

mgr Arkadiusz Zychewicz
 dr Barbara Siemiątkowska
 Instytut Podstawowych Problemów Techniki PAN

ZASTOSOWANIE ALGORYTMÓW ICP I SIFT W LOKALIZACJI ROBOTÓW MOBILNYCH

W pracy zaprezentowano zastosowanie algorytmów ICP (Iterative Closest Point) i algorytmu SIFT w lokalizacji robota mobilnego. W prowadzonych pracach przyjęto założenie, że robot przemieszcza się w przestrzeni 3D i jest wyposażony w skaner laserowy 3D. Klasyczny algorytm ICP jest czasochłonny, ale połączenie go z algorytmem SIFT znacząco zmniejsza czas obliczeń i jednocześnie zapewnia porównywalną dokładność wyniku. Ponieważ metoda SIFT wymaga zapisywania danych w postaci obrazu zaproponowano kilka metod przekształcenia skanu 3D do obrazu i przetestowano wpływ danej metody na poprawność oraz czas obliczeń.

THE APPLICATION OF ICP AND SIFT ALGORITHMS FOR MOBILE ROBOT LOCALIZATION

In this paper the applications of ICP (Iterative Closest Point) algorithm and SIFT algorithm for mobile robot localization are presented. It is assumed that the robot is equipped with 3D laser range scanner and it acts in 3D space. Classical ICP method is time consuming but using SIFT methods for features detection makes the process of localization more effective. Before the process of localization starts the 3D data are transformed into 2D image. In the paper a few method of data transformation are presented. The influence of data transformation in process classification is presented in the paper.

1. WSTĘP

Zagadnienie lokalizacji jest jednym z podstawowych zagadnień robotyki mobilnej. Na wybór odpowiednich metod ma wpływ rodzaj sensorów, w które wyposażony jest robot, oraz sposób reprezentacji otoczenia. Podstawowym źródłem informacji o przemieszczeniu się robotów mobilnych między punktami pomiarowymi jest odometria [1]. Istotną wadą tej metody jest to, że niedokładności pomiarów kumulują się w czasie. Błędy mogą mieć charakter systematyczny - wynikający z cech urządzenia (nierówności kół, ograniczonej rozdzielczości enkoderów) lub niesystematyczny. Mają one istotne znaczenie w sytuacji, gdy robot przemieszcza się na zewnątrz budynku, po powierzchni, która jest nierówna.

W latach siedemdziesiątych XX wieku, aby umożliwić większą elastyczność układów transportowych, w otoczeniu, po którym przemieszcza się robot zaczęto umieszczać sztuczne znaczniki aktywne (np. ultradźwiękowe) lub pasywne (np. obiekty o określonym kolorze i kształcie [2]). Metody te mogą być użyteczne tylko wtedy, gdy układ sztucznych znaczników w pomieszczeniu nie ulegnie zmianie i przynajmniej trzy punkty charakterystyczne są widoczne w każdym podobszarze, po którym porusza się robot. Kolejnym krokiem w rozwoju systemów lokalizacji jest wykrywanie charakterystycznych naturalnych elementów otoczenia. Rozwijane są metody, w których robot sam buduje reprezentację środowiska i wykrywa obiekty charakterystyczne. Algorytmy te dzielimy na dwie grupy:

- dopasowywanie skanów
- dopasowywanie cech.

Jedną z najbardziej znanych jest metod dopasowania skanów w przestrzeni 2D jest tworzenie histogramów kątowych [3], a w przestrzeni 3D opisywana w tym artykule metoda ICP.

W proponowanym w tej pracy algorytmie zaimplementowana jest metoda dopasowania cech, która składa się z trzech podstawowych etapów:

- wykrywanie na podstawie sensorów charakterystycznych elementów otoczenia
- dopasowanie wykrytych znaczników do elementów zapamiętanych w mapie
- określenie dokładności dopasowania i położenia robota.

Podstawowe problemy, jakie występują podczas jednoczesnego tworzenia mapy otoczenia i lokalizacji, to wybór reprezentacji sceny i cech oraz określenie metody dopasowywania. Istotny jest także wybór algorytmu, który w wyniku porównania położenia znaczników określi przemieszczenie robota.

2. ZBIERANIE DANYCH

W prowadzonych badaniach eksperymenty były przeprowadzane przy użyciu mobilnego robota „Elektron”, który został wykonany w Instytutach Automatyki i Robotyki oraz Elektroniki i Technik Informacyjnych Politechniki Warszawskiej [4]. Podstawowym sensorem umożliwiającym zbieranie informacji o otoczeniu jest dalmierz laserowy Sick LMS200, zamontowany na obrotnicy.



Rys. 1. Zdjęcia robota „Elektron” z laserem 3D oraz kamerą dookólną na górze

Skaner laserowy mierzy odległość do przeszkód dla kątów φ od -90° do $+90^\circ$, z rozdzielczością $\Delta\varphi=1,0^\circ$ lub $\Delta\varphi=0,5^\circ$. Konstrukcja obrotnicy umożliwia odchylenie od płaszczyzny poziomej od -15° do 90° . Dane są transmitowane ze skanera do jednostki centralnej. Istnieje możliwość dokonywania pomiaru w trybie ciągłym (skaner i obrotnica działają niezależnie) jak i skokowym (obrotnica jest zatrzymywana w czasie skanowania). Na rys. 1 przedstawiono zdjęcie robota z umieszczonym na nim dalmierzem laserowym firmy SICK.

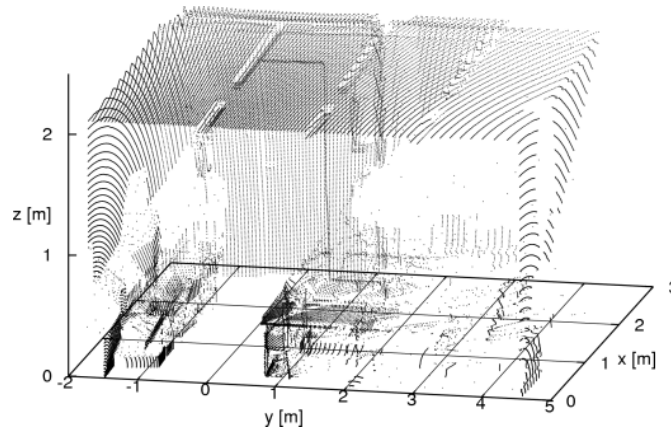
Wyniki pomiarów otrzymane są w układzie „pseudo-polarnym” $\{\varphi_i, \theta_i, r_i\}$, $i = 1, \dots, N$, gdzie φ jest poziomym kątem skanowania, θ jest kątem odchylenia obrotnicy, a r jest odległością do przeszkody. Dane mogą być przekształcane na postać kartezjańską za pomocą wzorów:

$$x = r \sin \varphi \cos \theta$$

$$y = r \cos \varphi$$

$$z = r \sin \varphi \sin \theta$$

i zapamiętane w postaci chmury punktów. Na rys. 2 przedstawiono przykładowe odczyty ze skanera, zapamiętane w postaci kartezjańskiej.



Rys. 2. Chmura punktów pojedynczej sceny. Robot znajduje się w środku układu współrzędnych, w punkcie (0, 0, 0).

3. LOKALIZACJA ROBOTA MOBILNEGO

3.1. Algorytm ICP

Ideą algorytmu ICP (*Iterative Closest Point*) [5, 6] jest znalezienie macierzy obrotu i wektora przesunięcia między dwoma zbiorami punktów pomiarowych w 3D we wspólnym układzie współrzędnych. Po dokonaniu transformacji na jednym z tych zbiorów chmury punktów zostaną dopasowane do siebie. Algorytm zakłada, że w obu zbiorach znajduje się istotna część wspólna skanowanej przestrzeni.

Każdy krok algorytmu składa się z kilku etapów. W pierwszej kolejności wyszukiwane są najbliższe sobie pary punktów z obu zbiorów. Przy czym uwzględnia się, że nie wszystkie punkty mają najbliższych sąsiadów w drugim zbiorze. Uwzględnia się to przypisując każdej parze odpowiednią wagę. Następnie wyliczana jest miara aktualnej rozbieżności między zbiorami:

$$f(\bar{R}, \bar{t}) = \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} w_{i,j} \|a_i - (\bar{R}b_j + \bar{t})\|^2,$$

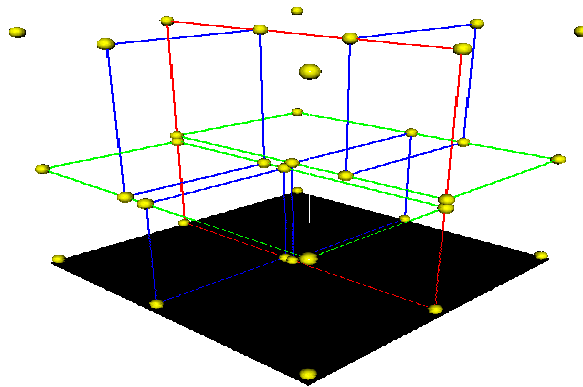
gdzie N_A i N_B to odpowiednio liczba punktów w zbiorze A i B , a $w_{i,j}$ jest wagą dopasowania, taką że $w_{i,j} = 1$ jeżeli punkty a_i i b_j są sobie najbliższe, natomiast w przeciwnym przypadku $w_{i,j} = 0$.

W kolejnym etapie obliczana jest transformacja, której celem jest zminimalizowanie miary funkcji $f(\bar{R}, \bar{t})$. Taką predykcję można wykonać używając jednej ze znanych metod:

- rozkład na wartości osobliwe
- kwaternionową metodę Horna
- metodę ortonormalnych macierzy Horna
- metodę podwójnych kwaternionów Walkera.

3.1.1. Algorytm ICP w robotyce

Niestety ICP w swojej oryginalnej formie nie może być bezpośrednio stosowany w robotyce, gdyż czas obliczeń rośnie w postępie kwadratowym w zależności od ilości punktów pomiarowych. Jedną z metod rozwiązania tego problemu jest odpowiednie ograniczenie liczby punktów. W artykule [6] znajduje się przegląd znanych metod osiągnięcia tego celu. Wszystkie zakładają istnienie siatki punktów na wejściu tj. zbiory punktów powinny znajdować się w prostokątnych macierzach nie zawierających pustych elementów. Jednak w wielu zastosowaniach uzyskanie siatki nie jest możliwe m.in. w robotyce. Zazwyczaj pomiary zawierają szum gaussowski i tzw. szum typu sól-pieprz, a część pomiarów jest odrzucana, co znacząco utrudnia zbudowanie siatki. Poza tym w robotyce otoczenie jest często skanowane tylko częściowo. Dlatego też zwykłe siatkowe metody nie radzą sobie z surowymi danymi.



Rys. 3. Przykład podziału przestrzeni przez drzewo-*kd* dla $k=3$
(źródło: http://pl.wikipedia.org/wiki/Drzewa_kd)

Do poszukiwania najbliższych punktów można wykorzystać drzewo-*kd* – algorytm podziału przestrzeni k wymiarowej na podprzestrzenie w taki sposób, by wyszukiwanie najbliższych sąsiadów było jak najszybsze.

Drzewo-*kd* jest drzewem binarnym, gdzie w każdym węźle znajduje się k -wymiarowy punkt. Każdy węzeł, który nie jest liściem tworzy hiperpłaszczyznę podziału, która dzieli przestrzeń na dwie podprzestrzenie. Punkty po lewej stronie hiperpłaszczyzny reprezentują lewe poddrzewo zaczynające się w tym węźle a prawe punkty, prawe poddrzewo. Kierunek hiperpłaszczyzny jest wybierany zgodnie z wektorem normalnym płaszczyzny. Najczęstszym wyborem miejsca podziału jest mediana zbioru. Takie podejście powoduje utworzenie dobrze zbalansowanego drzewa i zwiększa wydajność wyszukiwania najbliższych sąsiadów. Rys. 3 przedstawia *kd*-drzewo. Pierwszy podział (czerwony) dzieli główną komórkę (biała) na dwie podkomórki. Każda z nich jest następnie dzielona (zielony) na kolejne podkomórki. A te dzielone na kolejne podkomórki (niebieski). Gdy już nie można dalej dzielić, każda z ośmiu komórek jest tzw. liściem tj. elementem, który dalej nie podlega rozgałęzieniom. Zastosowanie tablicy z wcześniej znalezionymi najbliższymi sąsiadami dodatkowo przyspiesza działanie algorytmu ICP.

3.2. Algorytm SIFT

Algorytm SIFT [9, 10, 11] (skalo-niezmiennicze przekształcenie cech) oblicza na podstawie obrazu zbiór wektorów cech niewrażliwych na przesunięcie, skalowanie i obrót obrazu oraz częściowo niezmienniczych na zmiany oświetlenia jak również na lokalne geometryczne zniekształcenia. Algorytm ten składa się z kilku etapów. W pierwszej kolejności obliczane są

potencjalnie charakterystyczne punkty. Do wyznaczenia tych punktów stosuje filtrację kaskadową tj. tworzona jest piramida o coraz mniejszych rozdzielczościach. Na każdym poziomie piramidy ustawiane są kolejno obrazy uzyskane metodą odejmowania obrazów poddanych filtracji Gaussowskiej (**Difference of Gaussian**), przy użyciu coraz większych rozmyć. Na potencjalne punkty charakterystyczne typuje się te punkty, które są w swoim otoczeniu najmniejsze lub największe. Przy czym otoczenie rozumie się jako osiem sąsiadów na tym samym obrazie i po dodatkowe dziewięć z obrazu nad i pod w piramidzie.

Następnie punkty są sprawdzane pod kątem stabilności rozwiązania oraz dopasowuje się je ze względu na położenie i skalę. Umożliwia to odrzucenie źle wybranych punktów ze względu na zakłócenia.

Kolejny etap to określenie orientacji punktów charakterystycznych, co pozwala uniezależnić zbiór cech od obrotu obrazu. Orientację oblicza się na podstawie histogramu sporządzonego z punktów sąsiednich.

Ostatecznie dla tak wybranych punktów z określoną orientacją obliczany jest zbiór cech na podstawie gradientu i położenia punktów z otoczenia punktu charakterystycznego.

3.3. Zastosowanie algorytmu SIFT w ICP

Mimo zastosowania drzewa-*kd* z pamięcią (cache), które przyspiesza działanie algorytmu o ponad 50% , dla dużej ilości danych pochodzących ze skanera 3D algorytm ten potrzebuje stosunkowo dużo czasu (około 1-2 s). Dlatego też można ograniczać ilość danych wejściowych. Najprostszymi sposobami są:

- wybranie co n -tego punktu (np. z $n = 10$)
- wybranie losowego zbioru punktów.

Jednak może to prowadzić do otrzymania błędnego wyniku, gdyż tak wybrane punkty niekoniecznie mogą być charakterystyczne dla otoczenia.

Do wybrania takich punktów charakterystycznych można użyć algorytmu SIFT. W pierwszej kolejności należy przekształcić chmurę punktów w obraz. Można zastosować kilka metod np.:

- przekształcić odległości na skalę szarości
- obliczyć wektory normalne i odwzorować ich współrzędne na obraz RGB
- obliczyć wartość cosinus kąta między wektorami normalnymi i zrzutować na skalę szarości.

Następnie tak utworzony obraz jest wejściem do algorytmu SIFT, który obliczy punkty charakterystyczne i ich położenie na obrazie. Posiadając tak samo utworzony obraz z kolejnego skanu 3D oraz obliczone dla niego punkty charakterystyczne można odnaleźć te z nich, które występują na obydwu obrazach. Otrzymane w opisany sposób punkty charakterystyczne w przestrzeni kartezjańskiej będą stanowiły odpowiednie zbiory punktów do algorytmu ICP.

4. EKSPERYMENTY

Do przetestowania metody wybrano kilka lokalizacji. Pomiary zostały przeprowadzone zarówno w budynku jak i na zewnątrz. Aby sprawdzić najlepszy sposób przekształcania danych 3D w obraz dla każdej pary skanów wykonano kolejno:

- przekształcenie odległości na skalę szarości
- obliczenia wektorów normalnych i odwzorowania ich współrzędnych na obraz RGB

- obliczenia wartość kosinusa kąta między wektorami normalnymi i odwzorowania go na skalę szarości.

Wyniki działania algorytmu SIFT dla wyżej wymienionych przypadków przedstawia tabela 2. Jak można zauważyć w zależności od rodzaju przekształcenia zostały wybrane inne punkty charakterystyczne. A niektóre punkty na jednym obrazie zostały przypisane do więcej niż jednego punktu na drugim. Z powodu małej ilości wykrywanych punktów charakterystycznych, zjawisko nie jest pożądane. Dlatego też aby zminimalizować wpływ tego typu efektów oprócz samych punktów do analizy algorytmem ICP przekazywane są również punkty sąsiednie.

Tabela 1. Czasy działania algorytmów oraz ich dokładność

Zadanie	Czas obliczeń	Dokładność [ZEW]	Dokładność [WEW]
Oryginalny algorytm ICP	1940 ms	0,00 m	0,00 m
Losowy wybór (300punktów)	122 ms	0,2013 m	0,3072 m
Obraz głębi	39 ms	0,1916 m	0,1715 m
Wektory normalne	589 ms	0,2901 m	0,3551 m
Cosinusy	113 ms	0,7406 m	0,3519 m

Następnie wykonano pomiar czasu wykonania każdej z metod ograniczania ilości punktów. Wyniki przedstawiono w tabeli 1. Do sprawdzenia poprawności otrzymanego wyniku porównano go z wynikiem otrzymanym algorytmem ICP działającym na pełnym zestawie danych.

5. WNIOSKI

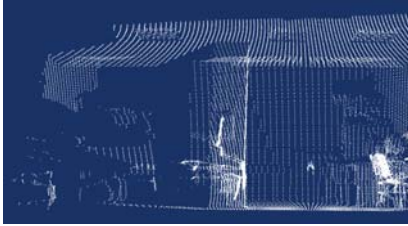
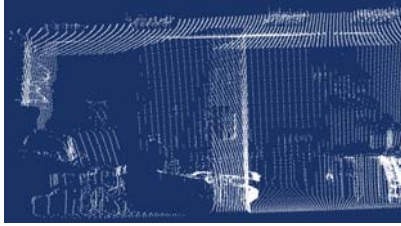
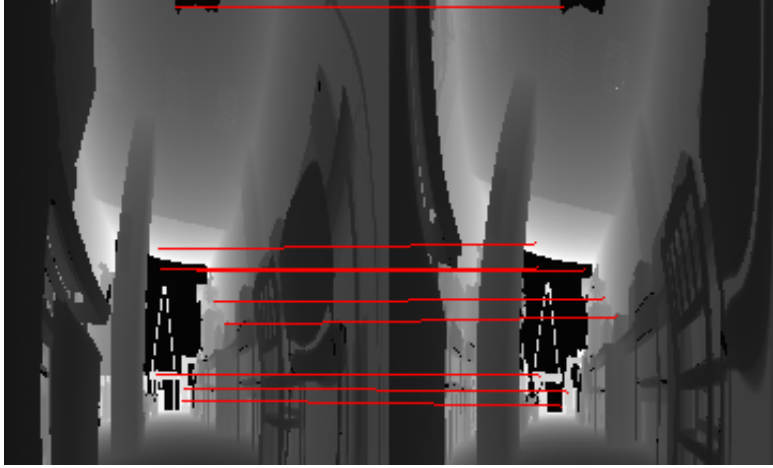
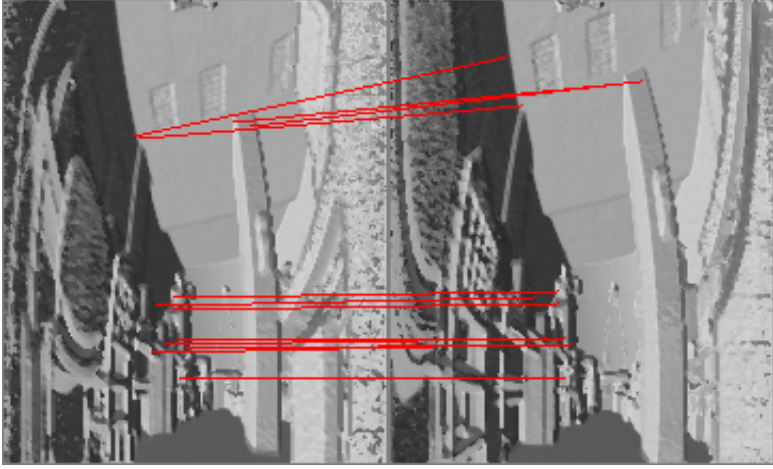
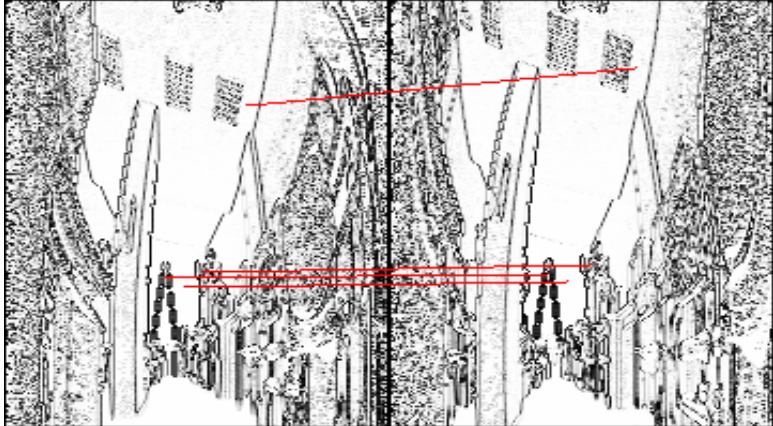
Przeprowadzone eksperymenty wykazały, że zastosowanie algorytmu SIFT do ograniczenia ilości punktów znacząco skraca czas obliczeń i jednocześnie charakteryzuje się lepszym wynikiem stosowania niż algorytm losowy. Połączenie i ICP oraz SIFT umożliwia dokonywanie lokalizacji robota w przestrzeni 6D w czasie rzeczywistym, umożliwia także rozwiązanie problemu SLAM dla robota przemieszczającego się w przestrzeni 3D. Z prowadzonych badań wynika, że najlepsze wyniki osiągnęte są w przypadku analizy mapy głębi. W przyszłości planujemy zastosowanie SIFT do analizy semantycznej danych oraz stworzenia algorytmu SLAM.

Prace zostały wykonane w ramach grantu MEiN. 4311/B/T02/2007/33

6. LITERATURA

- [1] Thrun S., Burgard W., Fox D.: *Probabilistic Robotics* (MIT Press, Cambridge, MA, 2005).
- [2] Borenstein J., Everett H., Feng L., Wehe D.: Mobile robot positioning: Sensors and techniques, *J. Robot. Syst.* 14, 231- 249, 1997
- [3] Weiss G., Puttkamer E.: A Map Based on Laserscans Without Geometric Interpretation, *Proceedings of the Intelligent Autonomous Systems*. pp. 403-407, 1995.
- [4] Jensfelt P., Christensen H.: Laser Based Position Acquisition and Tracking in an Indoor

Tabela 2. Wyniki eksperymentów

	Obraz P1	Obraz P2
Chmury punktów		
Obrazy głębi		
Wektory normalne		
Cosinusy		

- Environment, *Proceedings of the IEEE International Symposium on Robotics and Automation*, pp. 331-338, 1998.
- [5] Besl B. J., McKay N. D.: A method for registration of 3-D shapes, *IEEE Trans. Patt. Anal. Machine Intell.*, 18(5):540-547, 1996.
- [6] Rusinkiewicz S.: Real – Time Acquisition and Rendering of Large 3D Models. Ph.D. *Dissertation, Stanford University, 2001.*
- [7] Besl P., McKay N.: A Method of Registration of 3 – D Shapes. *Trans. PAMI*, Vol. 14, No. 2, 1992
- [8] Ke Y., Sukthankar R.: PCA-SIFT: A more distinctive representation for local image descriptors, in *IEEE CVPR*, vol. 2, pp. 506–513, 2004.
- [9] Mikolajczyk K., Schmid C.: A performance evaluation of local descriptors, in: *IEEE CVPR*, vol. 2, pp. 257–263, 2003.
- [10] Moradi M., Abolmaesumi P.: Medical image registration based on distinctive image features from scaleinvariant (SIFT) keypoints, in: *CARS*, pp. 1292, 2005.
- [11] Ke Y., Sukthankar R.: PCA-SIFT: A more distinctive representation for local image descriptors, in: *IEEE CVPR*, 2004, vol. 2, pp. 506–513.