

METODA ROZPOZNAWANIA PRZESZKÓD PRZEZ BEZZAŁOGOWY STATEK POWIETRZNY Z WYKORZYSTANIEM JEDNEJ KAMERY

LESZEK AMBROZIAK*, JAKUB CIEŚLUK*, ZDZISŁAW GOSIEWSKI**

Politechnika Białostocka, Instytut Lotnictwa***

Streszczenie

W artykule poruszony został problem omijania przeszkód i unikania kolizji przez bezzałogowy statek powietrzny z wykorzystaniem informacji zawartych w obrazie pobranym z jednej kamery. Zaprezentowany algorytm lokalizowania przeszkód wykorzystuje jedną z metod przepływu optycznego – gradientową metodę Lukas Kanade (LK), polegającą na analizowaniu podobieństw pomiędzy kolejnymi ramkami przetwarzanego obrazu. Wizyjny algorytm rozpoznawania przeszkód został zaimplementowany na systemie czasu rzeczywistego uClinux. Jego działanie zostało zweryfikowane na drodze symulacji komputerowej hardware-in-the-loop z modelem latającego skrzydła.

WPROWADZENIE

Lokalizowanie i rozpoznawanie przeszkód jest ważną częścią aplikacji urządzeń mobilnych, mających za zadanie działać autonomicznie w określonych warunkach swojej pracy. Do tej pory opracowanych zostało wiele metod pozwalających na wydobycie informacji o przeszkodzie, wykorzystując obraz pobrany z jednej kamery [7,8,9]. Są to metody oparte na segmentacji obrazu, wydobyciu głębi obrazu oraz grupa metod przepływu optycznego [2,3]. Metody bazujące na segmentacji obrazu wykorzystują barwę do opisu części obszaru niestanowiącego zagrożenia (np. barwę jezdni bądź nieba) i charakteryzują się dużą wydajnością obliczeniową, jednak małą dokładnością działania. Bardziej uniwersalne są metody służące do wydobycia głębi obrazu, które pozwalają na stworzenie mapy 3D badanego obszaru, dzięki czemu znane są informacje o odległości od widzianych przeszkód. Metody te wymagają jednak teksturowania środowiska, co pociąga za sobą konieczność tworzenia dobrej jakości zdjęć o jednolitym oświetleniu i wysokiej rozdzielczości. Może stanowić to duży problem, jeśli chcemy wykorzystać omawiane metody do sterowania urządzeniami mobilnymi. Ostatnią przedstawioną grupą, najbardziej uniwersalną ze względu na obszar działania, są metody przepływu optycznego [1]. Polegają one na porównywaniu kolejno pobranych ramek (klatek) obrazu i lokalizowaniu podobieństw między nimi. Metoda ta pozwala dość precyzyjnie określić odległości sterowanego obiektu od zlokalizowanej przeszkody. Ich wadą jest dość duża złożoność obliczeniowa algorytmu, którą da się jednak w pewnym stopniu zniwelować, stosując ograniczenie obszarów opisujących zagrożenie.

W niniejszym opracowaniu wybór metody został podyktowany głównie chęcią zastosowania jej do sterowania bezzałogowym aparatem latającym (UAV). W tym przypadku jedynym rozsądnym wyjściem była metoda przepływu optycznego, która pozwala zarówno zlokalizować przeszkodę jak i ocenić odległość. Dodatkowym atutem była możliwość pracy na obrazie o niewielkiej rozdzielczości i monitorowanie przeszkód w rzeczywistym czasie lotu jednostki UAV.

1. METODA LUCAS – KANADE

Metoda Optical Flow (pola przepływu optycznego) generuje wektor przepływu, który powstaje podczas tworzenia sekwencji obrazu. Aby otrzymać badaną strukturę otoczenia w przestrzeni 3D (wydobycie głębi obrazu) została wykorzystana metoda przepływu optycznego oparta na gradientowej metodzie Lucas-Kanade [1]. Pozwala ona na wygenerowanie rzadkiego pola przepływu pewnych lokalnych informacji pochodzących z określonego sąsiedztwa, które otacza interesujące nas punkty.

W metodzie LK zakłada się, że jasność punktu (x,y) obrazu jest stała w czasie, więc:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (1)$$

gdzie: δx , δy opisują pewne przesunięcia między ramkami obrazu w określonym przedziale czasu. Równanie różniczkowe dla (1) można zapisać w postaci:

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (2)$$

gdzie:

$$\frac{\partial x}{\partial t} = u, \quad \frac{\partial y}{\partial t} = v \quad (3)$$

są składowymi wektora prędkości.

Drugie założenie metody LK mówi, że ruch jest mały w następujących po sobie klatkach. Po rozwinięciu prawej strony równania (1) otrzymamy:

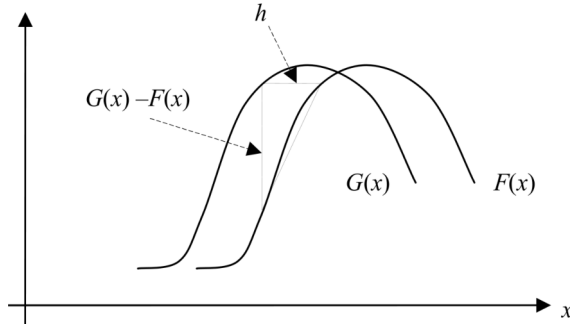
$$I(x, y, t) = I(x, y, t) + (\nabla I)^T (\delta x, \delta y) + \delta t I_t + O^2 \quad (4)$$

gdzie $\nabla I = (I_x, I_y)^T$; I_t - pochodne pierwszego rzędu, O^2 - pochodne wyższych rzędów które można zaniedbać. Po odjęciu od obu stron równania $I(x, y, t)$, pominięciu wyrazu O^2 i podzieleniu przez δt otrzymamy:

$$(\nabla I)^T \vec{v} + I_t = 0 \quad (5)$$

gdzie: $\nabla I = (I_x(x, t), I_y(y, t))^T$ jest przestrzennym gradientem funkcji intensywności, a $\vec{v} = (u, v)^T$ jest wektorem przepływu optycznego. Równanie (5) jest równaniem ograniczenia przepływu optycznego i definiuje pojedyncze, lokalne ograniczenie na wektor prędkości optycznej [6].

Do wyznaczenia wektorów przepływu optycznego użyto iteracyjnej piramidalnej wersji algorytmu Lucas-Kanade.



Rys. 1. Przeszczenie pomiędzy dwiema krzywymi

Mając dane funkcje $F(\vec{x})$ i $G(\vec{x})$, które opisują wartość pikseli w poszczególnych lokacjach obrazu \vec{x} (punkt (x,y)), można wyznaczyć wektor przesunięcia \vec{h} danego regionu zainteresowania. W tym przypadku należy znaleźć różnicę h pomiędzy dwiema krzywymi $F(x)$ i $G(x) = F(x-h)$ jak pokazano na Rys. 1. Dla małego h można zapisać:

$$F'(x) \approx \frac{F(x+h) - F(x)}{h} = \frac{G(x) - F(x)}{h} \quad (6)$$

gdzie:

$$h \approx \frac{G(x) - F(x)}{F'(x)} \quad (7)$$

Dodatkowo, aby znaleźć najmniejszą różnicę pomiędzy krzywymi $F(x)$ i $G(x)$ wyznaczamy:

$$E = \sum_x (F(x+h) - F(x))^2 = 0 \quad (8)$$

Wyliczając pochodną oraz rozwijając równanie (8) względem h otrzymamy:

$$0 = \frac{\partial E}{\partial h} \approx \frac{\partial}{\partial h} \sum_x [F(x) + hF'(x) - G(x)]^2 \quad (9)$$

Po zróżniczkowaniu otrzymamy:

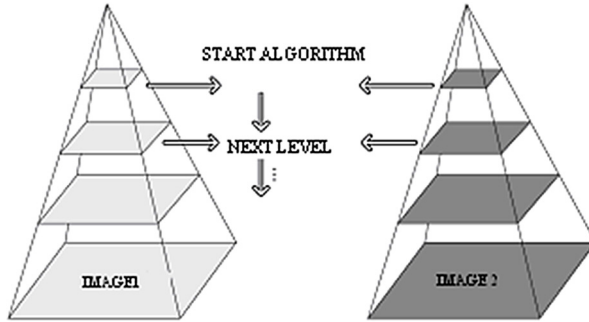
$$\sum_x 2F'(x)[F(x) + hF'(x) - G(x)] \approx 0 \quad (10)$$

Ostatecznie przybliżenie przeszczenia h wynosi :

$$h \approx \frac{\sum_x F'(x)[G(x) - F(x)]}{\sum_x F'(x)^2} \quad (11)$$

Wartość h , dla gradientu znajdującego się na dwóch kolejnych ramkach obrazu, jest wektorem przepływu optycznego tego obszaru. Ponieważ przesunięcie h przyjmuje niewielkie wartości, metoda LK jest metodą lokalną stosowaną do generowania „rzadkiego” pola. Wadą tego typu rozwiązania jest możliwość zgubienia większego ruchu, który może wychodzić poza obszar poszukiwania, czyli nie może być lokalizowany przez algorytm. Problem ten został zniwelowany, dzięki zastosowaniu rozwinięcia algorytmu LK w piramidy. Polega ono na badaniu obrazu

z różną rozdzielczością, zaczynając od najwyższego poziomu piramidy (o najniższej liczbie detali), schodząc do poziomu o największej liczbie detali, czyli dla najwyższej rozdzielczości obrazu, co przedstawia Rys. 2.



Rys. 2. Schemat dekompozycji hierarchicznej do analizy obrazu

2. LOKALIZACJA WEKTORÓW

Działanie algorytmu rozpoczyna się według założonej dekompozycji piramidalnej Gaussa, od badania obrazu o najniższej rozdzielczości. Przyjętych zostało $Lm = 4$ poziomów piramidy o rozmiarach $L_0 = 320 \times 240$, $L_1 = 160 \times 120$, $L_2 = 80 \times 60$, $L_3 = 40 \times 30$ pikseli. Dla każdego z obrazów wykonywane są operacje mające na celu zlokalizowanie wektorów przepływu optycznego \vec{v} .

W tym celu należy skorzystać ze wzoru (5) uwzględniając warunek na jak najmniejszy moduł wektora (8). Przyjmujemy, że ∇I jest wyliczane z drugiego obrazu $F(x)$, gdzie:

$$\nabla I = \begin{bmatrix} F_x \\ F_y \end{bmatrix} \begin{bmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial y} \end{bmatrix}^T \quad (12)$$

gdzie po podstawieniu do (4) mamy:

$$\sum_x \begin{pmatrix} F_x^2 & F_x F_y \\ F_x F_y & F_y^2 \end{pmatrix} \vec{v} - \begin{pmatrix} \delta I(x) F_x \\ \delta I(x) F_y \end{pmatrix} = 0 \quad (13)$$

Oznaczamy:

$$G = \sum_x \begin{pmatrix} F_x^2 & F_x F_y \\ F_x F_y & F_y^2 \end{pmatrix}, \quad \vec{b} = \sum_x \begin{pmatrix} \delta I(x) F_x \\ \delta I(x) F_y \end{pmatrix}$$

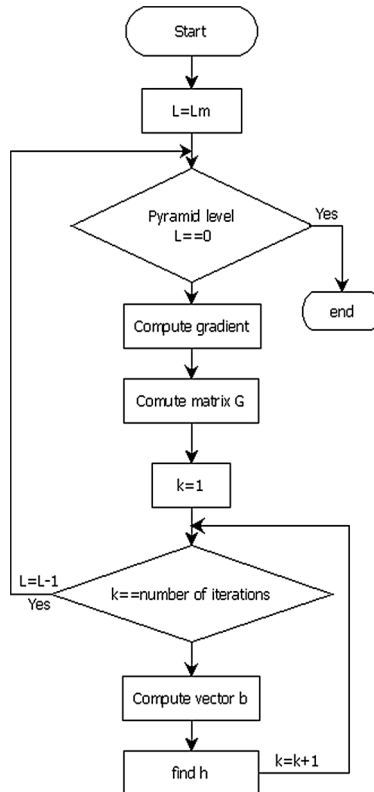
Zatem optymalny wektor przepływu optycznego wynosi:

$$\vec{v} = G^{-1} \vec{b} \quad (14)$$

Macierz G wyznaczamy tylko raz dla danego poziomu piramidy, ponieważ wersja piramidalna algorytmu zakłada, że przesunięcia pomiędzy dwoma obrazami są bardzo małe. Aby zwiększyć dokładność wyznaczania przemieszczenia h ze wzoru (9) potrzebna jest estymacja szerszej liczby wyników. Przeprowadzana w związku z tym jest większa liczba iteracji.

$$h_{k+1} = h_k + \frac{\sum_x w(x) F'(x+h_k) [G(x) - F(x+h_k)]}{\sum_x w(x) F'(x+h_k)^2} \quad (15)$$

W pętli iteracyjnej obliczana jest wartość wektora \bar{b} , aż do momentu uzyskania wyniku o ustalonej dokładności lub przekroczenia liczby przewidzianych powtórzeń. Ten krok algorytmu jest najbardziej złożony obliczeniowo, mimo że wprowadzono pewną optymalizację. Polega ona na sumowaniu wartości najpierw w kolumnach a następnie po wierszach. Otrzymany w tym przypadku dodatkowy błąd jest niewielki ze względu na dużą lokalność obliczeń. Algorytm działania omówionej aplikacji przedstawiony został na Rys. 3.

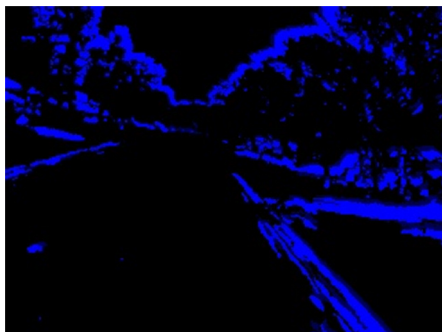


Rys. 3. Schemat działania metody LK

Efekt działania algorytmu przedstawiony został dla obrazu zapisanego w czasie lotu jednostki UAV w formacie avi o rozdzielczości 320x240 pikseli. Na Rys. 4 przedstawione zostały dwie wybrane ramki obrazu oraz na Rys. 5 efekt działania algorytmu Lucas-Kanade. Obszary opisujące ruch na obrazie zostały oznaczone kolorem niebieskim.



Rys. 4. Dwie przykładowe ramki obrazu

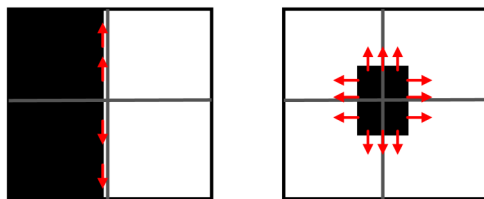


Rys. 5. Efekt działania algorytmu LK przedstawiający różnice pomiędzy obrazami z Rys. 4.

3. OPIS PRZESZKODY

Dokładności z jakimi wykreślane są wektory przepływu optycznego zależą przede wszystkim od kierunku lotu lub wykonywanych manewrów UAV. W przypadku lotu w linii prostej, wektory układają się jak na Rys. 6. Zlokalizowanie przeszkody możliwe jest, dzięki rozpoznaniu wektorów tworzących lustrzane odbicie względem osi symetrii środka obrazu. To podejście sprawdza się jedynie i wyłącznie w idealnych modelach przeszkód. W rzeczywistości należy przeprowadzić estymację wyników współrzędnych zlokalizowanych wektorów w poszczególnych częściach obrazu.

W przypadku manewrów UAV (Rys. 7) kierunek wektorów opisujących przeszkody przedstawiony jest zgodnie z wykonywanym ruchem kamery. Dla części obrazu, w kierunku której odbywa się manewr, np. prawo (Rys. 7a) wektory skierowane są wzdłuż poziomej osi środka obrazu. Podobna sytuacja zachodzi, gdy skręcamy w inną stronę wektory skierowane są w kierunku odpowiedniej osi przechodzącej przez centrum obrazu.



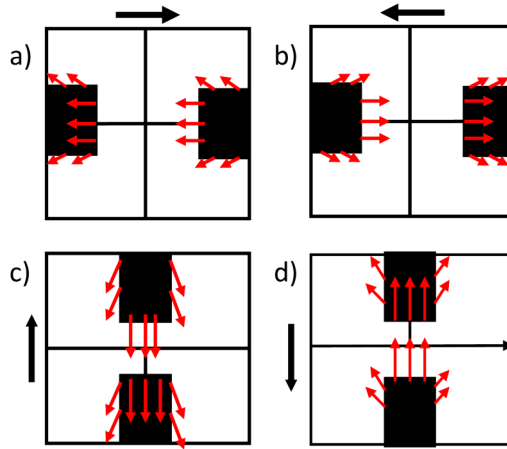
Rys. 6. UAV porusza się ruchem liniowym nie zmieniając trajektorii

Znając zachowanie parametrów wektorów przepływu optycznego, dla poszczególnych przypadków lokalizacji przeszkód na obrazie, można ustalić z pewną dokładnością ich obecność. Sprawą najbardziej istotną jest to, czy dana przeszkoda stanowi zagrożenie dla lecącej jednostki UAV. Problem ten rozwiązany został w trzech etapach.

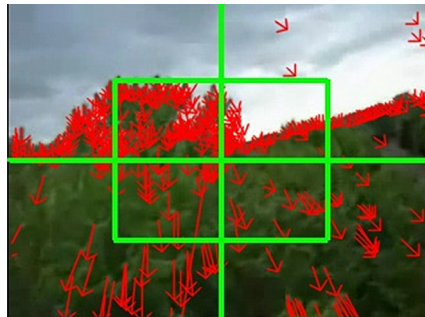
A. Lokalizacja przeszkód będących zagrożeniem poprzez wyodrębnienie obszarów zawierających istotne wektory przepływu optycznego

Pierwszy etap rozwiązany został w bardzo prosty sposób, który polega na wyodrębnieniu wektorów skierowanych jedynie do określonego, w pewnym przedziale, centrum obrazu. Wykorzystane zostały współrzędne wektorów (opisujące prostą przechodzącą przez dwa punkty) oraz współrzędne prostej poziomej, przechodzącej przez środek obrazu. Mając te dane wyznaczony został punkt wspólny obu prostych. Następnie wybrane zostały tylko te punkty, które

znajdują się w przedziale oddalonym o 20 pikseli od centrum obrazu. Pozwoliło to na wyodrębnienie wektorów przypisanych tym punktom.



Rys. 7. Zachowanie się UAV w czasie manewrów



Rys. 8. Wykreślone wektory przepływu optycznego

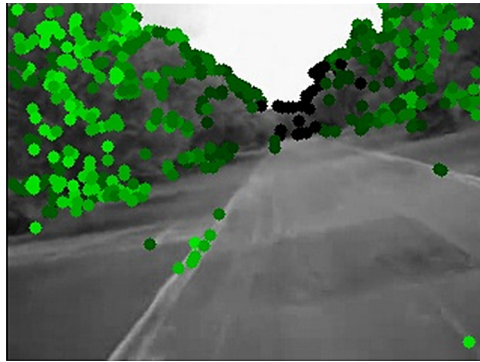
Ograniczenie liczby wektorów branych pod uwagę w działaniu algorytmu, znacznie przyspieszyło jego pracę. Przykładowo wyodrębnione wektory (z wyników otrzymanych na Rys. 8) dla określonych klatek obrazu przedstawione zostały na Rys. 9.



Rys. 9. Lokalizacja wektorów opisujących przeszkodę

B. Wnioskowanie odległości od przeszkody, dzięki parametrom długości wektora.

Kolejnym etapem jest wnioskowanie odległości przeszkody od jednostki UAV. W tym celu ponownie wykorzystano współrzędne wektorów przepływu optycznego, co pozwoliło na obliczenie ich modułu. Aby zwiększyć szybkość działania aplikacji należy ograniczyć tablicowanie danych. W tym celu płynnie porównywane są wyniki ze średnią wartością modułów lokalizowanych wektorów. Wartość średnia wyliczana jest na podstawie 20 pierwszych wektorów, a następnie uwzględnia moduły kolejnych. Kalkulacja odległości sprowadza się do określenia grupy, w której znajdują się dane wektory widziane na obrazie. Na Rys. 10 przedstawione zostało działanie algorytmu. Różnymi odcieniami zieleni oznaczono obszary o podobnych parametrach, dzięki czemu możliwe staje się wydobywanie głębi obrazu (przestrzeni 3D). Barwa jasna opisuje obiekty znajdujące się bliżej kamery. Im ciemniejszy kolor, tym przeszkody są bardziej oddalone od lecącego UAV.

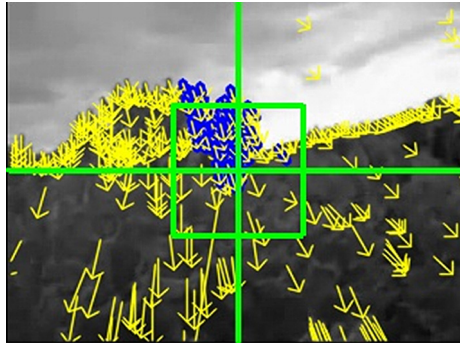


Rys. 10. Wnioskowanie odległości

Przy pomocy metody Optical Flow można najdokładniej odwzorować odległość w przypadku UAV poruszającego się ruchem liniowym, niewykonującym manewrów. Dodatkowo mając parametry odnośnie prędkości poruszania się UAV jest możliwość wyznaczenia zależności między prędkością, a długością wektora przepływu. Zależy ona od rozdzielczości obrazu. Im większa rozdzielczość, tym dokładność pomiaru się zwiększa. Wykozystanie jednak dodatkowej zależności jaką jest prędkość poruszania się samolotu jest bardzo kłopotliwe w implementacji. To właśnie z tego powodu moduły wektorów porównywane są bezpośrednio ze średnią wartością wszystkich wektorów. Pozwala to na traktowanie wektora jako jednego z elementów złożonego obrazu, a nie pojedynczego obiektu.

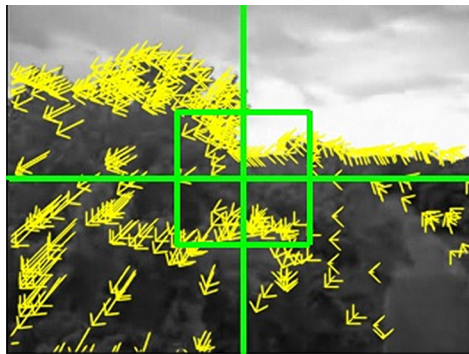
C. Decyzja o istnieniu zagrożenia.

Ostatnim etapem jest podjęcie decyzji o tym czy istnieje zagrożenie kolizji UAV z przeszkodą. Jest to zagadnienie, które łączy dwa wcześniej omówione etapy. W zależności od charakteru lotu UAV implementowany jest odpowiedni ciąg programu. W momencie, gdy generowany jest nowy punkt poza dotychczasową trajektorią lotu UAV (wykonywany odpowiedni manewr), uruchamiana jest procedura lokalizacji wektorów zbiegających do określonego w pewnym przedziale centrum obrazu. Natomiast, gdy UAV nie zmienia swojej trajektorii wykorzystany został algorytm wnioskowania odległości i tworzenia mapy przestrzennej obrazu. Jeśli obszar będący centrum obrazu będzie posiadał wektory o dużym module, to stanie się informacją o bliskiej obecności przeszkody. Przykład lokalizacji przeszkody przedstawiony został na Rys. 11. Zlokalizowane wektory przepływu optycznego ze względu na dużą wartość swego modułu przedstawiają obiekt znajdujący się w niebezpiecznej odległości od UAV.

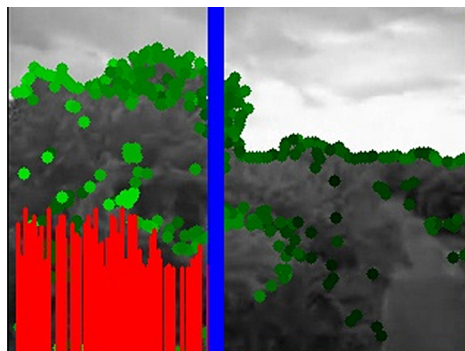


Rys. 11. Lokalizacja przeszkody

Obszar wolny od przeszkód wyznaczany jest na podstawie histogramu opisującego poszczególne moduły wektorów znajdujących się w danych kolumnach i wierszach obrazu. Odpowiednie progowanie badanego histogramu pozwala na zlokalizowanie grup wektorów o podobnych cechach (Rys. 12 - 13). Mając te informacje możliwe staje się określenie bezkolizyjnej trajektorii lotu jednostki UAV.



Rys. 12. Testowane wektory

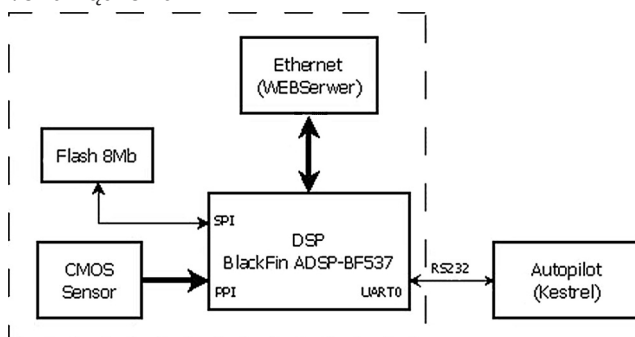


Rys. 13. Ocena odległości od przeszkody

Podjęcie decyzji o zmianie kierunku lotu wykonywane jest z pewnym opóźnieniem. W celu zniwelowania błędów badanych jest pięć kolejnych ramek obrazu, z których cztery mają wskazywać na obecność przeszkody przed lecącym UAV. Gdy dany warunek jest spełniony wysyłany jest sygnał do autopilota samolotu informujący o konieczności wykonania określonego manewru.

4. ROZWIĄZANIE SPRZĘTOWE

Prezentowany algorytm jest implementowany na urządzeniu wyposażonym w mikroprocesor Blackfin ADSP-BF537. Aplikacja działa pod kontrolą systemu uClinux wspierającą architekturę wybranego mikrokontrolera. Na urządzeniu zainstalowany został sterownik pamięci masowej Flash, sterownik obsługi plików JFFS2, sterownik interfejsu sieciowego Ethernet, serwer SSH oraz badany algorytm wizyjny. Działanie aplikacji na urządzeniu oraz podgląd i modyfikację niektórych parametrów w trybie działania programu można wykonać poprzez utworzony webservice urządzenia.

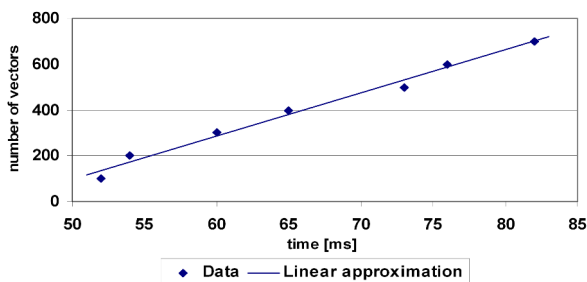


Rys. 14. Hardware diagram

Do komunikacji wizyjnego komputera pokładowego z autopilotem wykorzystane zostało złącze szeregowo RS232. Wypracowane na procesorze Blackfin sygnały decyzyjne trafiają w postaci odpowiednich ramek danych do autopilota, w celu przetworzenia ich na sygnały sterujące serwami i ruchem samolotu w określonym kierunku. Diagram przedstawiający hardware przedstawiony został na Rys. 14.

5. WYNIKI SYMULACYJNE

Sprawdzony został czas wykonywania obliczeń dla każdej ramki obrazu (Rys. 15). Pod uwagę brana była rozdzielczość oraz ilość lokalizowanych pikseli. Aby nie obciążać procesora dane odczytywane były z portu RS232, bez użycia webserwera urządzenia.

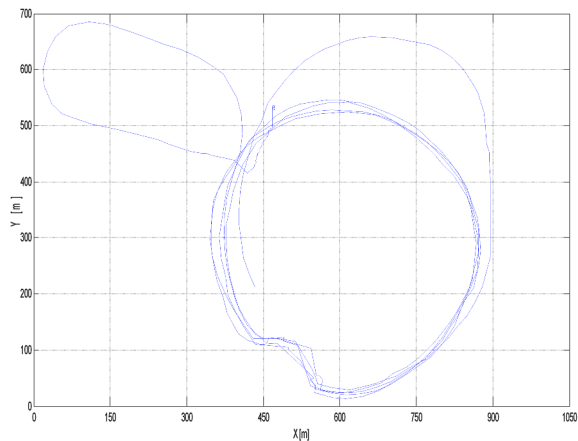


Rys. 15. Czas obliczeń dla danej liczby lokalizowanych pikseli

Największy wpływ na szybkość działania aplikacji ma rozdzielczość obrazu, która znacznie spowalnia obliczenia. Rozdzielczość, z jakiej korzystano, to 320x240 pikseli, gdzie wystarczająca

ilość lokalizowanych wektorów wynosi 300. Jak widać na wykresie czas wykonywania obliczeń dla jednej ramki obrazu wynosi około 60ms. Taki wynik pozwala na częstotliwość odświeżania obrazu rzędu 15 ramek/sek. Przy uwzględnieniu, że algorytm podejmuje decyzję o zmianie kierunku lotu na podstawie pięciu kolejnych ramek obrazu, to czas opóźnienia wynosi ok. 0.3s. Badane przez nas obiekty poruszają się z prędkością w granicach 17m/s. Podsumowując, aby przeszkoda została ominięta musi zostać zlokalizowana w odległości większej niż osiem metrów od lecącego samolotu.

Przeprowadzone zostały badania symulacyjne przy użyciu oprogramowania Virtualkokpit udostępnionego przez producenta autopilota kestrel. Ustalono trajektorię lotu samolotu jako okrąg o zadanym promieniu. Symulacja przeszkody polegała na załączeniu algorytmu wizyjnego w ustalonym momencie lotu jednostki UAV. Korzystano z zapisanych w pamięci flash urządzenia, kilkunastu klatek obrazu opisujących zbliżającą się przeszkodę. Przeprowadzone doświadczenie uwidacznia głównie prawidłowe działanie komunikacji między wykorzystywanymi urządzeniami. Wyniki wygenerowanej trajektorii lotu przedstawione zostały na Rys. 16.



Rys. 16. Symulacja trajektorii lotu

WNIOSKI

W artykule przedstawiony został algorytm rozpoznawania przeszkód wykorzystujący obraz pobrany z jednej kamery. Aplikacja zaimplementowana została na specjalnie zaprojektowanej platformie sprzętowej, której wymiary pozwalają na umieszczenie jej na UAV. Uwzględniono także moc obliczeniową zastosowanego urządzenia. Musi ono pozwalać na zastosowanie piramidalnej reprezentacji metody Lucas Kanade. Algorytm wymaga określonej ilości pamięci operacyjnej na zapis dwóch porównywanych ramek obrazu oraz jak największej mocy obliczeniowej mikroprocesora.

Napisana aplikacja spisuje się dobrze, gdy mamy do czynienia z pojedynczą przeszkodą bądź lotem w kanionie ulic. Problemy mogą pojawić się w przypadku licznych przeszkód o bardzo jednolitych teksturach np. wysokie budynki o jednolitej barwie. Wymogiem prawidłowego działania jest także zachowanie odpowiednich warunków atmosferycznych (opady, oświetlenie). Kolejną wadą tym razem wykorzystanej metody LK jest brak możliwości użycia stabilizacji cyfrowej. Omówione problemy można w pewnym stopniu rozwiązać dzięki zastosowaniu bardziej zaawansowanej optyki.

Pracę wykonano w ramach realizacji projektu rozwojowego nr OR00002911 "SIECIOCENTRYCZNY SYSTEM WSPARCIA ROZPOZNANIA I DOWODZENIA SYTUACJAMI KRYZYSOWYMI NA TERENACH ZURBANIZOWANYCH Z AUTONOMICZNYMI BEZZAŁOGOWYMI APARATAMI LATAJĄCYMI" finansowanego ze środków Ministerstwa Nauki i Szkolnictwa Wyższego.

BIBLIOGRAFIA

- [1] B. D. Lucas and T. Kanade. An iterative image- registration technique with an application to stereo vision. Proceedings of the 1981 DARPA Imaging Understanding Workshop, pp. 121-130, 1981.
- [2] G. Bradski and A. Kaehler. Learning OpenCV. O'Reilly Media, Inc., 2008.
- [3] I. Ulrich and I. R. Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," in AAAI/IAAI, 2000
- [4] P. C. Merrell, D.-J. Lee, and R. W. Beard, "Obstacle avoidance for unmanned air vehicles using optical flow probability distributions," Mobile Robots XVII, Vol. 5609, No. 1, pp. 13-22, 2004.
- [5] W. S. P. Fernando, L. Udawatta, P. Puthirana, „Identification of Moving Obstacles with Pyramidal Lucas Kanade Optical Flow and k means Clustering”, 2007
- [6] S.S. Beauchemin and J.L. Barron. The computation of optical flow. ACM Computing Surveys, 27(3):433 – 467, 1995.
- [7] A. Giachetti, M. Campani, and V. Torre, "The Use of Optical Flow for Road Navigation," IEEE Transactions. on Robotics and Automation., vol. 14, Issue.1, pp. 34-48, February 1998
- [8] D. Lieb, A. Lookingbill, and S. Thrun, "Adaptive Road Following using Self-Supervised Learning and Reverse Optical Flow," Stanford Artificial Intelligence Laboratory, Stanford University, 2005.
- [9] B. K. P. Horn, B. G. Schunck, "Determining Optical Flow," Artificial Intelligence Laboratory, MIT, Cambridge, MA, pp. 185-203, 1981

LESZEK AMBROZIAK, JAKUB CIEŚLUK, ZDZISŁAW GOSIEWSKI

OBSTACLE IDENTIFYING BY UNMANNED AERIAL VEHICLE WITH USING ONE CAMERA

Abstract

A problem of obstacles and collisions avoiding by unmanned aerial vehicles (UAVs), using the information provided from an image taken from one camera is presented in the paper. The optical flow method based on gradient method of Lukas-Kanade was used to obtain the tested structure in 3D space environment (extraction of image's depth). This method allow to generate local information flow patterns from particular neighborhood of surrounding points. Presented algorithm for obstacle location and estimation of its shape was developed and implemented on the workstation with real time uClinux system. Proposed vision based obstacle avoidance algorithm are demonstrated in simulation and in hardware in the loop flight tests on a fixed-wing UAV.