

METODA WEKTORYZACJI OBRAZU SCENY DLA AUTONOMICZNEGO ROBOTA

SCENE VECTORIZATION METHOD FOR AN AUTONOMOUS ROBOT

Piotr Śmigielski

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej
Katedra Informatyki Stosowanej
Al. Mickiewicza 3030-059 Kraków
e-mail: smigielski.piotr@gmail.com

Abstract: The aim of the studies was to design, implement and test the algorithm for vectorization of the scene, which is analyzed by an autonomous robot. The vectorization is the process of extraction of the contours of the objects located in the image taken by a camera. The result is stored in a computer's (robot's) memory as a sequence of points in euclidean space marking the examined objects' corners. During the tests, various images were used. Each one was previously pre-processed in a different way in order to reveal and highlight the contours of the objects. The vectorization process is divided into three steps which are described in detail in this article.

Keywords: vectorization, scene analysis, scene model, autonomous robot.

Wprowadzenie

Systemy wizyjne dla robotów stanowią obiekt badań od ponad 20 lat [1]. Systemy te mają wyjątkowe znaczenie dla robotów autonomicznych, także dla autonomicznych robotów latających (ang.: *Unmanned Aerial Vehicles* - w skrócie UAV). Roboty te wyposażone są w sensory służące do zbierania danych o otaczającym je środowisku [2]. Poza potrzebą zbierania danych niezbędnych do wykonania przez robota zadania, dane te umożliwiają robotowi znajdowanie bezkolizyjnej trasy między przeszkodami [3].

Jednym z kompleksowych zadań, przed którym stawiane są autonomiczne roboty jest tworzenie mapy badanego obszaru przy jednoczesnym bezkolizyjnym poruszaniu się w środowisku i określaniu na bieżąco własnego położenia w przestrzeni. Klasa takich zadań znana jest pod określeniem SLAM (*Simultaneous Localization and Mapping*). Do niej należą zadania określania lokalizacji oraz tworzenia mapy z wykorzystaniem systemów wizyjnych robota – Visual SLAM [4]. Niejednokrotnie robot

osadzony zostaje w środowisku zmieniającym się w czasie, co dodatkowo komplikuje tego typu zadania. Systemy przeznaczone do rozwiązywania tej klasy problemów posługują się kamerami, kamerami termowizyjnymi oraz dalmierzami w celu analizy sceny. Im dokładniejszy obraz zostanie przekazany z sensorów tym wierniejszy model otaczającego środowiska może zostać wygenerowany przez robota, co przekłada się na możliwość bardziej precyzyjnego określenia położenia w przestrzeni [5].

Wspomniane zadania są trudne również dlatego, że muszą być wykonywane z uwzględnieniem rygoru czasowego. W chwili, gdy robot porusza się, musi przetwarzać informacje możliwie najszybciej by poruszać się w bezpieczny sposób w skomplikowanym i często zmieniającym się środowisku. Te ograniczenia w połączeniu z limitami, jakimi obarczony jest robot w kontekście możliwości instalowania na nim wydajnych układów obliczeniowych stwarzają potrzebę projektowania efektywnych algorytmów przetwarzania

danych, oraz oszczędnych pamięciowo reprezentacji danych przestrzennych.

Proces wektoryzacji jest jednym z pierwszych etapów analizy sceny [6]. Prowadzi on do wyodrębnienia kształtów i cech obiektów znajdujących się w otoczeniu robota. Im dokładniejszy model wektorowy obiektów zostanie wygenerowany tym lepsze Oszacowanie pozycji w przestrzeni jest możliwe. Jednocześnie dokładniejszy modelu (mapa) sceny może zostać stworzona przez robota.

Przedmiotem badań opisanych w niniejszej publikacji było stworzenie algorytmu wektoryzacji obrazu, dedykowanego dla autonomicznego robota poruszającego się w trzech wymiarach – latającego lub nurkującego. Celem było stworzenie wydajnej obliczeniowo metody dającej oszczędną pamięciowo reprezentację, która może stanowić bezpośrednio wejście do dalszych etapów przetwarzania obrazu przez robota. Model wektorowy w ujęciu prezentowanym w poniższych rozdziałach składa się z reprezentacji poszczególnych obiektów obserwowanych przez kamerę robota. Każdy z obiektów reprezentowany jest przez ciąg punktów wyznaczających położenie kolejnych narożników bryły. Każdy taki ciąg definiuje łamaną zamkniętą na płaszczyźnie

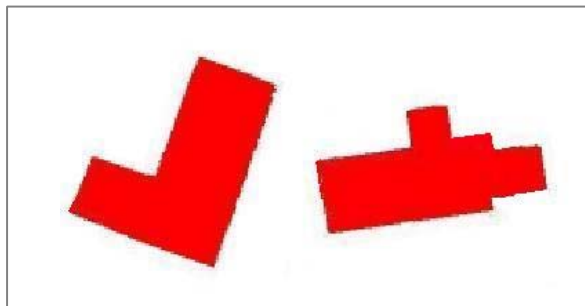
euklidesowej. Ostateczny wynik wektoryzacji danego zdjęcia niesie ze sobą informacje o kształcie, wielkości, skomplikowaniu poszczególnych brył oraz o relacjach przestrzennych między nimi.

Opis zagadnienia - opis algorytmu

Zaproponowana metoda przetwarza obraz w kilku etapach:

- 1 - wyodrębnienie krawędzi bryły na zdjęciu;
- 2 - iteracyjne wyznaczenie kolejnych, gęsto rozmieszczonych punktów na obrysie bryły;
- 3 - usunięcie zbędnych punktów, które leżą na wspólnej prostej i przybliżenie punktami miejsc, gdzie znajdują się narożniki bryły;
- 4 - przybliżenie lokalizacji narożników bryły poprzez post-processing otrzymanej w poprzednich etapach wektorowej reprezentacji.

Na dane wejściowe dla zaproponowanej metody nie są nałożone szczególne warunki. Niemniej jednak dla najbardziej zadowalających wyników zalecane jest wykorzystanie zdjęć poddanych pre-processingowi poprawiającego widoczność i kontrastowość brył, których kształt chcemy poddać wektoryzacji. Przykłady wyniku pre-processingu zaprezentowane zostały na rys. 1.



Rys. 1. Przykładowe zdjęcia po pre-processingu uwydatniającym kontury obiektów.

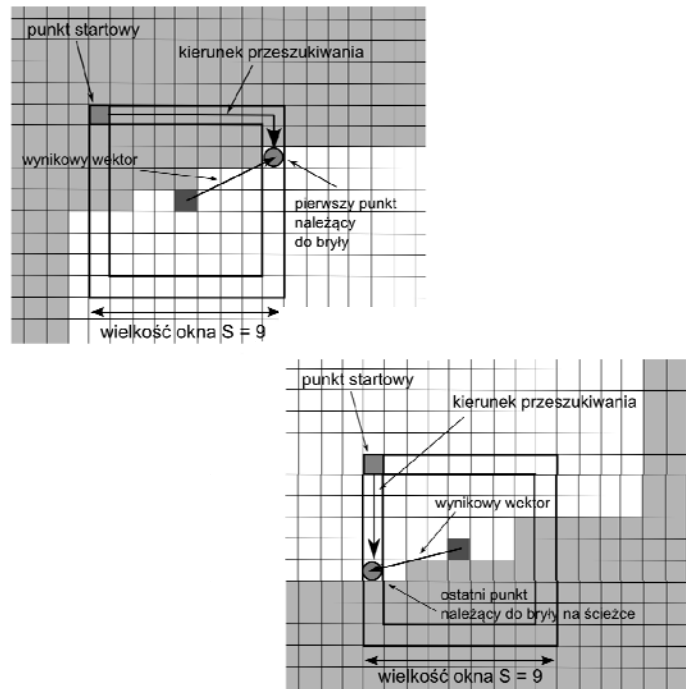
Prezentowany algorytm wektoryzacji może działać na obrazie dwukolorowym, będącym wynikiem silnego pre-processingu zdjęcia lub na obrazie z wyodrębnionymi krawędziami brył. W celu wyodrębnienia krawędzi brył podczas testów, wykorzystany został algorytm wzorowany na metodzie Canny Edge Detector [7]. Jest to metoda pozwalająca na uzyskanie bitmapowego obrazu gdzie piksel jednego

koloru określa tło sceny a piksel innego koloru miejsce krawędzi bryły. Bazuje ona na wyszukiwaniu najbardziej kontrastowych przejść barw na zdjęciu.

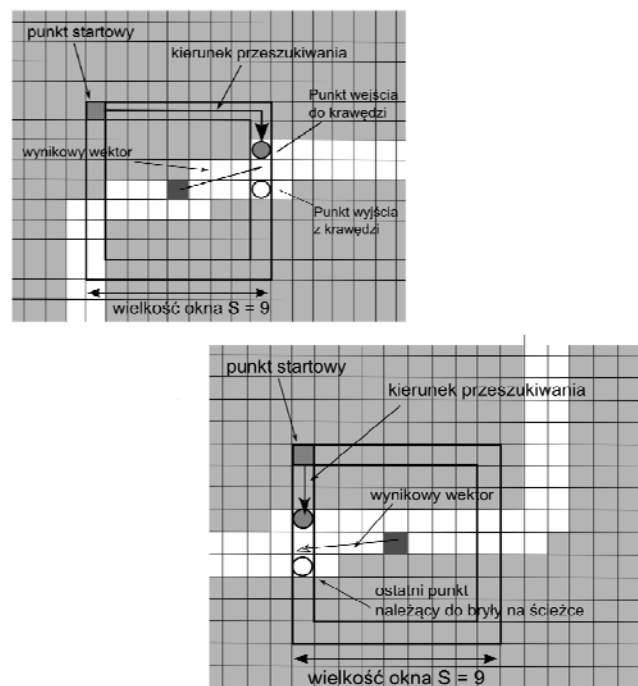
W dalszym etapie działania algorytmu wykorzystana zostaje bitmapa uzyskana w wyniku krawędziowania bądź silnego pre-processingu dającego w efekcie dwukolorową bitmapę. W kroku tym wyznaczany jest ciąg

punktów na obrysie każdej z wyodrębnionych brył. Do uzyskania pożądanego wyniku wykorzystana została struktura „okna”, która pozwala iteracyjnie podążać po linii sąsiadujących ze sobą pikseli wyznaczających obrys. Z tej linii wybierane są piksele w równej

odległości i pobierane są ich współrzędne. Te współrzędne wykorzystywane będą w dalszych etapach algorytmu jako punkty określające lokalizację bryły na obrazie. Pojęcie „okna” oraz sposób jego wykorzystania został przedstawiony na rys. 2 oraz rys. 3.



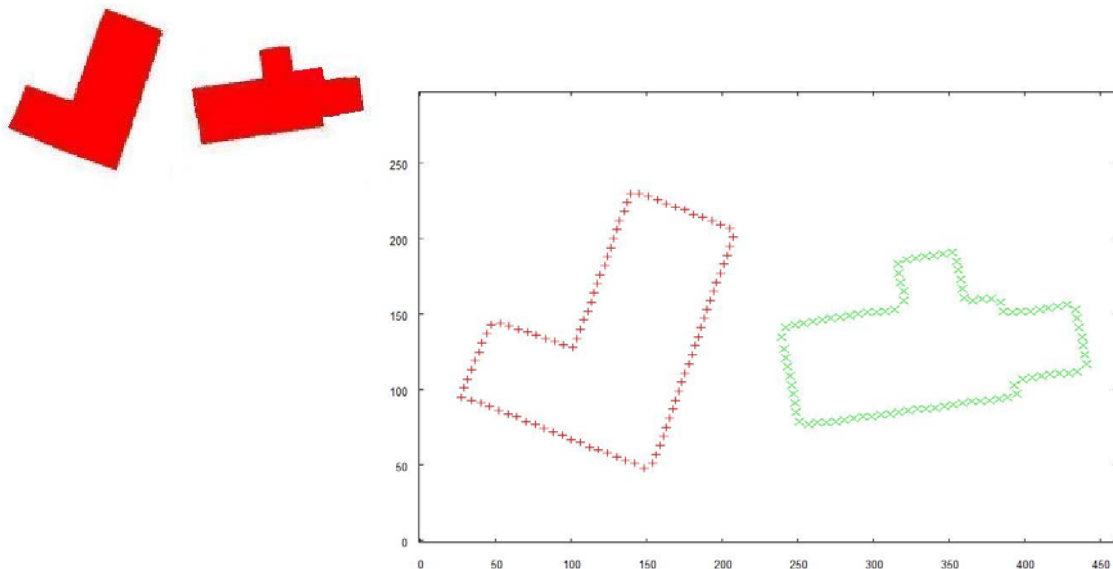
Rys. 2. Wykorzystanie „okna” do znajdowania kolejnych punktów na obrysie bryły uwidocznionej w wyniku silnego pre-processingu (piksel biały – powierzchnia bryły, piksel szary – tło obrazu).



Ryc. 3. Wykorzystanie „okna” do znajdowania kolejnych punktów na obrysie bryły po zastosowaniu metody krawędziowania (piksel biały – krawędź bryły, piksel szary – tło obrazu).

Poszczególne punkty wyznaczone w powyższym etapie definiują kolejne wektory wzdłuż

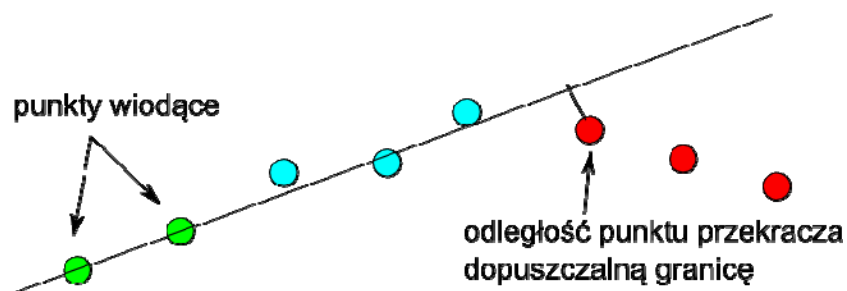
obrysu analizowanej bryły. Na rys. 4 przedstawione zostały wyniki tego etapu algorytmu.



Rys. 4. Wynik wyznaczania punktów na obrysie bryły. Wielkość obrazka wejściowego: 460x290 pikseli. Szerokość „okna” 13 pikseli (odległość między kolejnymi punktami: 7 pikseli).

Kolejnym etapem działania jest usuwanie zbędnych punktów, które znajdują się na wspólnej prostej. Warto zauważyć, że punkty takie nie wnoszą nic do szczegółowości opisu bryły. Istotne pozostają jedynie punkty przybliżające miejsce narożników obiektu, a tym samym jego kształt i rozmiary. Na rys. 5 przedstawiona jest zasada działania tej metody. Polega ona na określeniu prostej na podstawie

współrzędnych pierwszych dwóch punktów (punkty wiodące). Następnie usuwane są kolejne punkty leżące odpowiednio blisko wyznaczonej prostej. Kiedy odległość punktu przekroczy wielkość opisaną parametrem algorytmu punkt ten nie jest usuwany, lecz pozostaje on w zbiorze jako przybliżający miejsce narożnika bryły.



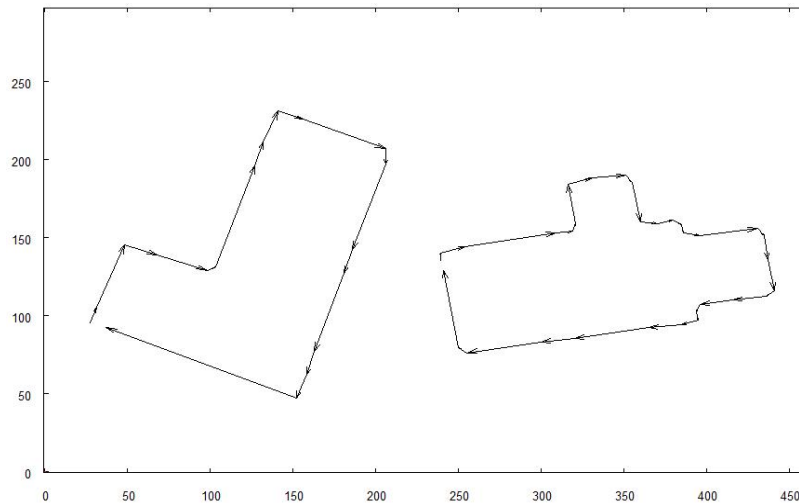
Ryc. 5. Sposób usuwania zbędnych punktów z obrysu bryły.

Na rys. 6 przedstawiony jest wynik tego etapu. Tym razem kolejne uzyskane wektory reprezentowane są przez strzałki, co umożliwia obserwację jak zmniejszyła się liczba punktów na obrysie, a co za tym idzie objętość pamięciowa reprezentacji bryły.

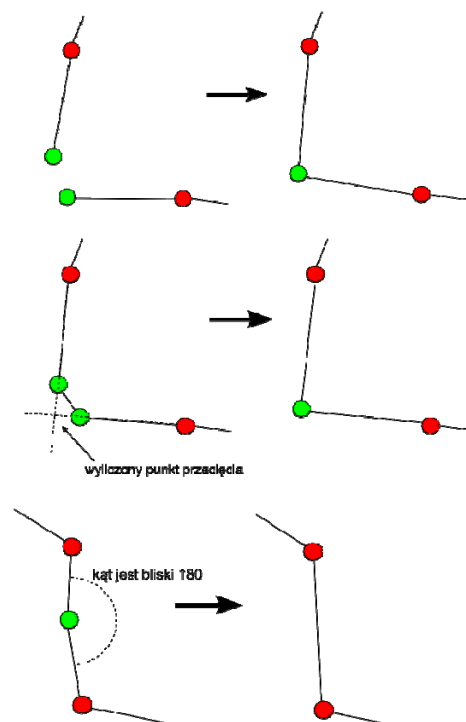
Ostatnim etapem działania algorytmu jest 3-stopniowy post-processing wektorowej reprezentacji. Ma on na celu poprawienie sposobu, w jaki reprezentacja przybliża miejsca narożników i załamań obrysu bryły. Rys. 7 przedstawia kolejne kroki tego etapu. W punkcie 1)

następuje złączenie punktów, które należą do osobnych łamanych a znajdują się dostatecznie blisko siebie (wielkość określona parametrem). Krok ten ma na celu poprawę wyniku działania poprzednich kroków algorytmu, między innymi wykorzystania „okna”, gdzie może dojść do wyodrębnienia pozornie spójnego obiektu jako dwóch lub więcej elementów. W punkcie 2) ma miejsce łączenie bliskich sobie punktów na jednej, spójnej łamanej (maksymalna odległość,

przy której łączymy punkty określona innym parametrem niż poprzednio). Pozycja wynikowego punktu wyliczona jest jako przecięcie prostych wyznaczonych przez wektory sąsiadujące z krótkim wektorem usuwanym w tym kroku. W punkcie 3) usuwamy punkt, który rozdziela dwa wektory, między którymi jest kąt bliski 180° (są prawie współliniowe). Odchył od 180° , poniżej którego następuje usunięcie definiowany jest przez parametr algorytmu.



Rys. 6. Wynik usuwania zbędnych punktów na obrysie bryły (bryła po lewej: 96 → 16 punktów, bryła po prawej: 91 → 25 punktów).

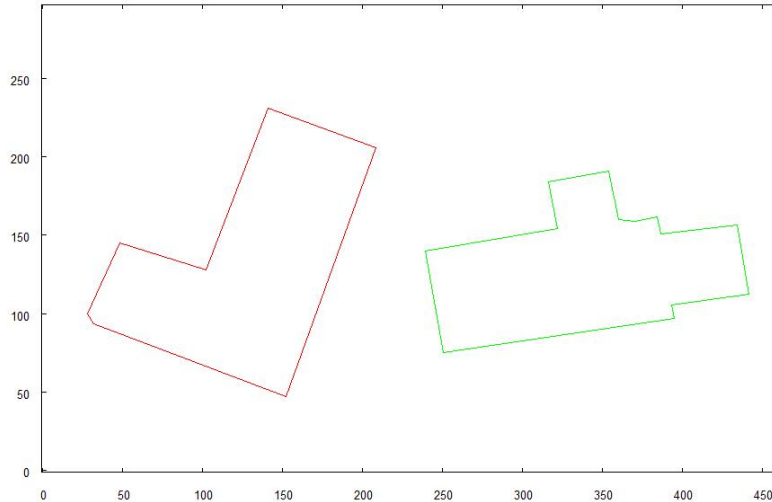


Rys. 7. Post-processing wektorów. Trzy kroki poprawiające przybliżenie punktów narożnych bryły przez współrzędne: od góry: 1- domykanie łamanych; 2 - zastępowanie bliskich punktów jednym; 3- usuwanie punktów na prostej linii.

Wyniki

Ostateczny wynik działania algorytmu dla danych zaprezentowanych na początku

poprzedniego rozdziału przedstawiony jest na rys. 8. Na rysunku punkty przybliżające narożniki brył połączone zostały liniami w celu ukazania kształtu brył.



Rys. 8. Ostateczny wynik działania algorytmu.

Dyskusja

Reprezentacja obiektów przedstawiona na rys. 6 daje dobre przybliżenie kształtów brył. Kolejne 3 kroki algorytmu mające na celu poprawienie przybliżenia narożników brył przez współrzędne punktów dają dużą poprawę w kwestii wielkości reprezentacji i przybliżenia kształtu brył przez ciąg punktów. Poprawa ta przekłada się na mniejszą ilość pamięci potrzebnej do działania algorytmu, co nie jest bez znaczenia przy implementacji algorytmu na robocie latającym poruszającym się w złożonym środowisku i przetwarzającym znaczne ilości informacji dochodzących w każdej sekundzie z sensorów wizyjnych.

Wynik algorytmu obrazuje też niedociągnięcia funkcjonowania algorytmu, które uzależnione są od doboru parametrów jego funkcjonowania (widoczne zdublowanie punktu w narożniku lewej bryły). Polem dla rozwoju opisanego algorytmu jest automatyzacja doboru wartości

parametrów, która może zostać oparta o rozróżnienie rozdzielczości obrazu, gęstości rozmieszczenia wyodrębnionych obiektów oraz wielkości ich samych.

Podsumowanie

Przedstawiona metoda daje dobre rezultaty dla obrazów poddanych pre-processingowi wyodrębniającemu kształty budynków. Algorytm działa w sposób szybki i deterministyczny (bazuje na prostych krokach dających przewidywalne wyniki). Reprezentacja będąca wynikiem jego działania – ciągi punktów na płaszczyźnie euklidesowej - jest oszczędna pamięciowo oraz umożliwia wykonywanie szybkich operacji matematycznych i przekształceń. Stanowi tym samym dobrą podstawę dla bardziej skomplikowanych operacji jakie robot autonomiczny musi wykonać poruszając się bezkolizyjnie w skomplikowanym lub zmieniającym się środowisku.

Literatura

1. Bonin-Font, F., Ortiz, A., Oliver, G., Visual navigation for mobile survey, *Journal of Intelligent and Robotic Systems*, 2008, 53, pp. 263–296.
2. Muratet, L., Doncieux, S., Briere, Y., Meyer, J.A., A contribution to vision-based autonomous helicopter flight in urban environments, *Robotics and Autonomous Systems*, 2005, 50, pp. 195–229.

3. Hrabar, S., Sukhatme, G. S., Corke, P., Usher, K., Roberts, J., Combined optic-flow and stereo-based navigation of urban canyons for a UAV. In Proceedings of the international conference on intelligent robots and systems IROS'2005, pp. 3309–3316.
4. Liu, J., Liu, D., Cheng, J., Tang, Y., Conditional simultaneous localization and mapping: A robust visual SLAM system, *Neurocomputing*, 2014, 145, pp. 269–284.
5. Strasdat, H., Montiel, J.M.M., Davison, A.J., Visual SLAM: Why filter? *Image and Vision Computing*, 2012, 30, pp. 65–77.
6. Bielecki, A., Buratowski, T., Śmigielski, P., 2013. Recognition of two-dimensional representation of urban environment for autonomous flying agents, *Expert Systems with Applications*, 2013, 40, pp. 3623–3633.
7. Canny, J. 1986. A Computational Approach To Edge Detection, *Pattern Analysis and Machine Intelligence*, 1986, 8, pp. 679–698.