

## SYSTEM MONITORINGU POMIESZCZEŃ ZAMKNIĘTYCH OPARTY NA ZESPOLE KAMER INTERNETOWYCH

Urszula ORSZULAK, Sylwia BABICZ<sup>1</sup>

Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Katedra Metrologii i Optoelektroniki, tel.: +48 58 347 18 86 e-mail: sylwia.babicz@eti.pg.edu.pl

**Streszczenie:** Artykuł opisuje system do monitoringu pomieszczeń zamkniętych oparty na zespole kamer internetowych. System umożliwia rejestrację obrazu za pomocą trzech kamer internetowych w momencie wykrycia ruchu i przekazuje w czasie rzeczywistym nagrany film użytkownikowi na dowolne urządzenie posiadające system Android. Dzięki takiemu rozwiązaniu można kontrolować dane pomieszczenie np. pokój dziecka, biuro czy też sklep. Umożliwia to podjęcie szybszej reakcji w z związku detekcją poruszającego się obiektu. System, po wykryciu ruchu w pomieszczeniu, rejestruje przez określony czas sekwencje obrazów. Poza opracowaniem odpowiedniego kodu analizującego obraz z kamery internetowej składa się z odpowiedniej infrastruktury sprzętowej oraz aplikacji przeznaczonej dla systemu Android.

**Słowa kluczowe:** system alarmowy, detekcja krawędzi, analiza obrazu.

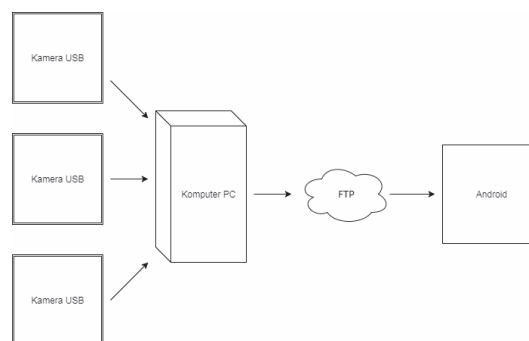
### 1. WPROWADZENIE

Opisywany system ma za zadanie przetwarzać obraz pochodzący z kamery (lub kilku kamer) internetowych i za pomocą odpowiednich algorytmów wykrywać ewentualny ruch w pomieszczeniu. Jedną z najprostszych metod wykrywania ruchu jest takie przetworzenie obrazu, które pozwoli na wykrycie zmian w zarejestrowanym obrazie tła. Należy przy tym pamiętać, że obrazem tła mogą być: regały z książkami, szafy, ściany z obrazami i oknami itp. Ponadto, ze względu na warunki atmosferyczne oraz obecność sztucznych źródeł światła natężenie oświetlenia w pomieszczeniu jest zmienne. Prezentowana metoda polega na wykryciu charakterystycznych krawędzi w obrazie tła (referencyjnym) i porównywanie ich z aktualnym obrazem pochodzącym z kamery. W przypadku wykrycia naruszenia krawędzi obraz z kamery jest rejestrowany i przechowywany w chmurze danych, co pozwala użytkownikowi na bieżąco reagować w sytuacji naruszenia monitorowanego pomieszczenia.

### 2. REALIZACJA SYSTEMU

W systemie wykorzystano trzy kamery internetowe USB (Microsoft LifeCam HD-3000 o rozdzielczości 1280x720 px, dwie kamery Selecline nagrywające film o rozdzielczości 640x480 px), jednostkę przetwarzającą obraz i przesyłającą dane na serwer FTP (ang. *File Transfer Protocol*) (którym jest komputer PC) i urządzenie wykorzystujące system Android (rys. 1). Za pomocą kamer USB rejestrowane są obrazy otoczenia. Następnie zostają

one poddane odpowiedniej analizie w celu wykrycia ruchu w nagrywanej sekwencji wideo. Jeżeli zostanie zauważony ruch następuje nagrywanie. W dalszym kroku rozpoczyna się przekazanie materiału na serwer FTP. Ostatnim elementem jest urządzenie z systemem Android, na którym można odebrać uzyskany materiał i go odtworzyć.



Rys. 1. Schemat ideowy systemu monitoringu

W ramach przetwarzania obrazu przetestowano trzy metody wykrywania krawędzi: operator Laplace'a, Sobela i algorytm Canny.

#### 2.1. Wybrane algorytmy krawędziowe

Krawędź stanowi granicę między dwoma regionami o różnych odcieniach jasności [1, 2]. Wykrywanie jej często jest dokonywane przy użyciu filtrów gradientowych. Krawędzie mogą być wykryte dzięki lokalizacji lokalnego ekstremum pierwszej pochodnej obrazu lub poprzez wybór miejsc zerowych drugiej pochodnej. W momencie, kiedy wykorzystuje się techniki oparte na gradiencie, często stosuje się także progowanie w celu usunięcia szumów. Wykorzystuje się także wygładzenie gaussowskie zwłaszcza, kiedy stosujemy algorytmy oparte na drugiej pochodnej, takie jak np. laplasjan.

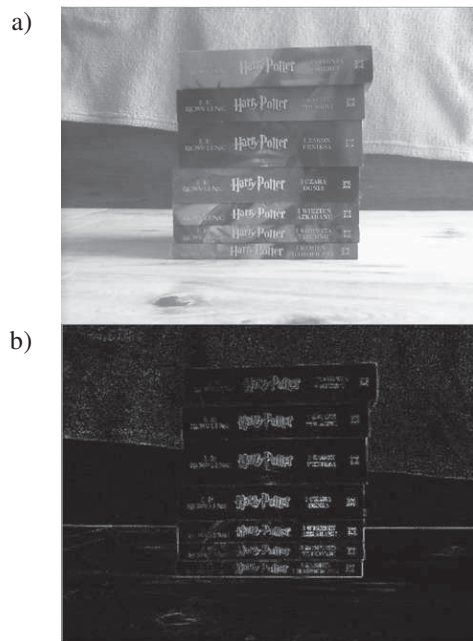
Laplasjan ciągłej funkcji dwóch zmiennych  $J(x,y)$  jest wyrażany za pomocą wzoru (1) [3]:

$$L[J(x, y)] = \frac{\partial^2 J(x, y)}{\partial x^2} + \frac{\partial^2 J(x, y)}{\partial y^2} \quad (1)$$

Jak widać, jest on sumą drugich pochodnych cząstkowych. Zastąpienie ciągłej pochodnej skończonymi przyrostami doprowadza do dyskretnej postaci wzoru i w efekcie maski Laplace'a (2) [4]:

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad (2)$$

Operator Laplace'a jest wrażliwy na szumy (rys. 2). Z tego względu przed jego zastosowaniem stosuje się wygładzanie obrazu operatorem Gaussa. W dodatku operator Laplace'a może produkować podwójny kontur, a kiedy jasność obrazu jest stała, staje się niewrażliwy na przestrzenną orientację krawędzi. Ze względu na te problemy opracowano metodę opartą na filtracji przy użyciu jądra Gaussa [4]. Rozwiązanie to zakłada, że stworzony operator można stroić dla określonej skali – do odpowiedniego poziomu detekcji krawędzi. Takie podejście umożliwia stworzenie opracowanie mapy krawędzi jako funkcji skali krawędzi.



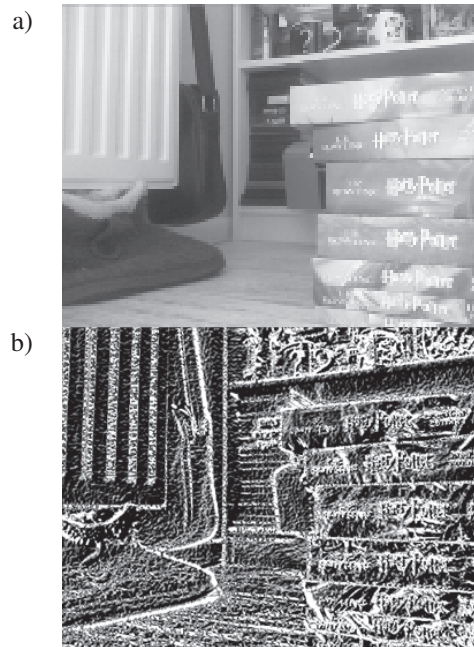
Rys. 2. a) Obraz oryginalny, b) efekt działania maski Laplace'a

Drugim rozważanym operatorem jest operator Sobela [5] (rys. 3), który podobnie jak maski Robertsa, umożliwia aproksymację pochodnych kierunkowych intensywności w ośmiu kierunkach, każdy co 45°. Maski przy filtracji Sobela są antysymetryczne w stosunku do kierunku wykrywanych krawędzi, zatem wystarczy znać maski dla 0°, 45°, 90° i 135°, gdyż pozostałe różnią się tylko znakiem (otrzymuje się je obracając podstawowe o 180°) (2):

$$\begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 \\ -2 & 0 & 2 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & -2 & -1 & 0 & -1 & -2 & -1 & 0 & -1 & -2 \end{pmatrix} \quad (2)$$

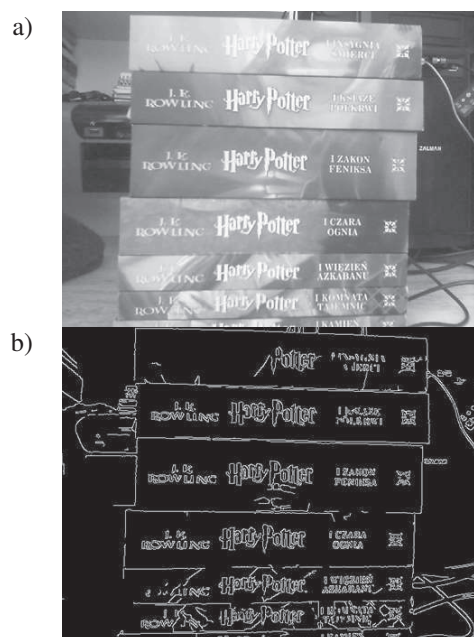
Ostatnim rozważanym sposobem wykrywania krawędzi jest algorytm Canny [6]. Jego autor, John Canny opublikował pracę [7], w której wyznaczył kryteria poprawnego wykrycia krawędzi w obrazie: dobra detekcja, dobra lokalizacja i odpowiedź tylko jednego operatora dla jednej rzeczywistej krawędzi. W przypadku algorytmu Canny'ego pierwszym etapem jest redukcja szumów filtrem Gaussa, a następnie wyznaczenie pierwszej pochodnej obrazu, jako pochodnej tej funkcji. Oba działania wykonywane są z wykorzystaniem masek

jednowymiarowych działających oddzielnie w kierunku pionowym i poziomym. W dalszym kroku „tłumione” są wartości nie-maksymalne (ang. *non-maximum suppression*). Jeżeli wartość odpowiedzi krawędzi będzie większa od wartości pikseli sąsiednich, leżących w tym samym kierunku, to piksel będzie pikselem krawędziowym. Końcowy etap to procedura binaryzacji z histerezą z wykorzystaniem dwóch wartości progów: wysokiego oraz niskiego (autor zaleca, aby wartość progu niskiego wynosiła od 33% do 50% progu wysokiego).



Rys. 3. a) Obraz oryginalny, b) efekt działania maski Sobela o orientacji x i y

Rezultaty działania algorytmu opartego na metodzie Canny'ego (rys. 4) zależą od efektywności użytego filtra gaussowskiego oraz rozmiaru okna, wartości odchylenia standardowego oraz progów. Wszystkie te parametry muszą być dopasowane do przetwarzanego obrazu.



Rys. 4. a) Obraz oryginalny, b) efekt działania algorytmu Canny

## 2.2. Wykrywanie ruchu w obrazie

Najprostszą i najczęściej stosowaną metodą wyszukiwania ruchu w obrazie jest sprawdzanie różnicy dwóch sąsiadujących kadrów. Obraz zarówno przed jak i po takim zabiegu powinien być odpowiednio przygotowany tzn. poddany binaryzacji i odpowiedniej filtracji. Ponadto należy zdecydować, czy dana zmiana w obrazie jest istotna.

Innym podejściem do wykrywania ruchu w obrazie na podstawie różnicy ramek jest wykorzystanie trzech kadrów [8, 9]. W tej metodzie zostają pobrane trzy kolejne ramki z pliku wideo. Następnie na podstawie różnicy pomiędzy nimi uzyskuje się kontur poruszającego się obiektu. Podejście to pozwala na usunięcie zakłóceń pochodzących z tła.

Oprócz wymienionych, można ponadto wyróżnić metodę MoG (ang. *Mixture of Gaussians*) [10, 11] oraz PSNR (ang. *Peak Signal to Noise Ratio*). Pierwsza wykorzystuje algorytm EM (ang. *Expectation – Maximization*) segregujący piksele na trzy grupy: tło (o najmniejszej wariancji), obiekt poruszający się (o największej wariancji), elementy tła (o średniej wartości wariancji). Druga dzieli piksele na podstawie współczynnika szczytowego stosunku sygnału do szumu.

Ze względu na konieczność wykorzystania szybkiego algorytmu działającego w czasie rzeczywistym zdecydowano się skorzystać z najprostszego rozwiązania realizującego różnicę między dwoma sąsiednimi kadrami.

## 2.3. Oprogramowanie

Oprogramowanie służące do analizy obrazu zostało podzielone na trzy projekty. Każdy z nich realizuje wykrywanie krawędzi za pomocą różnych algorytmów krawędziowych. Do ich realizacji została wykorzystana biblioteka OpenCV, zaś kod został napisany w języku Python w środowisku Microsoft Visual Studio 2015.

## 3. TEST OPRACOWANEGO SYSTEMU

W celu przetestowania wybranych algorytmów krawędziowych do realizacji systemu monitoringu pomieszczenia nagrano trzema kamerami internetowymi trzy niezależne ujęcia tej samej sytuacji, tj. pojawienie się zwierzęcia w monitorowanym pokoju. Następnie przeanalizowano zachowanie wykorzystanych algorytmów na zarejestrowanych ujęciach.

Testy przeprowadzono w naturalnie oświetlonym pomieszczeniu w słoneczny dzień oraz w nocy. Przeprowadzono 4000 testów dla każdego z rozpatrywanych algorytmów: po dwa tysiące w dzień i w nocy. W trakcie testów zapewniono nieustająco poruszający się obiekt w analizowanym kadrze w dzień i w nocy. Dokonano po tysiąc pomiarów dla obu warunków oświetleniowych. Co 66 ms sprawdzano obecność intruza. Za test braku obecności intruza przyjęto również po tysiąc pomiarów w obu warunkach oświetleniowych, każdy co 66 ms przy zapewnionym braku ruchu w pomieszczeniu. Efekt przeprowadzonych testów prezentuje Tablica 1.

Na podstawie przeprowadzonych testów można wnioskować, że operator Sobela, szczególnie w przestrzeni kolorów HSV, jest bardzo czuły i wykrywa ruch praktycznie za każdym razem nawet, kiedy ten ruch nie występuje (999 przypadków na 1000). Innymi słowy, algorytm wykorzystujący o operator Sobela niemal zawsze dokonuje detekcji ruchu. Nie można zatem uznać go za wiarygodnego również w trakcie testów w obecności obiektu ruchomego.

Przeciwnieństwem operatora Sobela, jest algorytm Canny. Algorytm Canny praktycznie nie dokonywał błędnej detekcji ruchu zarówno w dzień, jak i w nocy. Niestety, jego skuteczność w obecności ruchomego obiektu jest niezadowolająca, gdyż oscyluje wokół ok. 50% dla przestrzeni szarości i HSV. Zdecydowanie lepiej funkcjonuje on w przestrzeni barw RGB. W tym przypadku błędne wykrycie ruchu w dzień może zostać zaniedbane.

Zdecydowanie w testach najlepiej wypadł operator Laplace'a, gdzie dodano progowanie zwiększające poprawność działania algorytmu. Efekt programu wykorzystującego operator Laplace'a w przestrzeni szarości jak i HSV jest porównywalny. O ile w przestrzeni szarości rzadziej wykrywany jest błędnie ruch (1 raz na 1000 w dzień i 27 razy na 1000 w nocy), o tyle w przestrzeni HSV poprawna detekcja ruchu w nocy jest minimalnie lepsza (979 na 1000, a nie jak w przypadku przestrzeni szarości 973 razy na 1000). Jednocześnie algorytm wykazuje bardzo dużą skuteczność w obu przestrzeniach kolorów zarówno w dzień, jak i w nocy. Co więcej, operator Laplace'a nie wykrywa błędnie ruchu. Rozpatrując tylko ten rodzaj algorytmu, idealnym byłoby połączenie dwóch przestrzeni kolorów: zastosowanie skali szarości w dzień i HSV w nocy.

## 4. PODSUMOWANIE

System oparty na kamerach internetowych umożliwia monitorowanie pomieszczenia za pomocą wybranych algorytmów wykrywania krawędzi. Filmy zarejestrowane w wyniku wykrycia ruchu są przesyłane na serwer. Najefektywniejszym algorytmem krawędziowym do prezentowanego zadania okazał się być operator Laplace'a. Program bazujący o operator Laplace'a nie dokonuje błędnej detekcji ruchu, a jego skuteczność w przypadku obecności obiektu ruchomego w kadrze wynosi ponad 96%. W celu ulepszenia działania należałoby poprawić jego skuteczność w warunkach niedoskonałego oświetlenia, np. poprzez rozszerzenie histogramu każdego obrazu bądź też poprzez wymianę kamer internetowych na sprzęt o większej rozdzielczości. Algorytm Canny mógłby mieć zastosowanie tylko w dzień, gdyby wykorzystać go w przestrzeni barw RGB. Zdecydowanie najgorszym okazał się operator Sobela, którego zbyt wysoka czułość powoduje, że wykrywa on ruch niemal cały czas.

W pracy wykorzystano trzy z najbardziej podstawowych sposobów wykrywania krawędzi w celu przetestowania ich skuteczności w niskobudżetowym systemie monitoringu. Warto jednak rozpatrywać także inne, bardziej zaawansowane porównywania obrazów, niekoniecznie analizujące krawędzie. Do takich metod możemy zaliczyć algorytm SAD (ang. *Sum of Absolute Differences*) czy analizę korelacji tła i aktualnego obrazu [12, 13].

## 5. BIBLIOGRAFIA

1. Gatter M.: *Getting It Right in Print: Digital Pre-press for Graphic Designers* Laurence King Publishing, 2004.
2. Galer M., Horvat L.: *Digital Imaging: Essential Skills*, Focal Press, 2003.
3. Malina W., Smiatacz M.: *Cyfrowe przetwarzanie obrazów - Filtry cyfrowe, Filtry liniowe, Filtry wyostrzające, Operator Laplace'a*, Akademicka Oficyna Wydawnictwa EXIT, Warszawa, 2008, s. 74 - 77.

4. Malina W., Smiatacz M.: Cyfrowe przetwarzanie obrazów - Algorytmy krawędziowe, Algorytmy oparte na operatorze Gaussa, Laplasjan Gaussowski, Akademicka Oficyna Wydawnictwa EXIT, Warszawa 2008, s. 98 - 102.
5. Wilk J.: Techniki cyfrowego rozpoznawania krawędzi obrazów, Wydawnictwo Stowarzyszenia Polska - Wschód, Oddział Świętokrzyski, Kielce, 2009.
6. Malina W., Smiatacz M.: Cyfrowe przetwarzanie obrazów - Algorytmy wykrywania krawędzi, Algorytm Canny'ego, Akademicka Oficyna Wydawnictwa EXIT, Warszawa, 2008, s. 106 - 115.
7. Canny, J: A computational approach to edge detection, IEEE Transactions on pattern analysis and machine intelligence, No. 6, 1986, s. 679-698.
8. Ze-xian K., Han-hong J., Chao-liang Z.: Moving objects detection based on three frames differencing and GMM. Advanced Materials Research, Vol. 694-697, 2013, s. 1974-1977.
9. Xiaofeng L., Tao Z., Zaiwen L.: A novel method on moving-objects detection based on background subtraction and three frames differencing, 2010 Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on, Vol. 1, 2012, s. 148-151.
10. Yang K., Cai Z, Zhao L.: Algorithm Research on Moving Object Detection of Surveillance Video Sequence, Institute of Information Science and Engineering, Hebei University of Science and Technology, 2013, s. 308-309.
11. Bouwmans T., El Baf F., Vachon B.: Background Modeling using of Gaussians for Foreground Detection - A Survey, Bentham Science Publishers, 2008.
12. Cheezum M. K., Walker W. F., Guilford W. H.: Quantitative Comparison of Algorithms for Tracking Single Fluorescent Particles, Biophysical Journal, Vol. 81, Iss. 4, 2001, s. 2378-2388.
13. Friemel B. H., Bohs L. N., Trahey G. E.: Relative performance of two-dimensional speckle-tracking techniques: normalized correlation, non-normalized correlation and sum-absolute-difference, Ultrasonics Symposium, Proceedings, IEEE. Vol. 2. IEEE, 1995, s. 1481-1484.

Tablica 1. Zestawienie efektów przeprowadzonych testów

Algorytm			Canny			Laplace		Sobel	
Przestrzeń kolorów			Szarości	RGB	HSV	Szarości	HSV	Szarości	HSV
Ruch	Poprawne wykrycie	Dzień	560	932	516	999	964	999	999
		Noc	418	737	350	973	979	999	999
	Błędne wykrycie ruchu	Dzień	440	68	484	1	36	1	1
		Noc	582	263	650	27	21	1	1
Brak ruchu	Poprawne wykrycie	Dzień	1000	995	1000	1000	1000	1	1
		Noc	1000	1000	998	1000	1000	1000	1
	Błędne wykrycie ruchu	Dzień	0	5	0	0	0	999	999
		Noc	0	0	2	0	0	0	999
Prawdopodobieństwo poprawnej detekcji (ruch - dzień) [%]			56	93,2	51,6	99,9	96,4	99,9	99,9
Prawdopodobieństwo poprawnej detekcji (ruch - noc) [%]			41,8	73,7	35	97,3	97,9	99,9	99,9
Prawdopodobieństwo poprawnej detekcji (brak ruchu - dzień) [%]			100	99,5	100	100	100	0,1	0,1
Prawdopodobieństwo poprawnej detekcji (brak ruchu - noc) [%]			100	100	99,8	100	100	100	0,1

## THE INDOOR MONITORING SYSTEM BASED ON WEBCAMS

The article presents a monitoring system based on a set of webcams. The system allows to record the movie from three webcams at the moment of motion detection in real-time. User have an access to the movies by application on Android device. Three most popular edges detection algorithms were implemented and tested: Laplace's processing, Sobel mask and Canny algorithm. Authors elaborated the three edge detection algorithms and tested their functionality in presence and absence of moving object in different natural lighting conditions – day and night. For each situation (day/night, presence/absence of movement) one thousand single tests were carried out. The detection was conducted every 66 ms. The tests prove that the Laplace's operator is the most effective edge algorithm for this task. A program based on the Laplace operator does not perform erroneous motion detection and its efficiency in the presence of a moving object is over 96%. The Canny algorithm would only apply in the day if it were to be used in the RGB. Worst of all, Sobel's operator, has so high sensitivity that it detects the movement almost all the time event if it not occurs.

**Keywords:** monitoring, edge detection, image processing.