

Paweł KOWALSKI*, Robert SMYK*

WYKRYWANIE PRZEWODÓW NAPOWIETRZNYCH W OBRAZIE CYFROWYM W CZASIE RZECZYWISTYM Z UŻYCIEM FPGA

W artykule zaprezentowano projekt architektury oraz sprzętową implementację toru przetwarzania obrazu dedykowanego do wykrywania przewodów napowietrznych w czasie rzeczywistym. Detekcję przewodów zaimplementowano w postaci potokowej procedury sprzętowej przy użyciu algorytmów wykrywania krawędzi, a następnie ich redukcji. Projekt przetestowano w środowisku FPGA Intel Cyclone V. Przeanalizowano opóźnienia i złożoność sprzętową zsyntezowanej struktury w FPGA. Oszacowano również maksymalną szybkość przetwarzania obrazu z użyciem zaproponowanej implementacji.

SŁOWA KLUCZOWE: wykrywanie krawędzi, przetwarzanie obrazu, wykrywanie linii wysokiego napięcia, FPGA.

1. WSTĘP

Współcześnie energia elektryczna jest ogólnodostępna i powszechnie konsumowana. Z uwagi na rosnące zapotrzebowanie infrastruktura sieci elektroenergetycznej jest stale rozwijana. Podstawowym medium transmisji są przewody napowietrzne. W celu optymalnego wykorzystania istniejących linii oraz zapewnienia nieprzerwanych dostaw, wymagana jest kontrola istniejącej infrastruktury. Prowadzone są prace nad systemami wspomagającymi inspekcję linii wysokiego napięcia. Jednym z pomysłów jest wykorzystanie bezzałogowych statków latających klasy dron [1]. Tego typu obiekty zdolne są do przeprowadzania inspekcji na podstawie analizy danych wizyjnych. Wyposaża się je w moduł przetwarzania obrazu złożony z kamery oraz układu, w którym pracują algorytmy przetwarzania obrazu oraz ekstrakcji danych inspekcyjnych. W zakresie użycia dronów prowadzono również badania dotyczące wpływu odległości drona od linii na jego funkcjonowanie [2] oraz możliwości lądowania na przewodzie linii elektroenergetycznej [3]. Prototypowano też systemy ostrzegania oraz zapobiegania kolizji samolotów z przewodami [4].

Czynnikiem mogąącym zwiększyć dynamikę wzrostu zapotrzebowania na energię jest silnie rozwijająca się elektromobilność. Dąży się do zastępowania pojaz-

* Politechnika Gdańska

dów spalinowych elektrycznymi, tak jak to ma miejsce w przypadku pojazdów szynowych. Jednym z argumentów przemawiających za zwiększanie liczby samochodów elektrycznych jest obniżenie poziomu zanieczyszczeń generowanych przez ruch drogowy. Szersze wykorzystanie samochodów elektrycznych naturalnie powoduje wzrost zapotrzebowania na energię elektryczną. Może się to przyczynić do zwiększenia obciążenia linii elektroenergetycznych oraz ryzyka wystąpienia awarii, którym można przeciwdziałać modernizując linie oraz zwiększając ich kontrole w celu poprawy efektywności ich wykorzystania.

Głównym problemem związanym z użytkowaniem pojazdów elektrycznych jest magazynowanie oraz uzupełnianie energii elektrycznej. W obecnej fazie technologicznego rozwoju pojazdy takie mają niższy zasięg w porównaniu do samochodów spalinowych, także proces uzupełniania energii liczony jest w godzinach, więc trwa znacznie dłużej niż tankowanie pojazdów spalinowych. Obecnie, przy stosunkowo niewielkiej liczbie pojazdów elektrycznych, ich zapotrzebowanie na prąd jest pokrywane przez sieć ładowarek stacjonarnych. Jednak w przypadku wzrostu ilości samochodów elektrycznych do poziomu przewyższającego liczbę użytkowanych samochodów spalinowych, wybudowanie tak gęstej sieci ładowarek zapewniającej pokrycie energetyczne może okazać się zbyt kosztowne lub logistycznie bardzo trudne. Dodatkowo ładowanie poprzez ładowarki stacjonarne wiąże się z unieruchomieniem całego pojazdu na dłuższy czas. Celem skrócenia czasu ładowania zaproponowano wprowadzenie pojazdów wyposażonych w wymienne akumulatory, efektem są patenty stacji wymiany baterii w pojazdach [5–7]. Takie rozwiązanie pozwala na szybką wymianę baterii oraz kontynuowanie jazdy z naładowaną baterią. W tym czasie rozładowana bateria będzie ładowana w stacji ładowania. Pilotażowy program składający się z dwóch stacji wymiany baterii oraz trzech autobusów z wymiennymi akumulatorami został uruchomiony w Korei Południowej [8].

Głównym hamulcem rozwoju pojazdów elektrycznych jest ich cena, na którą znaczący wpływ ma koszt akumulatorów. Jest on w dużej mierze uzależniony od pojemności, która przekłada się na komfort użytkownika. Głównie problem dotyczy konieczności czasochłonnego ładowania oraz zapewnienia dostatecznej ilości energii dla odbiorników takich jak ogrzewanie, klimatyzacja czy radio przy zachowaniu możliwie długiego zasięgu pojazdu elektrycznego. Zwiększenie zasięgu oraz ograniczenie postojów związanych z ładowaniem osiągnąć jest poprzez zasilanie oraz doładowywanie akumulatorów podczas jazdy. Znane są dwa główne sposoby:

- zasilanie ze źródeł wchodzących w skład pojazdu, takich jak generatory spalinowe czy ogniwa fotowoltaiczne. Przykładem takich pojazdów są pojazdy hybrydowe,
- zasilanie z zewnętrznego źródła z wykorzystaniem urządzeń wchodzących w skład infrastruktury drogowej lub kolejowej – trolejbusy, pojazdy szynowe.

Znany jest prototyp instalacji przesyłu energii elektrycznej bezprzewodowo za pomocą cewek umieszczonych w jezdni [9] w wyniku czego powstało kilka torów testowych. Rozwiązanie to zostało wdrożone w Korei w postaci linii komunikacji miejskiej składającej się z autobusów zasilanych bezprzewodowo [10]. Problemem jest efektywność przesyłania energii, która maleje wraz ze wzrostem szczeliny powietrznej. W rozwiązaniach przeznaczonych do ładowania pojazdów samochodowych wynosi ona do ok. 80% [10, 11].

Innym sposobem przesyłania energii podczas jazdy jest wykorzystanie sieci trakcyjnych nad jezdniami. Efektywność przesyłu w tym przypadku znacznie przewyższa systemy bezprzewodowe. Rozwiązanie takie stosuje się od początku XX wieku do zasilania trolejbusów. W obecnej formie nie nadaje się ono jednak do masowego użytku publicznego. W przypadku systemu trolejbusowego podłączenie do sieci trakcyjnej następuje w trakcie postoju [12]. Ogólnodostępny system zasilania pojazdów z sieci trakcyjnej powinien funkcjonować automatycznie, co należy rozumieć jako możliwość podłączania oraz odłączania od sieci w czasie jazdy bez ingerencji kierowcy. Takie rozwiązanie może pozwolić na swobodne poruszanie się, wyprzedzanie oraz wymijanie innych uczestników ruchu drogowego. Obecne metody podłączania stosowane w trolejbusach nie są do tego przystosowane [12]. Podłączenie bez ingerencji kierowcy może zostać zrealizowane z wykorzystaniem systemu wizyjnego, który precyzyjnie zlokalizuje przewód oraz umożliwi podłączenie do niego głowicy odbieraka poprzez automatyczne nakierowanie jej na przewód. Obecnie prowadzone są badania nad zasilaniem samochodów ciężarowych z sieci trakcyjnej nad jezdnią, efektem jest stworzenie testowych odcinków elektrycznych autostrad w Niemczech, Szwecji i Kalifornii [13].

Zarówno system wspomagania inspekcji przewodów elektrycznych, jak i system automatycznego podłączania i utrzymywania pantografu na przewodzie wymagają do poprawnego funkcjonowania informacji o dokładnej pozycji przewodu względem pojazdu lub drona. Pozycję tą można określić na podstawie obrazu wizyjnego. W artykule przedstawiono implementację sprzętową algorytmu dedykowanego do szybkiego wykrywania przewodu w obrazie cyfrowym. Przedstawiona implementacja pozwala na określenie pozycji przewodu w czasie rzeczywistym.

W rozdziale 2 omówiono ogólną zasadę działania algorytmu. Rozdział 3 przedstawia opis toru przetwarzania obrazu. Rozdział 4 zawiera podsumowanie.

2. ALGORYTM WYKRYWANIA PRZEWODU

Wykonanie algorytmu wymaga przeprowadzenia dwóch kroków: krawędziowania i redukcji krawędzi. Do wykrywania krawędzi zastosowano algorytm obliczający nową wartość na podstawie dwóch pikseli. Podobny algorytm przedstawiono w [14] do wykrywania linii prostych. Algorytm ten został wybrany ze

względu na niewielką ilość zasobów niezbędnych do jego implementacji w układzie FPGA. Istotną cechą jest również bardzo krótki czas wykonywania. W proponowanym algorytmie krawędziowania użyto maskę P

$$P = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}. \quad (1)$$

Wykorzystana maska (1) ma rozmiar 1×3 . Umożliwia to zaprojektowanie filtru pracującego z opóźnieniem dwóch pikseli, co należy rozumieć, że w chwili otrzymania piksela o pozycji x znany jest wynik filtracji dla pozycji $x-2$. Wynika to z faktu, że do wyznaczenia krawędzi znajdującej się w miejscu piksela o indeksie x niezbędne są dwa piksele o indeksach $x-1$, $x+1$. W praktyce po odebraniu piksela $x+1$ następuje wykonanie obliczeń w czasie jednego taktu zegara, a więc trakcie przesyłania piksela $x+2$. Oznacza to opóźnienie dwóch pikseli. Formuła obliczeniowa wykorzystywana do wykrywania krawędzi ma następującą postać

$$z_{x,y} = v_{x-1,y} - v_{x+1,y}. \quad (2)$$

Wynikiem filtracji jest różnica wartości dwóch pikseli $v_{x-1,y}$ oraz $v_{x+1,y}$, gdzie x jest numerem kolumny, a y numerem wiersza. Filtracja w tym przypadku jest realizowana w oknie o szerokości trzech pikseli. Skuteczność przedstawionego algorytmu w dużej mierze zależy od kąta nachylenia krawędzi do wiersza pikseli. W opisywanym przypadku algorytm charakteryzuje się wysokim poziomem czułości przy wykrywaniu krawędzi prostopadłych do prostej, na której leży wiersz wczytywanych pikseli.

Przy identyfikacji przewodu pośród innych krawędzi w obrazie wykorzystano obserwację odnoszącą się do kształtu przewodu. Przewód w obrazie widoczny jest w postaci linii o niewielkiej krzywiźnie. Linia taka składa się z dwóch przeciwnych równoległych krawędzi, co w przypadku (2) daje rezultat o przeciwnym znaku. Wykrywanie znaku jest realizowane przez funkcję (3)

$$sng(a) = \begin{cases} 1 & \text{dla } a > 0 \\ 0 & \text{dla } a = 0 \\ -1 & \text{dla } a < 0 \end{cases}. \quad (3)$$

W finalnie przetworzonym obrazie środek przewodu umieszczany jest centralnie pomiędzy dwoma pikselami o odwrotnych wartościach. Ostatnim etapem krawędziowania jest progowanie wykonywane na tablicy Z , w rezultacie uzyskuje się tablicę Z' . $z'_{x,y}$ jest elementem tablicy Z' o współrzędnych (x, y) .

$$z'_{x,y} = \begin{cases} 0 & \text{jesli } |z_{x,y}| \geq T \\ z_{x,y} & \text{jesli } |z_{x,y}| < T \end{cases}. \quad (4)$$

W (4) wartości poniżej ustalonego progu T są zastępowane zerami, powyżej progu T pozostają bez zmian. Tablica Z' służy do utworzenia tablicy binarnej W . W początkowym etapie tablica inicjowana jest zerami. Następnie modyfikowana jest według zależności (5).

$$w_{\lfloor x+d/2 \rfloor, y} = 1 \quad \text{jesli} \quad \begin{cases} \forall h \in (x; x+d) : sng(z'_{x+h,y}) = 0 \\ sng(z'_{x+d,y}) = 1 \\ sng(z'_{x,y}) = -1 \end{cases}, \text{ dla } d=1, 2, \dots, d_{\max} \quad (5)$$

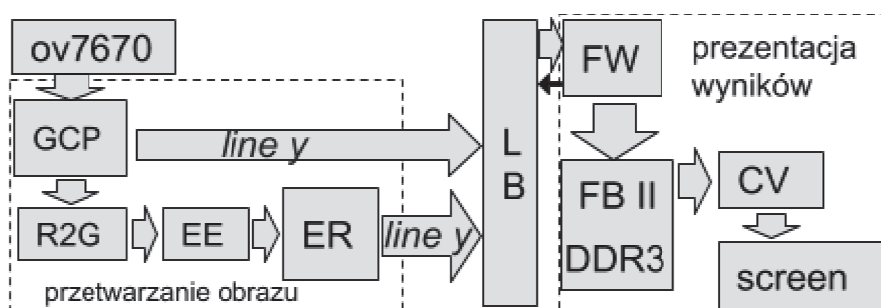
gdzie d jest liczbą naturalną określającą grubość wykrytej linii. Ograniczenie tej wartości z góry poprzez d_{\max} powoduje odfiltrowanie obiektów grubszych niż przewód. Wartość ta jest dobierana na podstawie przybliżonej grubości przewodu, zakres odległości od przewodu oraz parametrów obrazu.

3. TOR PRZETWARZANIA OBRAZU

Obraz cyfrowy przesyłany jest w postaci ciągu pikseli. W przypadku obrazu o rozdzielczości 640x480, przesyłanych jest w jednej klatce 307200 pikseli. W standardowym podejściu, z wykorzystaniem komputera PC oraz biblioteki OpenCV [15], obraz z kamery przesyłany jest do komputera i zapisywany w pamięci. Po zapisaniu obraz jest przetwarzany, a następnie wyświetlany, co daje minimalne opóźnienie równe czasowi przesyłania przez kamerę jednej klatki obrazu. Oznacza to, że pierwszy piksel obrazu wynikowego zostanie wysłany na ekran najwcześniej po wczytaniu całego obrazu. W zaproponowanej implementacji sprzętowej proces pozyskiwania i obróbki wykonywany jest w pełni potokowo. W tym przypadku maksymalne opóźnienie będzie równe czasowi przesyłania przez kamerę jednej linii obrazu.

Tor przetwarzania został zaimplementowany w układzie FPGA Intel Cyclone V wchodzącym w skład płyty rozwojowej Terasic DE10-Nano [16]. Surowy obraz pozyskiwany jest z kamery cyfrowej typu OV7670 [17]. Prezentacja wyników przetwarzania została zrealizowana poprzez jednoczesne wyświetlanie obrazu pochodzącego z kamery oraz obrazu przefiltrowanego.

Rys. 1 przedstawia schemat blokowy zaimplementowanego toru przetwarzania.



Rys. 1. Schemat toru przetwarzania

Zrealizowany układ składa się z dwóch podstawowych elementów: bloku przetwarzania obrazu oraz bloku prezentacji wyników. Elementem pośrednim

jest moduł *LineBuffer* (LB), w którym napływające dane w postaci surowego obrazu z kamery oraz obrazu finalnie przetworzonego składane są w jedną linię obrazu do prezentacji.

Prezentacja wyników polega na wyświetlaniu dwóch obrazów jednocześnie: surowy obraz z kamery oraz obraz przetworzony. Każdy o rozdzielczości 640x480. Zastosowana płyta rozwojowa wyposażona jest w jedno złącze HDMI, więc do prezentacji wyników wykorzystano jeden monitor o rozdzielczości 1280x720, pozwoliło to na umieszczenie dwóch obrazów o szerokości 640 pikseli obok siebie. W układzie zbudowano to na bazie rdzeni *Frame Buffer II IP Core* (FB II) oraz *Clocked Video IP Core* (CV) dostępnych w środowisku Altera Quartus 16.1. FB II odpowiada za buforowanie obrazu w pamięci RAM DDR3, na jego wejście należy przesłać kolejne piksele obrazu. Moduł *FrameWriter* (FW) pobiera kolejne piksele z bufora linii LB przesyłając je do FB II. CV odpowiada za przesyłanie obrazu przez HDMI do monitora. Piksele do wyświetlenia pobierane są z pamięci RAM DDR3 za pośrednictwem FB II. Przesłanie obrazu przez HDMI jest ostatnim etapem części odpowiedzialnej za prezentację wyników.

Wspomniany wcześniej etap przetwarzania obrazu rozpoczyna się od pobrania surowych danych z kamery. Zastosowana kamera przesyła dane za pomocą 8-bitowego interfejsu równoległego. Pojedynczy piksel przesyłany jest w dwóch paczkach danych zawierających łącznie 15-bitów znaczących, po 5-bitów na każdą składową RGB. Odbiór danych z kamery został zrealizowany w module *GetCamPix* (GCP). Generuje on na wyjściu 3 sygnały:

- wartość piksela w postaci trzech składowych RGB (po 5 bitów),
- współrzędna X (11 bitów),
- współrzędna Y (11 bitów).

Sygnały te przesyłane są jednocześnie do modułu LB oraz *RgbToGray* (R2G). R2G konwertuje w czasie rzeczywistym piksel do skali szarości z wykorzystaniem formuły (5)

$$Gray = 3 \cdot Red + 6 \cdot Green + Blue \quad (5)$$

Opis w języku HDL fragmentu modułu R2G odpowiedzialnego za konwersję z wykorzystaniem formuły (5) został przedstawiony na rys. 2.

```
process(clk)
begin -- 3*red + 6*green + blue
  if rising_edge(clk) then
    gray <= ("000" & red & '0') + ("0000" & red)
      + ("00" & green & "00") + ("000" & green & '0')
      + ("0000" & blue);
  end if;
end process;
```

Rys. 2. Konwersja piksela do skali szarości

Piksel w skali szarości wykorzystywany jest do filtracji z wykorzystaniem (2) w module *EdgeExtractor* (EE) przedstawionym na rys. 3.

```
if rising_edge(cam_pclk) then
  if cam_href='1' then
    P2 <= camGrayData;
    P1 <= P2;
    if P1 > camGrayData then
      edgeOut <= ("0" & P1) - ("0" & camGrayData);
    else
      edgeOut <= ("1" & camGrayData) - ("0" & P1);
    end if;
    edge_changed_sig <= NOT edge_changed_sig;
  end if;
end if;
```

Rys. 3. Fragment kodu modułu EdgeExtractor odpowiedzialnego za ekstrakcję krawędzi

Moduł EE (rys. 3) realizuje ekstrakcję krawędzi według (2). Sygnał *camGrayData* przechowuje aktualnie odczytany piksel z kamery (piksel o współrzędnej x), $P2$ piksel $x-1$, a $P1$ piksel $x-2$. W przypadku, gdy wartość piksela $x-2$ jest większa niż wartość piksela x MSB wyniku ma wartość 1, w przeciwnym przypadku 0. Jednocześnie wartość piksela w skali szarości zapisywana jest w sygnale $P2$, a ostatnia wartość z $P2$ jest przepisywana do $P1$. Odbywa się to w każdym takcie zegara kamery (*cam_pclk*). Po wykonaniu opisanej sekwencji sygnał *edge_changed_sig* zmienia się na przeciwny. Pozwala to na synchronizację EE z modułem *EdegeReductor* (ER). Wartość wynikowa *edge_out* generowana przez EE jest w kodzie ZM (znak modułu). Dalsza identyfikacja typu krawędzi bazuje na określeniu najbardziej znaczącego bitu (MSB). MSB równy 0 oznacza dodatnią, a MSB 1 ujemną krawędź. Pozostałe bity słowa to moduł.

Wynik ekstrakcji krawędzi wykorzystywany jest w module ER wyszukującym równoległe krawędzie. Fragment odpowiedzialny za redukcję z wykorzystaniem (4) i (5) został przedstawiony na rys. 4.

Moduł ten na wejściu przyjmuje między innymi piksel obrazu po ekstrakcji krawędzi:

- *edge_val* – wartość piksela,
- *edge_pos* – pozycja piksela.

Wyjście modułu to:

- *line_pos* – pozycja piksela po redukcji krawędzi,
- *line_val* – wartość piksela po redukcji.

W module realizowane jest wyszukiwanie krawędzi określonej przez sygnał *st*, a następnie najbliższej przeciwnej krawędzi. Dalej wyznaczana jest pozycja między tymi krawędziami (5). Główne operacje modułu wykonywane są w procesie ER. Schemat algorytmu znajduje się na rys. 5.

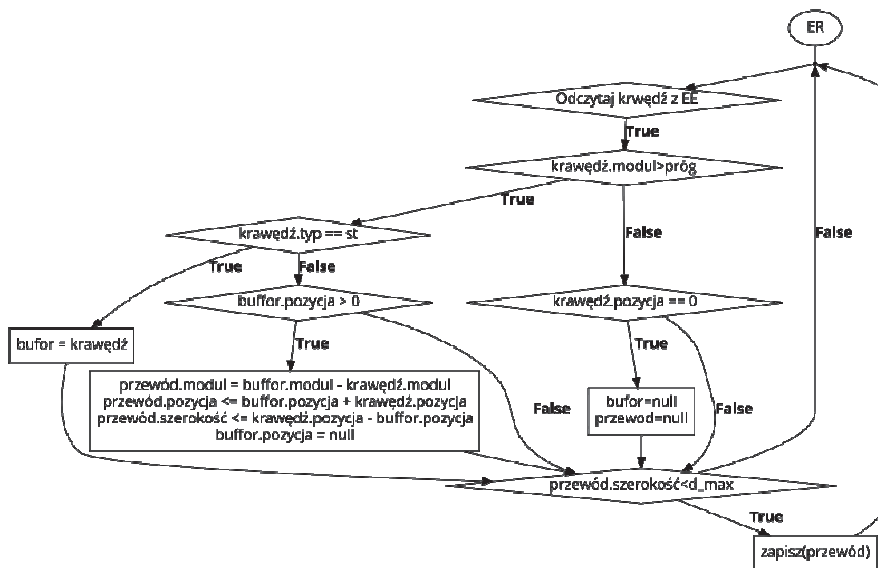
```

ER : process(clk)
begin
  if rising_edge(clk) then
    if (krawedz_modul > prog) then
      if ( krawedz_typ = st ) then
        bufor_modul <= krawedz_modul;
        bufor_pozycja <= krawedz_pozycja;
      else
        if (bufor_pozycja > 0) then
          przewod_modul <= ('0' & bufor_modul) - ('0' & krawedz_modul);
          przewod_pozycja <= ('0' & bufor_pozycja) + ('0' & krawedz_pozycja);
          przewod_szerokosc <= krawedz_pozycja - bufor_pozycja;
          bufor_pozycja <= (others => '0');
        end if;
      end if;
    else
      if (krawedz_pozycja = 0) then
        bufor_pozycja <= (others=>'0');
        przewod_pozycja <- (others->'0');
      end if;
    end if;
    krawedz_modul <= edge_val(edge_width 1 downto 0);
    krawedz_pozycja <= edge_pos;
    krawedz_typ <= edge_val(edge_width);
  end if;
end process ER;

line_pos <= przewod_pozycja(11 downto 1) when przewod_szerokosc < d_max;
line_val <= przewod_modul;

```

Rys. 4. Kod modułu EdgeReducer



Rys. 5. Schemat blokowy działania modułu EdgeReduction

Algorytm pracuje w pętli, pierwszym etapem jest odczytanie wyniku z EE do *krawędź*, a następnie sprawdzenie czy modul przekracza próg. W przypadku przekroczenia progu, kierunek krawędzi (3) porównywany jest z sygnałem *st*.

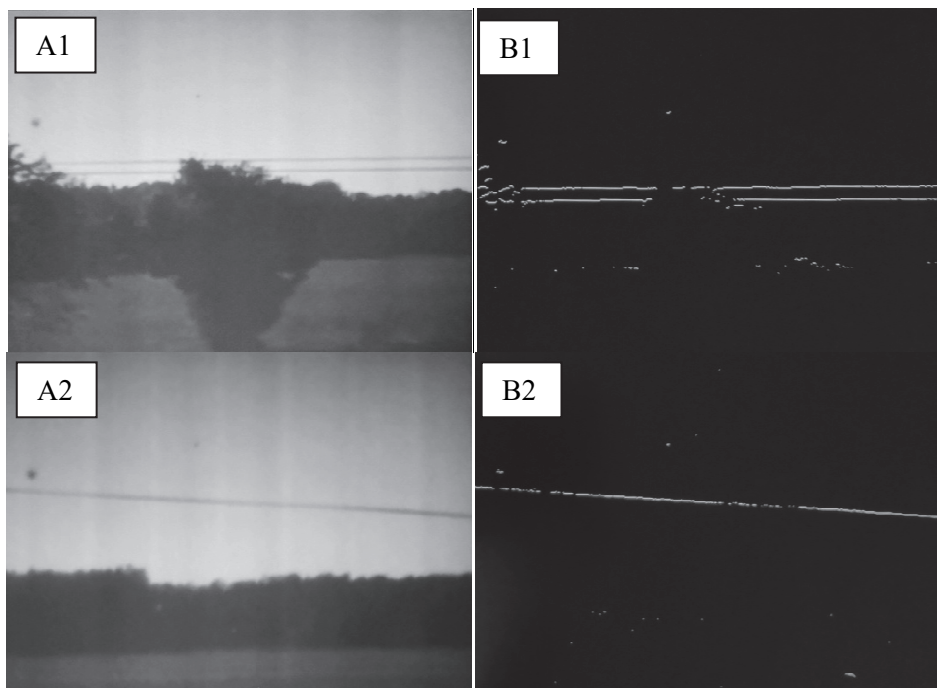
Jeżeli sygnały są identyczne krawędź jest zapisywana do buforu. W przeciwnym wypadku, jeśli bufor nie jest pusty, wyznaczana jest środkowa pozycja przewodu, a bufor jest czyszczony. Wykorzystanie sygnału *st* pozwala na wykrywanie ciemnych przewodów na jasnym tle jak i jasnych przewodów na ciemnym poprzez manipulację jej stanem. W praktyce wykrywanie ciemnych oraz jasnych przewodów można zrealizować poprzez powielenie struktury ER oraz przekazanie różnych sygnałów wejściowych *st*. Wyznaczone pozycje środkowe przewodów na bieżąco przesyłane są do modułu *LineBuffer* (LB). Moduł ten zawiera dwa rejestry: 15-bitowy o długości 640 elementów oraz dwa rejestry 8-bitowe o długości 640. Pozwala to na przechowywanie pełnej linii obrazu o rozdzielczości 1280x720 (HD), gdzie pierwsza połowa to surowy obraz z kamery, a druga to wykryte krawędzie. Rejestr 15-bitowy zapewnia buforowanie surowego obrazu. Dwa rejestry 8-bitowe wykorzystywane są do podwójnego buforowania obrazu przetworzonego. Na wejście modułu *LineBuffer* podawane są dwa sygnały:

- piksel kolorowy, pochodzący z kamery (GCP),
- pozycja wyliczona przez moduł (ER).

Zapis realizowany jest niezależnie w dwóch rejestrach.

Moduł LB połączony jest z FW. Zapis pojedynczej linii rozpoczyna się z chwilą zakończenia przesyłania pełnej linii przez kamerę, a więc w trakcie trwania sygnału synchronizacji, czyli przed rozpoczęciem przesyłania przez kamerę kolejnej linii obrazu. Pozwala to na pojedyncze buforowanie obrazu surowego. Jednak rozpoczęcie odczytywania obrazu przetworzonego przez FW może nastąpić już w trakcie przetwarzania kolejnej klatki. Aby zapewnić poprawne wyświetlanie nawet w takich sytuacjach, zastosowano podwójne buforowanie. W przypadku wykorzystania dwóch monitorów dla każdego obrazu mógłby zostać użyty pojedynczy bufor.

Rys. 6 przedstawia przykład wykrycia przewodu w obrazie z użyciem opisanej struktury. W eksperymencie próg T oraz maksymalna grubość przewodu d_{max} została dobrana doświadczalnie. Zilustrowano tam skuteczność przedstawionego rozwiązania przy wykrywaniu przewodów w obrazie rzeczywistym zarejestrowanym przy użyciu mobilnego nośnika latającego. Dalsze prace prowadzone są w kierunku opracowania algorytmu progowania oraz usuwania zakłóceń spowodowanych obiektami w tle.



Rys. 6. Przykład wykrywania przewodu w obrazie, A1, A2 – obraz zarejestrowany przez kamerę, B1, B2 – obraz poddany filtracji

3.1. Analiza opóźnień i złożoności sprzętowej

Opóźnienia dla zaimplementowanej struktury zostały określone przy użyciu narzędzia TimeQuest Timing Analyzer z pakietu Quartus Prime. Analizie poddano moduły bezpośrednio odpowiedzialne za realizację przedstawionego algorytmu, czyli *EdgeExtractor* oraz *EdgeReductor*. Inne moduły zostały pominięte. Dla modułów uzyskano maksymalną częstotliwość potokowania na poziomie 245MHz dla *EE* oraz 240 MHz dla *ER*. Odpowiada to strumieniowi obrazu o maksymalnym odświeżaniu 780 klatek/s przy rozdzielczości 640x480 lub 115 klatek/s dla obrazu FullHD (1920x1280).

Tabela 1. Wyniki syntezy w FPGA Cyclone V.

	ALM	Block memory bits	Registers	F _{max} [Mhz]	Max [fps]	
					640x480	FullHD
EE	15	0	5	245	797	118
ER	35	0	74	240	780	115

4. PODSUMOWANIE

W artykule przedstawiono sprzętowy algorytm wykrywania przewodów napowietrznych. Zaproponowana procedura charakteryzuje się bardzo niską latencją oraz wymaga niewielkiego narzutu na strukturę FPGA. Zrealizowany układ funkcjonuje w pełni potokowo. Skuteczność wykrywania zaproponowanego rozwiązania przetestowano eksperymentalnie w rzeczywistym środowisku.

LITERATURA

- [1] Wu Y., Zhao G., Hu J., Ouyang Y., Wang S., He J., Gao F., Wang S., Overhead Transmission Line Parameter Reconstruction for UAV Inspection Based on Tunneling Magnetoresistive Sensors and Inverse Models, *IEEE Transactions on Power Delivery*, ISSN 0885-8977, pp. 1937–4208, 2019.
- [2] Dong X., Qu F., Li Y., Wu Z., Chen Z., Lai Q., Liu G., Electric field simulation and discharge distance measurement between UAV and overhead line, in 2018 12th International Conference on the Properties and Applications of Dielectric Materials (ICPADM), Xi'an, 2018, pp. 633–636.
- [3] Miralles F., Hamelin P., Lambert G., Lavoie S., Pouliot N., Montfrond M., Montambault S., LineDrone Technology: Landing an Unmanned Aerial Vehicle on a Power Line, in 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 6545–6552.
- [4] Yetgin O.E., Senturk Z., Gerek O.N., A comparison of line detection methods for power line avoidance in aircrafts, in 2015 9th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, 2015, pp. 241–245.
- [5] Gu H.-L., Wei T.-T., Cheng L.-R., Battery charging and handling system for electric vehicles. 2000.
- [6] Ligrano R., Battery exchange station and a method of changing therein. 2008.
- [7] Park J.S., Kim W.-K., Park H.-J., Moon H.S., Choi W., Jeong J., Yu C.M., Do Yang J., Shin Y., Park J.-H., others, Battery exchanging method for electric vehicle. 2015.
- [8] Kim J., Song I., Choi W., An Electric Bus with a Battery Exchange System, *Energies*, Volume 8, Number 7, ISSN 1996-1073, 2015.
- [9] Rouissiya M., Abbassi I.E., Amghar B., Jaafari A., Darcherif A.-M., Saad A., Electrical energy wireless transfert: Application to electric roads, in 2017 15th International Conference on Electrical Machines, Drives and Power Systems (ELMA), Sofia, Bulgaria, 2017, pp. 309–313.
- [10] Jeong S., Jang Y.J., Kum D., Economic Analysis of the Dynamic Charging Electric Vehicle, *IEEE Transactions on Power Electronics*, Volume 30, Number 11, ISSN 0885-8993, 1941-0107, 2015.
- [11] Lee S., Huh J., Park C., Choi N.-S., Cho G.-H., Rim C.-T., On-Line Electric Vehicle using inductive power transfer system, in 2010 IEEE Energy Conversion Congress and Exposition, Atlanta, GA, 2010, pp. 1598–1601.
- [12] Zalewska A., Budowa i zasada działania trolejbusu na przykładzie Solaris Trollino 12, *Autobusy: technika, eksploatacja, systemy transportowe*, Volume 18, 2017.
- [13] Siemens AG, eHighway Innovative electric road freight transport. 2017.

- [14] Burns J. B., Hanson A. R., Riseman E. M., Extracting Straight Lines, IEEE Transactions on Pattern Analysis & Machine Intelligence, Volume 8, Number 4, 1986, pp. 425–455. .
- [15] Open Source Computer Vision Library, Reference Manual, 2014. .
- [16] Terasic, DE10-nano Cyclone V Soc with Dual-core ARM Cortex-A9 User Manual. 2018.
- [17] OV7670/OV7171 CMOS VGA(640X480) CameraChip with OmniPixel Technology Advances Information Preliminary Datasheet. 2005.

FPGA IMPLEMENTATION OF REALTIME DETECTION OF HIGH VOLTAGE LINES IN DIGITAL IMAGE

The paper presents the architecture design and hardware implementation of a custom image processing module dedicated for detection of high voltage lines in real time. It has been implemented in the form of a pipelined hardware procedure using edge detection and reduction algorithm. The design was tested in the Intel Cyclone V FPGA environment. Time and hardware complexity of the synthesized structure in FPGA were analyzed. The maximum image processing speed was also estimated using the proposed implementation.

(Received: 07.02.2019, revised: 05.03.2019)