

Paweł KOWALSKI\*, Robert SMYK\*

## ROZPOZNAWANIE ORAZ LOKALIZACJA W OBRAZIE PRZEWODÓW LINII WYSOKIEGO NAPIĘCIA

W pracy przedstawiono opracowany algorytm rozpoznawania oraz lokalizacji przewodów linii wysokiego napięcia na podstawie obrazu horyzontalnego. Procedura detekcji przewodu została podzielona na trzy etapy. Pierwszy etap zawiera algorytm wykrywania krawędzi wykazujący największą czułość na krawędzie poziome, a jednocześnie brak czułości na krawędzie pionowe. Efektem jest znaczna redukcja liczby wykrytych krawędzi w porównaniu do algorytmów wykrywających krawędzie we wszystkich kierunkach ze stałą czułością. Drugi etap polega na filtracji wykrytych krawędzi oraz klasyfikacji punktów do dalszej analizy. Na podstawie lokalnych różnic między sąsiadującymi pikselami wstępnie odrzucane są miejsca zawierające przypuszczalnie szum lub tło. Finalna klasyfikacja punktów reprezentujących przewód dokonywana jest z wykorzystaniem algorytmu znajdującego najdłuższą ścieżkę. Ostatni, trzeci etap polega na lokalizacji przewodu poprzez aproksymację jego kształtu na podstawie sklasyfikowanych punktów. Kontur wykrytego przewodu uzupełniany jest zbiorem pikseli reprezentującym przerwane ścieżki, zwiększając w ten sposób dokładność finalnej aproksymacji. Procedurę wykrywania przewodu zrealizowaną przy użyciu poszczególnych algorytmów przetestowano dla zdjęć rzeczywistych przewodów elektrycznych i stwierdzono jej pełną skuteczność.

SŁOWA KLUCZOWE: wykrywanie krawędzi, wykrywanie krzywych, wykrywanie przewodów wysokiego napięcia, aproksymacja kształtu przewodu, przetwarzanie obrazu.

### 1. WSTĘP

Obraz rejestrowany przez kamerę cyfrową zapisywany jest przy pomocy punktów zwanych dalej pikselami. Ich liczba na pojedynczym zdjęciu często przekracza kilka milionów. Obraz zawiera wiele informacji, między innymi o obiektach znajdujących się w polu widzenia kamery. Zapisane one są w postaci grup pikseli, których liczba może przekraczać kilka tysięcy. Poprzez przetwarzanie obrazu dokonywana jest ekstrakcja wybranych informacji. Jednym z głównych etapów przetwarzania jest rozpoznawanie oraz lokalizacja obiektów. Ich identyfikacja realizowana jest na podstawie cech charakterystycznych, takich jak kształt czy kolor. Znaleziony oraz zidentyfikowany obiekt może zostać

---

\* Politechnika Gdańska

przedstawiony i zapisany w postaci parametrów opisującego go modelu. Przykładowo, okrąg o średnicy tysiąca pikseli może zostać zapisany w postaci punktu centralnego oraz promienia, co odpowiada trzem liczbom. Przewód napowietrzny można zapisać w postaci modelu składającego się z kilku współczynników funkcji wielomianowej. Taka reprezentacja pozwala na znaczną redukcję zajmowanego miejsca w porównaniu do reprezentacji w postaci pikseli. Zestaw zamodelowanych obiektów wykorzystywany jest w dalszych etapach przetwarzania obrazu. Przykładem wykorzystania modeli obiektów może być wyznaczanie odległości, położenia, czy wielkości obiektów. W artykule został zaprezentowany sposób wykrywania przewodu oraz aproksymacji kształtu przewodu wysokiego napięcia na podstawie obrazu cyfrowego. Funkcjonowanie algorytmu zostało zweryfikowane eksperymentalnie.

W rozdziale 2 przedstawiono ekstrakcję krawędzi. W rozdziale 3 zaprezentowano procedurę wykrywania linii, której efektem jest zbiór punktów użytych następnie do aproksymacji kształtu przewodu, a w p.4 przedstawiono algorytm aproksymacji kształtu przewodu.

## 2. PROCEDURA WYKRYWANIA KRAWĘDZI W OBRAZIE CYFROWYM

Problem wykrywania linii jest szeroko poruszany w literaturze. Opisywane są uniwersalne algorytmy detekcji linii [1–4] jak i wersje specjalizowane. Znane są prace dotyczące specjalizowanych algorytmów takich jak wykrywanie prostych [2, 5] oraz krzywych [1]. Jeszcze bardziej specjalizowane algorytmy znajdują bezpośrednie zastosowanie w medycynie [6] czy też lotnictwie [7]. Opisywane są również sposoby wykrywania przewodów w obrazie [7–11], jednak w większości opierają się one na wykrywaniu linii prostych. Algorytmy przedstawione w artykule są dedykowane do wykrywania oraz modelowania krzywych przedstawiających przewody, jest to związane z planowanym zastosowaniem algorytmu do pomiaru odległości od przewodów linii wysokiego napięcia.

Algorytm wykrywania przewodów wykorzystuje aproksymację kształtu przewodu w obrazie cyfrowym. Obraz przewodu pochodzi z kamery ustawionej horyzontalnie. Typowo będzie on zbliżony do linii poziomej. W takim przypadku można przyjąć, że krawędzie pionowe w przetwarzanym obrazie są nieistotne i usuwać je już na etapie wykrywania krawędzi. Usunięcie krawędzi pionowych ma na celu redukcję danych w obrazie cyfrowym.

Omawiany algorytm zaprojektowano w celu zapewnienia przetwarzania obrazu w czasie rzeczywistym, przy czym głównym celem był możliwie krótki czas obliczeń przy zapewnieniu odpowiedniej skuteczności wykrywania przewodów. Proces wykrywania krawędzi jest jednym z pierwszych etapów przetwarzania obrazu. Na etapie tym ilość danych jest więc największa. Warto za-



uważyć, że w typowym podejściu przetwarzane są wszystkie piksele obrazu. Wykonując jedną operację na tym etapie, przykładowo dla zdjęcia o rozdzielczości 5 Mpx, operacja ta musi zostać wykonana 5 milionów razy. Pożądanym jest, aby algorytm wykrywania krawędzi wymagał możliwie niewielkiej liczby operacji. Zaproponowano użycie algorytmu wymagającego pojedynczej operacji obliczania różnicy dla każdego piksela. Jest to zmodyfikowana wersja algorytmu wykrywania krawędzi, który pierwotnie został użyty do wykrywania linii prostych [2]. W zmodyfikowanej wersji algorytmu maska w połączeniu z algorytmem redukcji krawędzi używana jest do wykrywania krzywych przedstawiających przewód. Przy obliczaniu nowej wartości piksela wykorzystywane są dwa piksele z obrazu wejściowego. Obliczany jest spłot z maską  $P_1$  o rozmiarze  $3 \times 1$  piksele, przedstawioną poniżej

$$P_1 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad (1)$$

W związku z tym zależność wykorzystywana do wykrywania krawędzi wykorzystuje następującą operację odejmowania

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

gdzie:  $z_{x,y}$  jest pikselem macierzy wyjściowej, powstałej w wyniku obróbki, a  $v_{x,y}$  i  $v_{x,y+2}$  są odpowiednio, pikselem macierzy wejściowej o współrzędnych  $(x, y)$  i pikselem o współrzędnych  $(x, y+2)$ .

Zależność (2) przedstawia sposób obliczania wartości wykrytej krawędzi zapisywanej w macierzy  $Z$  na podstawie obrazu wejściowego zapisanego w macierzy  $V$ .

### 3. SELEKCJA PUNKTÓW DO APROKSYMACJI

Aproksymacja kształtu przewodu jest procesem czasochłonnym. Wskazane jest, aby wcześniej jak najlepiej odfiltrować wejściowy zestaw danych. W tym celu w jednym kroku obliczeniowym wraz z wykrywaniem krawędzi wykonywany jest algorytm redukcji krawędzi. Zasada działania algorytmu jest następująca: przewód w obrazie reprezentowany jest przez dwie blisko położone równoległe krawędzie. Na podstawie tych krawędzi określany jest środek przewodu. Dwie krawędzie są następnie redukowane do jednej linii, co ma wpływ na efektywność dalszej aproksymacji. Proces wykrywania oraz redukcji krawędzi wymaga trzech etapów. Przykład realizacji tego procesu został przedstawiony na rys. 1.

a)		b)		c)		d)
180		-1		0		0
181		1		0		0
180		159		2		0
21		1		0		0
20	->	1	->	0	->	1
19		-4		0		0
23		-158		3		0
181		-1		0		0
182		0		0		0
182		-		-		-

Rys. 1. Przykład realizacji algorytmu redukcji krawędzi

Rys. 1 przedstawia przykładowe dane reprezentujące poziomy jasności pikseli w skali szarości w postaci fragmentu jednej kolumny obrazu z widocznym na nim przewodem. Kolejne kolumny zawierają wartości zapisywane w kolejnych krokach działania algorytmu wykrywania i redukcji krawędzi. Kolumna a) zawiera dane wejściowe (zarejestrowane przez kamerę). Wartość w kolumnie b) wyliczana jest z wykorzystaniem algorytmu wykrywania krawędzi (2) na podstawie dwóch pikseli z kolumny a). Użycie niewielkiej liczby pikseli oraz pojedynczej podstawowej operacji arytmetycznej zapewnia dużą szybkość działania. Kolumna c) zawiera dane po progowaniu. Funkcja realizująca to zadanie została przedstawiona poniżej.

$$threshold\_f(p) = \begin{cases} 0, & |p| < threshold\_val \\ 2, & p > threshold\_val \\ 3, & p < -threshold\_val \end{cases} \quad (3)$$

Na wejście funkcji progowania (3) kierowane jest wyjście algorytmu wykrywania krawędzi (1), (2). Jeśli wartość bezwzględna jest poniżej progu *threshold\_val*, funkcja przyjmuje wartość 0. W przeciwnym wypadku w zależności od znaku wartości *p* zwracana jest wartość 2 reprezentująca górną krawędź lub 3 reprezentująca dolną krawędź. Ostatnim krokiem jest wykonanie algorytmu redukcji krawędzi, który pomiędzy wartości 2 i 3 (w połowie odległości) wstawi wartość 1 reprezentującą wykryty przewód, jednocześnie usuwając dawne krawędzie 2 i 3. Efekt działania tego algorytmu prezentuje kolumna d). Opisany algorytm wykrywania i redukcji krawędzi gwarantuje, że w obrazie pozostaną jedynie linie o grubości 1 piksela. Pozwala to na proste wykrywanie linii poprzez znajdowanie dużych grup połączonych ze sobą pikseli. Omawianą operację opisano w postaci Algorytmu 1 wyrażonego pseudokodem zrealizowanym w formie funkcji rekurencyjnej `find_connected_pix`.

*Algorytm 1.* Algorytm poszukiwania najdłuższej ścieżki utworzonej przez przylegające do siebie piksele.

Wejście – współrzędne piksela startowego  $x, y$ .

Wyjście – lista pikseli tworzących najdłuższą ścieżkę.

```
find_connected_pix(x, y) {
  if (point(x, y) ∉ zbiór_krawędzi)
    return NULL;
  else {
    zbiór_krawędzi->delete(point(x, y));

    local_list->add(point(x, y));

    local_list->add_list  $\left( \max\_size \left( \begin{array}{l} \text{find\_connected\_pix}(x+1, y-1) \\ \text{find\_connected\_pix}(x+1, y) \\ \text{find\_connected\_pix}(x+1, y+1) \end{array} \right) \right)$ 

    return local_list;
  }
}
```

Funkcja `find_connected_pix` operuje na obrazie wynikowym po wykryciu i redukcji krawędzi. Parametry formalne reprezentują współrzędne wybranego punktu w postaci numeru kolumny  $x$  oraz wiersza  $y$ . Jeżeli punkt nie istnieje w zbiorze krawędzi, funkcja kończy działanie zwracając pustą listę (NULL). Jeśli punkt  $(x, y)$  istnieje w zbiorze wykrytych krawędzi, jest on z niego usuwany i dodawany do lokalnej listy pikseli. Następnie dla trzech najbliższych punktów, leżących po prawej stronie  $(x+1, y-1)$ ,  $(x+1, y)$ ,  $(x+1, y+1)$  wywoływana jest funkcja `find_connected_pix`. Liczby elementów zwróconych list są ze sobą porównywane. Następnie do lokalnej listy pikseli `local_list` dodawana jest najliczniejsza ze zwróconych list. Ostatnim etapem wykonywania funkcji jest zwrócenie lokalnej listy pikseli `local_list`, lista ta zawiera otrzymane współrzędne oraz listę pikseli wchodzących w skład najdłuższej ścieżki utworzonej z połączonych pikseli od punktu wejściowego  $(x, y)$  w prawo. Wywołując funkcję dla kolejnych pikseli reprezentujących krawędź, poczynając od lewej strony obrazu, zwrócone zostaną najdłuższe ścieżki. Przy czym usunięcie piksela z listy krawędzi po wywołaniu na nim funkcji `find_connected_pix` gwarantuje, że każda ścieżka zostanie zwrócona tylko raz. W przeprowadzonym eksperymencie aproksymacji poddawano grupy pikseli, których liczba przekraczała  $\frac{1}{4}$  szerokości całego zdjęcia.

#### 4. APROKSYMACJA PRZEWODU

Zbiór pikseli w postaci listy współrzędnych, tworzących ścieżkę prezentującą przewód elektryczny wykorzystywany jest do estymacji parametrów modelu



w postaci funkcji (4) opisującej kształt oraz pozycję przewodu na zdjęciu. Funkcję aproksymującą przyjęto w postaci wielomianu zmiennej  $x$ , gdzie  $x$  jest numerem kolumny pikseli

$$f(x) = \sum_{j=0}^n a_j x^j \quad (4)$$

Zastosowano aproksymację średniokwadratową wymagającą minimalizacji kryterium o postaci.

$$S = \sum_{i=0}^m \left[ y_i - \sum_{j=0}^n a_j x_i^j \right]^2 \quad (5)$$

gdzie:  $y$  jest numerem wiersza, w którym został wykryty przewód, a  $m$  wielkością próbki wejściowej (liczba pikseli stosowanych do aproksymacji). W celu wyznaczenia wartości parametrów  $a_j$ ,  $j=0, \dots, n$ , należy rozwiązać układ równań o postaci

$$\frac{\partial S}{\partial a_k} = -2 \left\{ \sum_{i=0}^m \left[ y_i - \sum_{j=0}^n a_j x_i^j \right] x_i^k \right\} = 0, \quad k=0, 1, \dots, n \quad (6)$$

Po uporządkowaniu (6) otrzymujemy następującą zależność

$$\sum_{j=0}^n \left[ \sum_{i=0}^m (x_i^j x_i^k) \right] a_j = \sum_{i=0}^m x_i^k y_i, \quad k=0, 1, \dots, n \quad (7)$$

Wartości parametrów modelu można uzyskać rozwiązując równanie (7). W celu łatwiejszej implementacji numerycznej równanie (7) można przedstawić w postaci macierzowej, jak niżej

$$A \cdot a = b \quad (8)$$

gdzie:  $a$  jest szukanym wektorem parametrów modelu. Macierze  $b$  oraz  $A$  obliczane są na podstawie współrzędnych pikseli zakwalifikowanych do aproksymacji. Elementy macierzy  $A$  i  $b$  wyznacza się z zależności

$$A_{k,j} = \sum_{i=0}^m x_i^{j+k}, \quad k=0, 1, \dots, n \quad (9a)$$

$$b_k = \sum_{i=0}^m x_i^k y_i, \quad k=0, 1, \dots, n \quad (9b)$$

Przedstawiony algorytm zrealizowano w programie w postaci sparametryzowanych funkcji, które zaimplementowano w standardzie ANSI C. Kod funkcji obliczających macierz  $b$  został przedstawiony na rys. 2.



```
1 double calculate_b(int k, double *x, double *y, int m){
2     int i;
3     double result=0;
4     for (i=0; i<m; i++)
5         result += pow(x[i],k)*y[i];
6     return result;
7 }
8 void fill_matrix_b(double *b, int n, double *x, double y[], int m){
9     int k;
10    for (k=0; k<=n; k++)
11        b[k]=calculate_b(k, x, y, m);
12 }
```

Rys. 2. Definicja funkcji fill\_matrix\_b oraz calculate\_b

Parametrami formalnymi funkcji są: wskaźnik do tablicy  $b$  reprezentującej macierz  $b$ , stopień wielomianu  $n$  i tablice współrzędnych  $x$  oraz  $y$  analogicznie do formuł (4-9b), oraz wielkość próbki  $m$  (5-9b), czyli rozmiar tablic współrzędnych  $x, y$ . Funkcja *fill\_matrix\_b* wywołuje dla każdej komórki tablicy  $b$  funkcję *calculate\_b*, która oblicza odpowiednią wartość wykorzystując (9b). Funkcje generujące wartości macierzy  $A$  zostały przedstawione na rys. 3.

```
1 double calculate_A(int k, int j, double *x, int m){
2     int i;
3     double result=0;
4     for (i=0; i<m; i++)
5         result += pow(x[i], j+k);
6     return result;
7 }
8 void fill_matrix_A (double *A, int n, double *x, int m){
9     int j,k=0;
10    for (j=0; j<=n; j++)
11        for (k=0; k<=n; k++)
12            A[j*(n+1)+k]=calculate_A(k, j, x, m);
13 }
```

Rys. 3. Definicja funkcji fill\_matrix\_A oraz calculate\_A

Jednym z parametrów formalnych funkcji jest wskaźnik do tablicy  $A$  reprezentującej macierz  $A$  (8, 9a). Znaczenie pozostałych parametrów jest identyczne jak w przypadku funkcji *calculate\_b* oraz *fill\_matrix\_b* (rys. 2). Funkcja *calculate\_A* oblicza pojedynczy element macierzy  $A$  według zależności (9a). Funkcja *fill\_matrix\_A* wypełnia tablicę  $A$  wartościami obliczonymi przy pomocy funkcji *calculate\_A*. Na rys.4 przedstawiono graficzną reprezentację macierzy  $A$  o rozmiarze  $4 \times 4$ .

$A_{0,0} = \sum_{i=0}^m x_i^0$	$A_{0,1} = \sum_{i=0}^m x_i^1$	$A_{0,2} = \sum_{i=0}^m x_i^2$	$A_{0,3} = \sum_{i=0}^m x_i^3$
$A_{1,0} = \sum_{i=0}^m x_i^1$	$A_{1,1} = \sum_{i=0}^m x_i^2$	$A_{1,2} = \sum_{i=0}^m x_i^3$	$A_{1,3} = \sum_{i=0}^m x_i^4$
$A_{2,0} = \sum_{i=0}^m x_i^2$	$A_{2,1} = \sum_{i=0}^m x_i^3$	$A_{2,2} = \sum_{i=0}^m x_i^4$	$A_{2,3} = \sum_{i=0}^m x_i^5$
$A_{3,0} = \sum_{i=0}^m x_i^3$	$A_{3,1} = \sum_{i=0}^m x_i^4$	$A_{3,2} = \sum_{i=0}^m x_i^5$	$A_{3,3} = \sum_{i=0}^m x_i^6$

Rys. 4. Macierz A o rozmiarze 4x4 używana do wyznaczania parametrów modelu

Rys. 5. Macierz A z zaznaczonymi komórkami o tych samych wartościach

Elementy macierzy z rys. 4 obliczono przy użyciu zależności (9b). Na rys. 5 zaznaczono 5 grup komórek macierzy  $A$  położonych na przeciwprzekątnych. Dla każdej z zaznaczonych grup sumy współrzędnych poszczególnych komórek są identyczne. Z zależności (9a) wynika, że wartości w takich komórkach są identyczne. Przykładowo, biorąc pod uwagę komórki  $A_{x,y}$  oraz  $A_{q,r}$ , występuje związek: jeżeli  $x+y=q+r$ , to  $A_{x,y} = A_{q,r}$ . Pozwala to na poprawę szybkości działania algorytmu poprzez przepisywanie powtarzających się wartości. Wprowadzając opisane usprawnienie wynikowa tablica wymaga obliczenia  $2n+1$  różnych elementów. Zoptymalizowana pod tym kątem funkcja `fill_matrix_A_v2` została przedstawiona na rys. 6.

```

1 void fill_matrix_A_v2(double *A, int n, double *x, int m){
2     int j, k, j0;
3     double tmp_A;
4     A[0] = calculate_A(0, 0, x, m);
5     for (j0=0; j0<=n; j0++){
6         for (j=j0, k=0, tmp_A = calculate_A(k, j, x, m); j>=0; j--, k++)
7             A[j*(n+1)+k]=tmp_A;
8         for (j=j0, k=n, tmp_A = calculate_A(k, j, x, m); j<=n; j++, k--)
9             A[j*(n+1)+k]=tmp_A;
10    }
11 }

```

Rys. 6. Definicja funkcji `fill_matrix_A_v2`



Zestaw parametrów formalnych oraz efekt działania funkcji `fill_matrix_A_v2` jest identyczny jak wcześniej opisywanej funkcji `fill_matrix_A`. Jednak poprzez zastosowanie optymalizacji polegającej na kopiowaniu powtarzających się wartości ograniczona została liczba wywołań najbardziej czasochłonnej funkcji `calculate_A`. Wraz ze wzrostem stopnia wielomianu oraz rozmiaru danych wejściowych, przy zastosowanej optymalizacji spada czas wykonania. Przykładowo dla wielomianu 3 stopnia, przed optymalizacją funkcja `calculate_A` była wywoływana 16 razy, po optymalizacji jest wywoływana 7 razy. Dla wielomianu 4 stopnia, proporcje te wynoszą odpowiednio 25 do 9.

Po wypełnieniu macierzy  $A$  oraz  $b$  na podstawie zależności (9a, 9b) można znaleźć współczynniki  $a$ . Znalezienie wartości współczynników  $a$  wymaga rozwiązania układu równań np. z wykorzystaniem algorytmu eliminacji Gaussa.

W celu przeprowadzenia eksperymentu zaimplementowano opisane algorytmy w C++ z wykorzystaniem biblioteki OpenCV[12]. Przeprowadzono testowe wykrywanie przewodów na zestawie zdjęć. Algorytm poprawnie realizuje aproksymację kształtu przewodów, w których krawędzie są reprezentowane przez jedną grupę połączonych pikseli. Jednak ciągłość pikseli reprezentujących krawędź przewodu może zostać przerwana, gdy przewód znajduje się on na tle obiektów o zbliżonym kolorze. Skutkiem jest zakwalifikowanie do aproksymacji jedynie części przewodu, co bezpośrednio wynika ze sposobu działania funkcji `find_connected_pix`.



Rys. 7. Przykład wpływu tła na przebieg procedury wykrywania i aproksymacji przewodu

Rys. 7 ilustruje wpływ obiektów tła na wybór punktów oraz aproksymację przewodu. Zbiór pikseli stanowiących obraz został podany na wejście algorytmu wykrywania oraz redukcji krawędzi. Następnie uruchomiony został algorytm selekcji punktów oraz aproksymacji. Rys. 7 przedstawia końcowy efekt działania wymienionych algorytmów, co uwidoczniło zaznaczając czerwoną linią zamodelowaną funkcję (4). Skrajne fragmenty przewodu zostały powiększone. Powiększenie z lewej strony pokazuje, że lewa strona przewodu została popraw-

nie zamodelowana. Przewód (czarna linia) jest w środku czerwonego zaznaczenia (model przewodu). Na powiększeniu z prawej strony widać, że przewód jest wyżej, niż zostało to zamodelowane przez funkcję. Lewa strona została zamodelowana poprawnie, ponieważ algorytm selekcji punktów do aproksymacji rozpoczyna swoje działanie od lewej strony obrazu. Po napotkaniu przeszkody (ciemne chmury, lampa w tle) nastąpiło przerwanie ciągłości linii, a tym samym do aproksymacji została przekazana jedynie część pikseli reprezentujących przewód.

W celu poprawy dokładności aproksymacji, wprowadzona została modyfikacja. Polega ona na rozszerzaniu listy aproksymowanych punktów na podstawie modelu funkcji. Pierwszym krokiem do rozszerzenia listy punktów jest utworzenie zbioru punktów leżących blisko zamodelowanego przewodu. Punkty, które powtarzają się w zbiorze krawędzi biorą udział w dalszej selekcji. Akceptowane poprzez dodanie do zbioru do aproksymacji są punkty wchodzące w skład najbliższej ścieżki pikseli. Operacje te są powtarzane aż do wykrycia oraz aproksymacji wszystkich punktów krawędzi przewodu.



Rys. 8. Wykryte przewody z zastosowaniem algorytmu aproksymacji

Rys. 8 przedstawia zdjęcie pokazujące efekt działania zmodyfikowanego algorytmu aproksymacji kształtu przewodu. Na zdjęciu zostały powiększone dwa skrajne fragmenty przewodu. W przeciwieństwie do rys. 7 pokrywają się one z zaznaczonym przez funkcję aproksymującą kształtem, co potwierdza skuteczność metody.

## 5. PODSUMOWANIE

W artykule przedstawiono opracowaną metodę aproksymacji kształtu przewodu na podstawie obrazu cyfrowego. Zaproponowano proces filtracji do wstępnej obróbki wykorzystujący szybki algorytm wykrywania krawędzi. Uzyskane krawędzie zostały wykorzystane do wyboru punktów aproksymacji, co zostało zrealizowane przy użyciu rekurencyjnej funkcji znajdującej najdłuższe ścieżki pikseli. Finalnie opracowany został algorytm aproksymacji kształtu

przewodu oraz zwiększania dokładności aproksymacji poprzez kilkakrotne użycie algorytmu aproksymacji i dodawanie znalezionych grup pikseli. Wszystkie algorytmy zaimplementowano programowo, a następnie przedstawiono zestawienie wyników przeprowadzonych eksperymentów. Przeprowadzone eksperymenty wykazały, że opracowane algorytmy umożliwiają wykrycie przewodu w obrazie.

## LITERATURA

- [1] Duda R. O., Hart P. E., Use of the Hough transformation to detect lines and curves in pictures, *Comm. ACM*, Vol. 15, Jan., 1972, pp. 11–15.
- [2] 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.
- [3] Grompone von Gioi R., Jakubowicz J., Morel J. M., Randall G., LSD: a Line Segment Detector, *Image Processing On Line*, 2012, pp. 35-55.
- [4] Grompone von Gioi R., Jakubowicz J., Morel J. M., Randall G., LSD: A Fast Line Segment Detector with a False Detection Control, *IEEE Transactions on Pattern Analysis and Machine Intelligence* Volume 32, Number 4, April 2010, pp. 722-732.
- [5] Huang Q., An Algorithm for Extracting Straight Lines based on Differences Process, 2008 International Conference on Computer Science and Software Engineering Computer Science and Software Engineering, Volume 2, 2008.
- [6] Nguyen U. T. V., Bhuiyan A., Park L. A., Ramamohanarao K., An effective retinal blood vessel segmentation method using multi-scale line detection, *Pattern Recognition*, March 2013.
- [7] Yetgin O. E., Gerek O. N., PLD: Power line detection system for aircrafts, *International Artificial Intelligence and Data Processing Symposium (IDAP) Artificial Intelligence and Data Processing Symposium (IDAP)*, Sept. 2017.
- [8] Guang Z., Jinwei Y., I-Ling Y., Farokh B., Robust Real-Time UAV Based Power Line Detection and Tracking, *IEEE International Conference on Image Processing (ICIP)*, Sept. 2016, pp. 744-748.
- [9] Candamo J., Kasturi R., Goldgof D., Sarkar S., Detection of Thin Lines using Low-Quality Video from Low-Altitude Aircraft in Urban Settings, *IEEE Transactions on Aerospace and Electronic Systems* Volume 45 Number 3, July 2009.
- [10] Karakose, E., Performance evaluation of electrical transmission line detection and tracking algorithms based on image processing using UAV, *International Artificial Intelligence and Data Processing Symposium (IDAP)*, 1-5 Sept. 2017.
- [11] Weiran C, Linlin Z., Jianda H., Tianran W., Yingkui D., High voltage transmission line detection for uav based routing inspection, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics Advanced Intelligent Mechatronics (AIM)*, July 2013, pp. 554-558.
- [12] Open Source Computer Vision Library, Reference Manual, 2014.

### POWER LINE DETECTION

The paper presents the algorithm for recognizing and locating high voltage power lines with the use of horizontal images. The algorithm has the highest sensitivity to horizontal edges and at the same time is insensitive to vertical edges. The first stage includes an edge detection algorithm. The second stage involves filtering of detected edges and classifying the points for further analysis. Using gradient method, the regions containing noise or background are discarded. The final classification of pixels representing the line is made using the longest path detection algorithm. In the third stage a shape of the wire is approximated based on the set of selected points. Then the contour of the detected wire is constructed with a set of pixels representing interrupted paths, thus increasing the accuracy of the final approximation.

*(Received: 07.02.2018, revised: 23.03.2018)*