



KoNaR

KOŁO NAUKOWE ROBOTYKÓW

SYSTEM WIZYJNY ROBOTA KLASY LINEFOLLOWER

Projekt z cyfrowego przetwarzania obrazów i sygnałów

Autor: *Grzegorz BIZIEL*

Prowadzący: *mgr inż. Łukasz MAŁEK*

Termin: wtorek parzysty 13:15

27 stycznia 2010

1 Problem

Rozwój techniki mikroprocesorowej oraz podnosząca się jakość przetworników obrazu pozwala na rozwiązywanie nowej puli problemów przy pomocy systemów wizyjnych. Postawionym zadaniem było zaprojektowanie algorytmu systemu wizyjnego dla robota klasy *linefollower* wydobywającego z obrazu z przed robota ścieżkę i przedstawiającego ją w zadanej parametryzacji.

2 Środowisko rozwoju algorytmu

Środowiskiem rozwoju był system Linuks oraz biblioteka OpenCV. Algorytm ma docelowo działać na płycie uruchomieniowej BeagleBoard, ale tworzony był na platformie PC z uwagi na większą elastyczność pracy na komputerze stacjonarnym. Biblioteka OpenCV dostarczyła większość podstawowych operacji przetwarzania (strukturę przechowywania obrazu, konwersje przestrzeni kolorów, operacje morfologiczne, wydobycie krawędzi). Kod był rozwijany w języku C++. Urządzeniem do akwizycji obrazu została tania kamera USB firmy Apollo oparta o przetwornik CMOS i układ Z-Star ZC0301.

3 Algorytm

3.1 Wstęp

Problemem postawionym przed algorytmem jest wydobywanie z obrazu ścieżki, którą ma podążać robot. W tym celu musi wydobyć z obrazu parametry toru przy jednoczesnym odrzuceniu elementów, które mogą się pojawić w polu widzenia robota lub zdobyć informację o przeszkodach blokujących tor. W tym celu wykorzystano operacje morfologiczne oraz ocenę momentów geometrycznych.

3.2 Wydobycie krawędzi

Podstawą dla wydobycia krawędzi z obrazu stał się jego charakter – silny kontrast między czarną linią toru i białym tłem. W związku z tym procedura wydobycia krawędzi została oparta o obraz binarny uzyskiwany w drodze progowania. Próg jest wybierany na podstawie danych statystycznych z poprzedniego przebiegu etapu 3.3. Jedną z operacji filtracji następuje już w tym etapie. Z uwagi na charakter sylwetki toru można na tym etapie usunąć grupę zaburzeń, która jak okazało się w testach była dość kosztowna do usunięcia w kolejnych etapach. W celu usunięcia drobnych podłużnych elementów obraz jest filtrowany przez parokrotne wykonanie operacji morfologicznego zamknięcia. Z takiego obrazu kontury są wydobywane na drodze różnicy obrazu wydobytych sylwetek oraz sylwetek poddanych erozji – tak zwana operacja *Top Hat*. Dużą zaletą użycia takiej procedury wydobycia krawędzi jest zapewnienie ciągłości konturu.

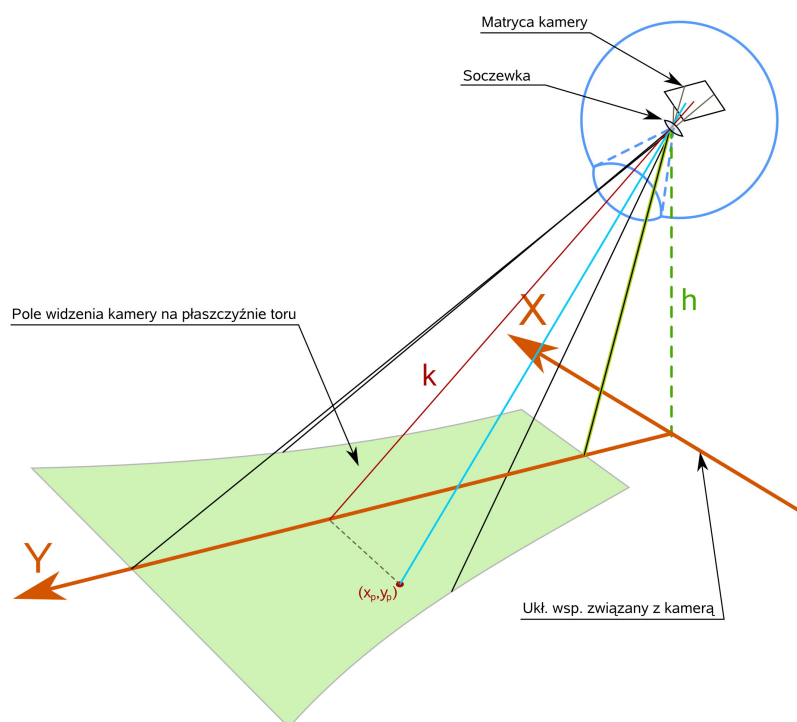
3.3 Wybór krawędzi

Obraz dostarczony do systemu może być obciążony różnymi zaburzeniami. Robot może chwilowo stracić poprawną orientację względem toru, przy krańcach

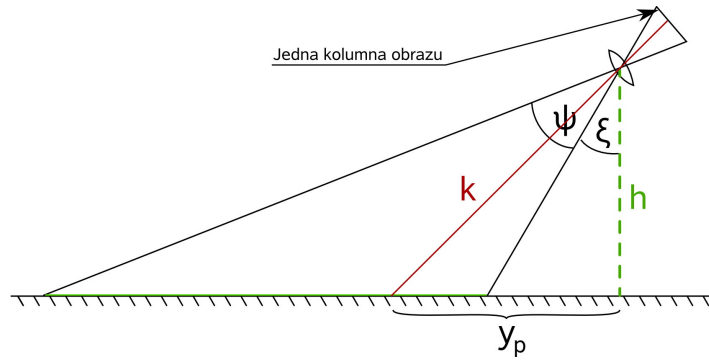
poła widzenia mogą pojawić się inne przedmioty lub sam tor może być zablokowany przez jakąś przeszkodę. W celu poprawnego wyboru krawędzi zbudowano klasyfikator. W celu minimalizacji kosztów obliczeniowych klasyfikatora obraz został podzielony na makrobloki i to w obrębie poszczególnych makrobloków dokonywano klasyfikacji. W każdym z nich wyliczono statystyki niskich rzędów. Na podstawie przewidywanych cech toru i eksperymentów stworzono funkcję zamieniającą wartości momentów na kategorie bloków. W drodze przeglądania typów bloków z sąsiedztwa wytypowano obszary z przeszkodami. Kontury sklasyfikowane jako przeszkody zostały usunięte. Z pośród pozostałych krawędzi wybierane są te o największej długości z uwzględnieniem progu minimalnej długości, ponieważ ewentualne fałszywe kontury, które mogły przedostać się przez filtrację są związane z drobnymi przeszkodami lub zabrudzeniami.

3.4 Zmiana parametryzacji

Dla wybranych z obrazu krawędzi dokonywana jest zmiana parametryzacji. Punkty składowe krzywych przenoszone są z układu odniesienia matrycy kamery do układu odniesienia krzyżującego się w punkcie na płaszczyźnie znajdującej się pionowo pod soczewką. Transformację wyprowadzono przy założeniu idealności zastosowanej optyki. Schemat ideowy sceny przedstawia rysunek 1. Na podstawie znanych kątów widzenia w pionie i poziomie oraz znanej wysokości soczewki i kąta pod jakim jest skierowana kamera zostały wyprowadzone wzory na odległość punktu od osi kamery.



Rysunek 1: Pole widzenia kamery na płaszczyźnie toru i układ współrzędnych kamery



Rysunek 2: Schemat widzenia kamery wzdłuż jednej kolumny obrazu

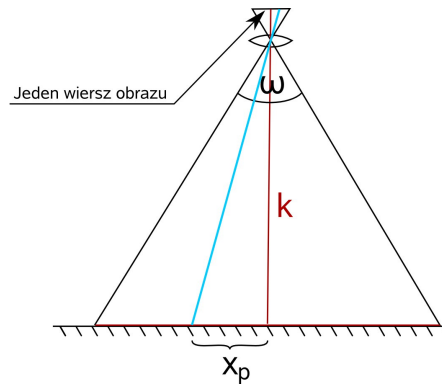
Dla jednej kolumny obrazu (rys. 2) odległość wyraża się jako iloczyn wysokości soczewki i tangensu części kąta widzenia dla danego piksela oraz kąta pola martwego przed kamerą (wzór 1).

$$y_p = \operatorname{tg}\left(\psi \frac{H-b}{H} + \xi\right)h \quad (1)$$

Gdzie ψ to kąt pola widzenia w pionie, ξ kąt pola martwego przed kamerą (obszar niewidoczny), H wysokość obrazu w pikselach, b numer linii punktu na obrazie, a h to wysokość soczewki nad powierzchnią toru.

Dodatkowo pomocniczo na potrzeby dalszych obliczeń wyliczana jest odległość od soczewki do danego punktu płaszczyźnie toru (wzór 2).

$$k = \frac{h}{\cos\left(\psi \frac{H-b}{H} + \xi\right)} \quad (2)$$



Rysunek 3: Schemat widzenia kamery wzdłuż jednego wiersza obrazu

Wykorzystując wcześniej wyliczone dane (wzór 2) można wyliczyć odległość od osi kierunku kamery (rys. 3) jako iloczyn k oraz tangens kąta odchylenia piksela od osi środkowej (wzór 3).

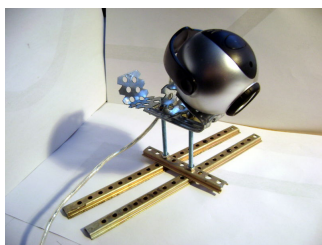
$$x_p = \operatorname{tg}\left(\omega \frac{1}{2} \frac{W \frac{1}{2} - a}{W \frac{1}{2}}\right)k \quad (3)$$

Gdzie ω to kąt pola widzenia kamery w poziomie, W szerokość obrazu, a numer kolumny na obrazie.

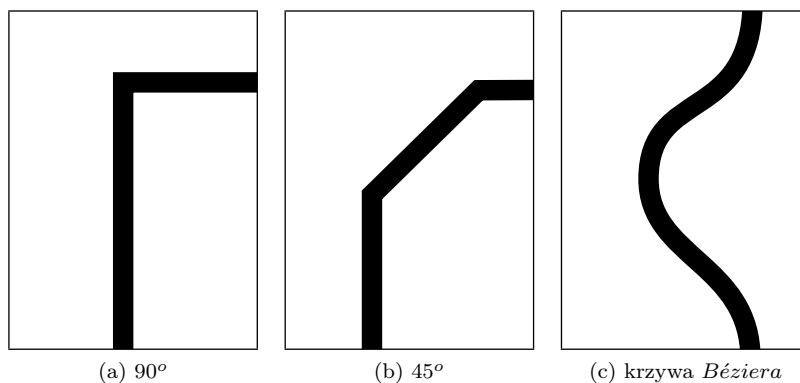
Ponieważ transformacja jest wykonywana dla niezmiennego kąta nachylenia kamery i jej wysokości, w związku z tym można było wyliczyć wartości dla wszystkich punktów na obrazie wcześniej i stworzyć *look-up table* transformacji pozwalającą usunąć powtarzające się obliczenia. Dodatkowo, aby zredukować ilość informacji o transformowanych krzywych są one aproksymowane łamanymi przy pomocy algorytmu Douglas'a-Peucker'a.

3.5 Testy

Na potrzeby testów zbudowano prostą podstawę zapewniającą stałą wysokość i nachylenie kamery względem podłoża (rys. 4). Przygotowano parę kart rozmiaru A4 z różnymi elementami potencjalnego toru – prostymi stykającymi się pod kątem 45 lub 90 stopni oraz z odcinkiem krzywej *Béziere* (rys. 5).

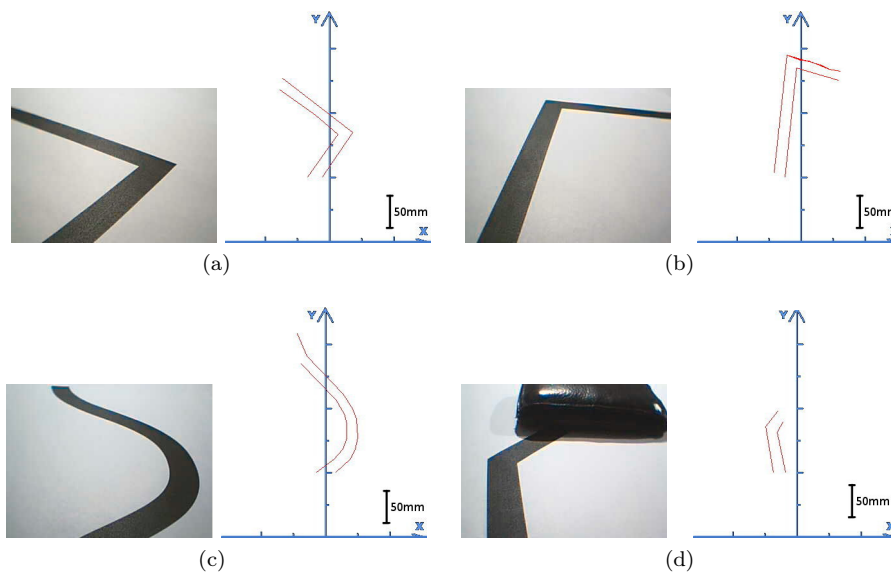


Rysunek 4: Podstawa kamery symulująca docelowy montaż



Rysunek 5: Plansze testowe

Wyniki kilku prób dla plansz przedstawia rysunek 6. Pary rysunków przedstawiają źródłowy obraz z kamery oraz krawędzie w parametryzacji związanej z kamerą. Efekty przedstawiane są w postaci unormowanej. System radzi sobie z krzywymi łamanymi jak i gładkimi. Wykrywa również przeszkody (rys. 6d). Stojak zapewnił następujące parametry pracy: długość pola martwego przed kamerą – 103mm, wysokość soczewki nad powierzchnią toru – 100mm, długość pola widzenia – 230mm, szerokość pola widzenia w jego połowie – 130mm.



Rysunek 6: Przykładowe wyniki

4 Wnioski

Zbudowany system wizyjny dostarcza dobrej jakości informacji o trasie. W związku z niską rozdzielczością przetwornika da się zauważyć drobne błędy odwzorowania dla punktów odległych – przy końcu pola widzenia. Reprezentacja krzywych gładkich jako łamanych daje zadowalające efekty przy znacznej redukcji ilości punktów opisujących krzywą. Przy doborze kamery do realizacji podjętego zadania trzeba wziąć pod uwagę możliwą rozdzielczość pracy ze względu na precyzję, ilość klatek na sekundę z uwagi na rozmycia związane z ruchem robota i jakość soczewki ograniczającą efektywne pole widzenia aberracjami sferycznymi. Ustawiając kąt nachylenia kamery do podłoża należy wziąć pod uwagę dostateczny zasięg pola widzenia z akceptowalną długością pola przed robotem, które zostanie wyłączone z analizy.

Literatura

- [1] Gonzalez R., Woods R., *Digital Image Processing*. Prentice-Hall, New Jersey, 2nd Edition, 2002.
- [2] Malina W., Smiatacz M., *Cyfrowe przetwarzanie obrazów*. AOW EXIT, Warszawa, 2008.
- [3] Choraś R., *KOMPUTEROWA WIZJA, Metody interpretacji i identyfikacji obiektów*. AOW EXIT, Warszawa, 2005.
- [4] *Dokumentacja biblioteki OpenCV*
<http://opencv.willowgarage.com/wiki/>