

Laboratorium 3: Proste metody przetwarzania obrazów II

Detekcja krawędzi¹

1. Wstęp

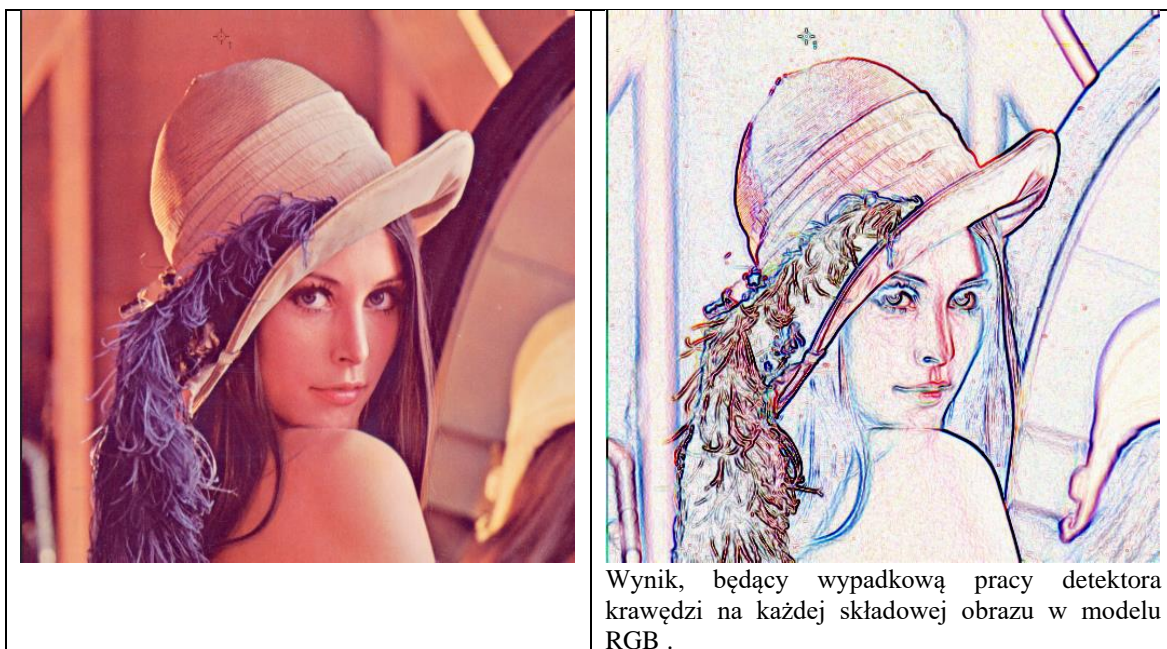
Krawędź na obrazie cyfrowym możemy określić jako *znaczące i nagłe zmiany* intensywności (zakładając obrazy w skali szarości) obrazu. Założenia dla obrazów kolorowych są takie same, ale w odniesieniu do rozważanych składowych koloru. Większość procedur detekcji krawędzi na obrazie wykorzystuje operatory pierwszej i drugiej pochodnej obrazu. Narzuca to wykorzystanie interpretacji obrazu jako funkcji dyskretnej dwóch zmiennych.

Poniżej przedstawiono wartości pochodnej pierwszego i drugiego rzędu dla funkcji dyskretnej jednej zmiennej, w przypadku obrazów cyfrowych ($\Delta x = 1$). Wartości pochodnej można obliczyć na kilka sposobów, w naszej pracy skorzystamy z tzw. pochodnej centralnej:

$$\frac{\partial}{\partial x} f(x) =_{df} f(x+1) - f(x-1)$$

$$\frac{\partial^2}{\partial x^2} f(x) =_{df} f(x+1) + f(x-1) - 2f(x)$$

Krawędzie na obrazie (zob. 1, przykład wykrycia krawędzi dla obrazu Lena) wykrywa się z różnych powodów – jeden z najciekawszych (najczęściej wykorzystywanych w praktyce) to rozpoznanie obiektów za pomocą detekcji oraz zamknięcia krawędzi na obrazie. Założenie dotyczy tego, że obiekty powinny być jednorodne i powinny wyróżniać się z otoczenia – tj. dobrze zamknięte przez swoją krawędź zewnętrzną.



¹ Wykorzystane obrazy w opracowaniu mogą być przedmiotem praw autorskich.



Rys. 1. Krawędzie na obrazie Lena

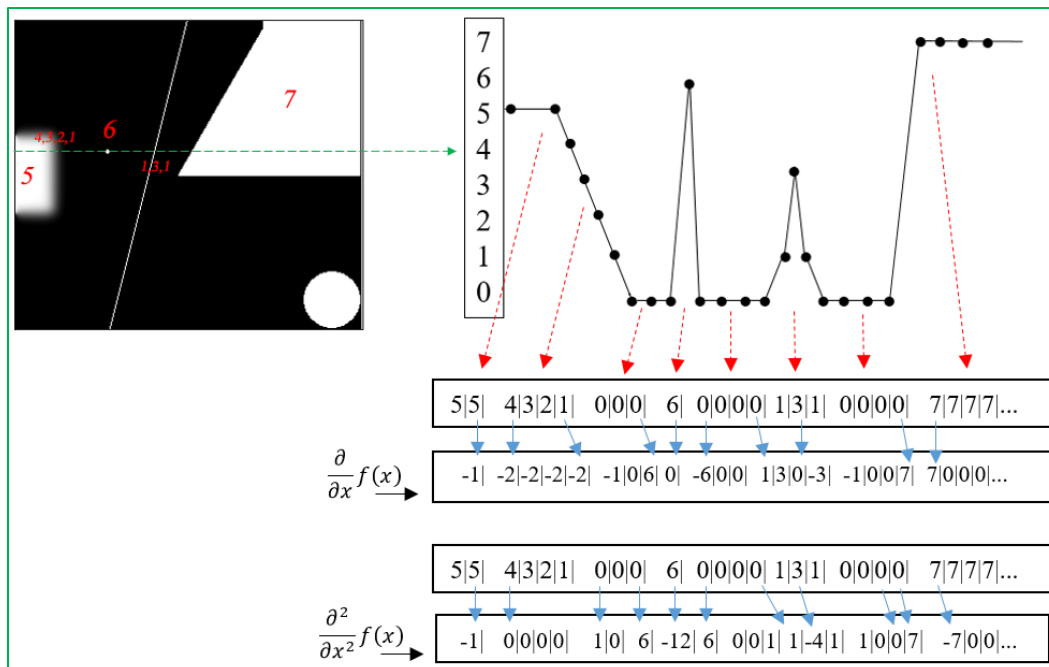
Widać pewną możliwość detekcji kapelusza, twarzy i części ramienia Leny poprzez odpowiednią analizę krawędzi. Inne elementy – np. pióra lub włosy, kompletnie nie nadają się do tego typu segmentacji obrazu. O ile obraz ‘Lena’ jest dość złożony względem segmentacji poprzez detekcję krawędzi, to dla pewnych zastosowań, wybór tego typu metod wydaje się naturalny (zob. rys. 2).



Rys. 2. Obiekty w naturalny sposób odseparowanie od tła krawędzią.

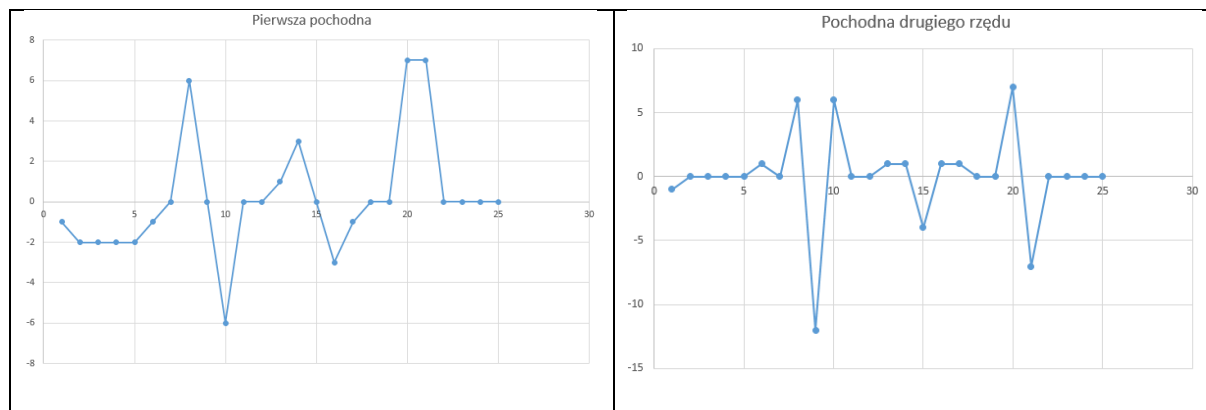
Aby zrozumieć rolę operatorów pierwszej i drugiej pochodnej w procesie detekcji krawędzi, należy rozważyć poniższy przykład (zob. rys. 3 - dla ułatwienia wizualizacji, przyjmujemy że kolor biały nie ma wartości 255, tylko 7). Rozważamy tylko wybrany wiersz obrazu na którym występują:

- 1) Łagodne przejście między obszarami: $5 \rightarrow 0$,
- 2) Pojedynczy punkt : 6,
- 3) Cienka linia : $1 \rightarrow 3 \rightarrow 1$,
- 4) Nagłe przejście między obszarami: $0 \rightarrow 7$,
- 5) Tło, obszar jednolity : 0.



Rys. 3. Interpretacja wartości pochodnych pierwszego i drugiego rzędu, względem wystąpienia różnych typów zmian intensywności w obrazie.

Wyraźnie widać pewną zależność zwaną własnością przejścia przez zero (ang. *zero crossing property*) dotyczącą miejsc, w których zlokalizowana jest krawędź (zob. rys. 4 – wykres wartości pierwszej i drugiej pochodnej).



Rys. 4. Własność przejścia przez zero. Możliwość lokalizacji krawędzi, wyjątkiem jest 'wejście' w obszar jednolity (o takiej samej wartości) w przypadku pierwszej pochodnej.

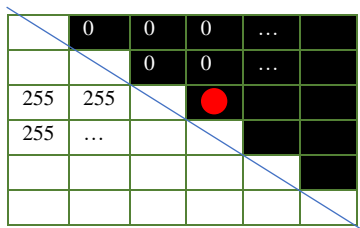
2. Detekcja krawędzi za pomocą wektora gradientu

Wektor gradientu funkcji dwóch zmiennych w punkcie określamy w sposób następujący (współrzędne wektora):

$$\nabla f(x, y) =_{df} \left[\frac{\partial}{\partial x} f(x, y), \frac{\partial}{\partial y} f(x, y) \right]$$

Intuicyjnie, wektor ten możemy wykorzystać do określenia kierunku oraz wartości zmiany funkcji w określonym punkcie. Zwrot wektora gradientu wskazuje kierunek najszybszego wzrostu wartości funkcji, natomiast jego długość wskazuje wartość tej zmiany. Można się więc domyślić, że w sposób naturalny wiele detektorów krawędzi wykorzysta własności wektora gradientu.

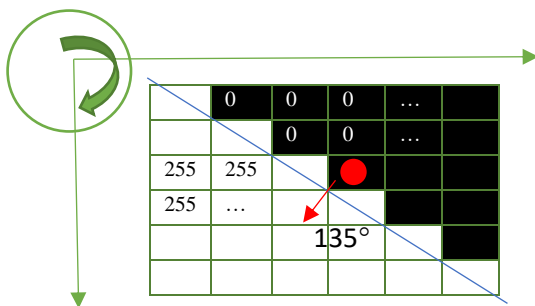
Aby lepiej zrozumieć zastosowanie gradientu na obrazie, rozważmy poniższy przykład (obliczenie gradientu w ‘czerwonym punkcie’, krawędź reprezentowana przez ‘niebieską linię’):



Dla uproszczenia wektor gradientu oznaczmy jako G , orientację kątem θ , a jego współrzędne odpowiednio: G_x i G_y .

$$G = [G_x, G_y]; G_x = f(x+1, y) - f(x-1, y) \text{ (czerwony punkt)} = -255; G_y = f(x, y+1) - f(x, y-1) = 255;$$

$$\text{Tym samym: } \|G\| = \sqrt{G_x^2 + G_y^2} = 255 * \sqrt{2} \approx 360.62; \theta = \arctan2(G_x, G_y) \approx 2.36 \text{ rad}; 2.36 * \left(\frac{180}{\pi}\right) \approx 135^\circ$$



Można zauważyć, że gradient jest skierowany prostopadłe do krawędzi, wskazując największe zmiany wartości jasności. Moduł wektora określać będzie wielkość tych zmian – im większy, tym większy skok intensywności jasności w obrazie.

3. Operator Sobela

Całkowicie naturalnym podejściem, wykorzystującym własności gradientu do detekcji krawędzi jest organizacja filtru splotowego, który będzie wykonywał na obrazie operację wyznaczania gradientu. Ciekawym operatorem jest tzw. filtr Sobela, zorganizowany jako kombinacja jednowymiarowego filtru Gaussa (wstępna redukcja szumu) oraz wartości pochodnej centralnej pierwszego rzędu:

Operator Sobela – wykrywanie krawędzi o orientacji pionowej (aproksymacja dla wartości G_x : filtr Gaussa 1D + pochodna pierwszego rzędu w kierunku x):

$$G_x : \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}_{3 \times 1} \times [-1 \ 0 \ 1]_{1 \times 3} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}_{3 \times 3}$$

Operator Sobela – wykrywanie krawędzi o orientacji poziomej (aproksymacja dla wartości G_y : pochodna pierwszego rzędu w kierunku y + filtr Gaussa 1D):

$$G_y : \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}_{3 \times 1} \times [1 \ 2 \ 1]_{1 \times 3} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}_{3 \times 3}$$

Obliczenie wartości G_x oraz G_y dla dowolnego punktu obrazu wykonywane jest jako konwolucja obrazu i odpowiedniej maski filtru Sobela.

Dodatkowo, rozumując w sposób analogiczny, można przygotować również maski analizujące krawędzie o orientacji ukośnych:

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \text{ oraz } \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

Posiadając już obliczone współrzędne wektora gradientu (G) można analizować jego kierunek i moduł. Wyjściem detekcji krawędzi za pomocą metody Sobela stanowi znormalizowana wartość modułu obliczonego gradientu, następnie przeskalowana do przedziału $[0, 255]$. Można również wykrywać krawędzie tylko w wybranym kierunku – np. x , wtedy maska dla kierunku y nie jest uwzględniana.

Przykłady działania operatora Sobela znajdziesz w dołączonym notebooku `Lab3_krawedzie.ipynb`.

4. Operator Canny'ego

Znanym problemem nie tylko operatorów Sobela, ale również innych detektorów krawędzi na obrazie cyfrowym, jest bardzo duża wrażliwość na szum oraz brak wyboru lokalizacji właściwych krawędzi (generowanie 'grubych' krawędzi). Często dochodzi do tzw. 'efektu spaghetti' – czyli wiele różnego rodzaju krawędzi na obrazie (również duża liczba 'krótkich' krawędzi), co w żaden sposób nie ułatwia zastosowanie algorytmów rozpoznania obiektów za pomocą detekcji krawędzi. Dlatego też zachodzi potrzeba wprowadzenia pewnego rodzaju decyzji, które 'odpowiedzi' pikseli na detektory krawędzi powinny być uwzględniane, a które nie.

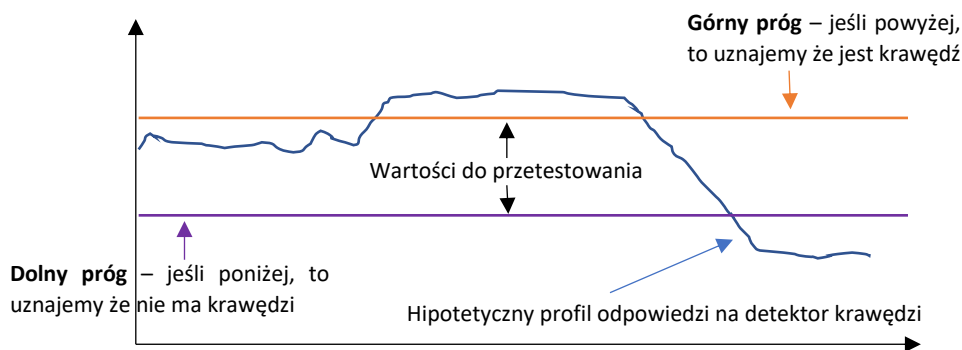
W swojej pracy naukowej (Canny, J., 1986), John F. Canny zaproponował ciekawe rozwiązanie dla powyżej wymienionych problemów uwzględniając informacje, które można wykorzystać z detektorów krawędzi – możliwość analizy wielkości i kierunku wektora gradientu.

Algorytm detekcji krawędzi Canny’ego opiera się na poniższych krokach:

1. ‘Preprocessing’ filtrem Gaussowskim.
2. Analiza lokalnych maksimów względem wartości gradientu (modułu wektora), uwzględniając kierunek krawędzi. Intuicyjnie: aby pozbyć się ‘grubych’ krawędzi, czyli pikseli które miały dużą odpowiedź na zastosowany detektor krawędzi, można analizować wartość dla piksela bieżącego względem jego sąsiadów i wybrać wartość maksymalną. Dodatkowo, można wykonać proces dla pikseli w kierunku krawędzi, tj. jeśli analizujemy krawędzie o orientacji pionowej (maską do detekcji krawędzi o orientacji pionowej), sensownie będzie pobrać wartości pikseli sąsiednich z lewej i prawej strony w poziomie. Wykonując tę czynność dla wszystkich pikseli, zlokalizujemy piksele o największej odpowiedzi w kierunku analizowanej krawędzi.

Wykorzystywanym operatorem detekcji krawędzi w przypadku filtru Canny’ego, to operatory Sobela w kierunku x oraz y (G_x , G_y). A więc wejściem algorytmu detektora Canny w punkcie 2, to wynik działania operatorów Sobela.

3. Progowanie z histerezą – hasło histereza dotyczy zjawiska opóźniania efektu w reakcji na inny czynnik. Intuicja jest wyjątkowo prosta: zakładamy dwa progi, względem odpowiedzi na detektor krawędzi (czyli modułu gradientu). Wszystkie krawędzie powyżej progu ‘górnego’ oznaczane są jako właściwe, poniżej dolnego progu jako niewłaściwe, natomiast wartości pomiędzy oznaczane są jako ‘kandydaci’ na krawędź (zob. rys. 5).



Rys. 5. Idea procesu progowania z histerezą

Kandydaci są testowani względem tego, czy są połączeni z pikselami reperującymi krawędzie powyżej progu górnego – jeśli tak, to kandydat jest właściwy (krawędź jest rozpoznawana w tym punkcie), jeśli nie to usuwamy kandydata. To podejmowanie decyzji poprzez dodatkowy proces względem wcześniej podjętej decyzji stanowi element histerezy.

Przykłady wyników detekcji krawędzi metodą Canny’ego znajdziesz w dołączonym notebooku Lab3_krawedzie.ipynb.

5. Detekcja krawędzi za pomocą Laplasjanu

Operator Laplace'a (Pierre Simon de Laplace) określony jest jako suma pochodnych cząstkowych drugiego rzędu (dla funkcji dwóch zmiennych):

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

Oczywiście, w przypadku funkcji dyskretnej mamy:

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Od razu widać, że dość prosto jest zrealizować maskę filtru konwolucyjnego, za pomocą którego realizowany jest powyższy operator:

0	1	0
1	-4	1
0	1	0

Filtr ten można stosować do detekcji krawędzi w kierunku poziomym i pionowym. Analogicznie, można określić maski dla filtrów detekcji krawędzi ukośnych:

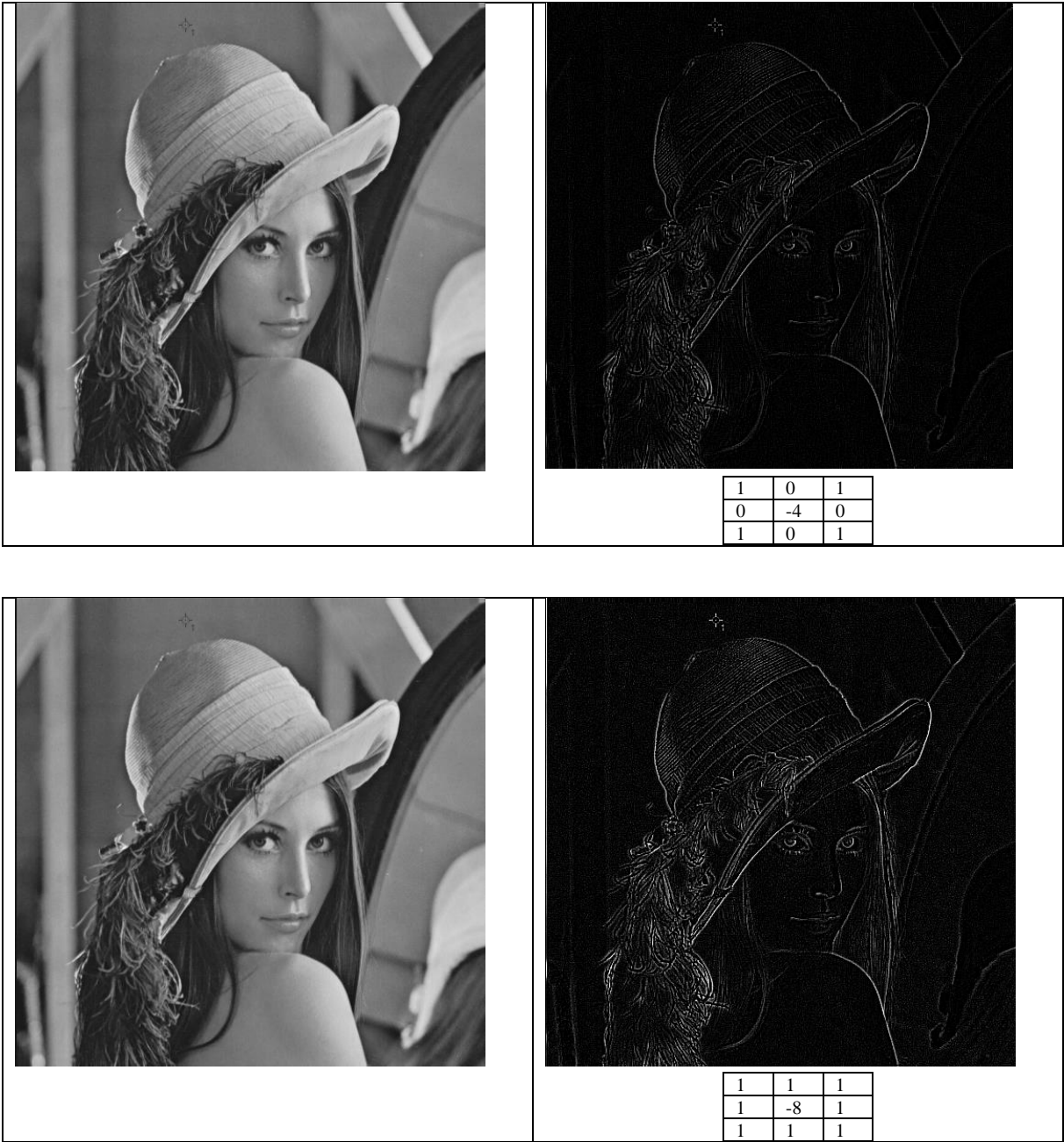
1	0	1
0	-4	0
1	0	1

Następnie, poprzez kombinację obu masek można określić filtr detekcji wszystkich krawędzi:

1	1	1
1	-8	1
1	1	1

Przykłady:





Literatura:

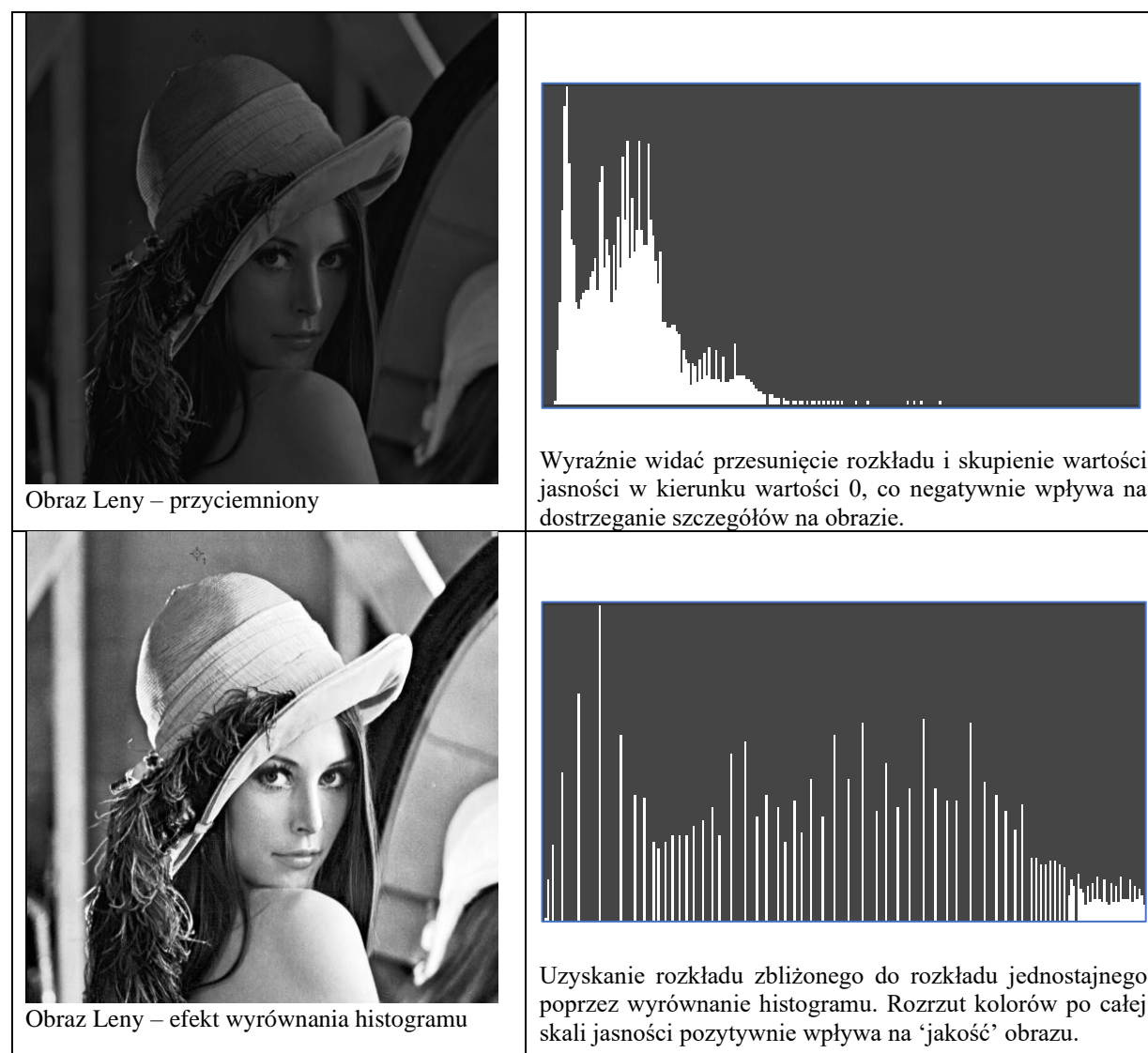
Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714 (1986).

Gonzalez, R. C., Woods, R. E.: Digital Image Processing. 4th Edition, MedData Interactive, Pearson, (2018).

Wyrównanie histogramu²

1. Wyrównanie histogramu

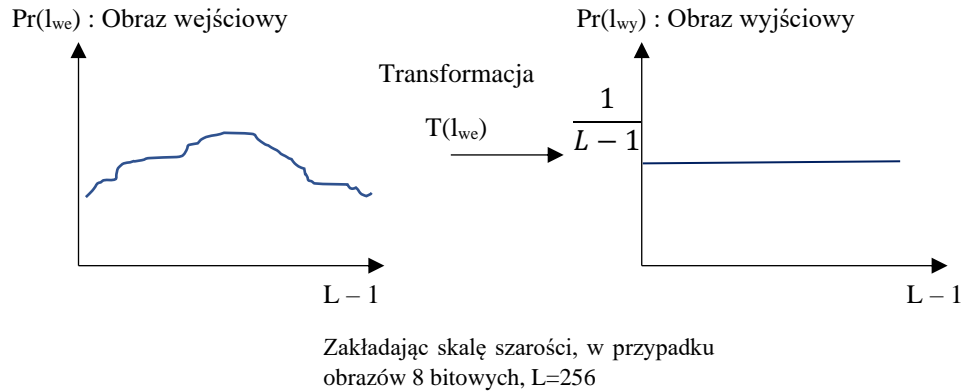
Jak już wiemy (zob. lab2 – temat ‘binaryzacja obrazu’), znormalizowany histogram obrazu interpretowany jest jako rozkład prawdopodobieństwa. Przy tej interpretacji, czyli interpretacji obrazu w kategoriach probabilistycznych, można wprowadzać różne techniki modyfikujące ten rozkład. Jeden z najbardziej znanych algorytmów modyfikacji obrazu przy wykorzystaniu jego histogramu polega na docelowym uzyskaniu dyskretnego rozkładu jednostajnego. Założenie dotyczy próby doprowadzenia do takiego samego prawdopodobieństwa wylosowania dowolnego koloru, co przekładać się będzie na zwiększenie zróżnicowania jasności (lub kolorów) na obrazie (zob. rys. 1).



Rys. 1. Efekt transformacji obrazu poprzez tzw. wyrównanie histogramu

² Wykorzystane obrazy w opracowaniu mogą być przedmiotem praw autorskich.

Generalnie chodzi tu o możliwość wprowadzenia takiej transformacji, która dla dowolnego rozkładu wejściowego, na wyjściu zbliży go do rozkładu jednostajnego:



Należy zauważyć, że prawdopodobieństwo realizacji zmiennej losowej obrazu wyjściowego, obliczane są względem dostępnej palety barw: $1/(L-1)$.

Założenia dla $T(l)$:

- Funkcja niemalejąca,
- Powinna to być transformacja, która będzie spełniać poniższą równość, względem wartości prawdopodobieństwa obrazu wyjściowego:

$$P(l_{wy}) =_{df} P(l_{we}) \left| \frac{\partial l_{we}}{\partial l_{wy}} \right| = P(l_{we}) \left| \frac{1}{\frac{\partial l_{wy}}{\partial l_{we}}} \right| = P(l_{we}) \left| \frac{1}{\frac{\partial T(l_{we})}{\partial l_{we}}} \right|$$

gdzie:

$$l_{wy} = T(l_{we}) = (L-1) \int_0^{l_{we}} P(w) dw$$

Sprawdźmy, czy taka transformacja T odpowiednio realizuje nasz cel – czyli prawdopodobieństwo dla każdej wartości jasności w obrazie wyjściowym na poziomie $1/(L-1)$:

$$\frac{\partial T(l_{we})}{\partial l_{we}} = \frac{\partial \left((L-1) \int_0^{l_{we}} P(w) dw \right)}{\partial l_{we}} = (L-1) P(l_{we})$$

następnie:

$$P(l_{wy}) =_{df} P(l_{we}) \left| \frac{1}{\frac{\partial T(l_{we})}{\partial l_{we}}} \right| =$$

$$= P(l_{we}) \frac{1}{(L-1)P(l_{we})} = \frac{1}{L-1}$$

A więc transformacja T spełnia stawiony cel, co oznacza że proces tzw. ‘wyrównania histogramu’ będzie generalnie bazować na kumulacji prawdopodobieństw wartości jasności z obrazu wejściowego przeskalowane przez wartość $L-1$.

Poniżej przykład obliczeniowy procesu wyrównania histogramu dla hipotetycznego obrazu:

Wartości jasności (l_{we}) $L = 6$	Liczba pikseli o określonej jasności	Prawdopodobieństwo $P(l_{we})$	$T(l_{we})$	Wartości wyjściowe (l_{wy}) – zaokrąglone funkcją floor()
0	790	0,199494949	0,997474745	0
1	1050	0,265151515	2,323232325	2
2	860	0,217171717	3,40909091	3
3	630	0,159090909	4,204545455	4
4	420	0,106060606	4,734848485	4
5	210	0,053030303	5	5

Ostatecznie, uzyskujemy następującą zmianę wartości jasności:

l_{we} wartość: 0 \rightarrow l_{wy} wartość: 0 ,

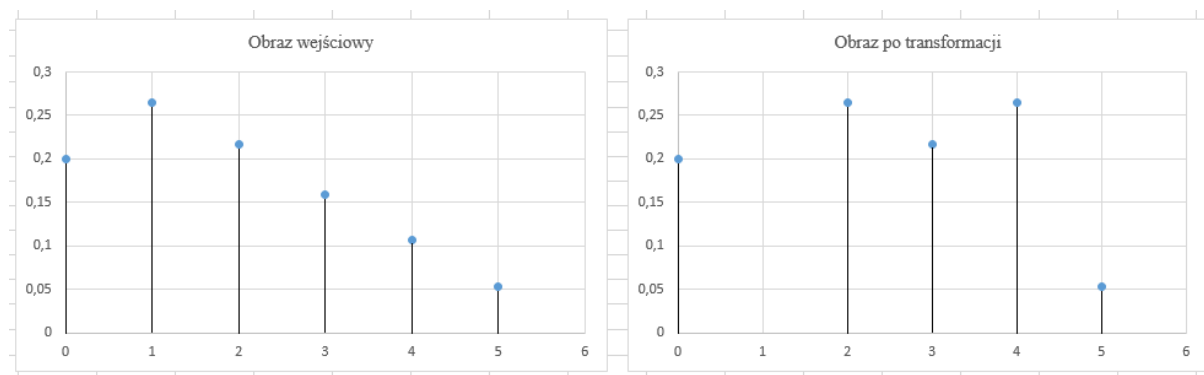
l_{we} wartość: 1 \rightarrow l_{wy} wartość: 2 ,

l_{we} wartość: 2 \rightarrow l_{wy} wartość: 3 ,

l_{we} wartość: 3 \rightarrow l_{wy} wartość: 4 ,

l_{we} wartość: 4 \rightarrow l_{wy} wartość: 4 ,

l_{we} wartość: 5 \rightarrow l_{wy} wartość: 5 .



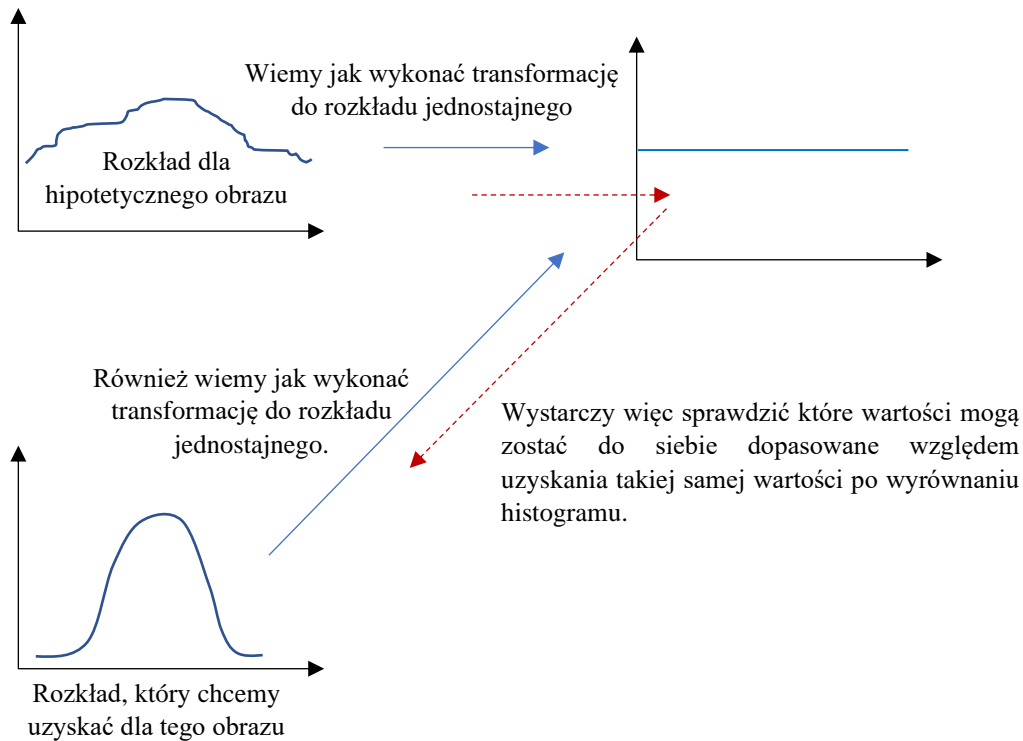
2. Transformacja do rozkładu docelowego (ang. *histogram matching*)

Naturalnym rozszerzeniem powyższego procesu jest możliwość tworzenia obrazów o zdanym rozkładzie prawdopodobieństwa (innym niż jednostajny). Proces umożliwiający osiągnięcie takiego celu jest bardzo prosty i sprowadza się do poniższych kroków:

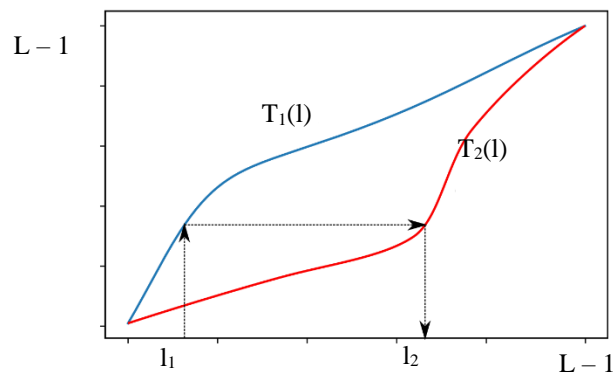
- Przekształcenie rozkładu analizowanego obrazu do rozkładu jednostajnego (‘wyrównanie histogramu’),

- Przekształcenie rozkładu żadanego dla obrazu do rozkładu jednostajnego ('wyrównanie histogramu'),
- Decyzja względem wyrównanego histogramu, które wartości jasności z obrazu wejściowego przekształcają się do których wartości z rozkładu docelowego.

Podstawowym założeniem tego procesu jest możliwość wykonania wyrównania histogramu dla dowolnego histogramu. Intuicja stojąca za procesem transformacji do rozkładu docelowego jest przedstawiona poniżej (zob. rys. 2) lub w kategoriach odpowiednich transformacji (rys. 3).



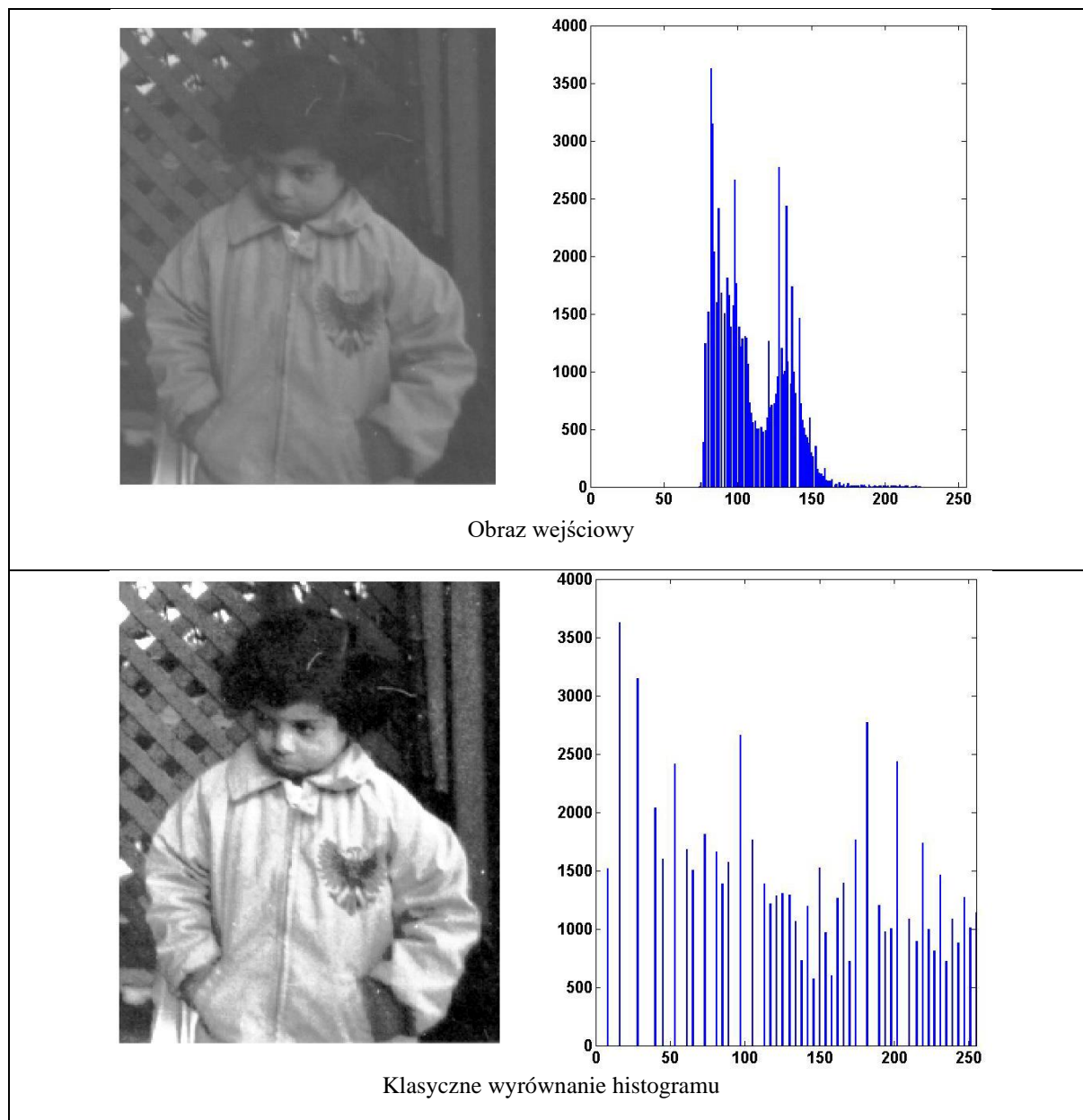
Rys. 2. Dopasowanie rozkładu, intuicja.



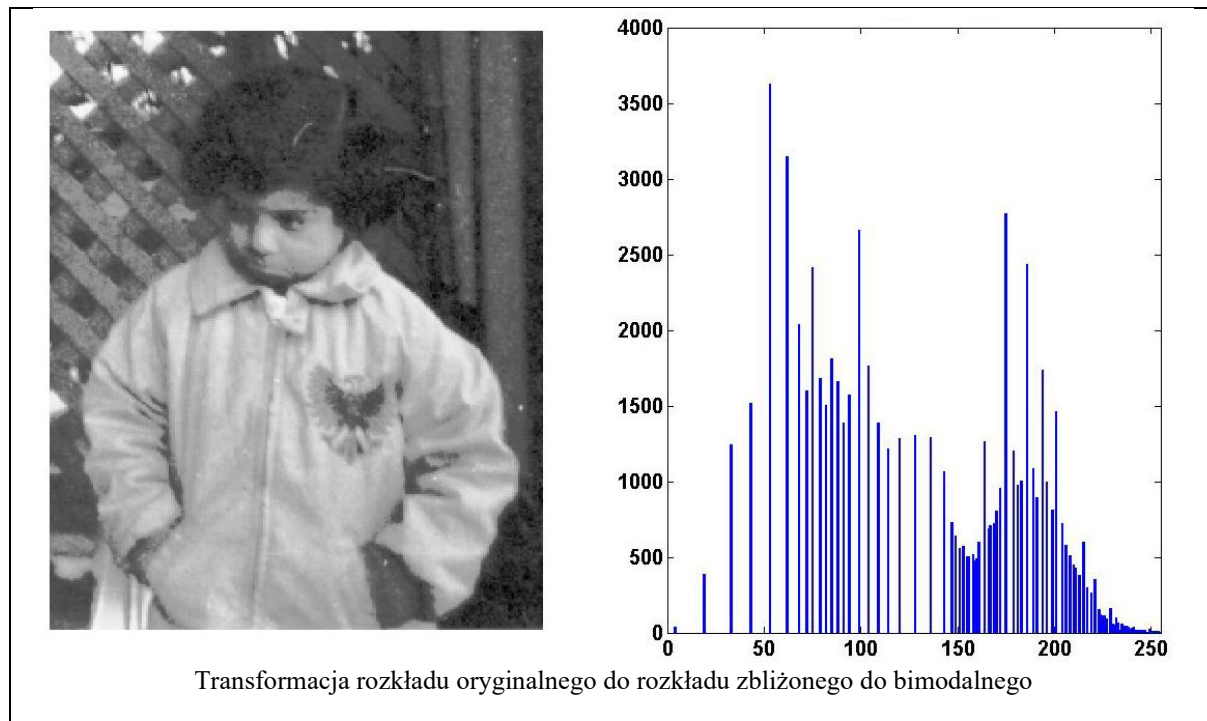
Rys. 3. Dopasowanie rozkładu, T_1 – transformacja dla obrazu wejściowego, T_2 – transformacja względem rozkładu docelowego. Wartość jasności l_1 przekształca się w l_2 .

Ewentualny problem może się pojawić gdy więcej niż jedna wartość z rozkładu docelowego zmienia swoją wartość po wyrównaniu histogramu. Wtedy, należy podjąć odpowiednią decyzję jak zmienić wartości jasności obrazu. Przykładowo: wartość jasności 9 z obrazu wejściowego po transformacji zmienia się np. na 12. Załóżmy również, że tę wartość przyjmują wartości z rozkładu docelowego: $\{10, 11, 12\}$. W takim przypadku, należy wybrać jedną wartość – np. najbliższą wartości 9 – czyli 10. A więc wartość 9 obrazu powinna być zmieniona na 10, aby uzyskać rozkład zbliżony do rozkładu docelowego.

Bardzo ciekawe może być wykorzystanie powyższej możliwości do wygenerowania rozkładu bimodalnego, który stanowi wejście do algorytmu binaryzacji Otsu (zob. lab2 – temat 'binaryzacja obrazu'). Poniżej przykład przekształcenia obrazu względem bimodalnego rozkładu docelowego³.



³ <https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/e56d6ceb-4a80-11e4-9553-005056977bd0/38e23fa7-e165-4987-8215-98cf3d3ef142/images/screenshot.jpg>



Literatura:

Gonzalez, R. C., Woods, R. E.: Digital Image Processing. 4th Edition, MedData Interactive, Pearson, (2018).