

Jacek Nowakowski*, Adam Mencwal**, Paweł Kośla**

Rekonstrukcja otoczenia robota mobilnego przy wykorzystaniu systemu stereowizyjnego

1. Wprowadzenie

Roboty mobilne systematycznie wkraczają w nowe dziedziny zastosowań, zarówno dotyczących sfery techniki, jak i szeroko pojętego życia codziennego. W początkowym okresie ich rozwoju głównymi dziedzinami ich zastosowań była eksploracja obszarów niedostępnych dla człowieka, głównie ze względów bezpieczeństwa, np. obszarów skażonych radioaktywnie. Z czasem po wykorzystanie robotów mobilnych sięgnęło wojsko i policja w sytuacjach zagrożenia bezpieczeństwa i życia ludzi, głównie jako zdalnie sterowane roboty do prac pirotechnicznych.

Jednocześnie były prowadzone prace badawcze i konstrukcyjne mające na celu rozszerzenie możliwości nawigacji robotów i dziedzin ich zastosowań. Różne konstrukcje robotów są szeroko opisywane w literaturze [5–7]. Na rysunku 1 są przedstawione przykłady różnych robotów mobilnych.



Rys. 1. Przykłady konstrukcji robotów mobilnych

* Katedra Informatyki Stosowanej, Politechnika Łódzka, Wyższa Szkoła Humanistyczno-Ekonomiczna w Łodzi

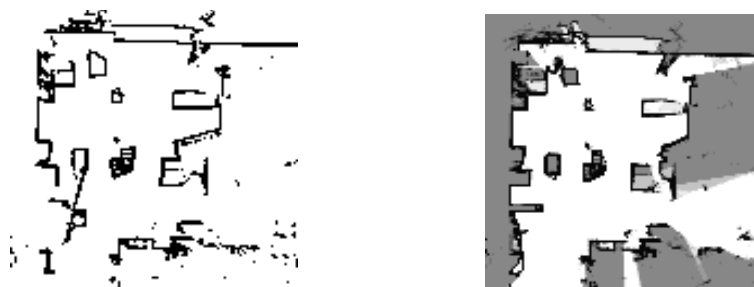
** Katedra Informatyki Stosowanej, Politechnika Łódzka w Łodzi

Inną dziedziną prac badawczych w zakresie robotów mobilnych są badania dotyczące minirobotów mobilnych, które znajdują szerokie zastosowanie na uczelniach wyższych do celów dydaktycznych. Roboty te ze względu na stosunkowo niską cenę i prostotę układu sterowania pozwalają na szerokie propagowanie zagadnień robotyki w środowisku akademickim. Jednym z robotów tego typu jest robot mobilny zbudowany w Instytucie Obrabiarstwa i Technologii Budowy Maszyn Politechniki Łódzkiej.

Istotną funkcjonalnie częścią tego robota jest podsystem stereowizyjny dokonujący aktualizacji mapy przestrzeni roboczej robota mobilnego. W niniejszym artykule zostały zaprezentowane zagadnienia teoretyczne związane z przetwarzaniem obrazów stereowizyjnych oraz główne koncepcje funkcjonowania podsystemu. Przedstawiono również wyniki działania aplikacji, interfejs użytkownika oraz parametry wydajnościowe programu.

2. Aktualizacja mapy otoczenia robota mobilnego

Warunkiem koniecznym do nawigacji robota mobilnego jest znajomość jego położenia w środowisku pracy. Jednym z zadań wykonywanych przez moduł nawigacyjny układu sterowania robotem mobilnym jest generowanie mapy przestrzeni roboczej. Przykłady map otoczenia przedstawia rysunek 2.



Rys. 2. Przykładowe mapy otoczenia robota mobilnego

Orientacja robota w otoczeniu i znajomość swojej pozycji i położenia przeszkód pozwala na zaplanowanie trajektorii ruchu robota do założonego celu. Wiele algorytmów sterowania robotami opiera się na mapie otoczenia aktualizowanej na bieżąco [6, 21, 22]. Obecnie stosowane algorytmy sterowania robotem mobilnym uwzględniają zmienność środowiska robota, ale traktują przeszkody, które pojawiają się na trajektorii robota lub zostają z niej usunięte tak samo jak wszystkie pozostałe [11, 12].

Wiele publikacji poświęconych procesowi identyfikacji ruchomych przeszkód dotyczy sytuacji spotykanych w ruchu drogowym [6, 22]. Foggia zaprezentował wstępne wyniki symulacji dotyczących opracowania działającego w czasie rzeczywistym systemu rozpoznawania przeszkód i ruchomych obiektów: MOOD – *Moving Object and Obstacle Detection* [5]. Sterowanie robotów mobilnych uwzględniające sygnały o otaczających robota przeszkodach z różnych źródeł, takich jak skaner laserowy, system kamer, czujniki ultradźwiękowe były tematem wielu publikacji [2, 4, 13, 14].

3. Metody wizyjne

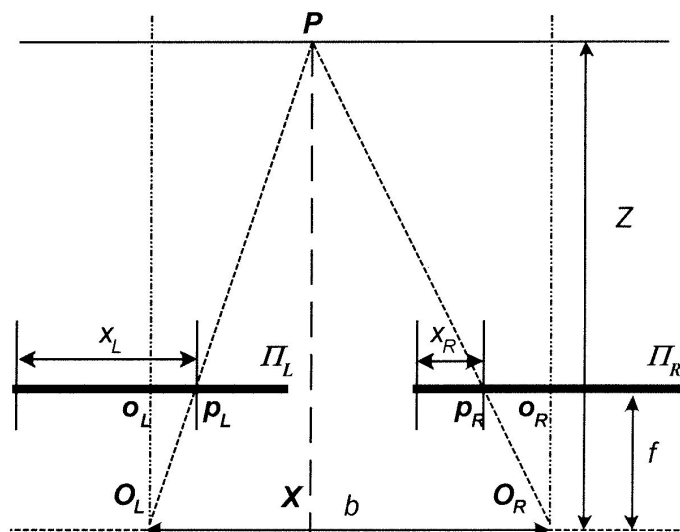
Wykorzystanie sygnału wizyjnego w nawigacji robotów mobilnych znajduje coraz szersze zastosowanie [10]. Jedną z możliwych konfiguracji jest taka, gdy kamera bądź zespół kamer, jest umieszczona na robocie mobilnym [1, 8, 13, 23].

Najczęściej używanymi systemami z wieloma kamerami są systemy stereowizyjne [7, 17, 18]. W przypadku najczęściej wykorzystywanych robotów mobilnych poruszających się po terenie płaskim, zasadniczą informacją uzyskiwaną z przetworzenia obrazu z kamer jest odległość obiektu od robota. Po wyznaczeniu odległości od obiektu, np. metodą dysparycji, możliwe jest wyznaczenie położenia robota.

3.1. Układ stereowizyjny

Bardzo obszerne i szczegółowe rozważania dotyczące problemów związanych z matematycznymi zagadnieniami z dziedziny przetwarzania obrazów trójwymiarowych i ich pozyskiwaniem zostały przedstawione w monografii [3].

W układach stereowizyjnych jednemu punktowi P (rys. 3) odpowiadają dwa jego obrazy P_L i P_R uzyskane odpowiednio przez kamerę lewą i prawą. Każdy z tych obrazów jest inny i przez to położenie obrazów tych samych punktów nie może być jednakowe. Różnica odległości obrazów tych samych punktów (obiektów) na prawym i lewym ekranie jest nazywana **dysparycją**. Jest to jeden z najważniejszych parametrów wykorzystywanych w procesach dopasowywania punktów, które prowadzą do rekonstrukcji trójwymiarowej sceny. Szczegółowe oznaczenia dotyczące rysunku 3 są podane w [3].



Rys. 3. Układ kanoniczny kamer stereowizyjnych [3]

Proces wyznaczenia mapy przeszkód w otoczeniu robota na podstawie uzyskanego obrazu z kamer stereowizyjnych można podzielić na następujące etapy:

1. Dopasowanie odpowiadających sobie punktów z dwóch obrazów kamer.
2. Wyznaczenie głębi poszczególnych odpowiadających sobie punktów.
3. Wyznaczenie mapy głębi dla obrazu skorygowanego.
4. Przekształcenie mapy głębi obrazu skorygowanego na mapę przeszkód.

Do celów nawigacji robotów mobilnych nie jest konieczne wyznaczenie parametrów geometrycznych obiektów trójwymiarowych. Badania są prowadzone dla robota mobilnego poruszającego się po terenie płaskim, niemającego możliwości pokonywania wzniesień, schodów, przemieszczania się ponad przeszkodami. Umożliwia to narzucenie pewnych ograniczeń dotyczących mapy głębi odtworzonego obrazu.

4. Konstrukcja robota mobilnego

Opracowane teoretycznie i sprawdzone podczas symulacji komputerowych algorytmy sterowania robotem były weryfikowane doświadczalnie na trójkołowym robocie mobilnym. Robot ten został zbudowany w Instytucie Obrabiarek i Technologii Budowy Maszyn Politechniki Łódzkiej a jego konstrukcja została przedstawiona m. in. w [11, 12, 20].

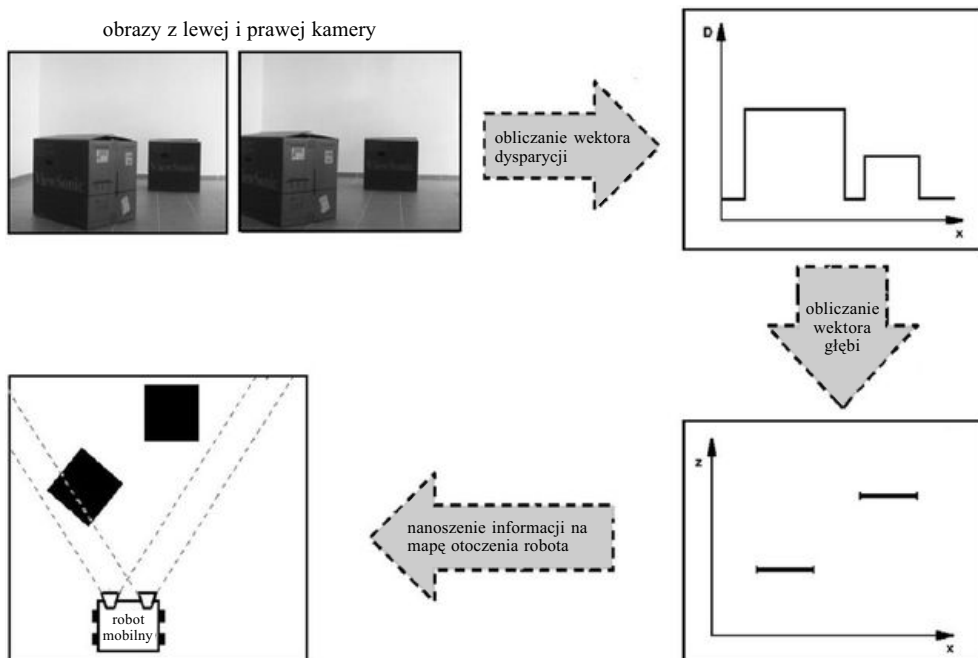
W skład systemu stereowizyjnego robota wchodzi zestaw kamer tworzących układ kanoniczny. Badania przeprowadzano korzystając z dwóch zestawów kamer:

- 1) Canon MV750i, DV(FireWire) o rozdzielczości 720×576 pikseli,
- 2) TechSolo 4810, USB o rozdzielczości 800×600 pikseli.

Parametry układu kanonicznego kamer zostały dobrane tak, by zapewnić rozdzielczość głębi 10 cm. Rozdzielczość ta jest równa podziałce mapy otoczenia robota wykorzystywanej w systemie nawigacji robota.

5. Analiza obrazu stereowizyjnego

Układ stereowizyjny to jeden z podsystemów robota mobilnego, którego celem jest aktualizacja mapy obiektów znajdujących się w otoczeniu robota. Celem analizy obrazu stereowizyjnego jest wyznaczenie odległości i położenia przeszkód względem robota. Pierwszym etapem przetwarzania obrazu stereowizyjnego jest wyliczenie wektora dysparycji dla poszczególnych elementów sceny obserwowanej przez kamery. Wartość dysparycji jest tym większa, im dany punkt sceny znajduje się bliżej obserwatora. Kolejnym etapem jest zredukowanie efektu perspektywy, przez którą obiekty tej samej wielkości znajdujące się w różnej odległości od obserwatora mają różną wielkość w rejestrowanym obrazie. Ostatecznie, wyliczony wektor posłuży do korygowania globalnej mapy obiektów w otoczeniu, po którym porusza się robot mobilny. Opisany schemat działania przedstawiony jest na rysunku 4.



Rys. 4. Schemat przetwarzania informacji w systemie stereowizyjnym robota mobilnego

5.1. Algorytm obliczania dysparycji

Najważniejszą częścią aplikacji jest algorytm obliczania współczynnika dysparycji dla każdego punktu obrazu z prawej kamery. Przy założeniu, że $L(x, y)$ i $R(x, y)$ to punkty obrazu odpowiednio z lewej i prawej kamery, dysparycja dla punktu $R(x, y)$ obliczana jest na podstawie maksymalnej wartości funkcji korelacji otoczenia tego punktu z otoczeniem wybranego punktu obrazu z lewej kamery $L(x, y)$. Wykorzystywane do tego celu były dwie miary podobieństwa:

- 1) SAD (*Sum of Absolute Differences* – suma modułów różnic),
- 2) SSD-N (*Sum of Squared Differences, Normalised* – znormalizowana suma kwadratów różnic).

Stosowanie drugiej miary dawało zdecydowanie lepsze wyniki w porównaniu z pierwszą pod względem poprawności odnajdywania podobnych fragmentów obrazów. Jednak złożoność obliczeniowa była zdecydowanie większa, co jest niekorzystnym parametrem w tego typu aplikacjach:

- SAD

$$\sum_{(i,j) \in U} |I_R(x+i, y+j) - I_L(x+i, y+d_y+j)|,$$

– SSD-N

$$\frac{\sum_{(i,j) \in U} \left(I_R(x+i, y+j) - I_L(x+i, y+d_y+j) \right)^2}{\sqrt{\sum_{(i,j) \in U} I_R(x+i, y+j)^2 \cdot I_L(x+i, y+d_y+j)^2}},$$

gdzie:

$I_R(x+i, y+j)$ – jasność piksela o współrzędnych $(x+i, y+j)$ w obrazie prawym,

$I_L(x+i, y+j)$ – jasność piksela o współrzędnych $(x+i, y+j)$ w obrazie lewym,

U – otoczenie badanego punktu,

d_y – dysparycja w pikselach.

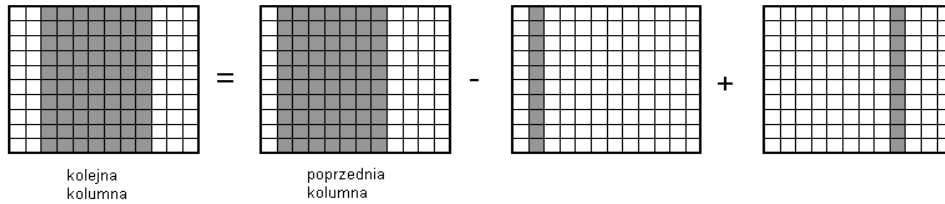
Otoczeniem w badanych przypadkach jest zbiór całych kolumn tworzących okno ROI (*Region of Interest*). Parametr szerokości okna jest określany przez użytkownika.

Ponieważ obraz z prawej kamery zawsze jest przesunięty w lewo względem obrazu z lewej kamery, nie ma potrzeby badania korelacji z wszystkimi obszarami lewego obrazu, a jedynie z tymi znajdującymi się na prawo od analizowanego fragmentu.

Parametryzowana jest maksymalna możliwa dysparycja ograniczająca strefę poszukiwania, dzięki czemu można wyeliminować fragmenty o większej dysparycji od założonej, ale przyspieszony zostaje czas działania algorytmu. Przeprowadzone wstępne badania efektywności działania algorytmu pozwoliły na znaczące ograniczenie analizowanego obszaru. Ze względu na konstrukcję robota i kąt widzenia układu optycznego kamer można ograniczyć obszar analizowanego obszaru usuwając z niego fragmenty obejmujące przestrzeń znacznie powyżej poziomej osi optycznej kamer, a więc i powyżej obrysu robota. Umożliwiło to zwiększenie szybkości przetwarzania danych z układu stereowizyjnego kamer do 10÷16 ramek/s.

Optimalizacja algorytmu

W celu przyspieszenia działania algorytmu znajdowania dysparycji poszczególnych kolumn ograniczono ilość operacji potrzebnych do obliczenia miary podobieństwa kolejnych fragmentów obrazów poprzez zapamiętywanie sum pośrednich. Miara podobieństwa była obliczana dla wydzielonych pionowych kolumn pikseli w obrazie. W przypadku sąsiednich kolumn otoczenie to częściowo jest wspólne, więc nie ma potrzeby ponownego obliczania wszystkich składowych funkcji. Rysunek 5 przedstawia przykład wyjaśniający działanie algorytmu optymalizującego dla wybranego otoczenia o szerokości 7 pikseli. Dla pewnej kolumny otoczenie, dla którego przeprowadza się obliczenia obejmuje kolumny od drugiej do ósmej. Dla kolejnej kolumny otoczenie o tej samej szerokości obejmuje kolumny od trzeciej do dziewiątej. Zatem mając zapamiętane wyniki obliczeń pośrednich otoczenie następnej kolumny otrzyma się z otoczenia kolumny poprzedniej, od którego odejmie się wynik dla kolumny drugiej, a doda wynik kolumny dziewiątej.



Rys. 5. Optymalizacja algorytmu

W toku przeprowadzonych eksperymentów wprowadzono również modyfikację miary SAD dopasowania obrazów. Modyfikacja polegała na dodatkowym dzieleniu przez sumę jasności pikseli badanego obszaru (zmodyfikowana miara określona została jako SAD-N). Nie zwiększając znacznie czasu przetwarzania, osiągnięto efekt jakościowo zbliżony do wersji SSD-N. Dodatkowo, dzięki tej modyfikacji możliwe było badanie podobieństwa obszarów o różnych wymiarach (efekt pojawiający się na krańcach obrazów).

SAD-N

$$\frac{\sum_{(i,j) \in U} |I_R(x+i, y+j) - I_L(x+i, y+d_y+j)|}{\sum_{(i,j) \in U} I_R(x+i, y+j)},$$

gdzie:

- $I_R(x+i, y+j)$ – jasność piksela o współrzędnych $(x+i, y+j)$ w obrazie prawym,
- $I_L(x+i, y+j)$ – jasność piksela o współrzędnych $(x+i, y+j)$ w obrazie lewym,
- U – otoczenie badanego punktu,
- d_y – dysparycja w pikselach.

5.2. Budowanie mapy przeszkód

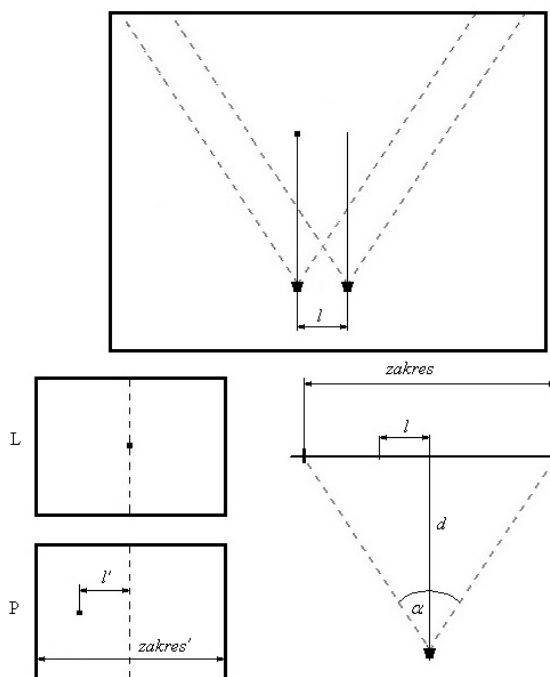
Na podstawie wartości dysparycji danego punktu w obrazie można stwierdzić w jakiej odległości od kamer znajdował się obiekt reprezentowany na obrazie przez piksel. Do ilościowego opisanie tej odległości potrzebne są dodatkowo informacje o szerokości obrazu, poziomym kącie widzenia kamery oraz odległości między dwiema kamerami (rys. 6).

Ponieważ przesunięcie (dysparycja) punktu na obrazie jest zawsze równe rozstawowi kamer, można z prostych zależności geometrycznych określić jaki jest zakres widoczności kamery w zależności od odległości od obiektywu. Pojęcie to definiowane jest jako maksymalna szerokość rejestrowanej prostopadłej płaszczyzny do osi kamery, która znajduje się w pewnej odległości od obiektywu.

$$\text{zakres} = \frac{l}{l'} \cdot \text{zakres}',$$

gdzie:

- zakres' – szerokość obrazu w pikselach,
- l' – dysparycja wyrażona w pikselach,
- l – odległość między osiami optycznymi kamer.



Rys. 6. Układ kamer stereowizyjnych i pomiar dysparycji

Na podstawie wartości dysparycji i wiedzy o kącie widzenia kamery można wyznaczyć odległość tej płaszczyzny od obiektywu, a więc można przeliczyć każdą wartość dysparycji na odpowiadającą jej odległość w rzeczywistości, wykorzystując przekształcenie trygonometryczne

$$d = \frac{\text{zakres}}{2 \cdot \operatorname{tg}\left(\frac{\alpha}{2}\right)},$$

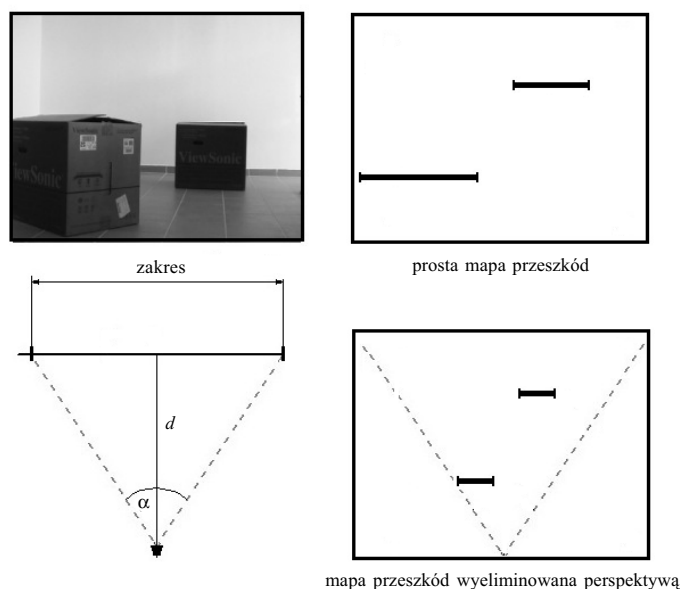
gdzie:

- zakres – zakres widoczności kamery,
- α – kąt widzenia kamery.

Mapa z wyeliminowanym efektem perspektywy

Wygenerowana mapa przeszkód na podstawie wartości dysparycji dla poszczególnych kolumn jest zniekształcona o efekt perspektywy. W celu odzwierciedlenia rzeczywistej przestrzeni wokół robota efekt perspektywy należy usunąć.

Procedura ta opiera się ponownie na wartości kąta widzenia kamery. Na jej podstawie wiadomo jaki jest maksymalny zakres widoczności w płaszczyźnie oddalonej od obiektywu o wartość d (rys. 7). Dzięki tej informacji poszczególne poziome linie mapy przeszkód (odzwierciedlające płaszczyzny w pewnej odległości od obiektywu) są skalowane z wykorzystaniem prostych przekształceń geometrycznych.



Rys. 7. Eliminowanie efektu perspektywy z mapy przeszkód

5.3. Aplikacja do wyznaczania mapy przeszkód z obrazu stereowizyjnego

Jako platformę systemową pod opracowaną aplikację wybrano system Windows XP, ze względu na wbudowaną usługę WIA (*Windows Image Acquisition*). Uniwersalność technologii WIA umożliwia stosowanie różnych interfejsów kamer, takich jak USB czy FireWire. Do zaprojektowania aplikacji wykorzystano platformę programistyczną .NET. Wybrany został język C# i platforma Visual Studio 2005. Ze względu na rozszerzenie funkcjonalności aplikacji i możliwość pracy z programem off-line w programie została zaimplementowana możliwość korzystania z zastępczego źródła obrazu stereowizyjnego, jakim są wcześniej nagrane stereowizyjne sekwencje wideo. Do obsługi plików filmowych zastosowano technologię DirectShow oraz użyto projekt DirectShowNet (<http://directshownet.sourceforge.net>), umożliwiający korzystanie z tej technologii dla języków platformy .NET. Wykorzystane biblioteki są ogólnodostępne na zasadzie licencji otwartego oprogramowania GNU LESSER GENERAL PUBLIC LICENCE Version 2.1.

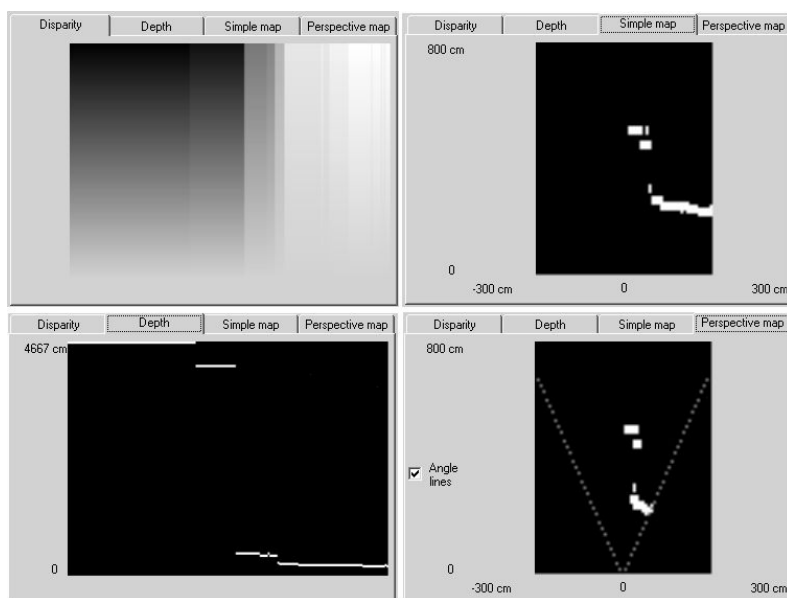
Opracowana aplikacja pozwala na:

- wybór trzech źródeł obrazów – rzeczywisty z kamer oraz pliki graficzne i wideo z wcześniej zarejestrowanymi scenami stereowizyjnymi;
- kalibrację kamer, czyli takie ustawienie, by osie optyczne kamer były równoległe;
- manipulację parametrami opisującymi układ optyczny (takimi jak wymiary rejestrowanego obrazu, ogniskowe układów optycznych kamer, odległość między kamerami, itp) oraz parametrami odpowiedzialnymi za tworzenie mapy przeszkód w zakresie

widoczności kamer; parametry te mają na celu regulację szybkości i dokładności analizy danych, oraz odpowiednie wyskalowanie wygenerowanej mapy do wymiarów rzeczywistego otoczenia robota.

6. Wyniki eksperymentów

Program umożliwia podgląd wyników działania algorytmów tworzenia wektora dysparycji, wektora głębi i mapy przeszkód. Na poszczególnych zakładkach prezentowana jest graficzna forma tych wyników (rys. 8).



Rys. 8. Okna aplikacji prezentujące kolejne etapy konstruowania mapy otoczenia robota

Wektor dysparycji (*Disparity*). Wyskalowany w skali szarości wektor zawierający przesunięcie w pikselach poszczególnych kolumn. Kolor biały oznacza maksymalne przesunięcie w danej scenie.

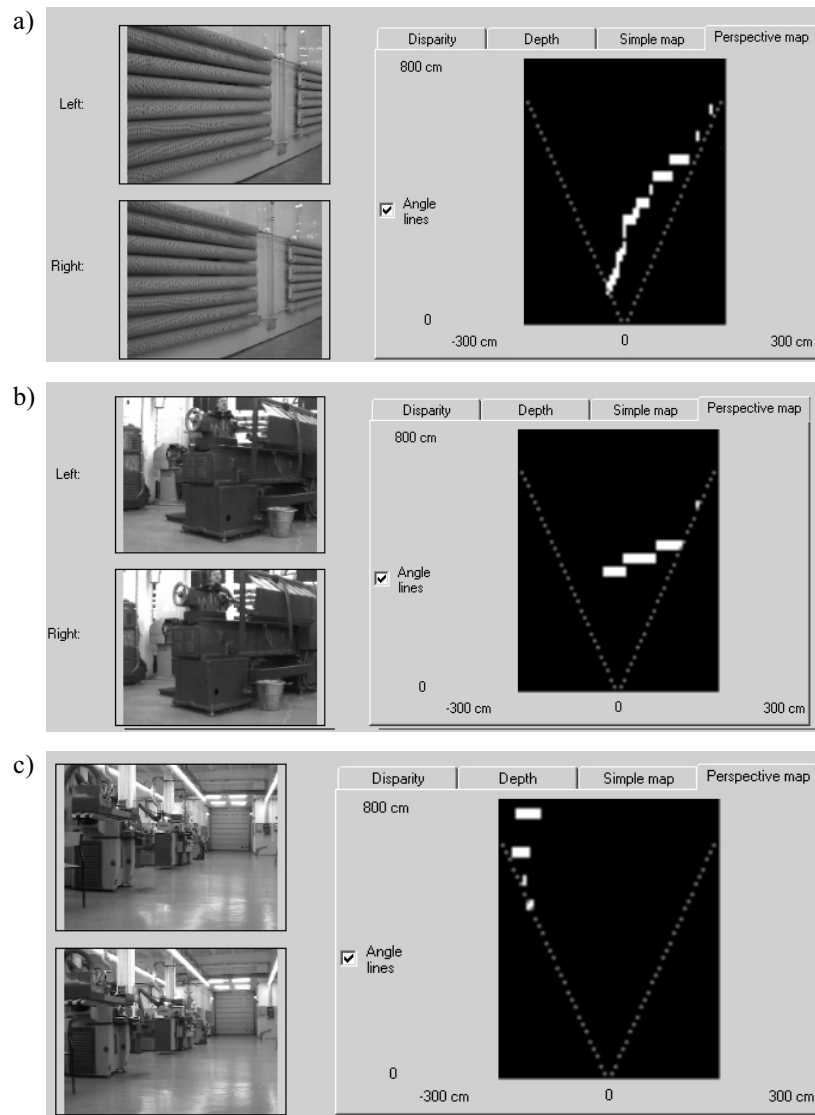
Wektor głębi (*Depth*). Zawiera graficzną ilustrację odległości przeliczonej na podstawie dysparycji i parametrów kamery. Maksymalna odległość jest odległością wynikającą z parametrów obrazu i kamery.

Prosta mapa przeszkód (*Simple obstacle map*). Jest to ograniczony i wyskalowany fragment wektora głębi na podstawie podanych parametrów mapy.

Mapa przeszkód z perspektywą (*Perspective obstacle map*). Jest to najistotniejszy wynik działania programu, najlepiej odzwierciedlający przestrzeń wokół robota. Prezentuje mapę przeszkód z uwzględnieniem zjawiska perspektywy. Mapa ta zależy od poziomego kąta widzenia kamery.

6.1. Mapy przeszkód rzeczywistych scen

Na rysunku 9 są przedstawione przykładowe wyniki przetwarzania sceny z otoczenia robota. Na kolejnych rysunkach są przedstawione przykładowe pary obrazów stereowizyjnych fragmentu hali technologicznej oraz przedstawiony w różnej formie graficznej wynik obliczenia dysparycji i mapy przeszkód z usunięciem efektu perspektywy.



Rys. 9. Przykładowe obrazy z kamer robota mobilnego wraz ze skonstruowaną mapą otoczenia (a, b, c)

7. Podsumowanie i wnioski końcowe

7.1. Szybkość przetwarzania danych stereowizyjnych

Program testowany był na kilku komputerach i przy wykorzystaniu kamer opisanych w rozdziale 4.

Komputery:

- Duron-AMD Duron 1,3 GHz, 640MB RAM (333MHz), HDD ATA133,
- Intel-Celeron 2,66 GHz, 512MB RAM, HDD ATA133.

W tabeli 1 przedstawiono czas potrzebny na analizę jednej pary obrazów stereowizyjnych.

Tabela 1
Prędkość przetwarzania obrazów stereowizyjnych

	Plik graficzny	Plik wideo	Kamery DV	Kamery USB
Duron	120 ms	350 ms	–	172 ms
Intel	47 ms	170 ms	90 ms	80 ms

Czas związany bezpośrednio z analizą obrazów to czas zbliżony do przypadku gdzie źródłem były pliki graficzne. W przypadku kamer czas przetworzenia jednej klatki jest zdecydowanie dłuższy. Związane jest to ze stosunkowo długim czasem wyodrębniania obrazu ze strumienia wideo. Przyczyną tego jest interfejs WIA, w którym jedynym możliwym sposobem przechwycenia klatki ze strumienia wideo jest zapisanie go na dysku a następnie wczytanie do pamięci. Interfejs USB czy FireWire nie wydaje się być znaczący, różnica między czasami dla kamery DV i USB związana jest raczej z rozdzielczością przechwytywanego obrazu.

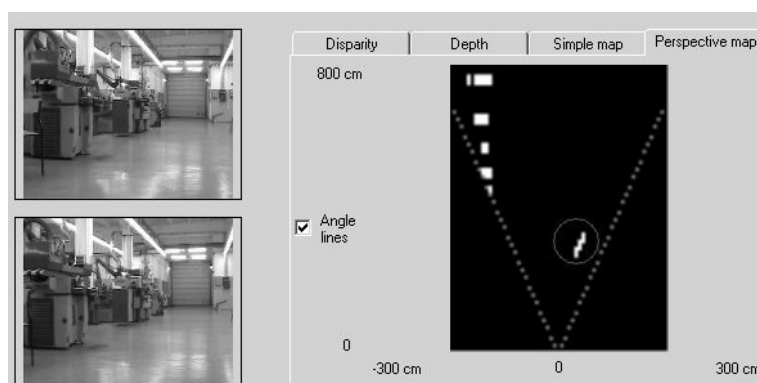
W przypadku źródła będącego plikiem wideo, długi czas przetwarzania obrazu związany jest z jednoczesnym obciążeniem procesora wynikającym z wyświetlania filmu.

Mimo niedogodności związanych z interfejsem WIA, osiągnięte wyniki dla kamer rzędu 10÷12 klatek na sekundę powodują, że analiza obrazów stereowizyjnych i budowa mapy przeszkód może się odbywać w czasie rzeczywistym.

Jak było wspomniane wcześniej, efektem działania procesu analizy obrazów stereowizyjnych jest wektor dysparycji, którego poszczególne wartości wskazują przesunięcie danej kolumny pikseli w drugim obrazie. Jako parametry tego procesu podawane jest maksymalne przesunięcie, podawane w procentowej szerokości obrazów. Większa wartość umożliwia rozpoznanie bliżej znajdujących się przeszkód, jednak kosztem dłuższego czasu analizy jednej sekwencji. Przeprowadzone testy doprowadziły do wniosku, że optymalną wartością jest 20%, umożliwiała ona rozpoznawanie obiektów od odległości około 1 metra (w zależności od parametrów kamery). W przypadku poruszania się robota w stosunkowo statycznej przestrzeni, odległość taka jest w zupełności wystarczająca, ponieważ przeszkoda zostanie zauważona dużo wcześniej, przy większej odległości.

7.2. Ograniczenia i problemy z wyznaczeniem dysparycji

Bardzo niekorzystnym zjawiskiem pojawiającym się w obrazach rejestrowanych za pomocą kamer są odbicia światła, szczególnie te od podłoża. Powodują one błędne traktowanie takiego odbicia jako przeszkody. By wyeliminować ten problem wskazane jest stosowanie filtrów polaryzacyjnych. Przykład sceny, na której wyraźnie widać odbicie światła w podłodze, jest pokazany na rysunku 10.



Rys. 10. Nieprawidłowe rozpoznanie przeszkody – efekt odbicia światła od podłoża

Na zdjęciach przedstawionych na rysunku 10 widoczne jest odbicie światła, co w efekcie dało obraz przeszkody. Innym rozwiązaniem tego problemu może być zwiększenie analizowanego sąsiedztwa. W celu wyeliminowania artefaktów związanych z odbiciem światła zwiększono zakres analizowanego okna do 20% szerokości obrazu, co przyniosło oczekiwany efekt, czyli odbicie światła w podłodze nie zostało rozpoznane jako przeszkoda.

Badania były realizowane w ramach grantu MNiI w latach 2004–2006.

Literatura

- [1] Betke M., Gurrivits L.: *Mobile Robot Localization Using Landmarks*. Proceedings of the IEEE International Conference on Robotics and Automation, 1994, 135–142
- [2] Bonci A., Leo T., Longhi S.: *Ultrasonic and Video Data Fusion for Mobile Robot Navigation*. Proceedings of the 10th Mediterranean Conference on Control and Automation – MED2002, Lisbon, Portugal
- [3] Cyganek B.: *Komputerowe przetwarzanie obrazów trójwymiarowych*. Warszawa, Akademicka Oficyna Wydawnicza EXIT 2002
- [4] Démonceaux C., Kachi-Akkouche D.: *Robust Obstacle Detection with Monocular Vision Based on Motion Analysis*. 2004 IEEE Intelligent Vehicles Symposium, University of Parma, Parma 2004, 527–532
- [5] Foggia P., Limongiello A., Vento M.: *A moving object and obstacle detection system in real time AVG and AMR applications*. ARAS Conference, Cairo 2005, 2190–2195

- [6] Lefaix G., Marchand E., Bouthemy P.: *Motion-based Obstacle Detection and Tracking for Car Driving Assistance*. IAPR Int. Conf. on Pattern Recognition, ICPR'2002, vol. 4, 74–77
- [7] Luong Q.T., Weber J., Koller D., Malik J.: *An integrated stereo-based approach to automatic vehicle guidance*. Proceedings of the Fifth International Conference on Computer Vision, 1995, 52–57
- [8] Mahkovic R., Slivnik T.: *Generalized Local Voronoi Diagram of Visible Region*. International Conference on Robotics and Automation, 1998 Leuven, Belgium, 349–355
- [9] Matthies L., Xiong Y., Hogg R., Zhu D., Rankin A., Kennedy B.: *A Portable, Autonomous, Urban Reconnaissance Robot*. International Conference on Intelligent Autonomous Systems, Venice, 2001
- [10] Murray D. Little J.: *Using real-time stereo vision for mobile robot navigation*. Autonomous Robots, vol. 8, Issue 2, 2000, 161–171
- [11] Podśędkowski L.: *Dynamiczne planowanie trajektorii robotów mobilnych w zmiennej przestrzeni roboczej*. Zeszyty Naukowe PŁ, nr 809, 1999
- [12] Podśędkowski L., Kwapisz L., Idzikowski M., Vizvary I.: *Sterownik ruchu robota mobilnego dla częściowo znanej przestrzeni roboczej*. PAR XII'00, 2000
- [13] Rennekamp T., Homeier K., Kroeger T.: *Distributed Sensing and Prediction of Obstacle Motions for Mobile Robot Motion Planning*. Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. Beijing, China 2006
- [14] Skrzypczyński P., Drapikowski P.: *Environment modelling in a multi – agent mobile system*. EUROBOT'99, 1999
- [15] Tomoaki Y., Akihisa O., Shin'ichi Y.: *Braille Block Detection for Autonomous Mobile Robot Navigation*. Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems
- [16] Wang D., Hamam Y.: *Optimal Trajectory Planning of manipulators with collision detection and avoidance*. The International Journal of Robotics Research, vol. 11, No. 5, October 1992, 460–468
- [17] Williamson T., Thorpe C.: *A trinocular stereo system for highway obstacle detection*. International Conference on Robotics and Automation (ICRA'99) 1999
- [18] Xiong Y., Matthies L.: *Error analysis of a real-time stereo system*. Conference on Computer Vision and Pattern Recognition, 1997. Proceedings, 1997 IEEE Computer Society, 1087–1093
- [19] Zhang Q., Pless R.: *Constraints for Heterogeneous Sensor Auto-Calibration*. Conference on Computer Vision and Pattern Recognition Workshop, 2004, 38
- [20] Idzikowski M.: *Laserowy system nawigacyjny robotów mobilnych*. Politechnika Łódzka, 2005 (praca doktorska)
- [21] Pfister S.T., Roumeliotis S.I., Burdick J.W.: *Weighted Line Fitting Algorithms for mobile robot map building and efficient data representation*. ICRA'03, 2003
- [22] Broggi A., Caraffi C., Porta P., Zani P.: *The Single Frame Stereo Vision System for Reliable Obstacle Detection used during the 2005 DARPA Grand Challenge on TerraMax™*. Proceedings of the IEEE Intelligent Transportation Systems Conference, Toronto, 2006, 745–752
- [23] Dutkiewicz P., Kiełczewski M.: *Sprzężenie wizyjne w sterowaniu grupą robotów mobilnych*. http://theta.ar-kari.put.poznan.pl/system_wizyjny.htm, 2003