinacją predykcji stanu i ważonej różnicy między wektorem obserwacji, a predykcją obserwacji. Metoda posiada następujące cechy: założono model liniowy, złożoność obliczeniowa metody wzrasta proporcjonalnie do kwadratu liczby obserwacji, konieczne jest jednoznaczne określenie początkowego położenia robota, przyjęto gaussowski rozkład szumu, co nie zawsze dobrze opisuje rzeczywistość.

W czasie II wojny światowej J. Neumann opracował metodę Monte-Carlo, która jest zwana także metodą quasi-bayesowską. Metody Monte-Carlo zostały niezależnie opracowane w wielu dziedzinach nauki np. fizyce, statystyce, technice. Od lat 90-tych XX wieku zaczęły zastępować filtr Kalmana w lokalizacji robotów mobilnych [6].

2. FILTR CZĄSTECZKOWY

W metodzie tej położenie robota w chwili t jest opisywane przez zbiór cząsteczek S^t :

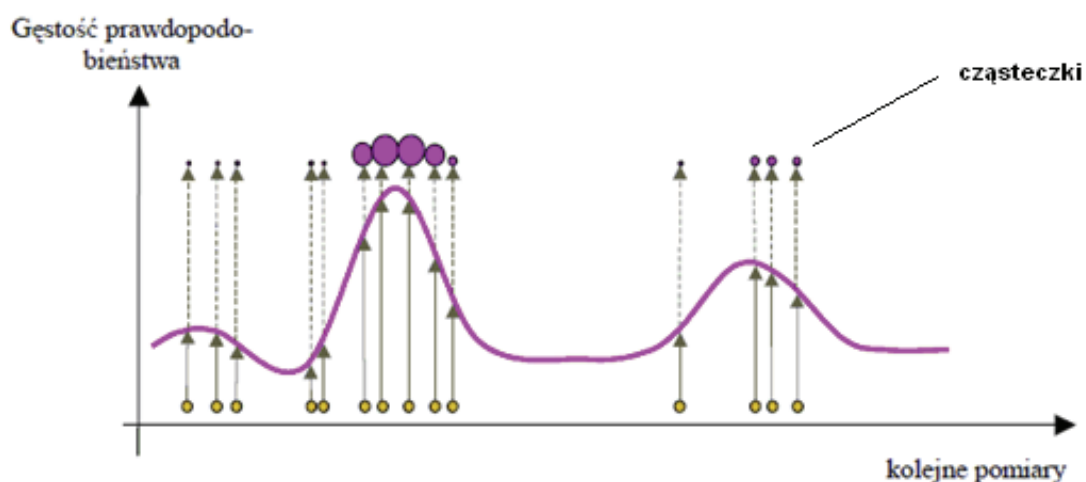
$$S^t = \{ x^i_t, w^i_t \} \quad i = 1, \dots, N, \quad (1)$$

x^i_t – określa położenie robota, w^i_t – jest wagą, która została przypisana i -tej cząsteczce.

Każda cząstka reprezentuje pewne możliwe położenie robota w chwili t i wagę w^i_t określającą przekonanie, że robot może się w danym punkcie znajdować. Rozkład cząstek odzwierciedla rozkład funkcji gęstość prawdopodobieństwa, że robot znajduje się w pewnym obszarze przestrzeni. Rys. 2 przedstawia ideę opisywanej metody.

W kolejnych krokach na podstawie wskazań sensorów odometrycznych u_t , dla każdej cząstki określone są przybliżone wartości x^i_{t+1} w chwili $t+1$. Następnie dla każdej wartości x^i_{t+1} obliczany jest współczynnik dopasowania w^i_{t+1} . Współczynnik ten określa, na ile przewidywane położenie cech otoczenia różni się od położenia obserwowanego w danym punkcie, w kolejnych krokach:

- wagi są normalizowane,
- określany jest nowy rozkład cząsteczek.



Rys. 2. Rozkład cząsteczek

Do podjęcia decyzji związanych z realizacją trasy robota konieczna jest jednoznaczna informacja o położeniu pojazdu, których mechanizm filtrów cząsteczkowych nie daje. W takim wypadku stosuje się jedną z metod oszacowania aktualnej pozycji: średnia ważona, najlepsza cząsteczka, średnia ważona w bliskim otoczeniu najlepszej cząsteczki.

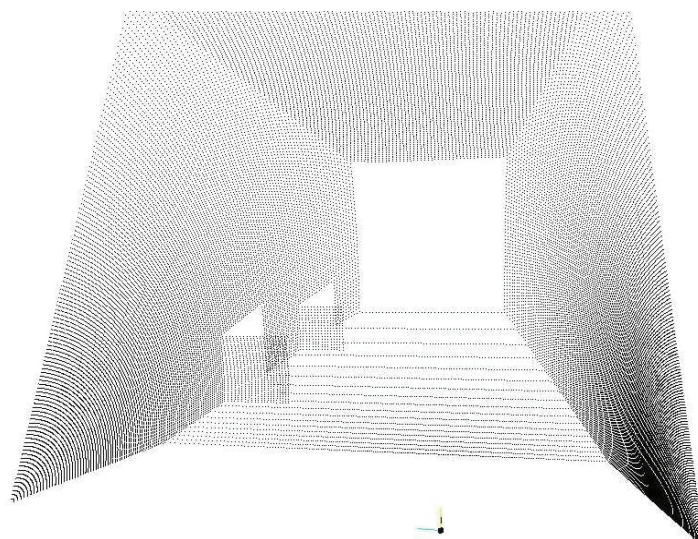
3. MODYFIKACJE FILTRU CZĄSTECZKOWEGO

Liczba wygenerowanych cząsteczek jest silnie zależna od niedokładności wskazań odometrii oraz liczby parametrów, które musimy estymować. Położenie robota przemieszczającego się w przestrzeni 3D określane jest przez 6 parametrów: położenie $[x, y, z]^T$, oraz kąty nachylenia do osi OX, OY, OZ : $[\alpha, \beta, \gamma]^T$. Przy dużej niedokładności wskazań sensorów odometrycznych potrzebujemy miliony cząsteczek, aby uzyskać zadowalającą dokładność. Przy tak dużej liczbie cząsteczek niemożliwe jest określanie przemieszczenia robota w czasie rzeczywistym.

W prowadzonych badaniach testowano dwie metody umożliwiające przyspieszenie czasu obliczeń:

- oddzielne korygowanie zmian orientacji i przesunięcia,
- wykonywanie obliczeń w sposób równoległy.

Eksperymenty przeprowadzono przy pomocy robota Elektron, który był wyposażony w dalmierz laserowy SICK [7]. Dane odebrane z dalmierza mają postać ciągu o długości równej liczbie odebranych punktów. SICK LMS 200 ma rozdzielczość skanowania $1^\circ, 0,5^\circ$ lub $0,25^\circ$ i dane odczytuje w zakresie $0-180^\circ$. Zamontowanie takiego lasera na głowicy obrotowej umożliwia otrzymanie informacji trójwymiarowej o otoczeniu, dane te zapisane są w postaci chmury punktów. Dla omawianego sposobu skanowania charakterystyczne jest zagęszczenie punktów przy osi obrotu głowicy. Na rys. 3 przedstawiono chmurę punktów otrzymaną, gdy robot został umieszczony w korytarzu wydziału Mechatroniki PW. W środowisku typu wnętrze pomieszczenia trzy wzajemnie prostopadłe kierunki główne wyznaczone są przez podłogę (sufit), układ ścian oraz elementów prostopadłych do ścian. Układ ten wyznacza wersory globalnego układu współrzędnych. Obserwując zmiany kierunków głównych możemy określić obrót robota wokół każdej z osi.

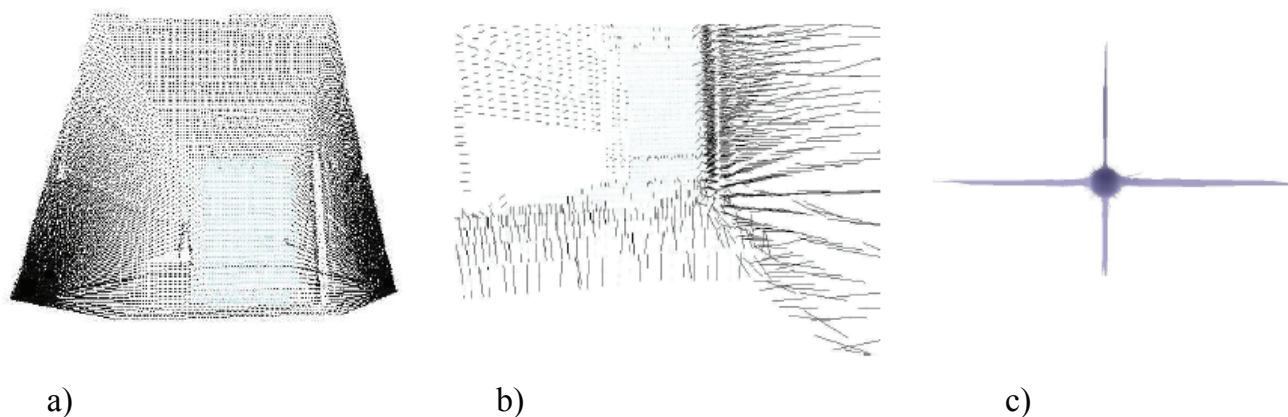


Rys. 3. Chmura punktów [8]

3.1 Określenie zmiany orientacji i przesunięcia

Lokalizacja robota składa się z dwóch procesów: na początku wyznaczana jest zmiana orientacji robota, a w kolejnym kroku przesunięcie robota wzdłuż każdej z osi. Każdy z tych procesów realizowany jest przy pomocy filtra cząsteczkowego.

Chmura punktów nie jest wygodna w procesie lokalizacji i zwykle jest ona zapisywana w innej formie. Jedną z nich jest określenie zbioru wektorów normalnych [9, 10].



Rys. 4. Zbiór wektorów normalnych: a) chmura punktów, b) wektory normalne, c) histogram kierunków normalnych [8]

Na rys. 4b przedstawiono zbiór wektorów normalnych utworzony na podstawie chmury punktów przedstawionej na rys 4a. Rys. 4c prezentuje histogram kierunków głównych. Im bardziej popularny jest dany kierunek, tym wyższy jest odpowiadający mu wierzchołek. Stosując filtr cząsteczkowy, możemy określić zmianę orientacji robota. W prowadzonych badaniach zmiany orientacji robota określa cząsteczka o największej wadze.

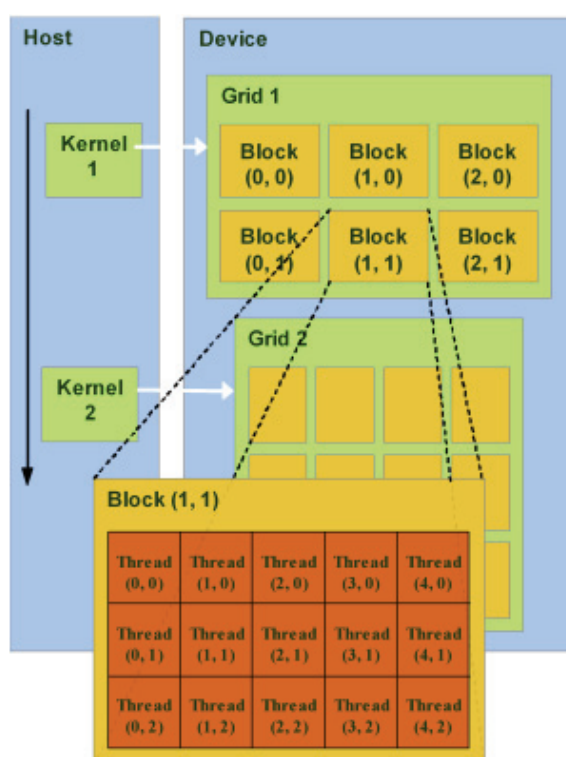
Znając orientację robota w globalnym układzie współrzędnych, możemy wyznaczyć przesunięcie robota wzdłuż każdej osi. Przyjęta metoda umożliwia znaczne zredukowanie

liczby cząsteczek. W następnym rozdziale opisane zostaną wyniki przeprowadzonych eksperymentów.

Metoda ta nie może być stosowana w środowisku nieuporządkowanym np. na zewnątrz budynków. Dlatego rozpoczęto badania dotyczące wykorzystania procesorów graficznych (technologii CUDA), które umożliwiają wykonywanie obliczeń w sposób równoległy.

3.2 Obliczenia równoległe

Procesory graficzne umożliwiają przetwarzanie informacji w sposób równoległy [11, 12]. Równoległość obliczeń na platformie CUDA otrzymywana jest poprzez wywołanie tzw. jądra jako zbioru wątków. Wątki pogrupowane są w bloki, te natomiast w tablice. Model wykonawczy technologii CUDA prezentuje rys. 5.

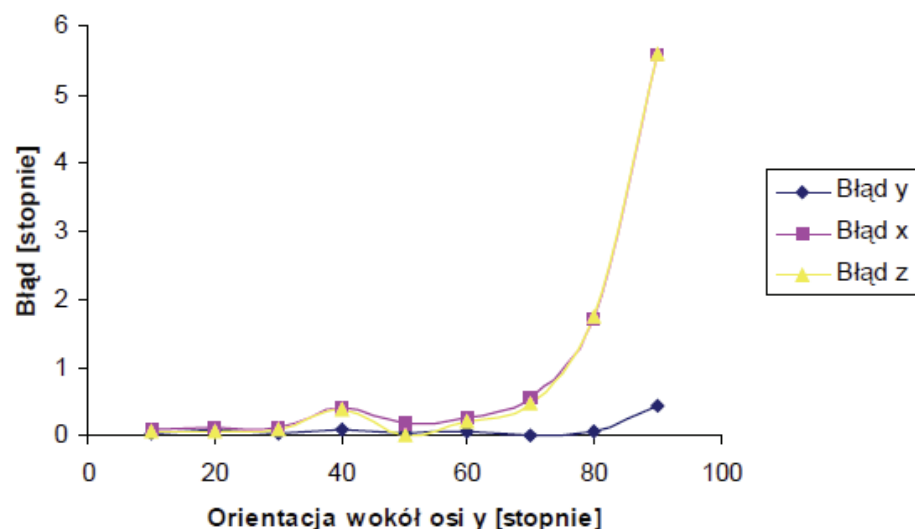


Rys. 5. Model wykonawczy CUDA (źródło: NVidia.com)

Zarówno bloki, jak i tablice mogą być jedno-, dwu- lub trójwymiarowe. Każdy wątek, blok i tablica posiada unikalny identyfikator, który można wykorzystać do wyznaczania pozycji przetwarzanego elementu w pamięci. Architektura CUDA zakłada całkowitą niezależność przetwarzania w obrębie bloku, co utrudnia zastosowanie platformy do obliczeń o charakterze ogólnym. W filtrze cząsteczkowym, każda z cząsteczek jest niezależna od pozostałych więc obliczenia współczynników dopasowania mogą odbywać się więc niezależnie. Wyniki wstępnych eksperymentów zostaną opisane w rozdz. 5.

4. Eksperymenty

Opisane w tym rozdziale wyniki eksperymentów zostały zaczerpnięte z [8]. Testowano wpływ szeregu parametrów na dokładność lokalizacji.



Rys. 6. Wpływ zmiany orientacji robota na dokładność otrzymanych wyników

Na rys. 6 przedstawiono wpływ rzeczywistego obrotu robota (względem osi OZ) na dokładność otrzymanych wyników. Obliczenia prowadzono dla 20 000 cząsteczek. Przy obrocie powyżej 60° błąd gwałtownie rośnie. Cecha ta wynika ze sposobu skanowania (rys. 3).

Kolejnym istotnym parametrem jest wpływ pola obserwowanych powierzchni głównych na dokładność określenia przemieszczenia robota. Analogicznie jak poprzednio obliczenia przeprowadzono dla 20 000 cząsteczek.

Tab. 1. Wpływ powierzchni płaszczyzn dominujących na dokładność lokalizacji

wysokość[m]	szerokość[m]	powierzchnia płaszczyzny[m ²]	wynik[m]	błąd[m]
0.005	0.5	0.0025	-2.7033	0.2967
0.01	0.5	0.005	-2.70709	0.29291
0.02	0.5	0.01	-2.71262	0.28738
0.03	0.5	0.015	-2.70217	0.29783
0.04	0.5	0.02	-2.99399	0.00601
0.05	0.5	0.025	-2.9988	0.0012
0.5	0.5	0.25	-3.00248	0.00248

Określenie przemieszczenia robota było wykonywane co 3 m. Zgodnie z intuicją, w sytuacji, gdy obserwowany jest niewielki fragment powierzchni błąd określania przemieszczenia jest duży (powyżej 10 %). Wyniki przedstawiono w tab. 1.

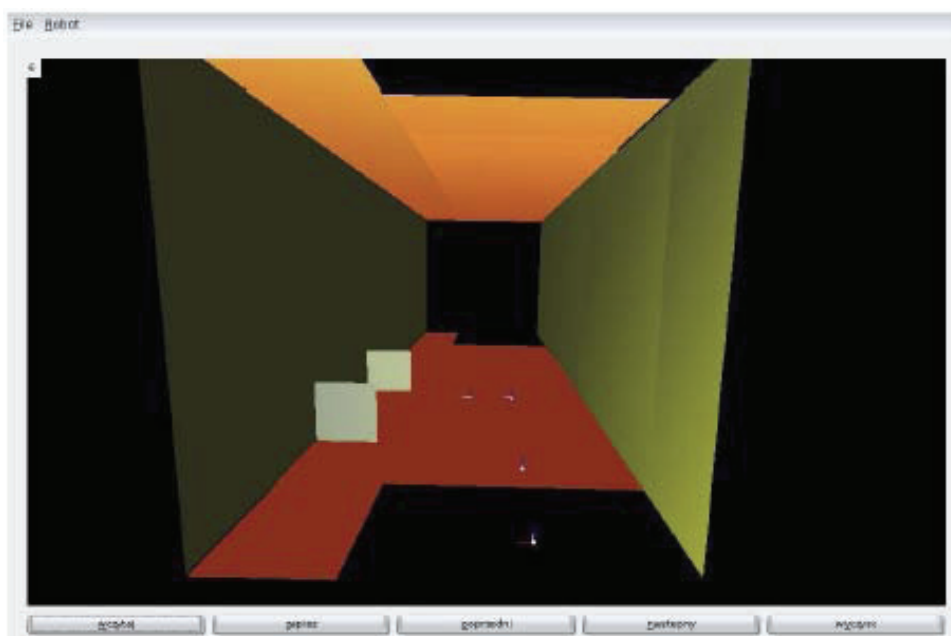
Tab. 2 zawiera wyniki badania wpływu przesunięcia robota na dokładność lokalizacji. Błąd rośnie, jeśli robot przejechał powyżej 4 m. Wynika to z tego, że z pola widzenia robota znikają płaszczyzny, które można dopasować.

Kolejnym parametrem jest określenie wpływu liczby cząsteczek na dokładność lokalizacji. Stwierdzono, że powyżej 100 000 cząsteczek i założonym 10 % błędzie odometrii dokładność lokalizacji nie wzrasta.

Tab. 2. Wpływ przesunięcia robota na dokładność lokalizacji

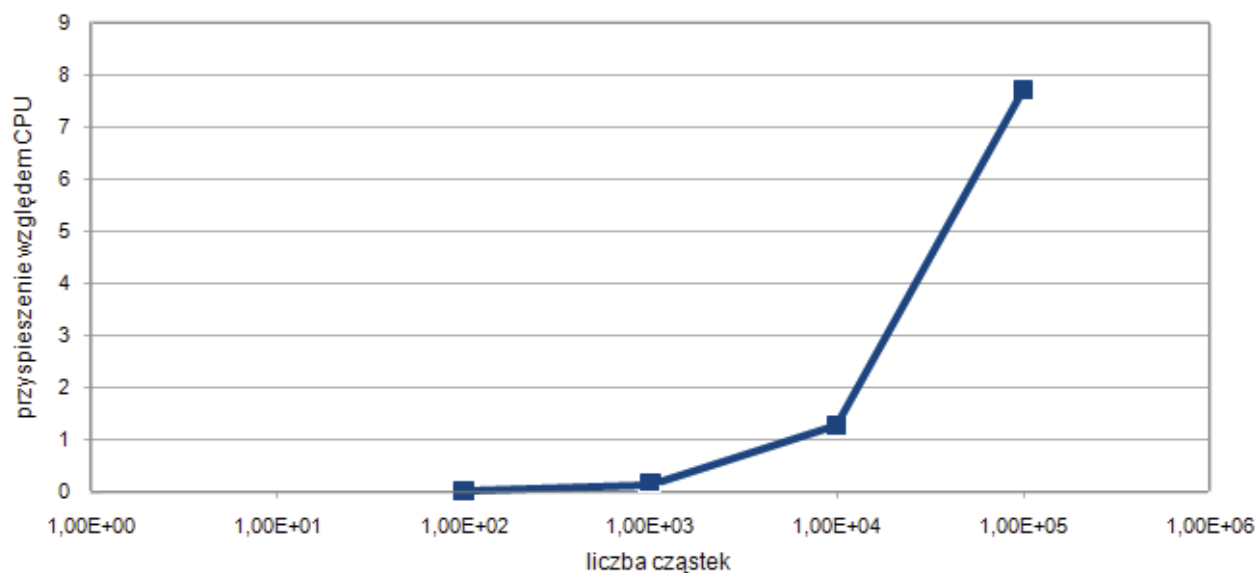
Lp.	Przeszczenie odniesienia [m]	Wynik x [m]	Wynik y [m]	Wynik z [m]	Błąd [m]
1.	0,5	-0,00194997	-0,0012705	0,49656	0,00343255
2.	1	0,00269155	-0,00556186	-1,00729	0,00730895
3.	1,5	-0,00188229	0,00207856	-1,50169	0,00169262
4.	2	-0,00129088	0,00077636	-1,9931	0,00689943
5.	2,5	0,00611422	-0,00014437	-2,4991	0,00089252
6.	3	-0,00045657	0,00165519	-2,9996	0,00039951
7.	3,5	-0,0018478	0,0011918	-3,49071	0,00928931
8.	4	0,0063206	0,00212154	-3,72036	0,27963403

Na rys. 7 przedstawiono mapę korytarza utworzoną na podstawie danych pochodzących ze skanera laserowego. Kropkami zaznaczono miejsca zbierania pomiarów. Obliczenia przeprowadzono dla 20 000 cząsteczek określających zmianę orientacji robota i 20 000 cząsteczek określających przesunięcie.



Rys. 7. Mapa korytarza

Oprócz obliczeń sekwencyjnych sprawdzono możliwość stosowania technologii CUDA w procesie lokalizacji. Na rys. 8 przedstawiono zależność między czasem obliczeń na komputerze z procesorem Intel Centrino (2 x 2,0 GHz) (algorytm sekwencyjny) i kartą graficzną NVidia GeForce 9300MG (implementacja równoległa). Na rys. 8 przedstawiono stosunek czasu obliczeń CPU do GPU dla różnej liczby cząsteczek. Stosowanie GPU przyspiesza proces lokalizacji w sytuacji, gdy liczba cząsteczek przekracza 10 000.



Rys. 8. Stosunek czasu obliczeń CPU do GPU dla różnej liczby cząsteczek

5. WNIOSKI

W powyższym artykule przedstawiono dwie implementacje filtra cząsteczkowego. W pierwszej metodzie proces obliczeń odbywa się sekwencyjnie. Orientacja i przemieszczenie robota określane są oddzielnie, dzięki czemu możliwe jest zmniejszenie liczby cząsteczek. W drugiej metodzie stosowana jest technologia CUDA. Wstępne wyniki eksperymentów wykazały, że stosowanie GPU ma sens wtedy, gdy liczba cząsteczek przekracza 10 000. Nasze dalsze prace będą dotyczyły zastosowania GPU w nawigacji robotów mobilnych.

6. BIBLIOGRAFIA

- [1] J. Borenstein, H. Everett, L. Feng, Where am I? Sensors and Methods for Mobile Robot Positioning. Univeristy of Michigan, 1996.
- [1] R. E. Kalman, A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering 82 (Series D), 1960, str. 35–45.
- [2] P. Skrzypczyński, Metody Analizy i Redukcji Niepewności Percepcji w Systemie Nawigacji Robota Mobilnego, Vol. 407 Rozprawy. Wydawnictwo Politechniki Poznańskiej, 2007.

-
- [3] R.C. Smith, P. Cheeseman (1987), On the representation and estimation of spatial uncertainty. *Int. J. Rob. Res.* 5(4), 56–68, 1987.
 - [4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.
 - [5] Zhang, Z., Iterative point matching for registration of free-form curves. *Rapports de Recherche 1658*, Institut National de Recherche en Informatique et en Automatique, 1992.
 - [6] I. Rekleitis, A Particle Filter Tutorial for Mobile Robot Localization, Technical Report TR-CIM-04-02, Center for Intelligent Machines, McGill University.
 - [7] Chojecki. R, Olszewski M., A Mobile Robot for Laboratory Purposes and Its Applications. *PAK*, 3 (2009), 55, 190–193.
 - [8] J. Syrczyński, Lokalizacja robota mobilnego w nieznanym otoczeniu metodą filtrów cząsteczkowych, Praca magisterska PW, 2009.
 - [9] B. Siemiątkowska, J. Szklarski, M. Gnatowski, A. Zychewicz, Budowa hybrydowej semantyczno-rastrowej reprezentacji otoczenia robota mobilnego na podstawie wskazań dalmierza laserowego 3D, *PAK*, 3:278–282, 2010.
 - [10] A. Borkowski, B. Siemiątkowska, J. Szklarski, Towards Semantic Navigation In Mobile Robotics. In: G. Engles, C. Lewerenz, W. Schafer, A. Schurr, B. Westfechtel (Eds.): *Graph Transformations and Model Driven Engineering – Essays Dedicated to Manfred Nafle*, LNCS 5765, Springer.
 - [11] S. H. Seyed, *Parallel processing and parallel algorithms: theory and computation*. Springer, 2000.
 - [12] Harris, Mark, *Optimizing Parallel Reduction in CUDA*, NVIDIA Developer Technology, <http://developer.download.nvidia.com>.