Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej Nr 61

Politechnika Gdańska

Wydział Elektrotechniki i Automatyki

Gdańsk 2018

doi: 10.32016/1.61.09

WYKRYWANIE PROSTYCH W OBRAZIE CYFROWYM Z WYKORZYSTANIEM TRANSFORMACJI HOUGHA

Paweł KOWALSKI[1], Robert SMYK[2]

1. Politechnika Gdańska, Wydział Elektrotechniki i Automatyki
   e-mail: pawel.kowalski2@pg.edu.pl
2. Politechnika Gdańska, Wydział Elektrotechniki i Automatyki
   e-mail: robert.smyk@pg.edu.pl

Streszczenie: Artykuł prezentuje eksperymentalną analizę wpływu szumu o założonym poziomie na skuteczność wykrywania prostych w obrazie przy użyciu algorytmu Hougha. Analizę przeprowadzono przy użyciu opracowanej aplikacji obejmującej realizację procedury generacji szumu oraz algorytmu automatycznie wyznaczającego liczbę pikseli w funkcji jasności w przestrzeni Hougha. Zbadano wpływ poziomu szumu na różnicę w liczbach pikseli tworzących prostą wejściową, a prostą tworzoną przez współliniowe piksele szumu.

Słowa kluczowe: przetwarzanie obrazu, transformacja Hougha, wykrywanie linii

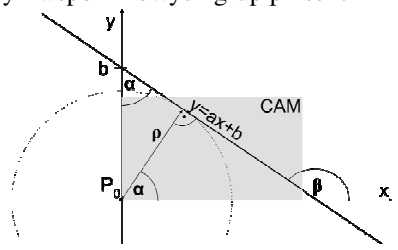
1. WSTĘP

W literaturze funkcjonuje wiele algorytmów dedykowanych do wykrywania linii w obrazie cyfrowym. Istnieje pewna grupa uniwersalnych algorytmów detekcji linii opierających się o algorytm Line Segment Detector [1], do tej grupy zaliczają się również algorytmy bazujące na transformacji Hougha [2,3]. Algorytmy wykrywania linii wykorzystuje się w wykrywaniu przewodów energetycznych w obrazie [4–7], a także mają zastosowanie w medycynie [8] czy też lotnictwie [4].

Jednym z istotnych zagadnień jest monitorowanie napowietrznych linii energetycznych w celu zwiększenia efektywności wykorzystania ich możliwości przesyłowych. Przy monitorowaniu z wykorzystaniem technik wizyjnych bardzo istotne jest efektywne wykrywanie linii w obrazie. Jednym z powodów konieczności gruntownego zbadania właściwości algorytmu Hougha było sprawdzenie jego przydatności w implementacji układowej FPGA, która to będzie stanowiła następny krok przyszłych prac badawczych. Istotne było też zbudowanie własnego prototypu programowego implementacji tego algorytmu oraz przebadanie jego odporności na szum przy wykrywaniu linii prostych.

2. TRANSFORMACJA HOUGHA

Obraz cyfrowy może być zapisany w postaci wektorowej, fraktalnej lub rastrowej. W reprezentacji wektorowej obraz złożony jest z obiektów takich jak odcinki, figury, krzywe opisanych za pomocą równań matematycznych [9]. Obraz fraktalny tworzą obiekty opisane zależnościami rekurencyjnymi [9]. Obraz rastrowy generowany jest na wyjściu urządzeń rejestrujących. Jest on złożony z punktów zwanych pikselami. Z każdym pikselem związana jest informacja o jego położeniu oraz wartości. Położenie określane jest przez indeksy kolumny x oraz wiersza y. Wartość w zależności od formatu może zawierać kilka składowych koloru (np. RGB) w przypadku obrazu kolorowego lub jedną (jasność) w przypadku obrazu w skali odcieni szarości. Szczególnym przypadkiem obrazu w skali odcieni szarości jest obraz binarny, w którym piksel może przyjąć jedną z dwóch wartości. W niniejszym artykule jako wartości piksela obrazu binarnego przyjęto 1 jako biały oraz 0 jako czarny. Obraz binarny wykorzystywany jest podczas przetwarzania do zapisu lokalizacji ważnych cech obrazu, np. wykrytych krawędzi [10], gdzie 1 oznacza wykrytą krawędź, a 0 brak krawędzi. Tego typu zbiór pikseli charakteryzuje się znacznie mniejszą zawartością informacji w porównaniu do obrazu RGB, co umożliwia wzrost efektywności przy implementacji skomplikowanych algorytmów. Jednym z takich algorytmów jest wykrywanie prostych, czyli współliniowych grup pikseli.



Rys. 1. Przykład linii prostej w obrazie cyfrowym

Rysunek 1 przedstawia prostą widoczną w obrazie rastrowym *CAM*. Kadr obrazu został wpisany w układ współrzędnych w taki sposób, że punkt (0,0) pokrywa się z rogiem obrazu, współrzędna x z numerem kolumny pikseli obrazu *CAM*, a współrzędna y z numerem wiersza obrazu *CAM*. W takim przypadku piksele obrazu *CAM* leżące na prostej spełniają równanie ogólne

$$y=ax+b\,, \tag{1}$$

gdzie a to kąt nachylenia, a b punkt przecięcia prostej z osią y. Kąt nachylenia wynika ze znanej zależności

$$a = tg\beta = -ctg\alpha\,. \tag{2}$$

Po podstawieniu (2) do równania ogólnego (1) otrzymuje się

$$y = -ctg\alpha \cdot x + b\,, \tag{3}$$

a po przekształceniu mamy

$$b = y + ctg\alpha \cdot x\,. \tag{4}$$

Do wyznaczenia kąta α można wykorzystać wartości ρ oraz b według zależności

$$\sin\alpha = \frac{\rho}{b}\,. \tag{5}$$

Po uporządkowaniu otrzymujemy odległość prostej od początku układu współrzędnych

$$\rho = \sin(\alpha) \cdot b\,, \tag{6}$$

gdzie po podstawieniu (4) do (6) mamy

$$\rho = \sin\alpha \cdot (y + ctg\alpha \cdot x)\,. \tag{7}$$

Uporządkowanie zależności (7) prowadzi do równania prostej w postaci normalnej

$$\rho = \sin\alpha \cdot y + \cos\alpha \cdot x\,. \tag{8}$$

Piksele leżące na określonej prostej można opisać za pomocą równania (1) lub (8) [2,3]. W obu przypadkach do zapisu prostej wykorzystywane są dwa argumenty a i b lub α i ρ. Biorąc pod uwagę prostą widoczną w obrazie *CAM*, przy zapisie w postaci ogólnej (1) a oraz b może przyjmować wartości z zakresu (-∞; +∞). W reprezentacji normalnej kąt α przyjmuje wartości z zakresu [0°; 360°). Odległość prostej od punktu P_0 jest nie większa niż odległość widocznego odcinka tej prostej od punktu P_0. Można więc wyznaczyć maksymalną wartość ρ jako odległość P_0 od najbardziej oddalonego punktu znajdującego się w *CAM*, odległość ta jest równa przekątnej kadru *CAM*. Oba współczynniki (α, ρ) mogą przyjmować skończone wartości, pozwala to na zapisanie prostej w postaci punktu (α, ρ) w dwuwymiarowej przestrzeni ograniczonej przez kąt 360° oraz odległość równą przekątnej ekranu.

3. IMPLEMENTACJA TRANSFORMACJI HOUGHA

Punkty (α, ρ) reprezentujące proste należą do przestrzeni R^2. W zrealizowanej implementacji parametry α oraz ρ zaokrąglane są do liczb całkowitych. Pozwala to na użycie ich do adresowania komórek macierzy o rozmiarze $\max(\rho) \times \max(\alpha)$. Macierz taka może posłużyć do przeprowadzenia procedury będącej pewnego rodzaju głosowaniem. Prezentacja wyników głosowania zrealizowana została poprzez utworzenia obrazu rastrowego, gdzie numer wiersza określa wartość parametru α, a numer kolumny wartość parametru ρ.

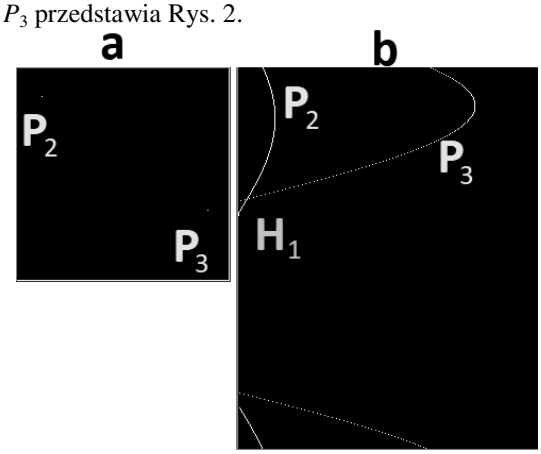
3.1. Głosowanie

W przestrzeni Hougha za pomocą jednego punktu można zapisać prostą z przestrzeni kartezjańskiej. Wykrywanie prostych realizowane jest z wykorzystaniem macierzy liczności, będącej swoistym rankingiem, w którym każdy piksel głosuje na przechodzące przez niego proste. Zestaw prostych generowanych jest na podstawie współrzędnych punktu z przestrzeni kartezjańskiej oraz zestawu wartości α. W przeprowadzonym eksperymencie uwzględniono wszystkie naturalne wartości α z zakresu [0°;360°). Wyznaczone proste zapisywane są w macierzy liczności w przestrzeni Hougha. Zapisanie prostej oznacza zwiększenie o jeden wartości komórki (α, ρ). W tak stworzonej macierzy wartość komórki o współrzędnych (α, ρ) odpowiada ilości pikseli leżących na prostej zdefiniowanej parametrami α, ρ. Kod funkcji *ToHough* odpowiadającej za budowę macierzy liczności został przedstawiony poniżej.

Algorytm 1. Funkcja ToHough

```
void ToHough(int x, int y, Mat hough){
  for(int alpha=0; alpha<180; alpha+=1){
    p=y*sin(alpha *CV_PI/180.0)
        +x*cos(alpha*CV_PI/180.0);
    if(p<0)
      hough.at<unsigned char>(alpha+180,-p)++;
    else
      hough.at<unsigned char>(alpha, p)++;
  }
}
```

Funkcja *ToHough* pozwala na transformację wybranego punktu (x,y) z przestrzeni kartezjańskiej do przestrzeni Hougha na podstawie zależności (8). Parametry formalne funkcji to współrzędne punktu - x, y oraz macierz punktów w przestrzeni Hougha - *hough*. Działanie funkcji polega na wyznaczeniu p na podstawie (8) dla zadanego kąta *alpha*, gdzie *alpha* przyjmuje wartości naturalne z zakresu [0°;180°). W przypadku ujemnej wartości p do tablicy zapisywany jest punkt (*alpha*+180°, -p), w przypadku dodatniej (*alpha*, p). Zapewnia to naturalne współrzędne komórek w macierzy Hougha oraz wyklucza potrzebę sprawdzania kątów z przedziału (180°;360°). Do graficznej prezentacji wyników wykorzystano obrazy rastrowe, w których punktem o współrzędnych (0,0) jest lewy górny róg obrazu. Współrzędna x odpowiada numerowi kolumny (numery rosną w prawą stronę), a współrzędna y numerowi wiersza (numery rosną w dół obrazu). W przypadku graficznej reprezentacji przestrzeni Hougha przyjęto, że odległość ρ odpowiada numerowi kolumny, a kąt α wyrażony w stopniach numerowi wiersza. Efekt działania funkcji *ToHough* dla obrazu zawierającego dwa punkty P_2 oraz P_3 przedstawia Rys. 2.



Rys. 2. Przykład transformacji dwóch punktów z przestrzeni kartezjańskiej do przestrzeni Hougha: a) obraz wejściowy, b) obraz po transformacji

Rysunek 2 przedstawia wynik transformacji punktów z przestrzeni kartezjańskiej do przestrzeni Hougha. Na Rys. 2a pokazano binarny obraz wejściowy zawierający dwa białe punkty P_2 oraz P_3 na czarnym tle. Wszystkie proste (o naturalnym kącie α) przechodzące przez punkt P_2 lub P_3

zostały przeniesione do przestrzeni Hougha i zapisane w postaci punktów. Wynik transformacji przedstawiono na Rys. 2b, gdzie zestawy punktów w przestrzeni Hougha utworzyły dwie krzywe. Krzywe przecinają się w punkcie H_1. Punkt ten wyznacza prostą przechodzącą przez P_2 i P_3 z Rys. 2a.

3.2. Transformacja odwrotna

Transformacja odwrotna pozwala zaprezentować wykrytą prostą na płaszczyźnie kartezjańskiej. W tym celu należy przekształcić punkt (α, ρ) z przestrzeni Hougha na postać ogólną prostej (1). Współczynnik a wyznaczany jest z wykorzystaniem formuły (2), natomiast b przy użyciu następującej formuły

$$b = \frac{\rho}{\sin(\alpha)} \quad . \tag{9}$$

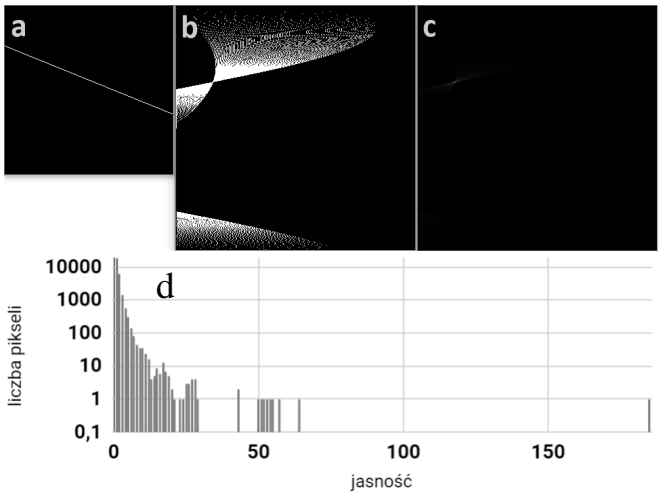
Po podstawieniu (2) oraz (9) do (1) otrzymujemy

$$y = -ctg(\alpha) \cdot x + \frac{\rho}{\sin(\alpha)} \, . \tag{10}$$

Punkty w przestrzeni kartezjańskiej spełniające zależność (10) leżą na prostej wyznaczonej przez punkt (α, ρ) w przestrzeni Hougha.

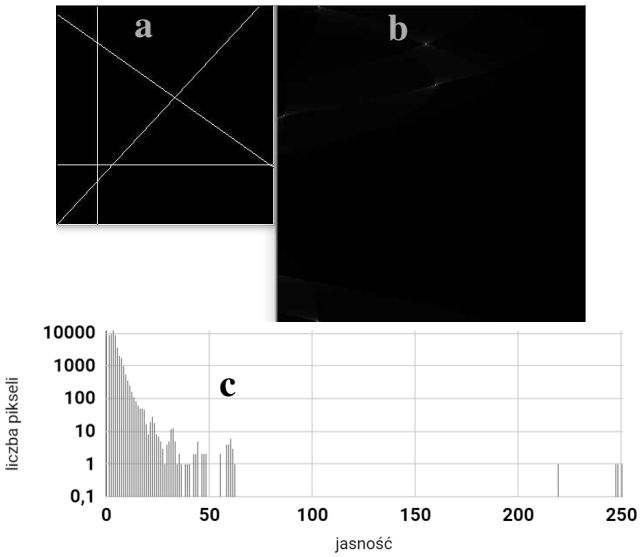
4. OPIS EKSPERYMENTU

Eksperyment zrealizowano stosując algorytm głosowania w przestrzeni Hougha, który został zaimplementowany w aplikacji testowej. Kod aplikacji zrealizowano w języku C++ z wykorzystaniem biblioteki OpenCV [11]. Wykrywanie prostych odbywa się na podstawie obrazu binarnego. Użyte w eksperymencie obrazy z prostymi zostały zaszumione na poziomie od 1% do 99% szumem typu sól i pieprz, następnie przebadana została skuteczność wykrywania tych prostych.



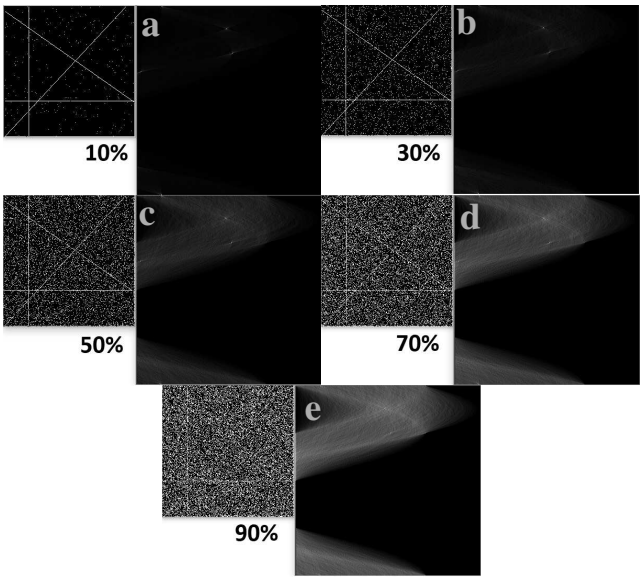
Rys. 3. Transformacja prostej z przestrzeni kartezjańskiej do punktu w przestrzeni Hougha:
a) obraz wejściowy, b) obraz po transformacji do binarnej macierzy w przestrzeni Hougha, c) obraz po transformacji do przestrzeni Hougha, d) liczba pikseli w funkcji jasności punktu w przestrzeni Hougha

Rysunek 3 przedstawia prostą w postaci grupy pikseli w przestrzeni kartezjańskiej (Rys. 3a). Dla każdego białego piksela został wygenerowany zestaw prostych przez niego przechodzących. Proste te zostały zapisane w przestrzeni Hougha w postaci punktów (Rys. 3b). Rys. 3b przedstawia przestrzeń Hougha w postaci obrazu binarnego, gdzie 1 (biały piksel) oznacza istnienie co najmniej jednego piksela przez który przechodzi prosta (α, ρ), a 0 (czarny piksel) oznacza, że dana prosta nie przechodzi przez żaden piksel obrazu z Rys. 3a. Na Rys. 3c jasność piksela reprezentuje liczbę pikseli z Rys. 3a przez które przechodzi dana prosta (wynik działania funkcji *ToHough*). Rys. 3d przedstawia wykres liczebności pikseli o danej jasności w przestrzeni Hougha, gdzie jeden piksel jest znaczenie jaśniejszy od pozostałych, posiada on jasność na poziomie 200. Piksel ten wyznacza prostą przechodzącą przez ok. 200 punktów, jest to prosta z Rys. 3a.

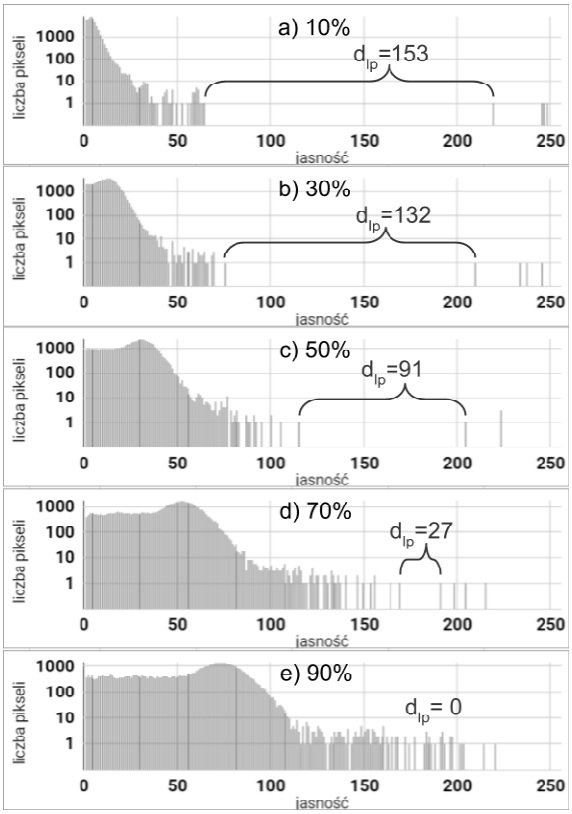


Rys. 4. Transformacja 4 prostych z przestrzeni kartezjańskiej do przestrzeni Hougha:
a) obraz wejściowy, b) obraz po transformacji do przestrzeni Hougha, c) liczba pikseli w funkcji jasności punktu w przestrzeni Hougha

Rysunek 4 przedstawia wynik transformacji obrazu (250x250 px) zawierającego 4 proste. Po przekształceniu obrazu do przestrzeni Hougha, ich równania można wyznaczyć na podstawie czterech pikseli o największych wartościach (najjaśniejsze punkty).

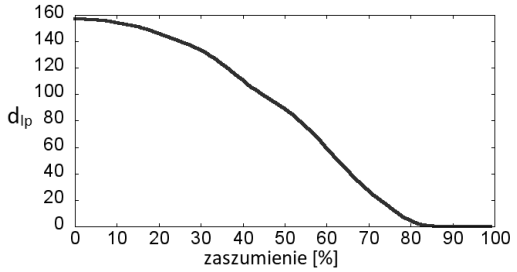


Rys. 5. Przykład transformacji obrazu o stopniu zaszumienia:
a) 10%, b) 30%, c) 50%, d) 70%, e) 90%



Rys. 6. Rozkład jasności pikseli w przestrzeni Hougha w zależności od stopnia zaszumienia

Rysunek 5 przedstawia wynik transformacji obrazów zaszumionych, a Rys. 6 rozkład jasności pikseli w przestrzeni Hougha oraz różnicę d_{lp} w liczbie pikseli tworzących najliczniejszą prostą powstałą z szumu oraz najmniej liczną prostą wejściową (Rys. 4a). W przypadku szumu na poziomie 70% (Rys. 6d) można z powodzeniem odczytać 4 punkty wyznaczające proste. Przy zaszumieniu na poziomie 90% (Rys. 6e) nadal dwa punkty o najwyższej jasności wyznaczają dwie proste z obrazu wejściowego, jednak pozostałe proste zostały zasłonięte szumem.



Rys. 7. Porównanie wpływu zaszumienia na d_{lp}

Wykres zaprezentowany na Rys. 7 przedstawia zależność różnicy d_{lp} od stopnia zaszumienia obrazu wejściowego. Przy szumie równym 70% d_{lp} wynosi 27. Aby znaleźć proste z obrazu wejściowego należy wybrać 4 komórki o najwyższych wartościach. W przypadku szumu przekraczającego 85% d_{lp} spada do 0, co oznacza że liczba pikseli przypadająca na najmniej liczną prostą wejściową nie jest większa niż liczba współliniowych pikseli szumu.

5. PODSUMOWANE

W artykule przedstawiono sposób transformacji punktu z przestrzeni kartezjańskiej do zestawu punktów w przestrzeni Hougha. Przeprowadzono eksperyment polegający na określeniu skuteczności działania transformacji Hougha przy zadanym poziomie szumu.

6. BIBLIOGRAFIA

1. Gioi R. G., 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, 2010, s. 722–32.
2. Duda R. O., Hart P. E.: Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM, 1972 s. 11–5.
3. Illingworth J., Kittler J.: A survey of the hough transform, Computer Vision, Graphics, and Image Processing, 1988 s. 87–116.
4. Yetgin O. E., Gerek O. N.: PLD: Power line detection system for aircrafts, International Artificial Intelligence and Data Processing Symposium (IDAP), 2017.
5. 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), 2016, s. 744-748.
6. 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, 2009.
7. 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), 2017.
8. 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, 2013.
9. Montusiewicz J: Zastosowanie dwuwymiarowej grafiki wektorowej i fraktalnej w projektowaniu, Postępy Nauki i Techniki, 2012, s. 47–60.
10. Kowalski P, Czyżak M.: Algorytmy wykrywania krawędzi w obrazie, Poznan University Of Technology Academic Journals Electrical Engineering, Poznań 2018, s. 243–54.
11. Open Source Computer Vision Library, Reference Manual, 2014.

STRAIGHT LINES DETECTION IN DIGITAL IMAGE USING HOUGH TRANSFORM

The paper experimentally analyzed the impact of noise level on the efficiency of straight lines detection using the Hough algorithm. The analysis was carried out in the own application containing the noise generation procedure and the algorithm that automatically determines the number of pixels as a function of brightness in Hough space. The impact of noise level on the difference in the number of pixels of input straight lines, and lines generated from noise was analyzed.

Keywords: image processing, Hough transform, line detection.