Piotr Szymczyk*, Magdalena Szymczyk*

Analiza sceny przy użyciu deskryptorów punktów charakterystycznych**

1. Wprowadzenie

Analiza sceny polega na wyodrębnieniu z obrazu obiektów oraz ich rozpoznaniu poprzez porównanie z obiektami znajdującymi się w bazie danych. Wyodrębnienie obiektów polega na odszukaniu charakterystycznych cech fragmentów obrazu, takimi kluczowymi elementami mogą być krawędzie, wierzchołki lub punkty charakterystyczne. W artykule zostaną zaprezentowane różne algorytmy stosowane w analizowaniu sceny wraz z przykładami ich działania. Na podstawie eksperymentów zostały sformułowane wnioski dotyczące dalszych badań nad ulepszeniem poszczególnych metod.

2. Przegląd algorytmów

2.1. Algorytmy detekcji krawędzi

Istnieje wiele algorytmów detekcji krawędzi. Są to algorytmy m.in. [15]:

- Sobel,
- Laplace,
- Canny [3],
- SUSAN,
- Prewitt,
- Roberts,
- Sharr,
- Marr-Hildreth.

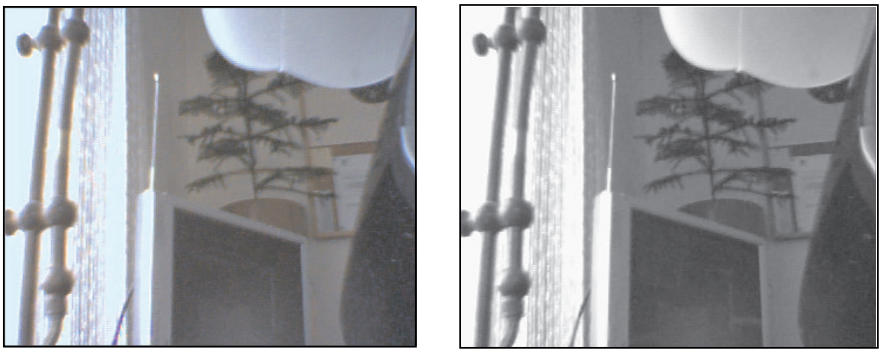
* AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki, al. A. Mickiewicza 30, 30-059 Kraków

** Projekt z Ministerstwa Nauki i Szkolnictwa Wyższego nr 0128/R/t00/2010/12

Algorytm Canny jest najczęściej używaną metodą detekcji krawędzi ponieważ żaden inny algorytm nie wykazał znaczącej przewagi nad tym algorytmem [8]. Algorytm ten przebiega w czterech fazach:

1) Redukcja szumów.
2) Szukanie natężenia gradientu obrazu.
3) Usuwanie niemaksymalnych pikseli.
4) Progowanie z histerezą.

Algorytm Canny działa na obrazie źródłowym, (rys. 1), który jest kodowany w odcieniach szarości. W obrazie wyjściowym (rys. 2) również kodowanym w odcieniach szarości zapisane są wykryte krawędzie. Jako parametry należy określić dwa progi (próg górny P_g, próg dolny P_d) oraz rozmiar macierzy splotu R_{ms}.



Rys. 1. Obraz oryginalny

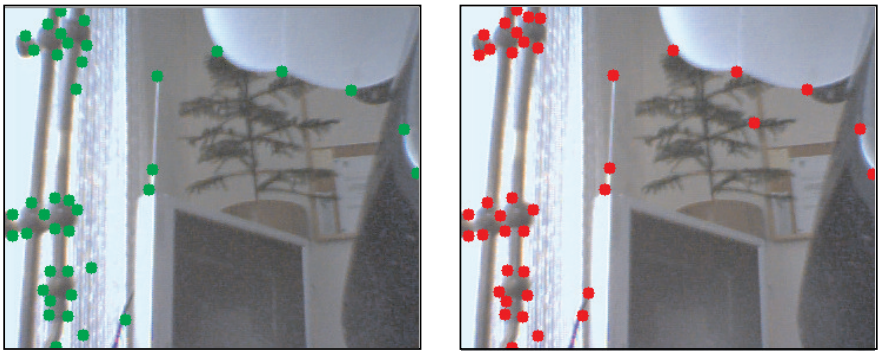
Rys. 2. Wynik działania algorytmu Canny (P_g = 158, P_d = 165, R_{ms}= 3) dla obrazu z rysunku 1

2.2. Algorytm detekcji wierzchołków

Niżej wymieniono wybrane algorytmy wykrywania wierzchołków:

- Moravec,
- Harris-Stephens,
- Shi-Tomasi (Kanade-Tomasi),
- LoG,
- DoG,
- DoH,
- Wang-Brady,
- SUSAN,
- Trojkovic-Hedley,
- AST, FAST,
- ASoD (Trujillo-Olague).

Algorytm Harris i Stephens jest udoskonalonym algorytmem algorytmu Moravec [7]. Natomiast algorytm Shi-Tomasi jest udoskonalonym algorytmem Harris i Stephens [9] (rys. 3).



Rys. 3. Wynik działania algorytmu Shi-Tomasi (z lewej) oraz Harris i Stephens (z prawej) dla obrazu z rysunku 1. Wierzchołki zaznaczono w postaci kropek na obrazie oryginalnym

W algorytmie Harris i Stephens na początku dla każdego piksela wyznaczana jest macierz kowariancji gradientu, a następnie obliczana jest wartość wyrażenia:

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \tag{1}$$

gdzie:

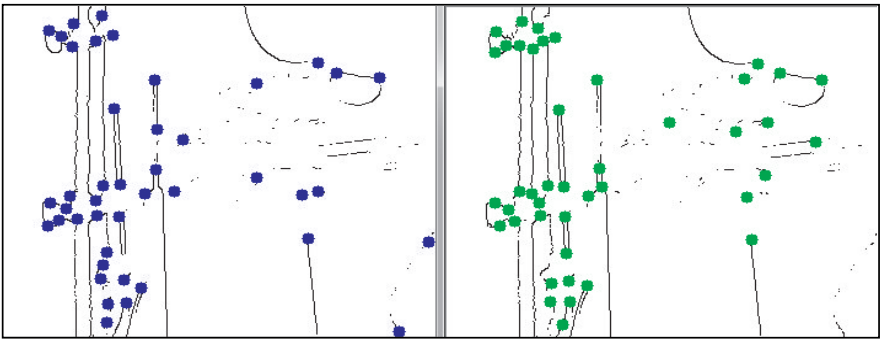
λ_1, λ_2 – wartości własne macierzy kowariancji gradientu,
k – współczynnik o wartości dobieranej doświadczalnie z przedziału od 0,04 do 0,15.

Wierzchołki wyznaczane są na podstawie lokalnych maksimów. W przypadku wierzchołków leżących zbyt blisko siebie algorytm pozostawia ten, który ma większe wartości własne.

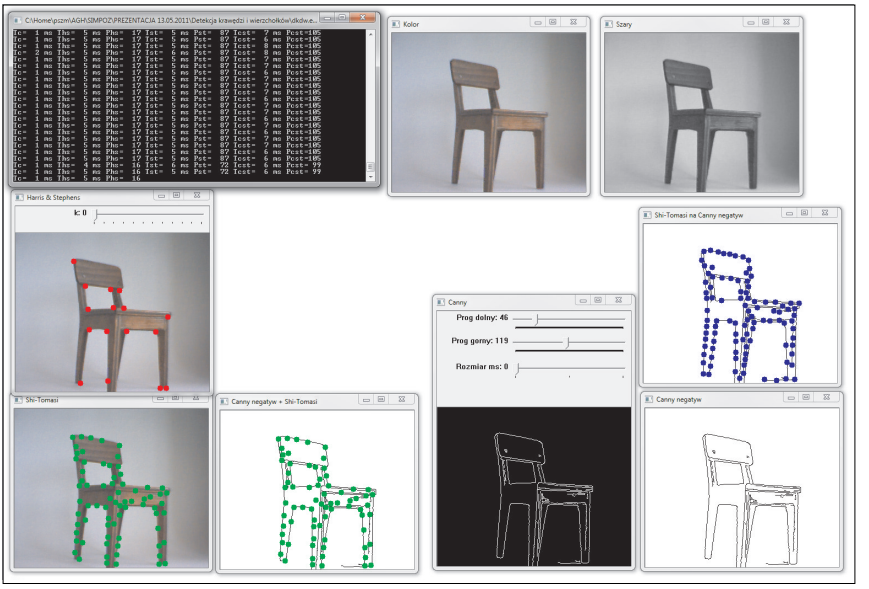
W algorytmie Shi-Tomasi oblicza się jedynie inne wartości na podstawie wyrażenia, które ma następującą postać:

$$R = \min(\lambda_1, \lambda_2) \tag{2}$$

Można zastosować algorytm detekcji wierzchołków Shi-Tomasi dla wcześniej przetworzonego obrazu za pomocą algorytmu detekcji krawędzi Canny, daje to jednak podobne efekty, jedynie można zauważyć wpływ zmiany parametrów algorytmu Canny na liczbę znalezionych wierzchołków. Wynik takiego złożenia pokazano na rysunku 4.



Rys. 4. Wynik złożenia algorytmu detekcji wierzchołków i krawędzi (z lewej) oraz wizualizacja wierzchołków wyznaczona wprost z oryginalnego obrazu pokazana na tle uzyskanym z algorytmu Canny (z prawej)



Rys. 5. Wynik zastosowania algorytmów Canny, Shi-Tomasi oraz Harris i Stephens

Średnie czasy przetwarzania obrazu z rysunku 5 dla algorytmu Canny wynosi 1 ms, Harris i Stephens 5 ms, Shi-Tomasi 5 ms, a dla Shi-Tomasi na obrazie wcześniej przetworzonym algorytmem Canny 7 ms. Są to, jak widać, bardzo krótkie czasy.

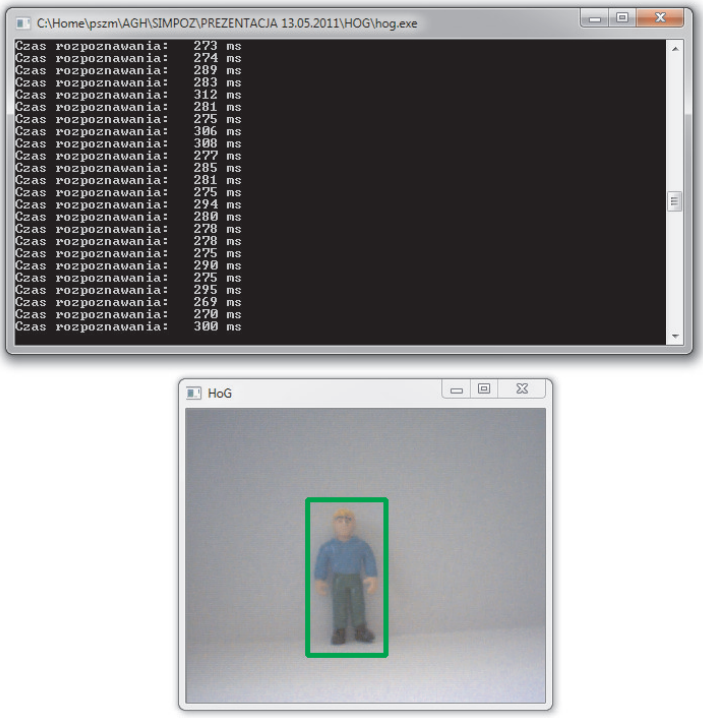
Do testów użyty został sprzęt o następujących parametrach:

– kamera USB kolorowa o rozdzielczości 352×288,
– komputer PC z procesorem Intel Core Duo T9400 2,53 GHz, 4GB RAM z Windows 7, OpenCV 2.2 [2,11,13], Code:Blocks 10.05.

Liczba wierzchołków wykrytych na tym obrazie wynosi odpowiednio, dla algorytmu Harris i Stephens 17, Shi-Tomasi 87, a dla Shi-Tomasi na obrazie wcześniej przetworzonym algorytmem Canny 103. Z tego eksperymentu wynika, że najlepszym rozwiązaniem jest zastosowanie algorytmu dla Shi-Tomasi na obrazie wcześniej przetworzonym algorytmem Canny, ponieważ rozwiązanie takie daje najwięcej wierzchołków.

2.3. Algorytm HoG

Algorytm HoG (*Histogram of Oriented Gradients*)jest bardzo często stosowany z powodzeniem do detekcji osób [4] (rys. 6).

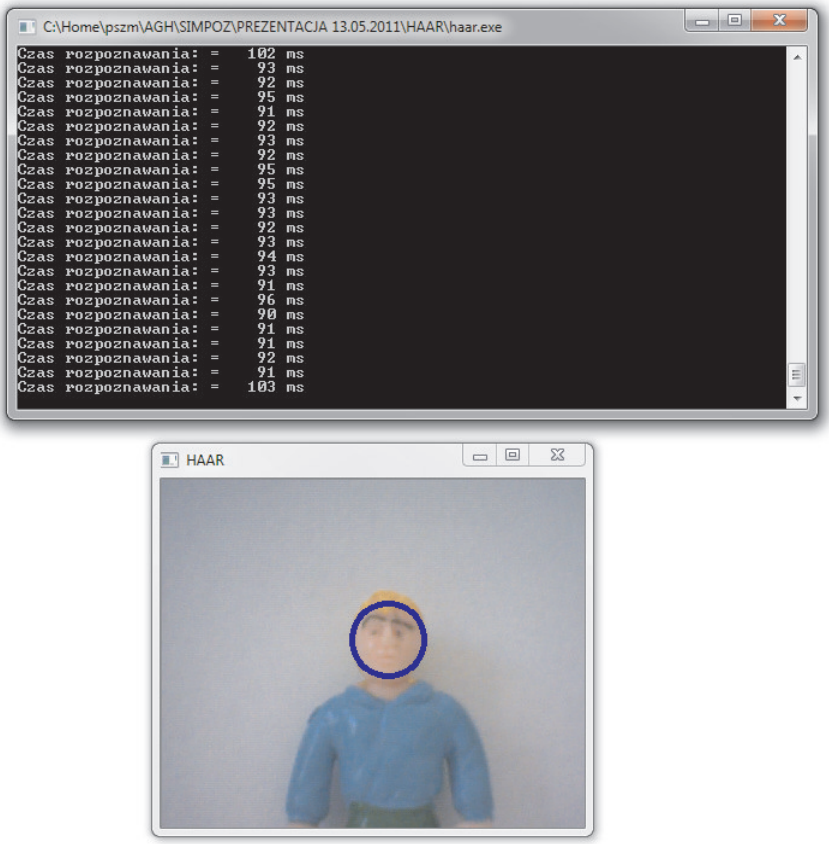


Rys. 6. Działanie algorytmu HoG

Czas przetwarzania wahał się od 269 do 312 ms.

Algorytm HoG bazuje na wykrywaniu kierunków zmian gradientu dla obrazu podzielonego na małe obszary.

2.4. Algorytm Haar



Rys. 7. Działanie algorytmu Haar

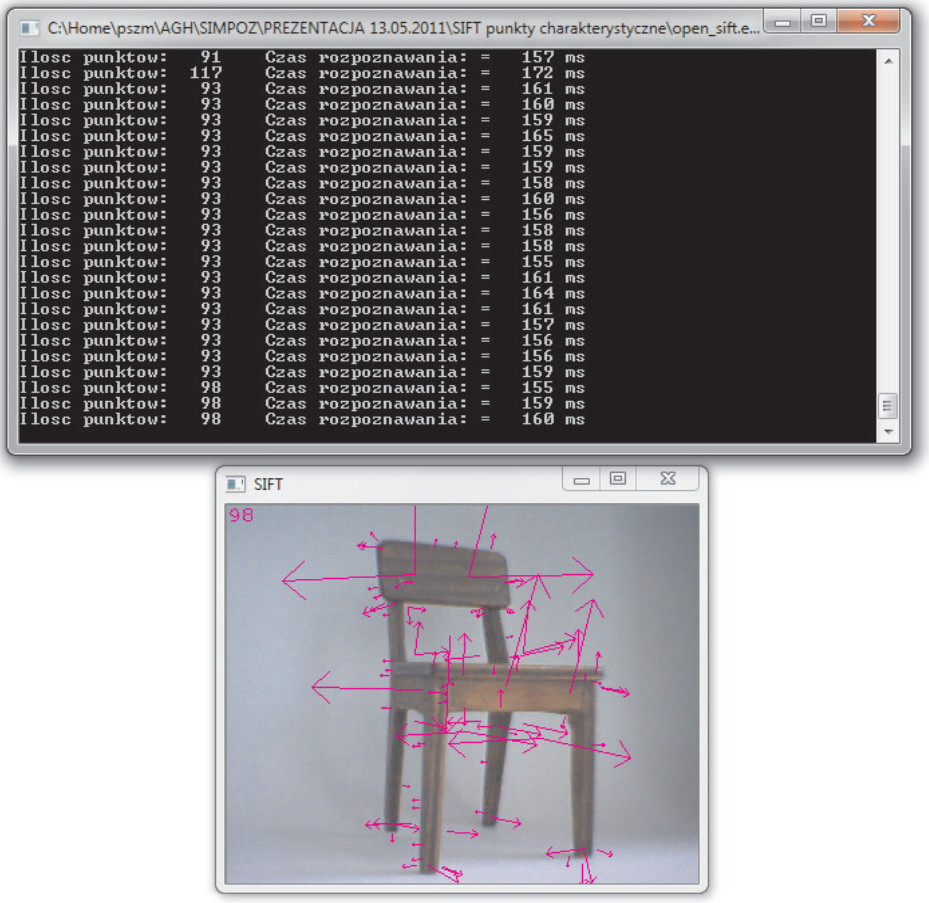
Czas przetwarzania wahał się w przedziale od 91 do 103 ms.

Algorytm ten oparty jest na wykorzystaniu falek Haara do wykrywania twarzy [16] (rys. 7).

2.5. Algorytm SIFT

SIFT (*Scale Invariant Feature Transform*) została zaproponowana przez Davida Lowe'a w 1999 r. [10]. Wyznacza dla analizowanego obrazu zbiór punktów charakterystycznych

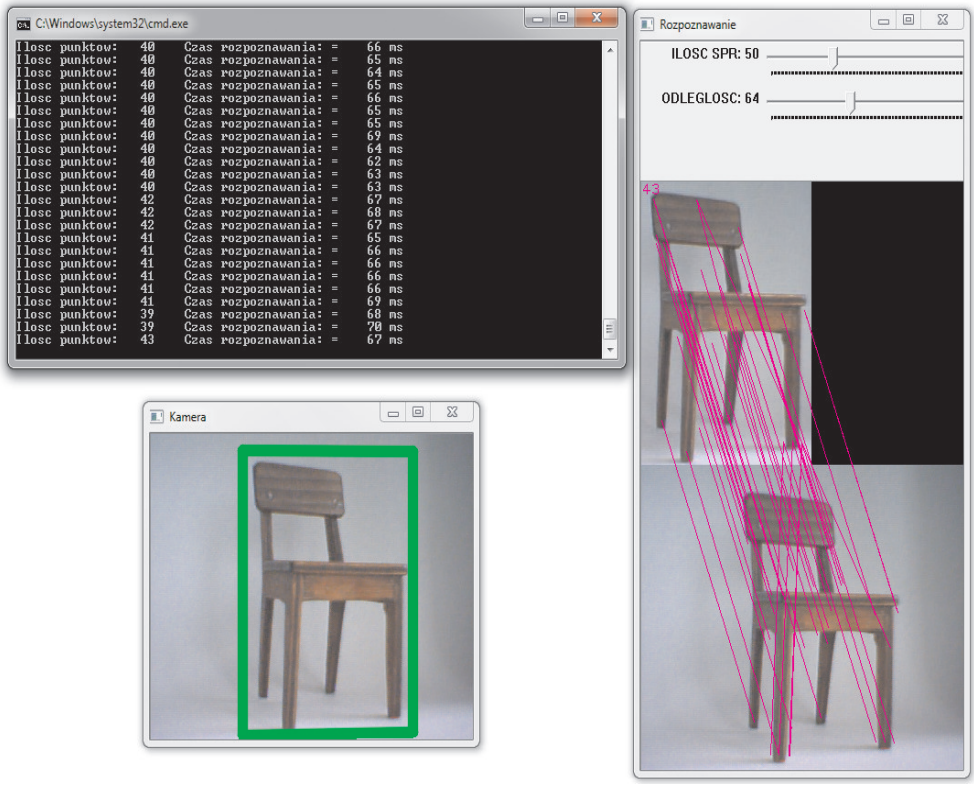
i opisuje je za pomocą deskryptorów w postaci 128-elementowych wektorów. Metoda ta jest inwariantna ze względu na skalę i obrót. Jest stosowana do automatycznego tworzenia zdjęć panoramicznych, rozpoznawania obiektów na obrazie, rekonstrukcji scen 3D, lokalizacji w przestrzeni itp. [12].



Rys. 8. Wynik działania algorytmu SIFT z zaznaczonymi punktami charakterystycznymi

Na rysunku 8 pokazano wyznaczanie punktów charakterystycznych SIFT. Czas przetwarzania obrazu wyniósł około 95 ms, a liczba punktów charakterystycznych około 155. Rysunek 9 z kolei pokazuje proces rozpoznawania obiektów. Linie w oknie „Rozpoznawanie" oznaczają skojarzone punkty charakterystyczne z obrazu wzorcowego i obrazu z kamery, prostokąt w oknie „Kamera" pokazuje obrys rozpoznanego obiektu.

Czas rozpoznawania wyniósł około 65 ms, a liczba punktów charakterystycznych zgodnych pomiędzy obrazem wzorcowym a analizowanym wyniósł około 41.



Rys. 9. Wynik rozpoznawania obiektu przy użyciu algorytmu SIFT

2.6. Algorytm SURF

SURF (*Speeded Up Robust Features*) jest algorytmem detekcji i opisu obrazu przez punkty charakterystyczne [5, 6]. Pierwszy raz zaprezentowany przez Herberta Baya w 2006 [1]. Jest używany do rozpoznawania obiektów i rekonstrukcji scen 3D. Jest częściowo wzorowany na SIFT. Standardowa wersja SURF jest kilka razy szybsza od SIFT [14]. SURF jest inwariantny ze względu na skalę i obrót

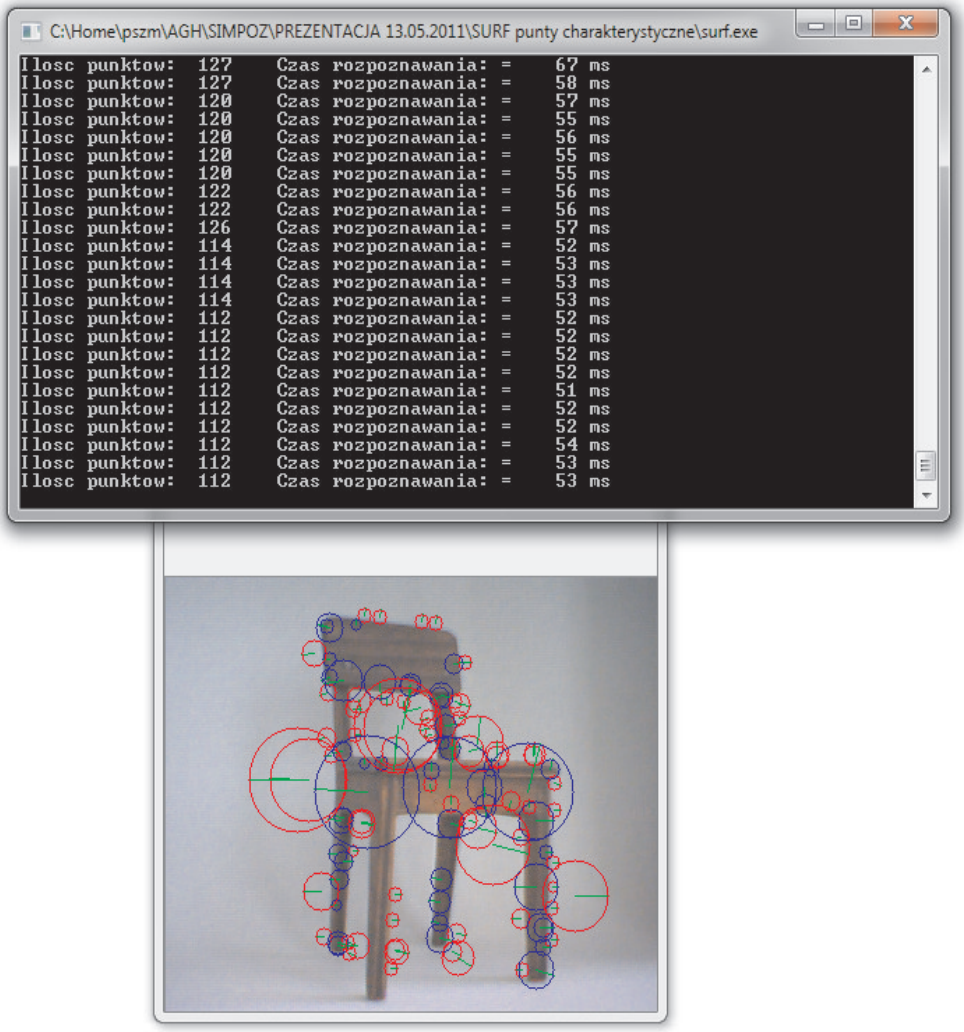
Algorytm działa w dwóch fazach. W pierwszej wyznaczane są punkty charakterystyczne, a w drugiej dla każdego punktu obliczany jest wektor 64-elementowy będący jego deskryptorem.

Szybkość działania algorytmu uzyskano między innymi poprzez zastosowanie scałkowanego obrazu oraz aproksymacji filtrami blokowymi wyznaczania wyznacznika Hessianu.

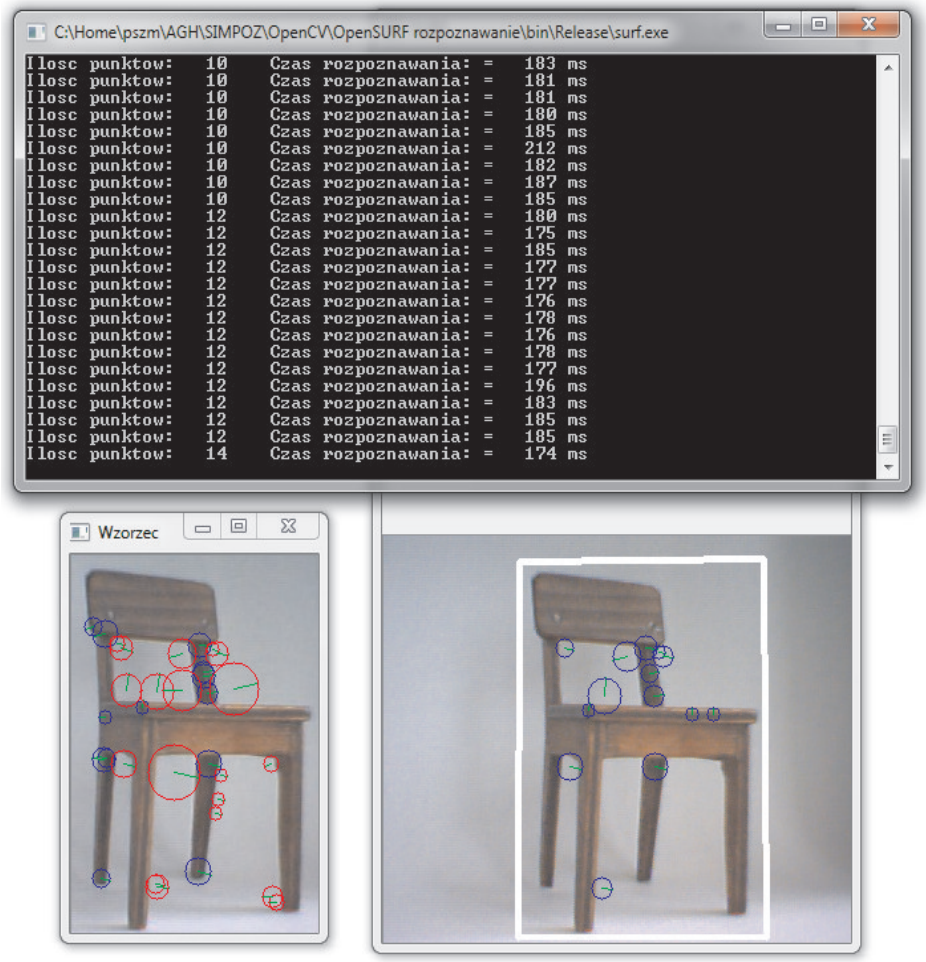
Na rysunkach 10 i 11 pokazane są punkty charakterystyczne wyznaczone za pomocą algorytmu SURF. Kolor czerwony okręgu oznacza jasny obszar na ciemnym tle, a niebieski

ciemny na jasnym, długość zielonej linii wskazuje ważność punktu, a jej zwrot kierunek – wyznaczony przez algorytm jako kierunek zmiany jasności obszaru.

Na rysunku 10 widzimy wyznaczanie punktów charakterystycznych SURF, czas przetworzenia obrazu to około 54 ms, a liczba wyznaczonych punktów charakterystycznych to około 55. Rysunek 11 pokazuje wykorzystanie algorytmu SURF do rozpoznawania obiektów. W czasie około 180ms zostaje rozpoznany obiekt na bazie około 12 podobnych punktów charakterystycznych.



Rys. 10. Wynik działania algorytmu SURF z zaznaczonymi punktami charakterystycznymi



Rys. 11. Wynik rozpoznawania obiektu przy użyciu algorytmu SURF

3. Wnioski

W pracy przedstawiono implementacje następujących algorytmów:

- Canny,
- Harris-Stephens,
- Shi-Tomasi (Kanade-Tomasi),
- HoG,
- Haar,
- SIFT,
- SURF.

Programy te przetwarzały obraz z kamery. Prowadzone testy potwierdziły ich wysoką skuteczność w zadaniach rozpoznawania obiektów na analizowanym obrazie. Dzięki pracy online można zaobserwować czasy przetwarzania obrazu dla każdego z nich oraz efekty działania. Czasy przetwarzania są bardzo krótkie nawet na zastosowanej platformie sprzętowej i programowej. Daje to nadzieję, że w przypadku zastosowania większych rozdzielczości i lepszego sprzętu będzie możliwe analizowanie obrazu online nawet przy rozpoznawaniu kilku obiektów równocześnie. Zauważono również, że wyniki działania tych algorytmów zależą w sposób istotny od ustawienia ich parametrów pracy. Wydaje się więc zasadne wzbogacenie tych algorytmów o mechanizmy automatycznego doboru tych parametrów, ale zagadnienie to nie jest proste i wymaga odrębnej pracy. Planuje się również prowadzenie dalszych prac mających na celu rozpoznawanie wielu obiektów oraz tego samego obiektu widzianego z różnych stron.

Literatura

[1] Bay H., Ess A., Tuytelaars T., Luc Van Gool, *SURF: Speeded Up Robust Features*, Computer. Vision and Image Understanding (CVIU), vol. 110, No. 3, 2008, 346–359.
[2] Bradski G., Kaehler A., *Learning OpenCV*. O'Reilly Media, Inc., Sebastopol, 2008.
[3] Canny J., *A Computational Approach To Edge Detection*. IEEE Trans. Pattern Analysis and Machine Intelligence, 8, 1986, 679–714.
[4] Dalal N., *Finding People in Images and Videos*. Ph.D. Thesis 2006.
[5] Evans Ch., *Notes on the OpenSURF Library.* 2009.
[6] Hess R., *An Open-Source SIFT Library*. MM'10 October 25–29, Firenze Italy, 2010.
[7] http://en.wikipedia.org/wiki/Corner_detection.
[8] http:// pl.wikipedia.org/wiki/Wykrywanie_krawędzi.
[9] http://www.aishack.in/2010/05/the-shi-tomasi-corner-detector/.
[10] Lowe D.G., *Distinctive Image Features from Scale-Invariant Keypoints.* International Journal of Computer Vision, 2004.
[11] *OpenCV Reference Manual v.2.2.*
[12] Pawlik P., Mikrut S., *Wyszukiwanie punktów charakterystycznych na potrzeby łączenia zdjęć lotniczych*. SLOK, 2006.
[13] Rafajłowicz E. (ed.), *Algorytmy przetwarzania obrazów i wstęp do pracy z biblioteką OpenCV*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2009.
[14] Schweiger F., Zeisl B., Georgel P., Schroth G., Steinbach E., Navab N.: *Maximum Detector Response Markers for SIFT and SURF*. VMV, 2009.
[15] Szeliski R., *Computer Vision: Algorithms and Applications*. Springer, 2010.
[16] Tuytelaars T., Mikolajczyk K., *Local Invariant Feature Detectors: A Survey*. Foundations and Trends in Computer Graphics and Vision, vol. 3, No. 3 (2007), 2008 177–280.