Cyfrowe przetwarzanie obrazów i sygnałów
Wykład 8
AiR III

Joanna Ratajczak

KCiR (W4/K7)

[1] Niniejszy dokument zawiera materiały do wykładu z przedmiotu Cyfrowe Przetwarzanie Obrazów i Sygnałów. Jest on udostępniony pod warunkiem wykorzystania wyłącznie do własnych, prywatnych potrzeb i może być kopiowany wyłącznie w całości, razem ze stroną tytułową.

Reprezentacja obrazu w bazie F–C

W. Frei, C. Chen, *Fast Boundary Detection: A Generalization and New Algorithm*, IEEE Trans. Computers, 1977.

W ośmiospójnym otoczeniu punktu (x, y) funkcję obrazu f można traktować jako dziewięciowymiarowy wektor

$$\mathbf{f}_8(x, y) \in \mathcal{R}^9$$

o składowych równych wartościom f w poszczególnych punktach tego otoczenia.

$$\begin{array}{|c|c|c|}\hline f(x-1,y-1) & f(x,y-1) & f(x+1,y-1) \\ \hline f(x-1,y) & f(x,y) & f(x+1,y) \\ \hline f(x-1,y+1) & f(x,y+1) & f(x+1,y+1) \\ \hline \end{array} \rightarrow \mathbf{f}_8 = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_9 \end{bmatrix}$$

Reprezentacja obrazu w bazie F–C

Jego reprezentacja w bazie h_i

$$\mathbf{f}_8(x,y) = \sum_{i=1}^{9} a_i h_i$$

ma składowe

$$a_i = \frac{(\mathbf{f}_8(x,y), h_i)}{(h_i, h_i)}.$$

Baza kształtów lokalnych

$$h_1 = \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \qquad (h_1, h_1) = 8$$

$$h_2 = \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} \qquad (h_2, h_2) = 8$$

$$h_3 = \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \qquad (h_3, h_3) = 8$$

Baza kształtów lokalnych

$$h_4 = \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} \qquad (h_4, h_4) = 8$$

$$h_5 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \qquad (h_5, h_5) = 4$$

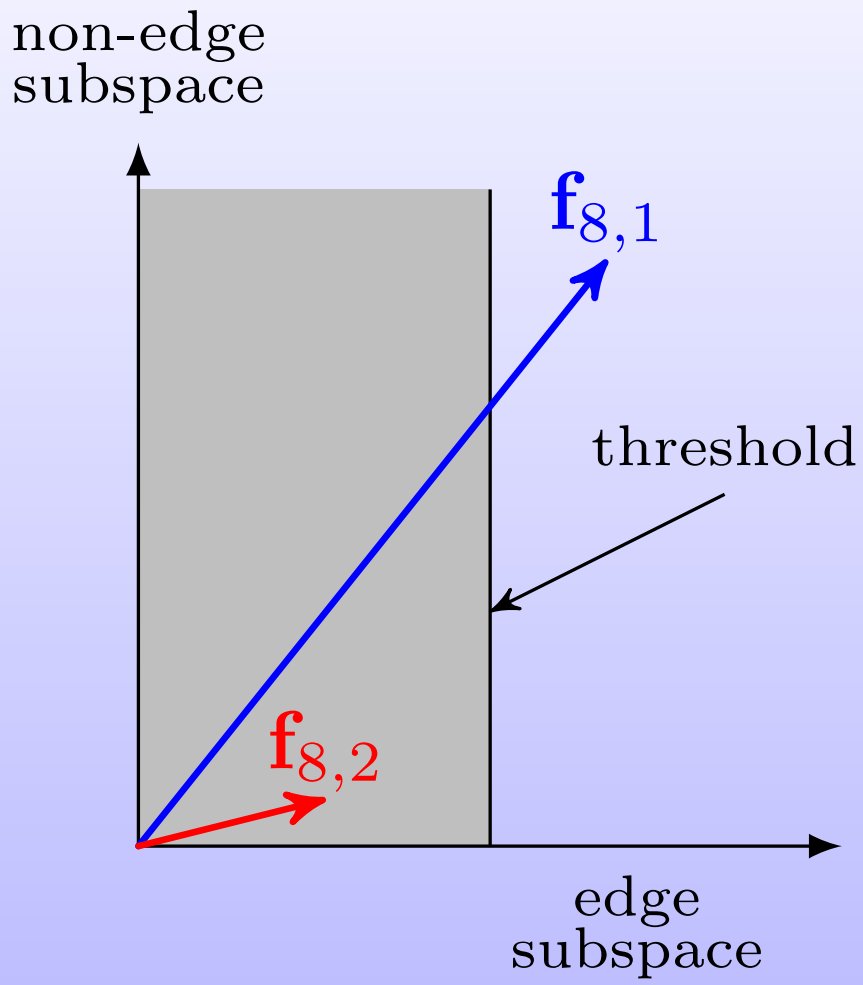
$$h_6 = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \qquad (h_6, h_6) = 4$$

Baza kształtów lokalnych

$$h_7 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \qquad (h_7, h_7) = 36$$

$$h_8 = \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} \qquad (h_8, h_8) = 36$$

$$h_9 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad (h_9, h_9) = 9$$


non-edge
subspace
$\mathbf{f}_{8,1}$
threshold
$\mathbf{f}_{8,2}$
edge
subspace

Baza F–C a wykrywanie krawędzi

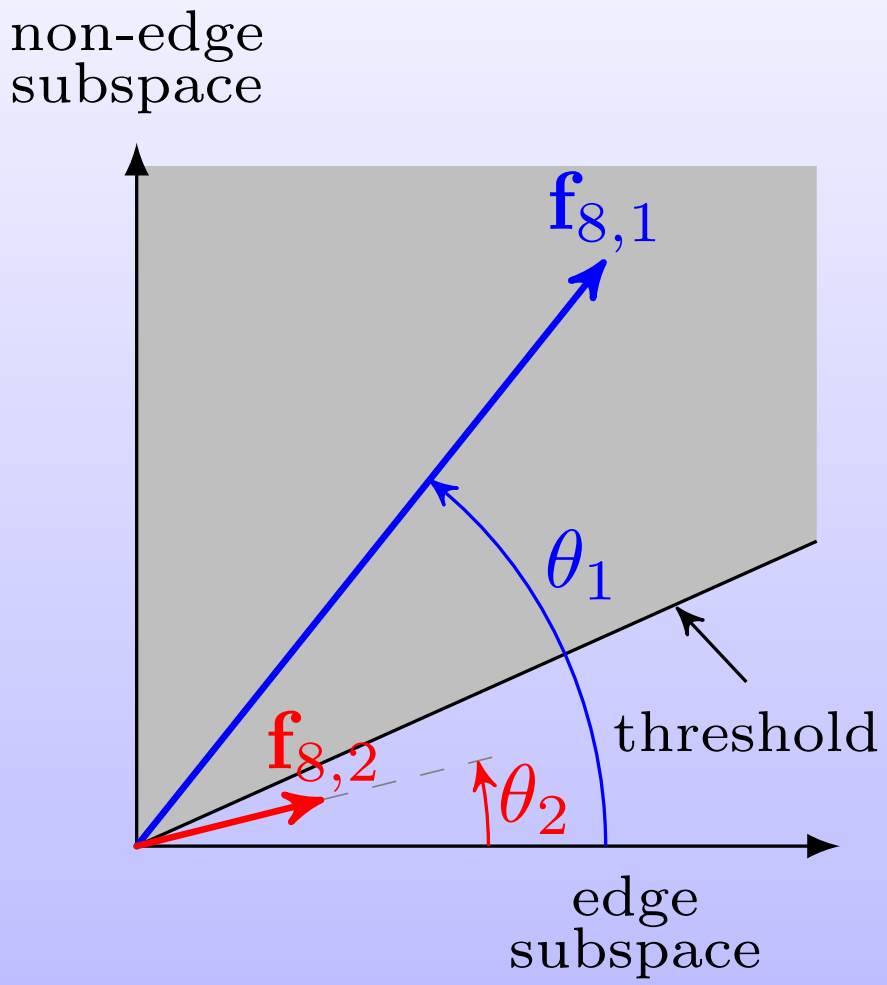
Podprzestrzeń „krawędziowa” jest rozpięta przez h_1 i h_2

$$E = \text{span}\{h_1, h_2\}.$$

Kryterium przyjęcia punktu (x, y) za element krawędzi stanowi kąt pomiędzy wektorem $\mathbf{f}_8(x, y)$ a podprzestrzenią E.

$$\theta = \arccos \sqrt{\frac{\sum_{i=1}^{2} (\mathbf{f}_8, h_i)^2}{\sum_{j=1}^{9} (\mathbf{f}_8, h_j)^2}}$$

Można założyć wartość progową tego kąta, poniżej której uznamy zasadność przyjęcia (x, y) jako elementu krawędzi.


non-edge
subspace
$\mathbf{f}_{8,1}$
θ_1
threshold
$\mathbf{f}_{8,2}$
θ_2
edge
subspace

Baza F–C a wykrywanie krawędzi

Równoważnie

$$|\mathbf{f}_E| = \sqrt{(\mathbf{f}_8, h_1)^2 + (\mathbf{f}_8, h_2)^2}$$

$$|\mathbf{f}_N| = \sqrt{\sum_{i=3}^{9} (\mathbf{f}_8, h_i)^2}$$

$$\theta = \operatorname{arc\,tg} \frac{|\mathbf{f}_N|}{|\mathbf{f}_E|}$$

Detektor Frei–Chen

- Nieczuły na długość wektora $\mathbf{f}_8$.
- Wykrywa krawędzie w ciemnych obszarach.
- Wykrywa subtelne krawędzie.
- ”Mocne” krawędzie oznaczone są cienką linią.
- Czuły na zakłócenia.

Detektor krawędzi Canny

Canny J., *A Computational Approach to Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 1986.

Cele:

1. **skuteczne wykrywanie** – znaleźć jak najwięcej rzeczywistych krawędzi. Przy czym błędem jest zarówno detekcja fałszywych krawędzi jak i pomijanie rzeczywistych krawędzi.
2. **poprawna lokalizacja** – punkt sklasyfikowany jako punkt krawędzi powinien być jak najbliższy środkowemu punktowi rzeczywistej krawędzi.
3. **minimalny nadmiar** – oznaczyć każdą krawędź pojedynczą linią, nie oznaczać szumów.

Detektor krawędzi Canny

Etapy przetwarzania

1. Redukcja szumu poprzez zastosowanie filtru Gaussa $G(x, y, \sigma)$ parametryzowanego przez odchylenie standardowe σ. σ odpowiada za rozmiar maski użytej do filtracji i ma wpływ na stopień eliminacji szumów (a przy okazji krawędzi) w obrazie.
2. Wyznaczenie gradientu jasności obrazu (dowolny filtr gradientowy: Sobel, Roberts, Prewitt)

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} \nabla_x f \\ \nabla_y f \end{bmatrix}$$

oraz wyznaczenie modułu (wartości) i kierunku gradientu

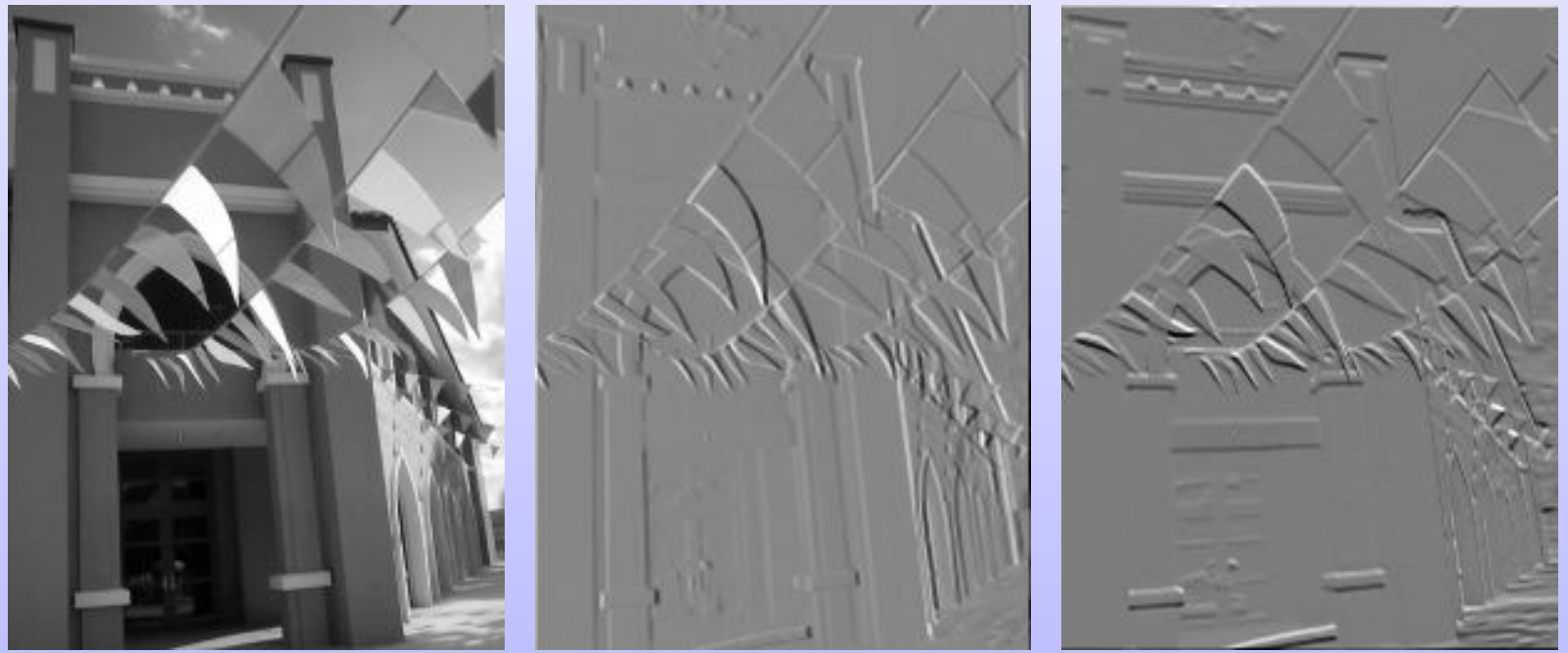
$$|\nabla f(x, y)| = \sqrt{(\nabla_x f)^2 + (\nabla_y f)^2} \qquad \alpha(x, y) = \frac{\nabla_y f}{\nabla_x f}$$

Detektor krawędzi Canny

3. Odrzucenie punktów nie będących maksimami gradientu wzdłuż jego kierunku – tłumienie niemaksymalne.
4. Śledzenie segmentów krawędzi z progowaniem podwójnym (progowanie z histerezą). Usunięcie z obrazu słabych krawędzi, ale jednocześnie generowanie ciągłych krawędzi.

Składowe gradientu z filtracją

$$\sigma = 0.5 \rightarrow h_{3\times3}$$

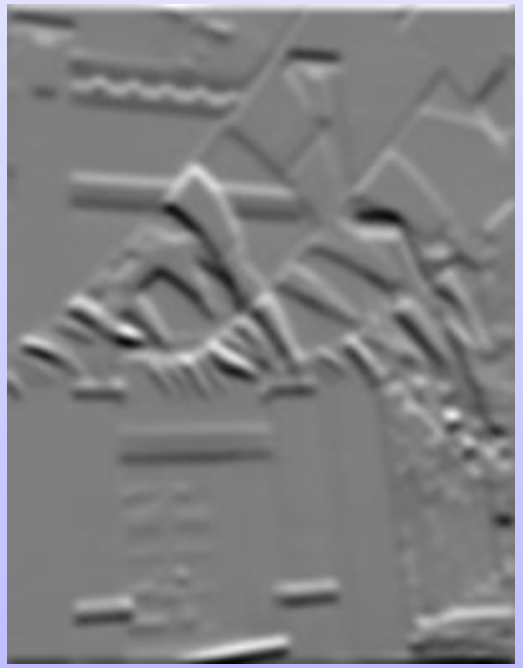


http://suraj.lums.edu.pk/~cs436a02/CannyImplementation.htm

Składowe gradientu z filtracją

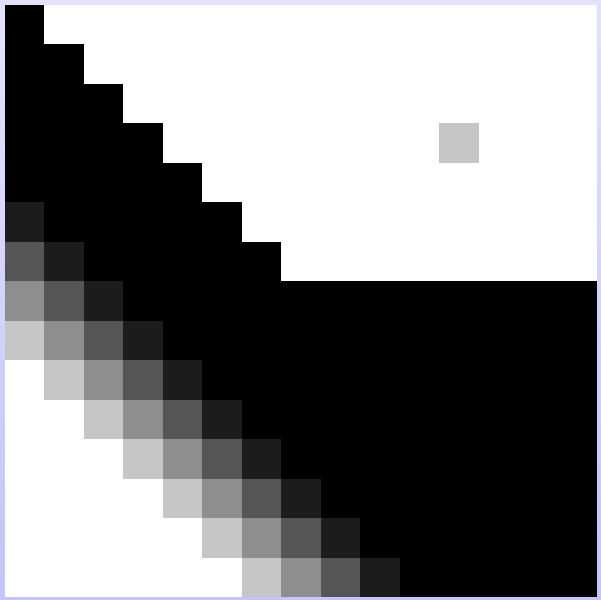
$$\sigma = 2 \rightarrow h_{9\times 9}$$

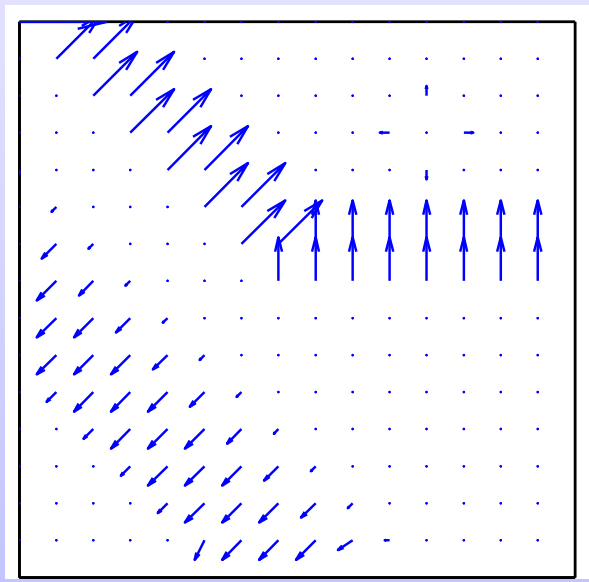




Moduł i kierunek gradientu

0	9	9	9	9	9	9	9	9	9	9	9	9	9	9
0	0	9	9	9	9	9	9	9	9	9	9	9	9	9
0	0	0	9	9	9	9	9	9	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9	9	7	9	9	9
0	0	0	0	0	9	9	9	9	9	9	9	9	9	9
1	0	0	0	0	0	9	9	9	9	9	9	9	9	9
3	1	0	0	0	0	0	9	9	9	9	9	9	9	9
5	3	1	0	0	0	0	0	0	0	0	0	0	0	0
7	5	3	1	0	0	0	0	0	0	0	0	0	0	0
9	7	5	3	1	0	0	0	0	0	0	0	0	0	0
9	9	7	5	3	1	0	0	0	0	0	0	0	0	0
9	9	9	7	5	3	1	0	0	0	0	0	0	0	0
9	9	9	9	7	5	3	1	0	0	0	0	0	0	0
9	9	9	9	9	7	5	3	1	0	0	0	0	0	0
9	9	9	9	9	9	7	5	3	1	0	0	0	0	0

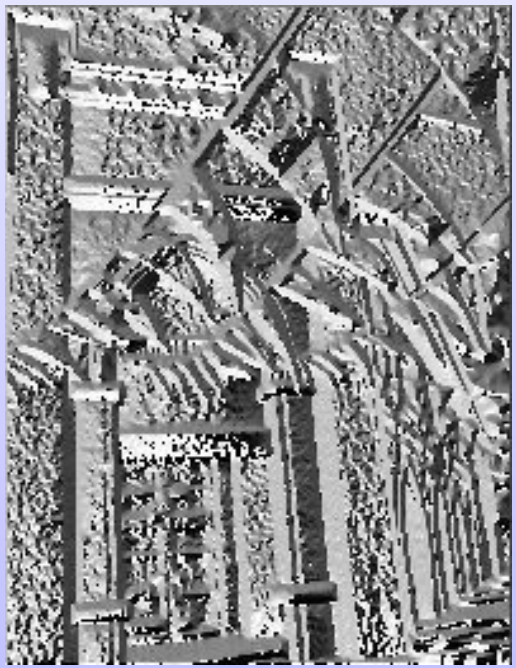
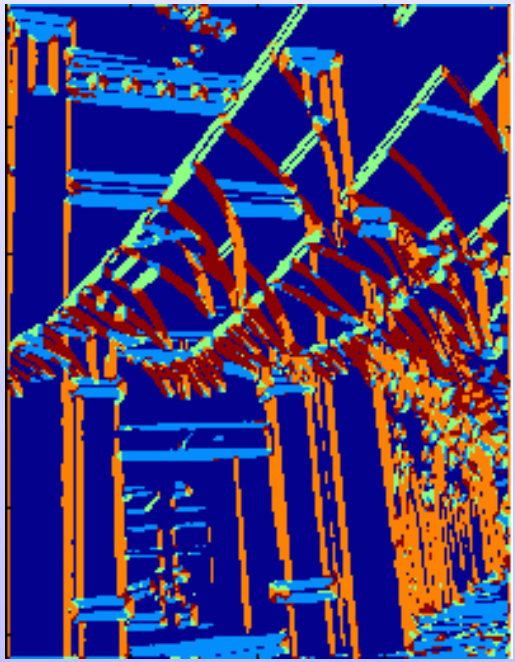




Moduł gradientu

$\sigma = 0.5$ $\sigma = 2$

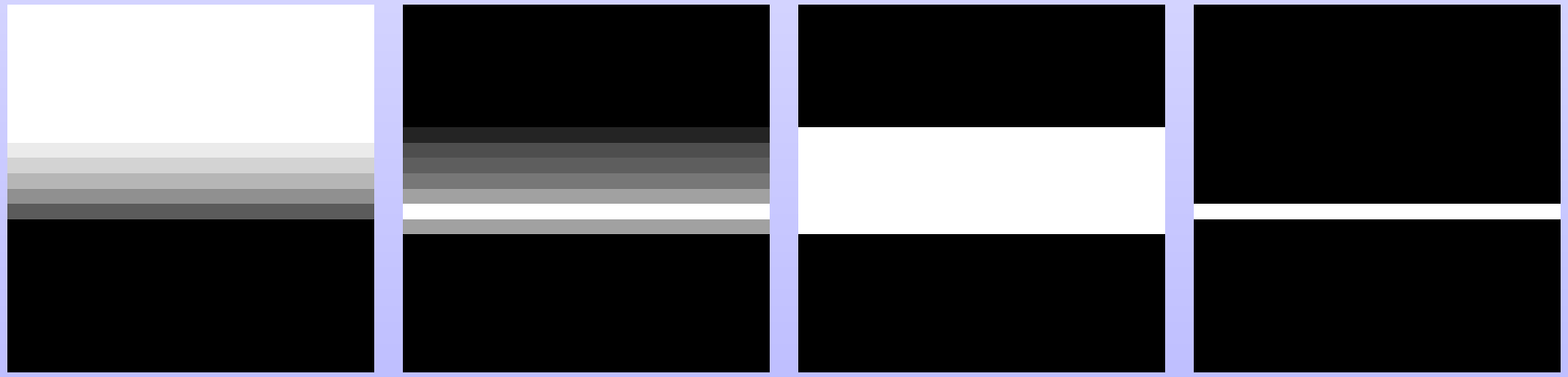
Kierunek gradientu

Tłumienie niemaksymalne

Cel: generowanie przez detektor pojedynczej odpowiedzi dla każdej rzeczywistej krawędzi w obrazie.

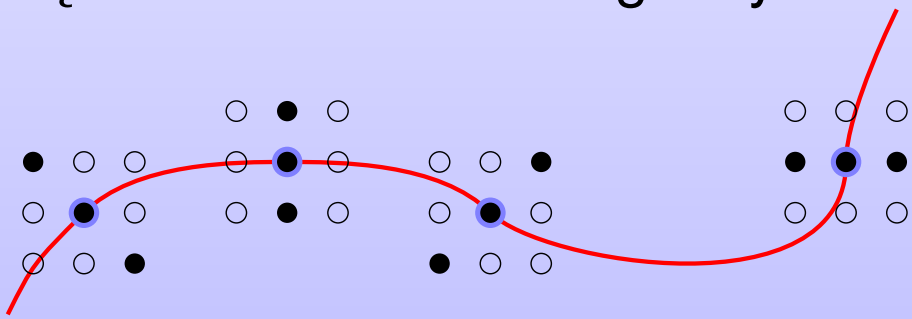
Zapewnia, że pikselem krawędzi jest piksel, którego wartość jest maksymalna w danym kierunku gradientu.



Tłumienie niemaksymalne

Algorytm

1. Przeglądaj wartości gradientu $|\nabla f(x, y)|$ dla kolejnych pikseli. Dla piksela o niezerowej wartości wyznacz kierunek gradientu.
2. Dla aktualnie przeglądanego piksela sprawdź dwóch najbliższych sąsiadów w kierunku zgodnym z kierunkiem gradientu.



3. Jeśli wartość gradientu dla aktualnego piksela jest większa od obu sąsiadów zachowaj ją w przeciwnym wypadku przypisz pikselowi wartość zero.

Usuwanie niemaksimów

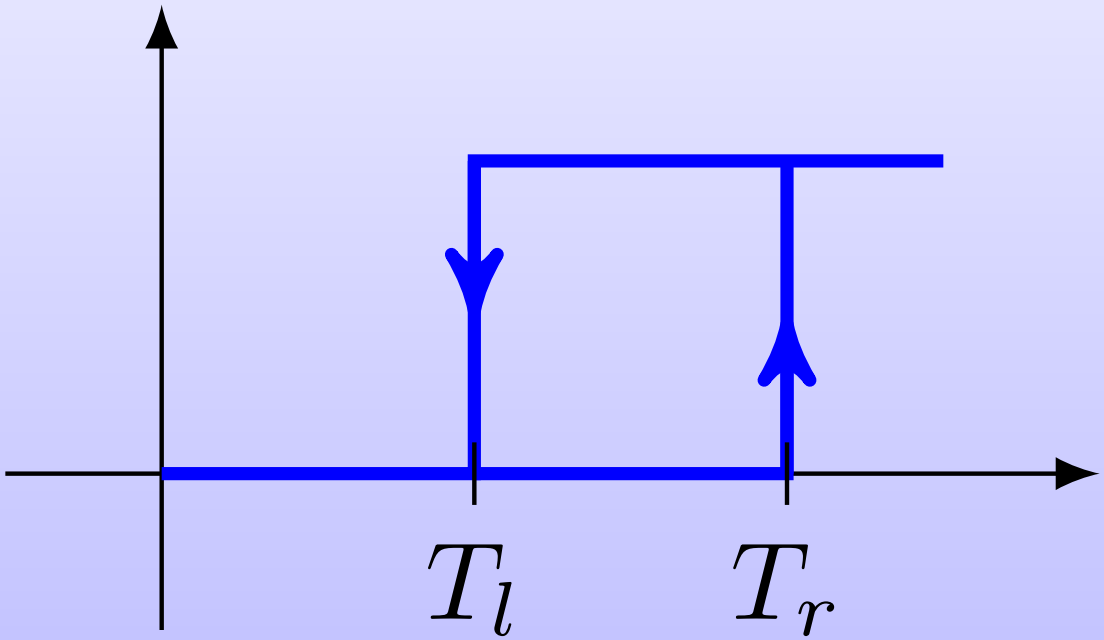
Śledzenie z progowaniem podwójnym

Cel: zachowanie punktów dla których wartość gradientu jest duża i odrzucić te dla których wartość gradientu jest mała.

Problem z krawędziami, które posiadają ciemniejsze fragmenty.

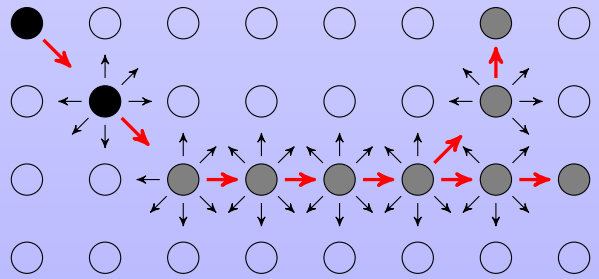
Wykorzystujemy progowanie z histerezą.

Progowanie podwójne (histerezowe)



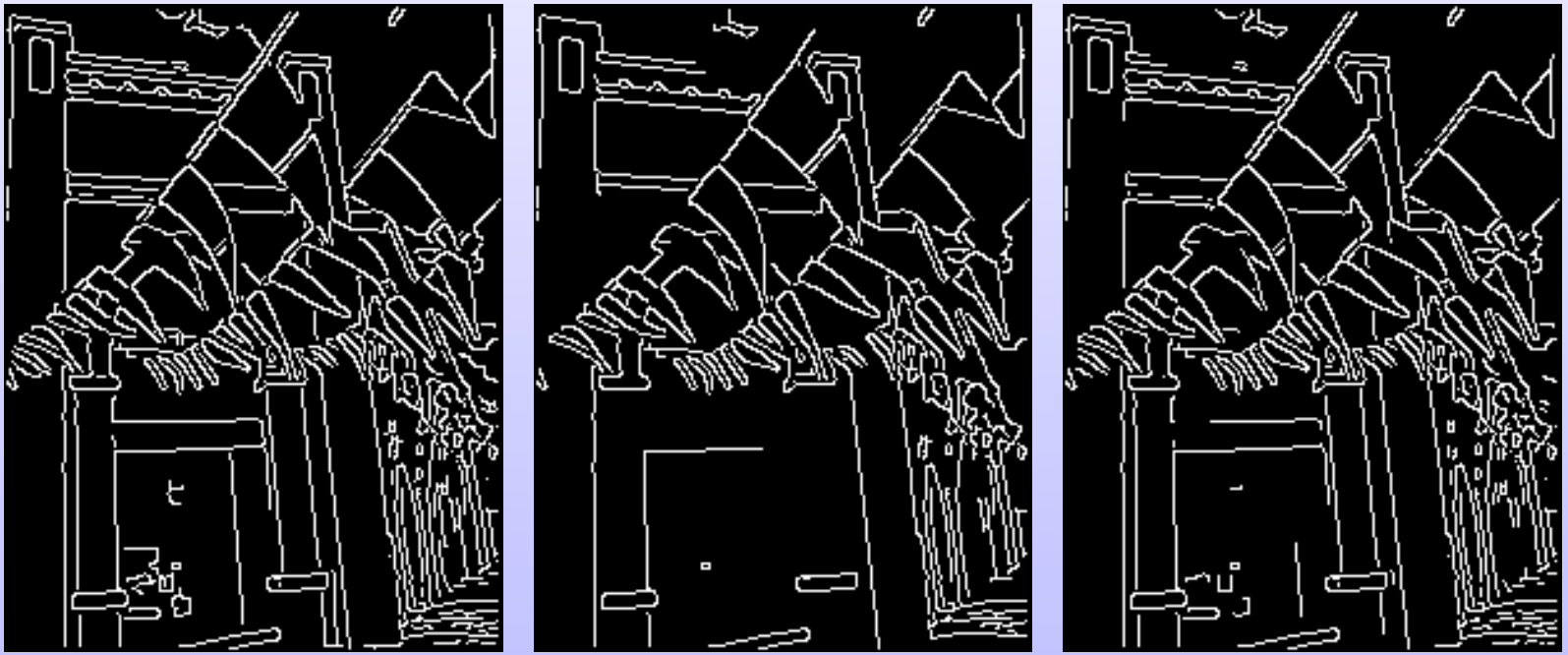
Wykorzystujemy rekurencyjne śledzenie krawędzi:

1. Przeglądamy obraz z lewa na prawo, z góry do dołu.
2. Piksel, którego wartość (gradientu) jest powyżej ustalonego progu radykalnego T_r jest przyporządkowywany do krawędzi.
3. Przeglądamy „sąsiadów”. Wszyscy, którzy posiadają wartość większą od T_l stają się nowymi początkami śledzenia.
4. Warunek STOPu: wartość gradientu sąsiadów jest poniżej T_l lub piksel był już analizowany.



Śledzenie z progowaniem podwójnym

$T_r = 50,\ T_l = 10;$ $T_r = 100,\ T_l = 10;$ $T_r = 50,\ T_l = 40;$



http://suraj.lums.edu.pk/~cs436a02/images/anim.mpg

▸ Animacja

Wybór cech

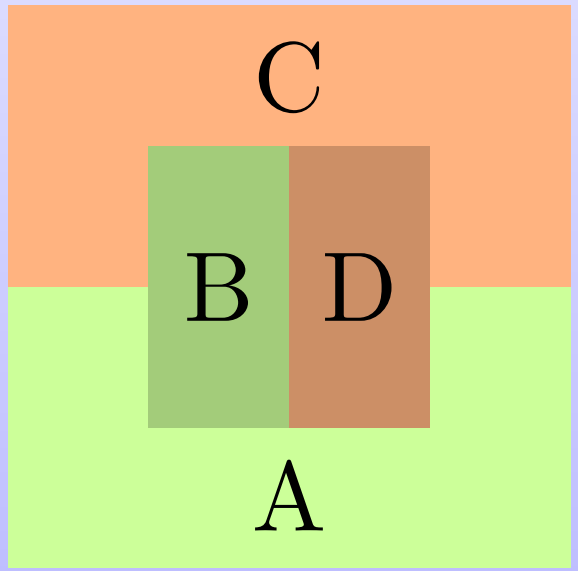
- Modele barw

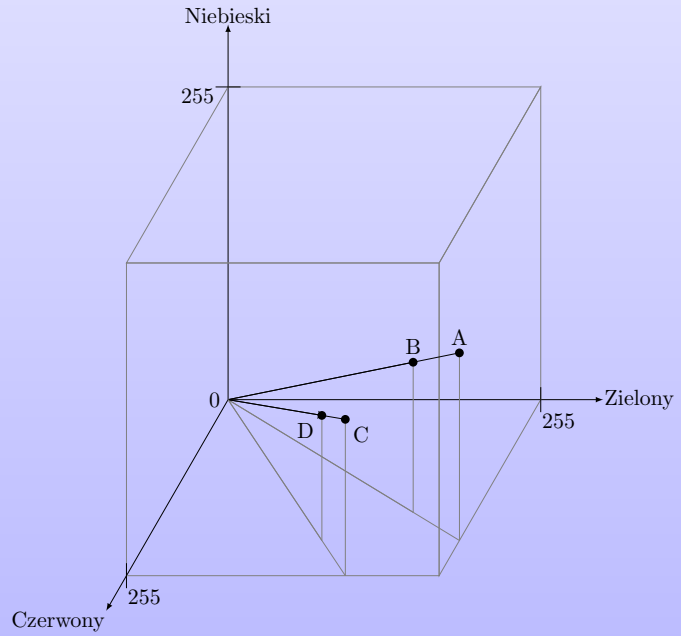
model	kierunek oświetlenia	natężenie oświetlenia	kolor oświetlenia	refleksy
RGB	+	+	+	+
HSV	-	-	+	-

- Miara podobieństwa

Wybór cech

- Modele barw
- Miara podobieństwa





Metodologie

1. Konwersja do skali szarości.
2. Suma gradientów dla poszczególnych składowych RGB. Dobre do aplikacji dla obrazów, w których kolory i obiekty są dobrze zdefiniowane.
3. Kombinacja gradientów poszczególnych składowych HSV. Mniej czułe na cienie w obrazie.
4. Zmodyfikowane operatory Robertsa, Sobela, itp.

Metodologie

5. Gradienty dla wybranej miary podobieństwa (np. odległości euklidesowej, miary kątowej).
6. Metody statystyczne – średnia z elementów wektora, mediana z elementów, itp.
7. Gradient wektorowy, w otoczeniu punktu centralnego bierzemy max odległość euklidesową.
8. Metody hybrydowe – w oparciu o ideę fuzzy logic wykorzystujemy więcej niż jedną reprezentację (np. RGB + HSV).

Wybrane operatory

Zmodyfikowany operator Robertsa

- skala szarości

$$|\nabla f(x,y)| = \sqrt{(f(x,y) - f(x+1,y+1))^2 + (f(x+1,y) - f(x,y+1))^2}$$

- kolor

$$\mathcal{G}_c f(x,y) = \max\left(\left|\vec{f}(x,y) - \vec{f}(x+1,y+1)\right|, \left|\vec{f}(x+1,y) - \vec{f}(x,y+1)\right|\right)$$

Wybrane algorytmy

Zmodyfikowany operator Sobela

- skala szarości

$$\nabla_x f(x,y) = \frac{1}{2}\left(f(x+1,y) - f(x-1,y)\right)$$
$$\nabla_y f(x,y) = \frac{1}{2}\left(f(x,y+1) - f(x,y-1)\right)$$
$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- kolor

$$\mathcal{G}_{xc} f(x,y) = \frac{1}{2}\left(\vec{f}(x+1,y) - \vec{f}(x-1,y)\right)$$
$$\mathcal{G}_{yc} f(x,y) = \frac{1}{2}\left(\vec{f}(x,y+1) - \vec{f}(x,y-1)\right)$$
$$\mathcal{G}_c f(x,y) = \max\left(\left|\mathcal{G}_{xc} f(x,y)\right|, \left|\mathcal{G}_{yc} f(x,y)\right|\right)$$

Wybrane algorytmy

Zmodyfikowany filtr medianowy

- Obraz jest przeglądany z oknem o ustalonym rozmiarze.
- Wartość środkowego piksela zastępowana jest medianą ze zbioru sąsiedztwa.
- Należy przyjąć jakąś metodę porządkowania, np. wyznaczamy sumaryczną odległość wektora w stosunku do pozostałych wektorów opisujących piksele w otoczeniu.
- Wyznaczoną odległość porządkujemy rosnąco.
- Wyjściem filtru jest mediana, tzn. wektor, który minimalizuje odległość od wszystkich pozostałych wektorów.