

Projekt aplikacji mobilnej umożliwiającej umieszczenie wirtualnej grafiki w rzeczywistym położeniu

Podstawy przetwarzania obrazów

Michał Aniserowicz

6 stycznia 2014

Streszczenie

Celem artykułu jest opisanie zbioru koncepcji, które posłużą do implementacji algorytmu sztucznej inteligencji grającego w grę Scrabble w języku polskim. W artykule zostały przeanalizowane i porównane dane zawarte w dwóch głównych słownikach wyrazów do gier dla języka polskiego, przedstawione dane statystyczne ułatwiające wprowadzanie heurystyk do algorytmu, a także opisane metody niezbędne do wyznaczania wszystkich możliwych kombinacji ruchów w danej turze. Autor omawia również podział rozgrywki na fazy gry i przybliża podejście, które pozwala uzyskiwać najlepsze wyniki na każdym etapie rozgrywki.

1 Wstęp

2 Dyskretny splot macierzy

Podstawową operacją wykorzystywaną podczas przetwarzania obrazów cyfrowych jest dyskretny splot macierzy. Pozwala on przefiltrować obraz, tzn. uwydatnić lub ukryć niektóre jego cechy, takie jak zaszumienie.

Operacja ta przebiega w następujących krokach:

1. Wybór obrazu, który zostanie poddany

splotowi.

2. Określenie:

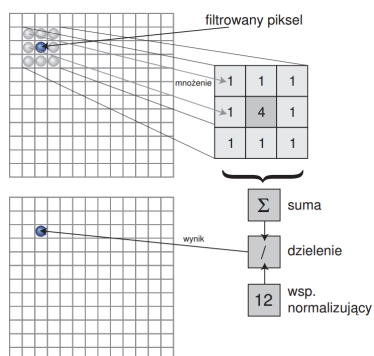
- macierzy (tzw. maski) filtru, najczęściej o niewielkich rozmiarach, np. 3×3 lub 5×5 ;
- współczynnika normalizującego, najczęściej równego sumie wartości komórek maski.

3. Obliczenie wartości każdego piksela obrazu wynikowego poprzez nałożenie maski na dany piksel, tj.:

- (a) pomnożenie wartości pikseli obrazu wejściowego przez wartości odpowiadających komórek maski;
- (b) zsumowanie otrzymanych wartości;
- (c) podzielenie sumy przez współczynnik normalizujący;
- (d) ustawienie wyniku jako wartości odpowiedniego piksela obrazu wynikowego.

Neleży zauważyć, że wartość współczynnika normalizującego równa sumie wartości komórek maski gwarantuje wartość wyniku mieszczącą się w przedziale dopuszczalnych wartości piksela (np. 0 – 255)

Przebieg operacji dyskretnego splotu został zobrazowany na Rysunku 1.



Rysunek 1: Operacja dyskretnego splotu macierzy. Źródło: [2].

2.1 Przykład - Rozmycie Gaussa

Przykładem praktycznego zastosowania operacji dyskretnego splotu macierzy jest operacja Rozmycia Gaussa. Pozwala ona wygładzić obraz, tzn. zredukować jego zaszumienie. Odbywa się to kosztem utraty ostrości.

Przykładowe maski stosowane w tej operacji przedstawia Rysunek 2.

$$\begin{matrix} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 4 & 2 & 1 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix} \\ \text{(a)} & \text{(b)} \end{matrix}$$

Rysunek 2: Maski stosowane w operacji Rozmycia Gaussa: (a) 3×3 ; (b) 5×5 .

Maski te mają następujące cechy:

- znaczenie piksela maleje wraz z jego odległością od środka maski według funkcji Gaussa;
- wartość piksela wynikowego jest uśrednieniem wartości odpowiadające go piksela wejściowego i wartości jego

sąsiadów¹.

Przykład działania Rozmycia Gaussa został przedstawiony na Rysunku 3.



(a)



(b)

Rysunek 3: Przykład działania Rozmycia Gaussa: (a) obraz wejściowy; (b) obraz wynikowy. Źródło: [3].

3 Detekcja krawędzi

Operacji dyskretnego splotu macierzy używa się również w celu wykrycia krawędzi na obrazie. W tym przypadku stosowane są filtry, które pozwalają aproksymować pochodne kierunkowe intensywności obrazu (gradienty). Pojedyncza maska filtru wykrywa gradienty obrazu dla pojedynczego kierunku. Powszechnie stosowane filtry, takie jak operatory Prewitt i Sobela, różnią się jedynie liczbą i rodzajem masek.

Rysunek 4 przedstawia przykładowy wynik zastosowania filtru wykrywającego krawędzie.

¹Filtry posiadające tę cechę nazywa się filtrami uśredniającymi.



(a)



(b)

Rysunek 4: Przykładowy wynik działania filtru wykrywającego krawędzie w obrazie: (a) obraz wejściowy; (b) obraz wynikowy. Źródło: [4].

3.1 Krzyż Robertsa

Jednym z najwcześniej opracowanych algorytmów wykrywania krawędzi jest Krzyż Robertsa. Został on zaproponowany w 1963 r. przez Lawrence'a G. Robertsa. Wykorzystuje się w nim dwie maski, z których każda wykrywa krawędzie skierowane w innym kierunku. Maski te zostały przedstawione na Rysunku 5.

$$\begin{matrix} \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} \\ (a) \end{matrix} \quad \begin{matrix} \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \\ (b) \end{matrix}$$

Rysunek 5: Krzyżu Robertsa - maski: (a) dla kierunku 45°; (b) dla kierunku 135°.

Wynikowy obraz otrzymuje się poprzez obliczenie różnic modułów wartości odpowiadających sobie pikseli z obrazów powstałych przez zastosowanie masek.

Algorytm cechuje się niską złożonością obliczeniową. Jest on jednak mało odporny na szum, tzn. powoduje wykrycie wielu krawędzi powstałych w wyniku zaszumienia obrazu wejściowego.

3.2 Operator Prewitt

Kolejnym algorytmem wykrywania krawędzi jest operator Prewitt. Został on zaproponowany w 1966 r. przez Judith M. Prewitt.

Wykorzystuje zbiór czterech masek, wykrywających krawędzie skierowane w czterech kierunkach (0°, 45°, 90°, 135°). Maski te przedstawia Rysunek 6.

$$\begin{matrix} \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \\ (a) \end{matrix} \quad \begin{matrix} \begin{bmatrix} 0 & +1 & +1 \\ -1 & 0 & +1 \\ -1 & -1 & 0 \end{bmatrix} \\ (b) \end{matrix}$$

$$\begin{matrix} \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \\ (c) \end{matrix} \quad \begin{matrix} \begin{bmatrix} +1 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} \\ (d) \end{matrix}$$

Rysunek 6: Operator Prewitt - maski: (a) dla kierunku 0°; (b) dla kierunku 45°; (c) dla kierunku 90°; (d) dla kierunku 135°.

Operator Prewitt cechuje się dużo lepszą skutecznością i odpornością na szum od Krzyża Robertsa, ale jest bardziej wymagający obliczeniowo.

3.3 Operator Sobela

Najczęściej stosowanym filtrem wykrywającym krawędzie na obrazie jest operator Sobela. Został on opracowany w 1968 r. przez Irwina Sobela. Od operatora Prewitt różni się jedynie wagami komórek masek (Rysunek 7).

W porównaniu z operatorem Prewitt, zastosowanie operatora Sobela powoduje otrzymanie obrazu bardziej wygładzonego.

$$\begin{array}{cc} \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} & \begin{bmatrix} 0 & +1 & +2 \\ -1 & 0 & +1 \\ -2 & -1 & 0 \end{bmatrix} \\ \text{(a)} & \text{(b)} \\ \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} & \begin{bmatrix} +2 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \\ \text{(c)} & \text{(d)} \end{array}$$

Rysunek 7: Operator Sobela - maski: (a) dla kierunku 0°; (b) dla kierunku 45°; (c) dla kierunku 90°; (d) dla kierunku 135°.

3.4 Operator Scharra

Kolejnym filtrem wykrywającym krawędzie jest operator Scharra, zaproponowany w 2000 r. przez Hanno Scharra. Od operatorów Prewitt i Sobela różni się on jedynie wagami komórek masek (Rysunek 8).

$$\begin{array}{cc} \begin{bmatrix} -3 & 0 & +3 \\ -10 & 0 & +10 \\ -3 & 0 & +3 \end{bmatrix} & \begin{bmatrix} 0 & +3 & +10 \\ -3 & 0 & +3 \\ -10 & -3 & 0 \end{bmatrix} \\ \text{(a)} & \text{(b)} \\ \begin{bmatrix} +3 & +10 & +3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} & \begin{bmatrix} +10 & +3 & 0 \\ +3 & 0 & -3 \\ 0 & -3 & -10 \end{bmatrix} \\ \text{(c)} & \text{(d)} \end{array}$$

Rysunek 8: Operator Scharra - maski: (a) dla kierunku 0°; (b) dla kierunku 45°; (c) dla kierunku 90°; (d) dla kierunku 135°.

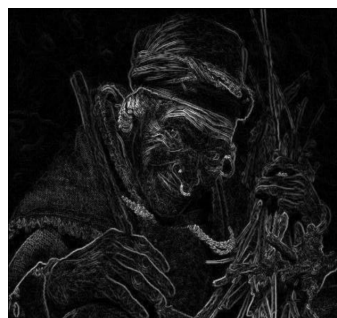
W porównaniu z operatorem Sobela, operatora Scharra lepiej wykrywa kierunek krawędzi na obrazie.

3.5 Przykłady

Przykłady działania omówionych algorytmów przedstawia Rysunek 9.



(a)



(b)



(c)



(d)

Rysunek 9: Przykłady działania filtrów wykrywających krawędzie na obrazie: (a) obraz wejściowy; (b) wynik działania Krzyża Robertsa; (c) wynik działania operatora Prewitt; (d) wynik działania operatora Sobela. Źródło: [5].

4 Algorytm Canny'ego

Najczęściej stosowanym algorytmem wykrywania krawędzi jest obecnie algorytm Canny'ego, powstały w 1986 roku. Jego autorem jest John F. Canny.

Algorytm ma w założeniu osiągać trzy cele:

- dobra detekcja – wykrycie jak największej liczby rzeczywistych krawędzi,
- dobra lokalizacja – oznaczenie danej krawędzi jak najbliżej jej rzeczywistego położenia,
- minimalna odpowiedź – oznaczenie danej krawędzi tylko raz; brak krawędzi powstałych w wyniku zakłóceń.

Kroki algorytmu są następujące:

1. Wygładzenie obrazu.
2. Obliczenie modułu gradientu obrazu.
3. Usunięcie niemaksymalnych pikseli.
4. Progowanie z histerezą.

4.1 Wygładzenie obrazu

Celem pierwszego kroku jest redukcja zaaszumienia obrazu wejściowego.

Wygładzenie obrazu polega na zastosowaniu filtru Gaussa (patrz sekcja 2.1). Odchylenie standardowe filtru (σ) jest parametrem algorytmu - im większa wartość σ , tym mniej fałszywych krawędzi znajdzie się na obrazie wynikowym.

Przykładowy obraz otrzymany w tym kroku został przedstawiony na Rysunku 10.

4.2 Obliczenie gradientu

Kolejny krok na celu znalezienie potencjalnych krawędzi na obrazie. Polega on na zastosowaniu jednego z filtrów wykrywających krawędzie (patrz sekcja 3).



(a)



(b)

Rysunek 10: Przykładowy rezultat wygładzenia obrazu ($\sigma = 3$): (a) obraz wejściowy; (b) obraz wygładzony. Źródło: [6].

Oprócz pozycji krawędzi, należy również zapamiętać ich kierunek. Przykładowy wynik tego kroku algorytmu przedstawia Rysunek 11.

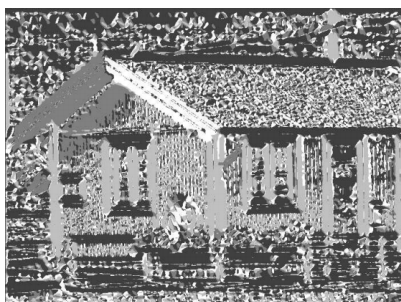
4.3 Usunięcie zbędnych pikseli

Celem trzeciego kroku algorytmu jest zmniejszenie grubości krawędzi otrzymanych w poprzednim kroku tak, aby wszystkie krawędzie miały grubość jednego piksela.

Usunięcie zbędnych pikseli odbywa się poprzez porównanie każdego piksela obrazu z dwoma pikselami sąsiednimi i zachowaniu jego wartości tylko wtedy, gdy jego jasność jest większa od jasności obu sąsiadów (w przeciwnym wypadku jasność piksela jest ustawiana na 0). Piksele sąsiednie wyznaczane są na podstawie informacji o



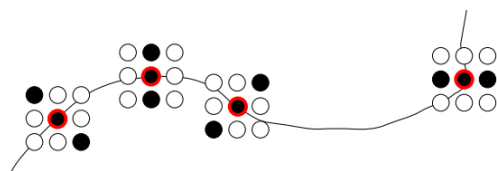
(a)



(b)

Rysunek 11: Przykładowy rezultat drugiego kroku algorytmu Canny’ego: (a) potencjalne krawędzie; (b) kierunek gradientu (ten sam odcień szarości przypisany krawędzion o tym samym nachyleniu). Źródło: [6].

kierunku gradientu (patrz Rysunek 12).

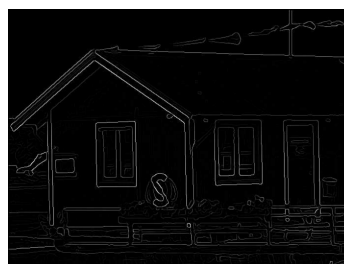


Rysunek 12: Zasada wyznaczania sąsiadów pikseli leżących na krawędzi na podstawie informacji o kierunku gradientu. Źródło: [6].

Przykładowy wynik tego kroku algorytmu przedstawia Rysunek 13.

4.4 Progowanie z histerezą

Ostatni krok algorytmu ma na celu usunięcie zbyt ciemnych krawędzi przy zachowaniu ciemnych fragmentów jasnych krawędzi



Rysunek 13: Przykładowy rezultat usunięcia zbędnych pikseli z obrazu. Źródło: [6].

(stąd użycie słowa “histereza”²). Przebiega on następująco:

1. Określenie dwóch progów: T_1 , T_2 (są to dwa kolejne, obok σ , parametry algorytmu).
2. Zaakceptowanie krawędzi, dla których moduł gradientu jest większy od T_2 .
3. Usunięcie krawędzi, dla których moduł gradientu jest mniejszy od T_1 .
4. Rekurencyjne usunięcie krawędzi, dla których moduł gradientu jest mniejszy od T_2 , i które nie przylegają do żadnej z już zaakceptowanych krawędzi.

4.5 Przykładowe rezultaty

Przykładowe rezultaty działania algorytmu Canny’ego przedstawia Rysunek 14.

5 Podsumowanie

- Wykrywanie krawędzi (a także wiele innych operacji) sprowadza się do zastosowania dyskretnego spłotu macierzy
- Algorytmy zaproponowane w latach 60., czy 80. są nadal stosowane

²Wyjaśnienie zjawiska histerezy dostępne pod adresem: <http://pl.wikipedia.org/wiki/Histereza>.



(a)



(b)



(c)



(d)

Rysunek 14: Przykładowe wyniki algorytmu Canny'ego: (a) obraz wejściowy; (b) wynik dla $\sigma = 3, T_1 = 75, T_2 = 125$; (c) wynik dla $\sigma = 1, T_1 = 75, T_2 = 125$; (d) wynik dla $\sigma = 3, T_1 = 25, T_2 = 75$. Źródło: [6].

- Operator Sobela i algorytm Canny'ego to obecnie najpopularniejsze metody wykrywania krawędzi

Literatura

- [1] *Komputerowa analiza i przetwarzanie obrazów*, Tadeusiewicz R., Korohoda P., Kraków, Wydawnictwo Fundacji Postępu Telekomunikacji, 1997, 83-86476-15-X.
- [2] *Filtracja obrazów rastrowych* [online], Steć P., <http://www.uz.zgora.pl/~pstec/files/filtracja.pdf> [dostęp: styczeń 2013].
- [3] *Rozmycie Gaussowskie* [online], Encyklopedia Artifice, http://encyklopedia.artifice.pl/index.php?title=Rozmycie_Gaussowskie [dostęp: styczeń 2013].
- [4] *Krzyż Robertsa* [online], Wikipedia, http://pl.wikipedia.org/wiki/Krzyż_Roberta [dostęp: styczeń 2013].
- [5] *Study and Comparison of Various Image Edge Detection Techniques* [online], Maini R., Aggarwal H., <http://wwwmath.tau.ac.il/~turkel/notes/Maini.pdf> [dostęp: styczeń 2013].
- [6] *Wykrywanie cech w obrazach cyfrowych* [online], Bołdak C., <http://aragorn.pb.bialystok.pl/~boldak/DIP/CPO-W04-v01-50pr.pdf> [dostęp: styczeń 2013].