

Piotr MRÓZ, Daniel KUJAWA
Uniwersytet Zielonogórski
Instytut Metrologii, Elektroniki i Informatyki

SYSTEM POMIAROWO–STERUJĄCY DO STEROWANIA MODELEM CIĘŻARÓWKI Z ROZPOZNAWANIEM OBRAZU Z KAMERY

W Instytucie Metrologii, Elektroniki i Informatyki trwają prace nad rozbudową systemu pomiarowo–sterującego nadzorującego pracę czterech zestawów modeli ciągników siodłowych z naczepami. Prace prowadzone są w kierunku automatycznego podjeżdżania ciągnika do dowolnej naczepy. W związku z tym pierwszym krokiem jest rozpoznawanie naczepy na podstawie obrazu z kamery. W artykule przedstawiono rozbudowę struktury systemu pomiarowo–sterującego o system rozpoznawania obrazu. Dodatkowo przedstawiono wyniki badań rozpoznawania naczepy na obrazie z kamery przeprowadzonych dwoma metodami: porównania ze wzorcem i zastosowania znaczników ArUco pod kątem możliwości wykorzystania ich w układach sterowania modelami ciężarówek.

THE MEASUREMENT AND CONTROL SYSTEM FOR TRUCK MODEL CONTROL WITH IMAGE RECOGNITION FROM THE CAMERA

At the Institute of Metrology, Electronics and Computer Science, the research is underway to develop a measurement and control system supervising the work of four sets of tractor models with semi-trailers. The works are carried out in the direction of the tractor automatically approaching any semi-trailer. Therefore, the first step is to recognize the semi-trailer based on the camera image. The article presents the expansion of the measurement and control system structure with an image recognition system. Additionally, are presented the results of research the trailer recognition on the image from the back camera made by two methods: comparison with the pattern and the use of ArUco markers. The research is made for its use in truck models.

1. WSTĘP

W Instytucie Metrologii, Elektroniki i Informatyki trwają prace nad rozbudową układu pomiarowo–sterującego przeznaczonego do sterowania zestawami czterech modeli ciągników siodłowych z naczepami pokazanymi na rys. 1 [1]. Cały czas trwają prace nad rozbudową oprogramowania i struktury systemu umożliwiające realizację nowych funkcji zarówno przez ciągniki siodłowe jak i naczepy.

Prace prowadzone są w kierunku automatycznego podjeżdżania ciągnika do dowolnej naczepy. W tym celu konieczne jest rozpoznanie naczepy, wyznaczenie jej środka a następnie opracowanie i implementacja algorytmu sterowania podjazdem samochodu do naczepy. W związku z tym rozbudowano układ pomiarowo–sterujący o system rozpoznawania obrazu. Umieszczony jest on w kabinie ciągnika.

W artykule przedstawiono rozbudowę systemu oraz wyniki badań rozpoznawania naczep.

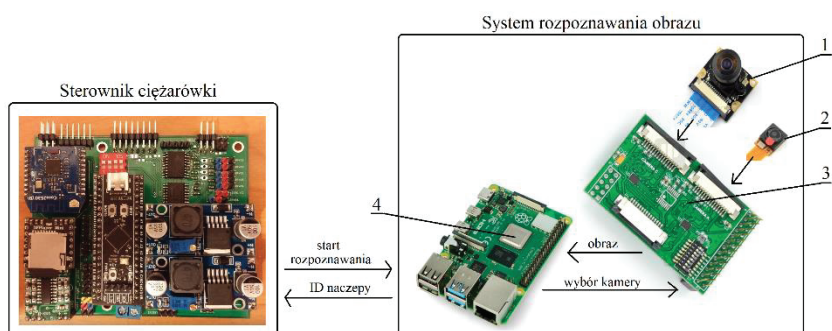
2. SYSTEM ROZPOZNAWANIA OBRAZU

Na rys. 2 pokazano strukturę systemu pomiarowo–sterującego z systemem do rozpoznawania obrazu. Dwie kamery: przednia – szerokokątna 1 i druga – zwykła 2 podłączone są do przełącznika kamer 3. Przełącznik kamer 3 połączony jest z procesorem Raspberry Pi 3+ 4, który odpowiedzialny

jest za rozpoznawanie obrazu. System ten jest połączony ze sterownikiem ciężarówki. Do rozpoznawania naczepy wykorzystywana jest kamera tylna 2.



Rys. 1. Widok modeli ciągników siodłowych z naczepami
Fig. 1. View of models of tractors with semi-trailers



Rys. 2. Struktura systemu pomiarowo–sterującego z systemem do rozpoznawania obrazu
Fig. 2. The structure of the measurement and control system with the image recognition system

Po wykryciu naczepy w siodle ciągnika sterownik ciężarówki przesyła do systemu rozpoznawania obrazu rozkaz rozpoczęcia rozpoznawania. Po rozpoznaniu naczepy lub po upływie określonego czasu system rozpoznawania obrazu zwraca kod naczepy lub informację, że jej nie rozpoznano.

3. ROZPOZNANIE NACZEPY PRZEZ PORÓWNANIE ZE WZORCEM

Pierwszą koncepcją było porównanie obrazu z kamery z wzorcami zapisanymi w pamięci komputera. Do przetwarzania obrazów wykorzystano bibliotekę OpenCV [2]. Podczas testów okazało się, że przy zastosowaniu tej metody jest wiele czynników wpływających na skuteczność rozpoznania naczepy. Są to między innymi kąt pod jakim wykonano zdjęcie naczepy, jasność zdjęcia, jego kontrast czy układ światłocienia [3, 4]. Wszystkie te czynniki powodują, że konieczne jest zapisanie wielu wzorców tej samej naczepy. Im jest więcej wzorców, tym większa szansa na poprawne rozpoznanie naczepy. Niestety zwiększenie liczby wzorców bezpośrednio wydłuża czas niezbędny do rozpoznania. Dodatkowo czas ten wydłuża się ze względu na liczbę różnych naczep. W wyniku testów ustalono, że zadawalające jest zastosowanie 9 wzorców na jedną naczepę. Przy wprowadzeniu po 9 wzorców dla 4 naczep czas rozpoznania naczepy wynosi około 10 s. Wprowadzono korekty do algorytmu wyszukujące w obrazie narożniki naczepy [5]. Niestety działania te nie przyspieszyły procesu rozpoznania naczepy.

Z punktu widzenia układu sterowania czas ten jest zbyt długi, dlatego rozwiązanie to nie może być wykorzystane w modelach ciężarówek.

4. ROZPOZNANIE NACZEPY Z ZASTOSOWANIEM ZNACZNIKÓW ArUco

Inną koncepcją jest zastosowanie znaczników ArUco. Metoda ta polega na umieszczeniu na naczepie znacznika, na którym będzie zapisany kod indywidualny dla każdej naczepy. Znacznik ArUco umieszczony został w przedniej części naczepy. Widok naczepy z umieszczonym na niej znacznikiem pokazano na rys. 3.

Po otrzymaniu przez procesor Raspberry Pi rozkazu poprzez interfejs UART wywoływana jest funkcja rozpoznawania kodu na znaczniku pokazana na rys. 4. W pierwszej kolejności przechwytywany jest obraz z kamery umieszczonej w tylnej części kabiny ciągnika siodłowego. Następnie na obrazie tym rozpoznawane są znaczniki ArUco. Jeżeli nie zostanie rozpoznany żaden znacznik, lub wykrytych będzie więcej niż jeden, to przyjmuje się, że kod ma wartość 0xff. Rozpoznawanie powtarza się do czasu, gdy rozpoznany będzie tylko jeden znacznik a jego kod jest zwracany przez funkcję. Dodatkowo sprawdzane jest, czy nie przyszedł rozkaz przerwania procesu rozpoznawania kodu. W takim przypadku zwracana jest wartość znaku równa 0xff. Oznacza ona brak rozpoznanego kodu. Na podstawie odczytanego kodu procesor sterujący pracą ciągnika siodłowego określa adres naczepy, do której podjechał i będzie sterował za pośrednictwem interfejsu bezprzewodowego ZigBee.

Do rozpoznawania obrazów wykorzystano dedykowane biblioteki współpracujące z biblioteką OpenCV [6]. Zastosowanie takiego rozwiązania sprawiło, że sam program jest znacznie prostszy i bardziej przejrzysty w porównaniu z metodą opisaną w p. 3, a rozpoznanie znacznika zajmuje mniej niż 1 s. Dzięki temu możliwe jest wykorzystanie tej metody w praktyce.

5. KIERUNKI DALSZYCH PRAC

Przy stosowaniu biblioteki rozpoznającej znaczniki ArUco oprócz zapisanego kodu na znaczniku możliwe jest określenie również współrzędnych jego środka. Przewiduje się, że informacja o środku znacznika, a co za tym idzie i naczepy, zostanie wykorzystana przy opracowaniu i implementacji algorytmów sterujących modelem podczas autonomicznego podjeżdżania ciężarówka do naczepy.

Podczas podjeżdżania prawdziwym ciągnikiem siodłowym do naczepy kierowca zmaga się z dwoma zadaniami: regulacji wysokości siodła ciągnika względem sworznia królewskiego (zaczepek) naczepy, oraz osiowego podjeżdżania ciągnika względem naczepy. Trwają prace nad opracowaniem systemu automatycznie regulującego wysokość siodła oraz wspierającego kierowcę podczas podjeżdżania do naczepy. Do realizacji drugiego zadania wykorzystuje się doświadczenia zebrane podczas realizacji opisanego w artykule rozpoznawania znaczników ArUco. W tym przypadku znacznik umieszczony w osi naczepy jest punktem odniesienia dla systemu informującego kierowcę o pozycji ciągnika siodłowego względem osi naczepy.

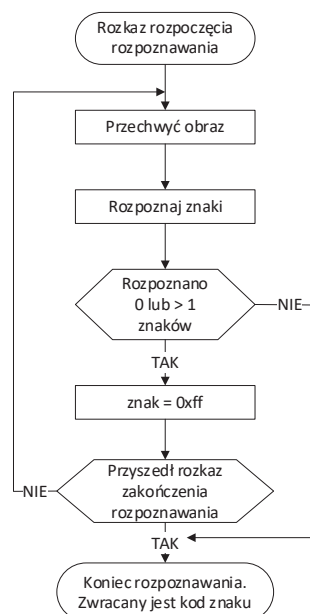
6. PODSUMOWANIE

System pomiarowo–sterujący do sterowania modelami ciągników siodłowych z naczepami znajdującymi się w Instytucie Metrologii, Elektroniki i Informatyki jest rozbudowywany o możliwość automatycznego podjeżdżania ciągnika do naczepy. Pierwszym etapem prac jest rozpoznawanie naczepy, do której podjeżdża ciężarówka. Z tego powodu prowadzono prace nad sposobami identyfikacji naczepy przez ciągnik. Zdecydowano się na rozpoznawanie w oparciu o obraz pozyskiwany z kamery umieszczonej w tylnej części kabiny samochodu.

W pierwszym kroku porównywano obraz z kamery z wzorcami zapisanymi w pamięci komputera. Badania wykazały, że musi być wiele wzorców, by uzyskać zadawalającą rozpoznawalność naczepy. Okazało się, że wraz z liczbą wzorców znacznie rośnie czas rozpoznawania, co powoduje, że metoda ta nie może być wykorzystana.



Rys. 3. Widok naczepy z umieszczonym na niej znacznikiem ArUco
Fig. 3. View of the semi-trailer with the ArUco sign



Rys. 4. Algorytm funkcji rozpoznawania kodu na znaczniku
Fig. 4. The algorithm of the code recognition function on the marker

W związku z powyższym zdecydowano się na umieszczenie na przedzie naczepy znacznika ArUco. Rozpoznawalność naczepy z takim znakiem jest bardzo wysoka, a czas rozpoznawania jest pomijalnie krótki. Powoduje to, że zdecydowano się na wykorzystanie tej metody przy rozpoznawaniu naczepy.

Metoda ta zostanie wykorzystana przy opracowywaniu systemu wspierającego podjeżdżanie „prawdziwego” ciągnika siodłowego do naczepy. W tym przypadku wyznaczana jest współrzędna środka naczepy, na podstawie której system informuje kierowcę o pozycji ciągnika siodłowego względem osi naczepy.

LITERATURA

1. <http://www.wuztruck.imei.uz.zgora.pl/> pobrany 2020-05-06
2. Kaehler, A. & Bradski, G. 2016, Learning OpenCV 3: computer vision in C with the OpenCV library, " O'Reilly Media, Inc."
3. Bay, H., Ess, A., Tuytelaars, T. & Van Gool, L. 2008, "Speeded-up robust features (SURF)", Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346-359.
4. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D. & Petkovic, D. 1995, "Query by image and video content: The QBIC system", computer, vol. 28, no. 9, pp. 23-32.
5. Mair, E., Hager, G.D., Burschka, D., Suppa, M. & Hirzinger, G. 2010, "Adaptive and generic corner detection based on the accelerated segment test", European conference on Computer visionSpringer, pp. 183.
6. <https://www.learnopencv.com/augmented-reality-using-aruco-markers-in-opencv-c-python/> pobrany 2020-05-06