

SYMULACJA I TESTOWANIE ALGORYTMÓW WYKRYWANIA KRAWĘDZI NA POTRZEBY LOTU WŚRÓD KANIONU ULICY

Mariusz BOGDAN*

*Katedra Automatyki i Robotyki, Wydział Mechaniczny, Politechnika Białostocka, ul. Wiejska 45 C, 15-351 Białystok

m.bogdan@pb.edu.pl

Streszczenie: W artykule testowano i analizowano metody wykrywania krawędzi w środowisku programowym Matlab'a. Skupiono się głównie na metodach gradientowych. W celu uproszczenia informacji zawartej w zarejestrowanych obrazach (klatkach) zaproponowano, także opis analityczny wykrywanych krawędzi – przeszkód (transformata Hough'a).

1. WSTĘP

Obraz wideo (sekwencja obrazów) dostarcza ogromnej ilości informacji. Ilość ta jest zbyt duża, aby system wizyjny w pełni radził sobie z przetwarzaniem wszystkich informacji. W miarę wzrostu rozdzielczości przetworników, gwałtownie rośnie strumień dostępnych danych, a co się z tym wiąże ilość potencjalnych operacji. Konfiguracja sprzętowa mikrosamolotu (głównie miniaturyzacja oraz ograniczona ilość energii zasilającej układy) wymusza pewną konstrukcję systemu wizyjnego (poważne wyzwanie technologiczne). Wszystkie montowane systemy muszą odznaczać się bardzo wysoką skalą integracji, a wiele z nich musi stanowić integralne elementy struktury mikrosamolotu (Sibiński, 2009). Przykładem takiego systemu może być wprowadzenie tak zwanej Aktywnej Wizji (AW, ang. Active Vision). Ideą AW jest selektywne ograniczenie analizowanych obszarów do tych, które są w danej chwili interesujące (np. charakterystyczne punkty otoczenia). Oczywiście informacja o otoczeniu musi być pozyskana w możliwie najszybszy sposób, a także powinna posiadać odpowiednią reprezentację (właściwą formę dla układu podejmującego decyzję). Strumień wideo w postaci nieprzetworzonej nie nadaje się do tego celu – przede wszystkim ze względu na ilość danych. W celu umożliwienia przetwarzania i analizy obrazów w czasie rzeczywistym na platformie mikrosamolotu, wydobycia parametrów wpływających na jego lot, należy zmienić reprezentację danych z rastrowego (gdzie jeden piksel odpowiada uśrednionej jasności danego obszaru obserwowanego przez kamerę) na wektorowy – tj. strukturalny i niezależny od skali opis obserwowanej sceny. Przy czym, i należy to podkreślić, nieodzownym jest uproszczenie procesu przetwarzania, które może polegać na wektoryzacji jedynie obserwowanych krawędzi. Takie podejście jest powszechnie stosowane w wielu systemach wizyjnych ze względu na to, że poprawne odwzorowanie krawędzi pozwala na wydobycie najistotniejszych informacji o kształtach obserwowanych obiektów, przeszkód itp.

2. ALIZA I PRZETWARZANIE OBRAZÓW W MATLABIE

Obraz można traktować jako dwuwymiarową funkcję $L(x, y)$ intensywności oświetlenia, której wartość L (amplituda) we współrzędnych (x, y) określa intensywność (jasność) obrazu w tym punkcie. Ponieważ światło jest formą energii, dlatego wartość funkcji jest nieujemna i skończona. Wartość funkcji intensywności zależy od iluminacji i odbicia (Wiatr, 2002):

$$L(x, y) = i(x, y) r(x, y) \quad (1)$$

gdzie: $i(x, y)$ – iluminacja w punkcie (x, y) , $r(x, y)$ – odbicie w punkcie (x, y) .

W celu zmniejszenia złożoności obliczeń, a tym samym ich przyspieszenia, do badań wykorzystywano jedynie obraz monochromatyczny, powstały na wskutek przekształcenia-uproszczenia obrazu barwnego RGB. Intensywność obrazu monochromatycznego wyrażona jest w poziomie szarości L (ang. *grey level*) a zakres jej zawiera się w granicach $[L_{min}, L_{max}]$. Zakres poziomu szarości wynika z własności rzeczywistego toru wizyjnego, a w szczególności z zakresu zmian sygnału z kamery CCD, wzmocnienia tego sygnału i parametrów przetwornika wizyjnego. W programie Matlab wartości $L_{min} = 0$ odpowiada poziomowi czerni natomiast $L_{max} = 255$ poziomowi bieli. Dla potrzeb cyfrowego przetwarzania obraz jest zatem skwantowany zarówno ze względu na współrzędne, jak również na tak zdefiniowane poziomy szarości. Obraz ten można przedstawić jako tablicę o rozmiarach: liczba wiersz w , liczba kolumn k (N_w, N_k):

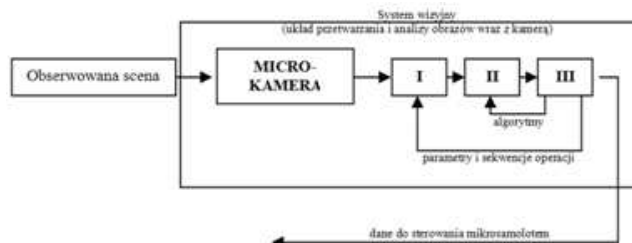
$$L(x, y) = \begin{bmatrix} L(1, 1) & L(1, 2) & \dots & L(1, N_k) \\ L(2, 1) & L(2, 2) & \dots & L(2, N_k) \\ \dots & \dots & \dots & \dots \\ L(N_w, 1) & L(N_w, 2) & \dots & L(N_w, N_k) \end{bmatrix} \quad (2)$$

Każdy z elementów tablicy (piksel) przechowuje skwantowany poziom szarości (z zakresu od 0 do $2^B - 1$). Dla potrzeb niniejszego zastosowania tj. przetwarzania i analizy obrazów w czasie rzeczywistym, przyjęto 8-bitową roz-

dzielczość poziomów szarości: $B=8$ (tzn. piksele w sygnale wizyjnym zmieniają się w zakresie 0 – 255). Jest to też najczęściej stosowana rozdzielczość, która w zupełności wystarczy do przetwarzania obrazów przeznaczonych do wspomnienia sterowania mikrosamolotu w czasie rzeczywistym. Źródłem obrazu wizyjnego we współczesnych systemach sensorycznych jest głównie przetwornik CCD (Wiatr, 2002; Piszczek i Rutyna, 2007).

W algorytmach analizy obrazów można wyróżnić kilka poziomów przetwarzania i analizy obrazu. W zależności od czasu przetwarzania na docelowym, dedykowanym układzie wizyjnym zamontowanym na mikrosamolocie testowane będą trzy poziomy przetwarzania i analizy obrazów (Rys. 1):

- (I) Najniższy poziom przetwarzania obrazów. Wstępnie przetwarza sygnał wizyjny (ang. *preprocessing*). Cel: eliminacja zakłóceń, detekcja krawędzi, ustalenie poziomów szarości pikseli należących do obiektu (np. przeszkody na podstawie histogramu, równoważenie histogramu itp.).
- (II) Średni poziom analizy obrazów. Obejmuje on segmentację obrazu, lokalizację obiektów, rozpoznanie kształtu obiektu i wyróżnienie cech charakterystycznych tego kształtu.
- (III) Najwyższy poziom. Przeprowadza analizę złożonej sceny w sensie analizy ruchu obiektu i na bieżąco steruje obiektem oraz pozostałymi poziomami systemu wizyjnego (zadaje algorytmy oraz parametry i sekwencje operacji).



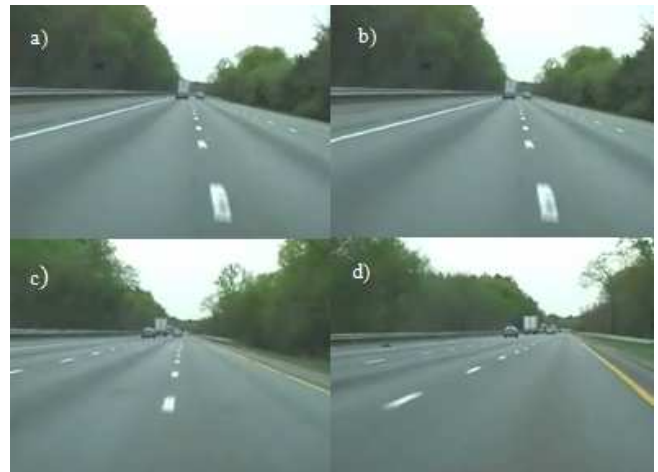
Rys. 1. Poziomy przetwarzania i analizy obrazów systemu wizyjnego czasu rzeczywistego (Wiatr, 2002)

Przedstawiona na Rys. 1 struktura systemu wizyjnego ukazuje ponadto występujące powiązania (sprzężenia) pomiędzy różnymi poziomami systemu wizyjnego. Wyniki (II) i (III) etapu przetwarzania obrazu mogą mieć wpływ na kolejność wykonywanych operacji na (I) najniższym poziomie oraz na algorytmy przetwarzania na (II) poziomie. Przetwarzania i analizy sekwencji obrazów dokonano w Matlabie (schemat blokowy w simulinku, oraz w m-plikach). Dane wejściowe to plik wideo pochodzący z kamery zamontowanej na samochodzie (docelowo strumień wideo rejestrowany za pomocą kamery zamontowanej na samolocie). Rozmiar to 640x480 (30 FPS). Zakłada się także możliwość redukcji liczby klatek na sekundę, lub rozdzielczości, tak by późniejsza implementacja działała w czasie rzeczywistym. Ponadto algorytm ma działać w oparciu jedynie o sąsiednie klatki (podejście bezpośrednie – brak wyliczania konkretnych cech). Zbadano także możli-

wość wykorzystania transformaty Hougha w celu określenia (wyznaczenia) bezkolizyjnego lotu wśród kanionu ulicy.

3. OBRAZY WIDEO PODDAWANE TESTOWANIU I ANALIZIE

Testowaniu i analizie poddano plik wideo (Tab. 1). Przedstawia on sekwencję obrazów zarejestrowanych za pomocą kamery umieszczonej na samochodzie (Rys. 2). Ruch kamery odbywa się równolegle do powierzchni drogi. Opisany powyżej przypadek posłuży do opracowania i testowania uniwersalnej metody wspomaganie lotu poprzez wykorzystanie informacji zawartej w obrazach (strumień wideo), rejestrowanych za pomocą mikrokamery umieszczonej na mikrosamolocie - nawigacja mikrosamolotu podczas lotu wśród budynków – „kanionu ulicy”.



Rys. 2. Wybrane klatki z analizowanego wideo

Tab. 1. Parametry wideo

Ustawienia wideo	Plik_2
Kodek	mpeg4
Rozdzielczość	320x240
Bitrate(bitów/sek)	768
FPS (klatek/sek)	30
czas trwania (sek)	22

Badania symulacyjne przeprowadzono zarówno przy użyciu skryptów pisanych w Matlabie (m-plików) oraz w Simulinku w Video and Image Processing Blockset (schematy blokowe). Video and Image Processing Blockset zapewnia możliwość szybkiego projektowania, symulacji oraz weryfikacji algorytmów i systemów przetwarzania obrazów. Z punktu widzenia uniwersalności modelowania współpracuje on z Real-Time Workshop (automatyczna generacja kodu ANSI/ISO C).

Ponadto tworzenie kodu zarówno w środowisku Simulinka oraz Matalaba (m-pliki) ułatwić może znacznie proces implementacji opracowanych algorytmów na docelowy układ przetwarzania i analizy obrazów na potrzeby nawigacji mikrosamolotu (testowanie oraz modyfikacje). W przypadku m-plików istnieje dodatkowo możliwość częściowego wykorzystania kodu. Ze względu na charakter przetwa-

rzanej informacji parametry symulacji określono na: fixed-step (stało-krokową) oraz discrete - no continuous state (dyskretną).

3.1. Wykrywanie krawędzi

Detekcja krawędzi stanowi istotny element w operacjach przygotowawczych do procesu głębszej analizy obrazu. Przekształcony obraz do postaci wyróżnionych krawędzi znacznie ułatwia logiczną interpretację obrazu (Piszczek i Rutyna, 2007; Tadeusiewicz, 1997). Celem tego przekształcenia jest znalezienie granic obiektów (przeszkód) znajdujących się na scenie (Rys. 3).

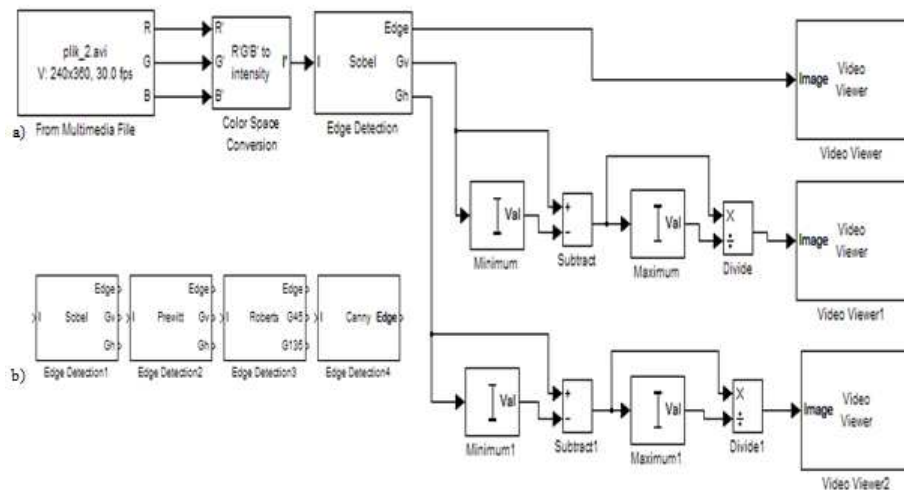
Na schemacie z Rys. 3a można wyróżnić następujące kroki:

1. Blok („From multimedia file”) - wczytany plik video z katalogu, typ danych wyjściowych - dane typu double.
2. Sekwencja obrazów barwnych (model RGB) przekształcona zostaje na obrazy intensywności (odcieni szarości - obrazy monochromatyczne) wg wzoru: $0.2989 * \text{rgb_img}(:, :, 1, n) + 0.5870 * \text{rgb_img}(:, :, 2, n) + 0.1140 * \text{rgb_img}(:, :, 3, n)$; n – poszczególne klatki – dla danego pliku $0.2989 * \text{plik_2}(:, :, 1, n) + 0.5870 * \text{plik_2}(:, :, 2, n) + 0.1140 * \text{plik_2}(:, :, 3, n)$;

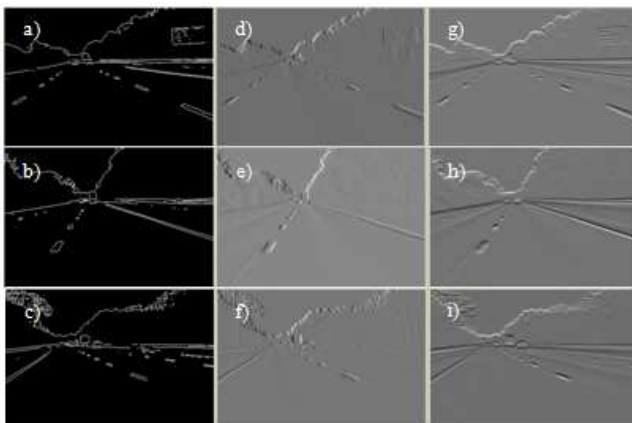
Do wykrywania krawędzi obiektów wykorzystano blok „Edge Detection” (Rys. 3b). Blok ten znajduje piksele w miejscach, w których wielkość gradientu intensywności jest większa niż wartość progowa. Miejsca te zazwyczaj występują na granicach obiektów.

Parametry bloku wykrywania krawędzi metodą Sobel’a:

- Output type - Binary image and gradient components - jako dane wyjściowe zwracany jest obraz binarny (0 – tło, 1 – obiekt – krawędź) oraz komponenty (części) gradientu (Gv – w kierunku pionowym i Gh – w kierunku poziomym; wartości są tego samego typu jak w przypadku macierzy wejściowej); wymiar macierzy Gv i Gh oraz Edge jest identyczny jak macierzy wejściowej;
- Select the Edge thinning check box - zmniejszenie grubości krawędzi na obrazie wyjściowym. Opcja ta wymaga dodatkowego czasu przetwarzania i zasobów pamięci.



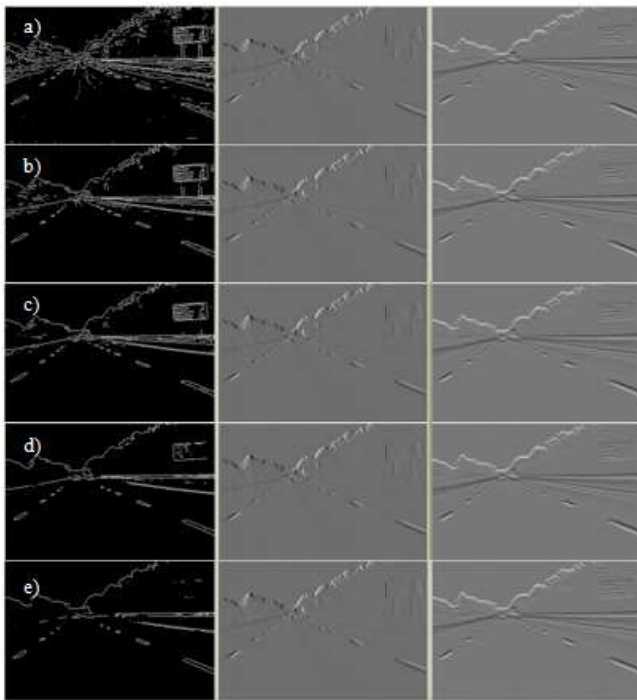
Rys. 3. Schemat blokowy wykrywania krawędzi (a); b) bloki detekcji krawędzi metodą gradientową: Sobel’a, Prewitt’a, Roberts’a, Canny’ego (MATLAB Users’ Guide)



Rys. 4. Detekcja krawędzi metodą Sobel’a. Czas trwania symulacji: początek symulacji – a), d), g); 5 sekund – b), e), h); 10 sekund – c), f), i). Wyniki dla macierzy: „Edge” – a), b), c); Gv – d), e), f); Gh – g), h), i)

Na Rys. 4 pokazano rezultaty wykrywania krawędzi metodą Sobel’a. Natomiast na Rys. 5 przedstawiono wyniki symulacji dla różnych wartości mnożnika; parametru służącego do automatycznego, przy użyciu średniej wielkości kwadratu gradientu obrazu, wyznaczania wartości proggu odcięcia.

Na podstawie badań symulacyjnych określono zalety i wady stosowania metod gradientowych do wykrywania krawędzi. Do niewątpliwych zalet należy zaliczyć brak uwydatnienia zakłóceń (przeciwieństwo filtracji górnoprzepustowej). Jednak w obrazach o małej kontrastowości mogą pojawić się kłopoty z interpretacją wyników.



Rys. 5. Zmiana wartości mnożnika: a) 0.1; b) 0.5; c) 1; d) 4; e) 10

4. TRANSFORMACJA HOUGH'A

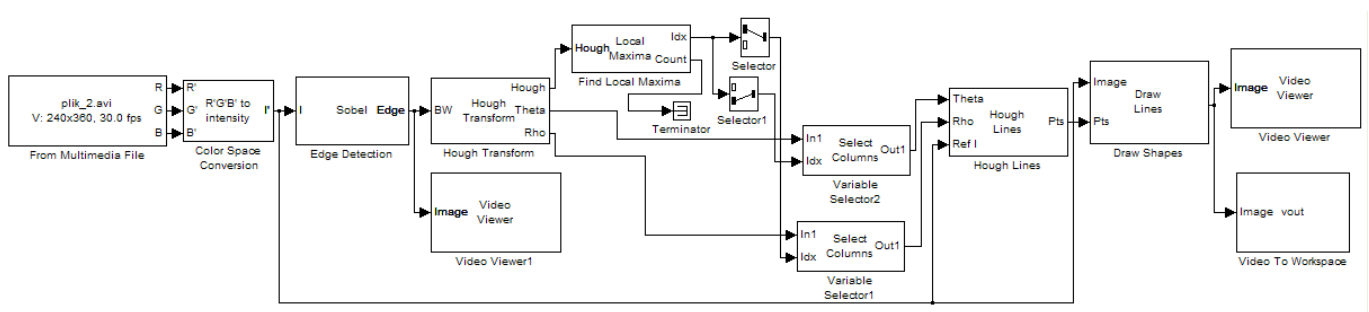
Transformacja Hougha jest metodą detekcji współliniowych punktów leżących na prostej. Każda prosta może być jednoznacznie przedstawiona za pomocą dwóch parametrów. Przestrzeń tych parametrów nosi nazwę przestrzeni Hougha. Najczęściej wykorzystywanymi parametrami są współczynniki rho, theta opisujące równanie prostej w postaci normalnej (Doros, 2003; Kyewook, 2006, Li i inni, 2004; Yu i Jain, 1997):

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (3)$$

gdzie: rho – promień wodzący, theta – kąt tworzony przez rho z osią OX.

Do własności transformaty Hougha należy zaliczyć (MATLAB Users' Guide; Kyewook, 2006):

- prostej w przestrzeni kartezjańskiej odpowiada punkt w przestrzeni Hougha;
- punktowi w przestrzeni kartezjańskiej odpowiada krzywa sinusoidalna w przestrzeni Hougha;
- punkty leżące na tej samej prostej (w przestrzeni kartezjańskiej) korespondują z sinusoidami przechodzącymi przez wspólny punkt w przestrzeni Hougha.



Rys. 6. Schemat blokowy do wyznaczenia wartości prostej dla każdej klatki filmu

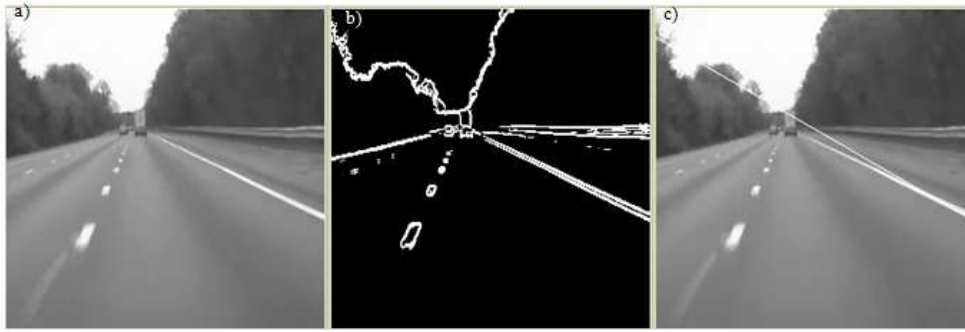
Metoda wyliczania transformaty Hougha składa się z następujących kroków:

- przez każdy badany (różny od zera) punkt obrazu prowadzony jest pęk prostych przechodzących przez ten punkt;
- każda z tych prostych transformowana jest do przestrzeni Hougha i tworzy tam punkt o współrzędnych rho, theta w ten sposób, że każdy punkt obrazu pierwotnego (pęk prostych) jest odwzorowany w sinusoidalną krzywą w przestrzeni Hougha;
- przestrzeń Hougha jest przestrzenią akumulacyjną tzn. punkty sinusoidalnych krzywych, wygenerowanych dla różnych punktów obrazu pierwotnego, dodają się w miejscach, w których krzywe te przecinają się. Powstałe w ten sposób (w przestrzeni Hougha) maksima odpowiadają zbiorom punktów, należących do jednej prostej. Wartość maksimum odpowiada ilości współliniowych punktów;
- współrzędne rho, theta tego maksimum jednoznacznie określają położenie prostej na obrazie pierwotnym.

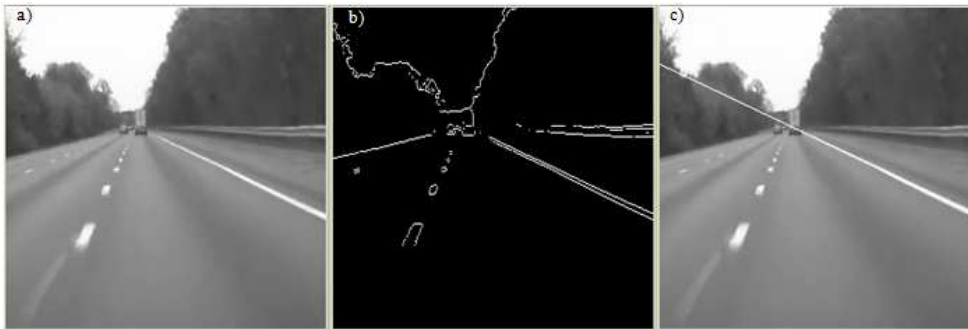
Opis i testowanie schematu Rys. 6:

- Blok („From multimedia file”) - wczytany plik video z katalogu, typ danych wyjściowych - dane typu double;
- Sekwencja obrazów barwnych (model RGB) przekształcona zostaje na obrazy intensywności (odcieni szarości - obrazy monochromatyczne) wg wzoru: $0.2989 \cdot \text{rgb_img}(:, :, 1, n) + 0.5870 \cdot \text{rgb_img}(:, :, 2, n) + 0.1140 \cdot \text{rgb_img}(:, :, 3, n)$; n - poszczególne klatki – dla danego pliku $0.2989 \cdot \text{plik_2}(:, :, 1, n) + 0.5870 \cdot \text{plik_2}(:, :, 2, n) + 0.1140 \cdot \text{plik_2}(:, :, 3, n)$;
- Do wykrywania krawędzi wykorzystano metodę gradientową Sobel'a;
- Początkowo testowano znalezienie jednej - głównej prostej (blok „Find Local Maxima” – parametr poszukuje jednego maksimum) – wynik symulacji przedstawiono na Rys. 7.

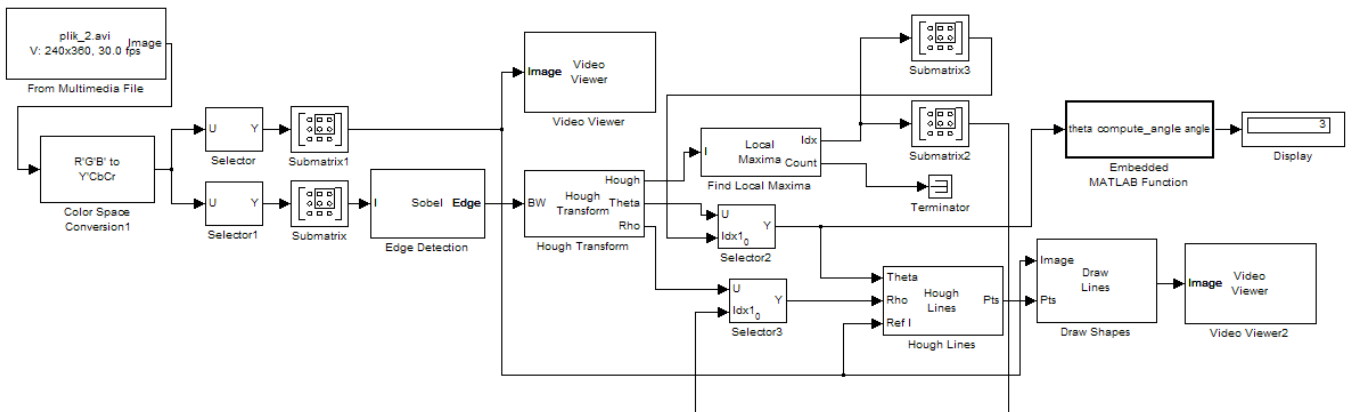
Zmiany parametrów bloków (testowanie różnych ustawień) ze schematu przedstawionego na Rys. 7 doprowadziły do poprawy dopasowania prostej do krawędzi. Podwyższono wartość mnożnika (parametr służący do obliczenia wartości progu odcięcia) a tym samym (Rys. 8b) zmniejszono liczbę punktów opisujących krawędzie.



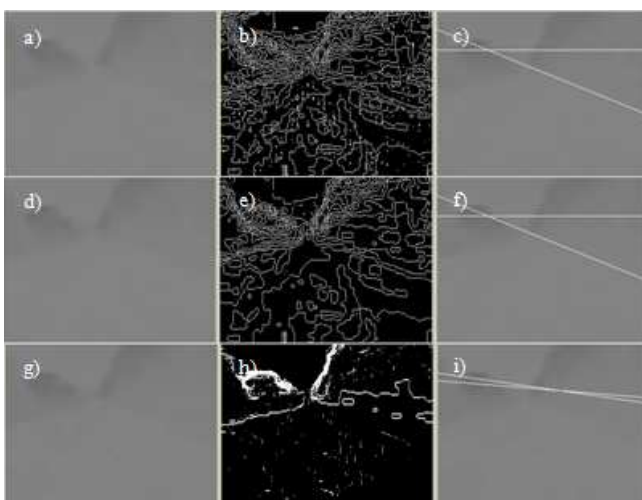
Rys. 7. Przykład poszukiwania prostej dla obrazu: a) klatka w 5 sekundzie trwania wideo; b) wykryte krawędzie; c) wykryta prosta



Rys. 8. Przykład poprawy dopasowania prostej: a) klatka w 5 sekundzie trwania wideo; b) wykryte krawędzie; c) wykryta prosta



Rys. 9. Schemat blokowy do wyznaczenia wartości dwóch prostych dla każdej klatki filmu



Rys. 10. Przykład dopasowania dwóch prostych dla obrazu (klatka w 5 sekundzie trwania wideo: a), d), g) składowa Y; b), e), h) wykryte krawędzie dla składowej Cb; c), f), i) dwie proste

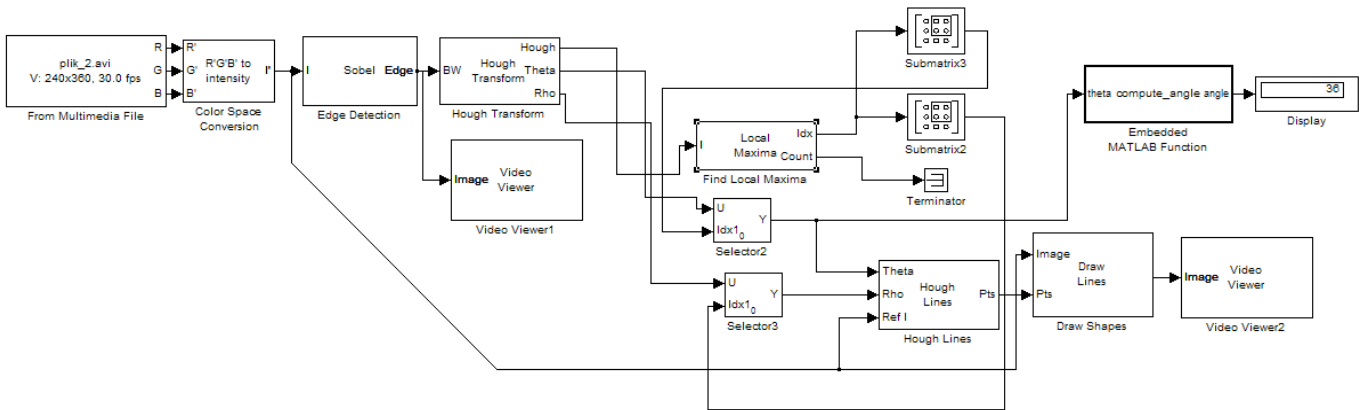
Można zatem stwierdzić, że na jakość dopasowania prostych reprezentujących krawędzie wydatnie wpływa próg odcięcia. Podobne podejście zaprezentowano na schemacie przedstawionym na Rys. 9, z tą różnicą, że wczytane wideo (a dokładniej jego klatki) zamieniano na model barwny YCbCr, dodatkowo zwiększono liczbę poszukiwanych maksimów (prostych) do 2.

Model YCbCr (przestrzeń kolorów) używany jest do cyfrowego przesyłania oraz przechowywania obrazów i wideo. Wykorzystuje on do tego trzy typy danych: Y – składową luminancji, Cb – składową różnicową chrominancji Y-B, stanowiącą różnicę między luminancją a kolorem niebieskim, oraz Cr – składową chrominancji Y-R, stanowiącą różnicę między luminancją a kolorem czerwonym. YCbCr nie jest bezwzględną przestrzenią kolorów, jest sposobem na opisanie informacji na podstawie danych o kolorze w postaci RGB. Wartość YCbCr można określić tylko na podstawie tych danych. Celem zabiegu przedstawionego na Rys. 9 było oddzielenie informacji

o jaskrawości od informacji o kolorze poprzez wykorzystanie bloków „Selector” oraz „Selector1” – separacja Y (luminancji) i Cb (chrominancji) z sygnału głównego.

Na Rys. 10 pokazano wyniki poszukiwania (wyznaczenia) dwóch prostych w oparciu o schemat blokowy przed-

stawiony na Rys. 9. Przykładowa klatka została wyznaczona w 5 sekundzie trwania wideo. Wykrywania krawędzi dokonano dla wartości mnożnika równej 0,1 – Rys. 10b, 1 – Rys. 10e, 4 – Rys. 10h.



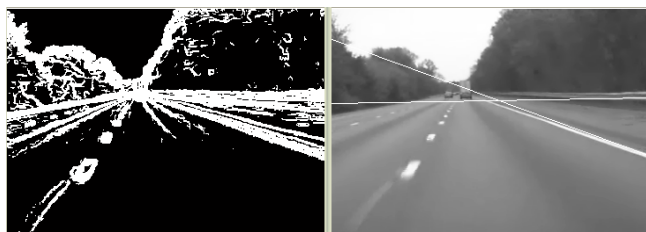
Rys. 11. Schemat blokowy do wyznaczenia wartości dowolnej liczby prostych opisujących krawędzie dla każdej klatki filmu

W przypadkach przedstawionych na Rys. 10, o kącie nachylenia prostych decyduje wartość mnożnika (ustalenie wartości progu odcięcia przy wyznaczaniu krawędzi obrazów).

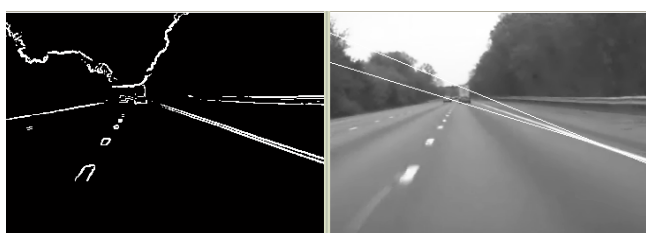
Na Rys. 11 przedstawiono schemat dający możliwość wyznaczenia dowolnej liczby prostych oparty na modelu RGB – jest to pewna modyfikacja schematu z Rys. 9.

Określenie wartości parametrów wpływających na dokładność i jakość odwzorowania krawędzi odbyło się poprzez testowanie schematu blokowego z Rys. 11 i polegało głównie na zmianie wartości mnożnika oraz redukcji grubości krawędzi. Rezultaty przedstawiono na poniższych rysunkach:

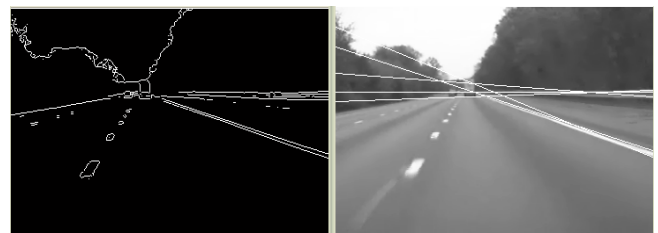
- wartość mnożnika 0,1 – bez opcji redukcji grubości krawędzi (blok „Edge detection”), poszukiwane 2 maksima (2 proste);



- wartość mnożnika 4 – bez opcji redukcji grubości krawędzi (blok „Edge detection”), poszukiwane 2 maksima (2 proste);



- wartość mnożnika 4 – z opcją redukcji grubości krawędzi (blok „Edge detection”), poszukiwane 5 maksima (5 prostych);



5. PODSUMOWANIE

Celem zaprezentowanego w artykule podejścia do przetwarzania obrazów na potrzeby lotu wśród kanionu ulic jest znaczne uproszczenie obrazu, które pozwoli na przetwarzanie go w czasie rzeczywistym. Ponadto umożliwi wydobywanie tylko tej informacji, która będzie przydatna w czasie lotu mikrosamolotu. Przy opisie analitycznym wykrytych krawędzi istotną rolę odgrywa proces zamiany obrazu monochromatycznego na obraz binarny (piksele z przedziału 0 – 255 przekształcane są na wartości 0, 1; gdzie 1 reprezentuje krawędź). Zależy on głównie od jakości rejestrowanego obrazu. Ponadto zastosowano transformację Hougha do znajdowania najdłuższych linii występujących w obrazie rastrowym. Celem takiego podejścia było sprowadzenie każdej klatki wideo do postaci łatwo interpretowanej (linie proste). Ponadto bardziej złożona implementacja transformacji Hough’a (moduł wizyjny oparty na regułach, współpracujący z autopilotem) umożliwi wyznaczenie bezkolizyjnego obszaru lotu poprzez wykrycie i uwzględnienie w czasie lotu fragmentów ścian, krawędzi ulic - przeszkód (w rozważaniach przyjęto, że mikrosamolot będzie poruszał się po ulicy otoczonej budynkami).

LITERATURA

1. **Doros M.** (2003), *Przetwarzanie obrazów*, Wyd. Wyższej Szkoły Informatyki i Zarządzania, Warszawa.
2. **Kyewook L.** (2006), *Application of the Hough Transform. Report*, University of Massachusetts, Lowell.
3. **Li Q., Zheng N., Cheng H.** (2004) Springrobot: A prototype autonomous vehicle and its algorithms for lane detection, *IEEE Transactions on Intelligent Transportation Systems*, No. 5(4), 300-308.
4. **Piszczyk M., Rutyna K.** (2007), Potencjał informacyjny obrazów pozyskiwanych z bezpilotowej platformy latającej, *PAK 2007 nr 9bis*, 173-176.
5. **Sibiński K.** (2009), Mikroelektromechaniczne latające roboty - dziś i jutro, *XLVIII Sympozjon - Modelowanie w mechanice*, Wisła.
6. **Tadeusiewicz R.** (1997), *Komputerowa analiza i przetwarzanie obrazów*, Wyd. Fundacji Postępu Telekomunikacji, Seria: Społeczeństwo Globalnej Informacji, Kraków.
7. **Wiatr K.** (2002), *Sprzętowe implementacje algorytmów przetwarzania obrazów w systemach wizyjnych czasu rzeczywistego*, AGH, Kraków.
8. **Yu B., Jain, A. K.** (1997) Lane boundary detection using a multiresolution Hough transform, *International Conference on Image Processing (ICIP'97)* Vol. 2, 748-756, Washington.
9. **MATLAB** Users' Guide

SIMULATION AND TESTING OF EDGE DETECTION ALGORITHMS FOR FLIGHT AMONG STREET CANYON

Abstract: In the article the methods for edge detection in the Matlab's software environment were tested and analyzed. The primary focus is placed on the gradient methods. In order to simplify the information comprised in the registered images (frames) an analytical description of the detected edge-obstacles was proposed (Hough's transform).

Pracę wykonano w ramach realizacji projektu rozwojowego nr 0059/R/T00/2008/06 finansowanego ze środków Ministerstwa Nauki i Szkolnictwa Wyższego.