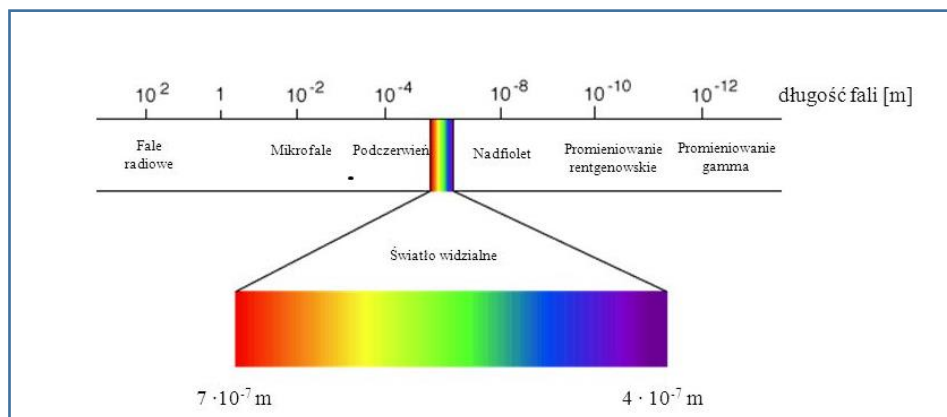


Laboratorium 2: Proste metody przetwarzania obrazów I

Modele barw¹

1. Wstęp

Barwa (kolor), którą widzimy, to nasza percepcja odbieranego przez ludzkie oko promieniowania elektromagnetycznego z zakresu światła widzialnego. Zakres długości fal elektromagnetycznych spektrum widzialnego to od około 400 nm – 700 nm (zob. rys. 1).



Rys. 1. Zakres światła widzialnego.

Jedną z teorii biologicznych mówiących o spostrzeganiu barw, zakłada reakcję grup komórek wzrokowych znajdujących się w siatkówce oka zwanych czopkami (łac. *coni*) na barwę *czerwoną*, *niebieską* i *zieloną*. Te właśnie trzy barwy, będziemy nazywać barwami podstawowymi (bazowymi). W grafice komputerowej i przetwarzaniu obrazów stosowane są różne rodzaje tzw. *modeli barw*. Model barw jest geometrycznym lub matematycznym sposobem na zapis koloru. Istnieje wiele różnych modeli barw, które znajdują różne zastosowania.

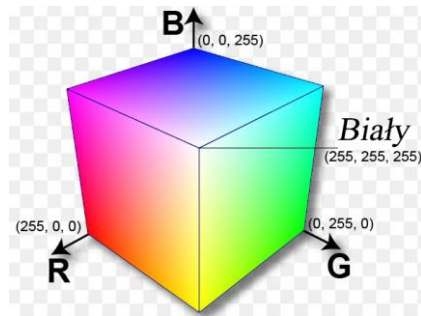
Celem niniejszego ćwiczenia jest zapoznanie się z możliwościami transformacji między modelami barw oraz ich zastosowaniem w przetwarzaniu i analizie obrazów.

2. Modele barw

– RGB

Najbardziej znanym i stosowanym modelem barw jest model RGB (ang. *red*, *green* i *blue*), w którym składowe koloru to barwy podstawowe (zob. rys. 2)

¹ Wykorzystane obrazy w opracowaniu mogą być przedmiotem praw autorskich.

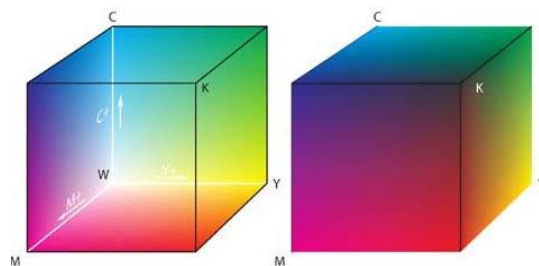


Rys. 2. Geometryczna reprezentacja modelu barw RGB

Model RGB jest tzw. addytywnym modelem barw i jest podstawowym rozwiązaniem w przypadku urządzeń wyświetlających obraz. Kiedy spostrzegamy kolor z określonego źródła światła, np. takiego jak telewizor lub monitor komputera, to mówimy o kolorze addytywnym. Kolor addytywny generowany jest na poprzez zmieszanie różnych długości fal światła w różnych proporcjach. W modelu RGB aby uzyskać różne barwy należy dodawać poszczególne składowe koloru. Barwa biała jest wynikiem zmieszania wszystkich składowych w maksymalnych wartościach, reprezentowana jako trójka: $(255_R, 255_G, 255_B)$.

– *CMYK*

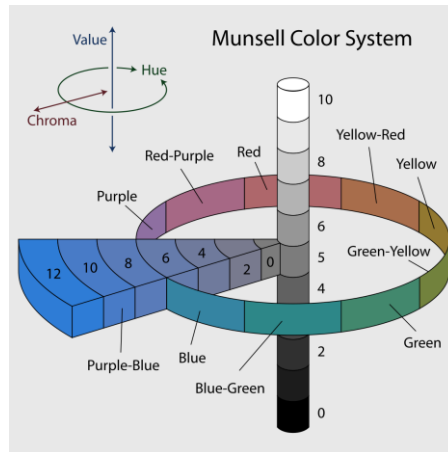
Odwrotne podejście, czyli model subtraktywny odnosi się do spostrzegania barw poprzez odbicie światła od obiektów, które również go absorbują. Takim modelem barw jest np. model CMYK (*cyjan, magenta, żółty, czarny*; rys. 3). Model ten stosowany jest w drukarstwie, a wrażenie barwy uzyskane poprzez spostrzeganie porcji odbitego światła z białej powierzchni, przechodzącą przez półprzezroczysty nadruk (działający jak filtry) składowymi kolorów modelu CMYK.



Rys. 3. Model CMYK

– *Model Munsella*

Powyższe modele barw znajdują szerokie zastosowanie w praktyce, ale istotnymi w kontekście analizy i przetwarzania obrazów, są modele barw najbardziej zbliżone do ludzkiej percepcji barwy. Prekursorem tych modeli jest model autorstwa prof. Alberta H. Munsella, na bazie którego wprowadzono znane i często stosowane w analizie obrazu cyfrowego modele barw HSV i HSL. W modelu Munsella jako pierwszy rozbito kolor na założone barwy podstawowe ułożone obok siebie w kole, jasność barwy oraz jej nasycenie i przedstawiono tę koncepcję geometrycznie w 3D (zob. rys. 4)




Rys. 4. Model barw Munsella

Koncepcja tworzenia koloru w tym modelu została stworzona według następujących założeń:

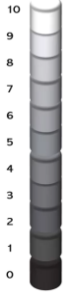
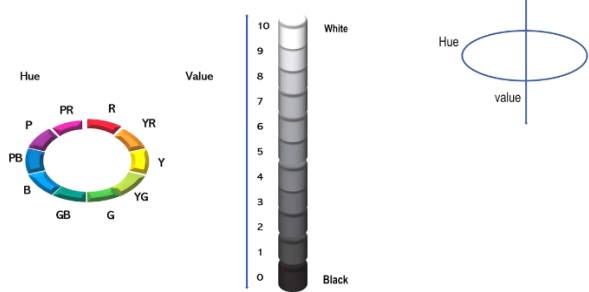
Rozpoznanie wartości barw podstawowych – składowa zwana **Hue** (barwa):

- podział na zbiór kolorów podstawowych w założeniu Munsella²:

<p>Hue</p>  <p>Koło barw podstawowych – barwy podstawowe + wtórne</p>	<p>Red</p> <p>Yellow</p> <p>Green</p> <p>Blue</p> <p>Purple</p> <p>Barwy podstawowe</p>	<table border="0"> <tr><td>R</td><td></td><td>Red</td></tr> <tr><td>YR</td><td></td><td>Yellow-Red</td></tr> <tr><td>Y</td><td></td><td>Yellow</td></tr> <tr><td>YG</td><td></td><td>Yellow-Green</td></tr> <tr><td>G</td><td></td><td>Green</td></tr> <tr><td>GB</td><td></td><td>Green-Blue</td></tr> <tr><td>B</td><td></td><td>Blue</td></tr> <tr><td>PB</td><td></td><td>Purple-Blue</td></tr> <tr><td>P</td><td></td><td>Purple</td></tr> <tr><td>PR</td><td></td><td>Purple-Red</td></tr> </table> <p>Barwy podstawowe + wtórne (zmieszanie sąsiednich podstawowych)</p>	R		Red	YR		Yellow-Red	Y		Yellow	YG		Yellow-Green	G		Green	GB		Green-Blue	B		Blue	PB		Purple-Blue	P		Purple	PR		Purple-Red
R		Red																														
YR		Yellow-Red																														
Y		Yellow																														
YG		Yellow-Green																														
G		Green																														
GB		Green-Blue																														
B		Blue																														
PB		Purple-Blue																														
P		Purple																														
PR		Purple-Red																														

Kwantyzacja wartości jasności – składowa zwana **Value** (koduje wartości jasności).

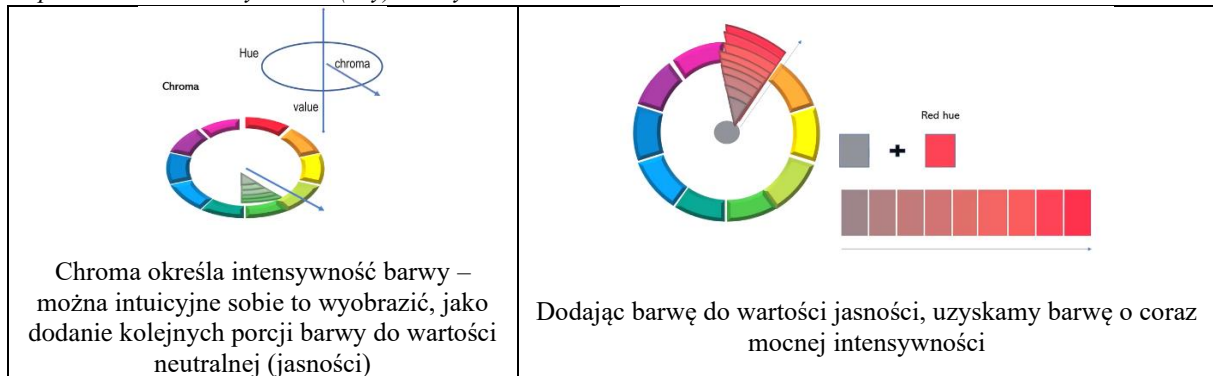
- podział na wartości jasności (odcieni szarości, względem czerni i bieli):

 <p>Wartości jasności: {0, ..., 10}</p>	 <p>Widać relację pomiędzy Hue i Value. Koło barw podstawowych ulokowane jest na środku przedziału Value (wartość 5)</p>
--	--

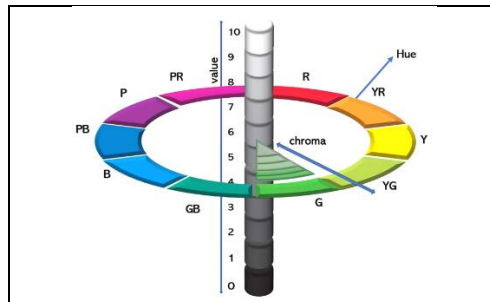
Wprowadzenie pojęcia nasycenia barwy – tzw. **Chroma**.

² Źródło wykorzystanych obrazów : <https://www.youtube.com/watch?v=MWbQirKEUU>

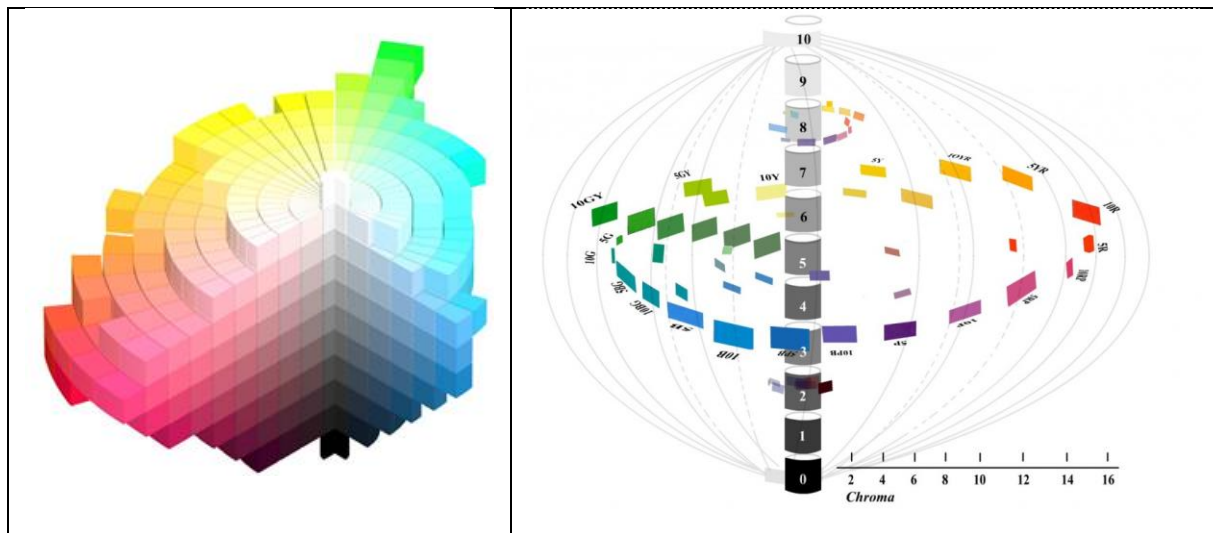
- wprowadzenie intensywności (siły) barwy:



Podsumowanie:

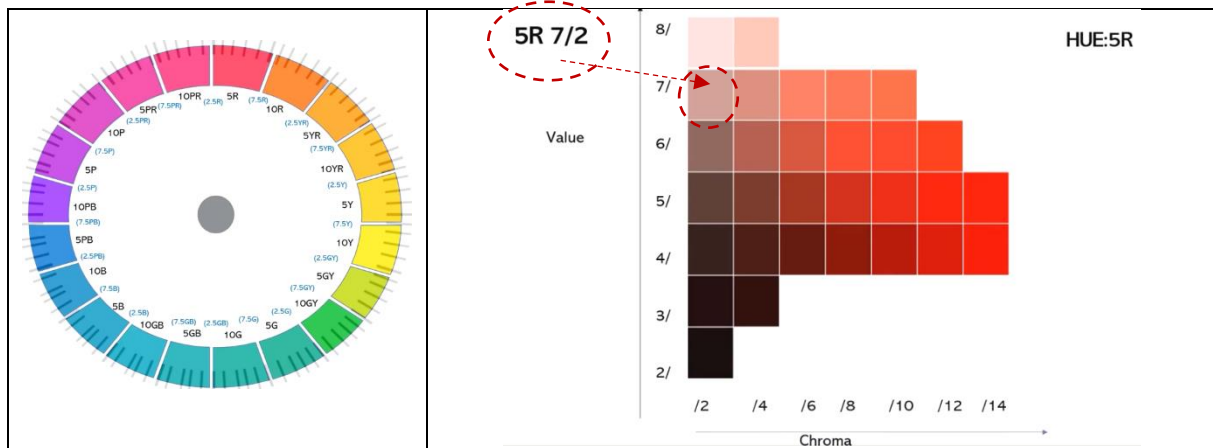


Dodatkowo, Munsell zauważył że niektóre barwy są ‘silniejsze’ niż inne, tj. generują więcej kolorów (potrzeba więcej ‘chromy’ aby uzyskać barwę). Np. barwa czerwona jest ‘silniejsza’ niż zielona, co odzwierciedlane jest geometrycznie w 3D jako struktura nieregularna (zob. rys. 5).



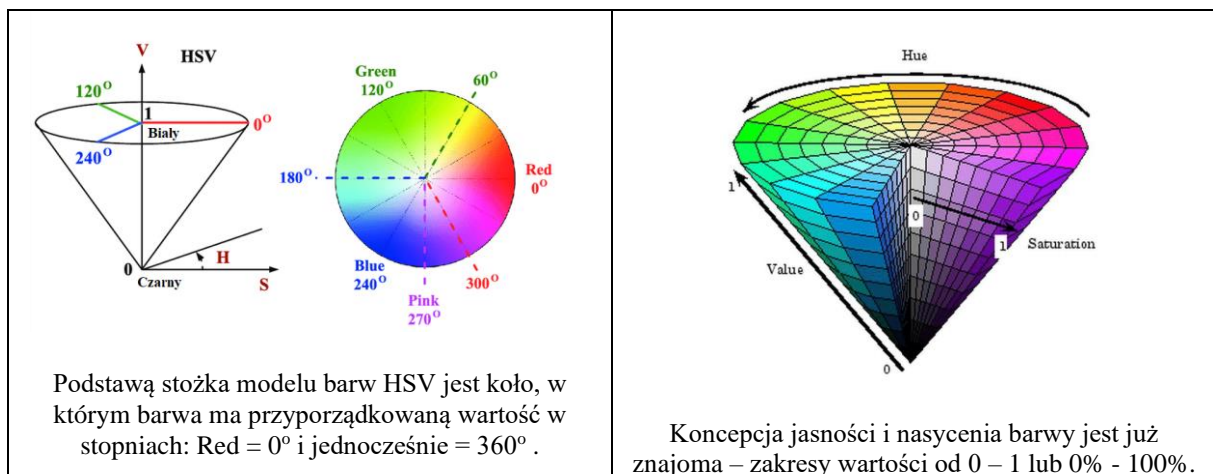
Rys. 5. Geometryczne przedstawienie struktury modelu barw Munsella

Dzięki tej koncepcji, można jednoznacznie zidentyfikować barwę poprzez odpowiednie kodowanie (dodatkowo zastosowano podział wartości barw podstawowych i wtórnych na 10 części każdą). Zapis barwy odbywa się poprzez podanie wartości Hue, a następnie Value/Chroma:



3. Przetwarzanie i analiza obrazów

Rozwinięciem i często stosowanym w dziedzinie przetwarzania obrazów cyfrowych modelu Munsella stanowią modele barw *HSV* i *HSL*. Poniżej, geometryczna interpretacja modelu HSV (zob. rys. 6).



Rys. 6. Geometryczna interpretacja modelu barw HSV

Dlaczego modele wywodzące się z pracy Munsella są tak istotne w przetwarzaniu i analizie obrazów ?

Model HSV umożliwia lepsze zrozumienie koncepcji koloru na poziomie logicznym, co wykorzystywane jest w różnych algorytmach analizy i przetwarzania obrazów. Dla przykładu, analizę obrazu w kierunku wykrywania krawędzi można dokonać pracując na składowej jasności modelu, segmentację poprzez progowanie można wykonać na składowej nasycenia barwy, zakładając że obiekty różnią się od tła względem mocniejszego nasycenia barwy itd.

Oczywiście, aby to dokonać, należy w sensie informatycznym umożliwić konwersję logiczną z jednego modelu barw do innego. Poniżej przedstawiono przykład takiej konwersji:

<i>RGB → HSV</i>	<i>HSV → RGB</i>
<p><u>Algorytm</u></p> <p><i>Wejście:</i> znormalizowane wartości R, G, i B: r, g, b ∈ [0, 1]</p> <p><i>Wyjście:</i> h ∈ [0°, 360°], s, v ∈ [0, 1] (jeśli s = 0, h jest nieokreślone)</p> <p><i>Działanie:</i> max := Max{r, g, b}, min := Min{r, g, b},</p> <p>v := max,</p> <p>Jeśli (max = 0) To s := 0, wpp. s := (max – min)/max ;</p> <p>Jeśli (s = 0) To h := (nieokreślony), wpp. :</p> <p>{ d := max – min; Jeśli (r = max) To h := (g – b)/d; wpp. Jeśli (g = max) To h := 2 + (b – r)/d; wpp. Jeśli (b = max) To h := 4 + (r – g)/d;</p> <p>h := 60 * h; Jeśli (h < 0) To h := h + 360; }</p> <p>Koniec.</p>	<p><u>Algorytm</u></p> <p><i>Wejście:</i> h ∈ [0°, 360°] lub nieokreślony s, v ∈ [0, 1]</p> <p><i>Wyjście:</i> r, g, b ∈ [0, 1]</p> <p><i>Działanie:</i> Jeśli (s = 0) To Jeśli (h jest nieokreślony ∉ [0°, 360°]) To r, g, b := v; wpp. wyprowadź błąd.</p> <p>wpp. { Jeśli (h = 360) To h := 0; wpp. h := h/60;</p> <p>zm := Floor(h); ulm = h – zm; (część ułamkowa)</p> <p>p := v * (1 – s); q := v * (1 – s*ulm); t := v * (1 – s*(1 – ulm));</p> <p>W przypadku, gdy – zm :</p> <p>0 : { r := v; g := t; b := p; } 1 : { r := q; g := v; b := p; } 2 : { r := p; g := v; b := t; } 3 : { r := p; g := q; b := v; } 4 : { r := t; g := p; b := v; } 5 : { r := v; g := p; b := q; } }</p> <p>Koniec.</p>

4. Inne modele

Istnieją również inne modele kolorów, wykorzystywane w różnych zastosowaniach praktycznych: YUV, YIQ (stosowane w telewizji i systemach wideo), YCbCr (cyfrowe obrazy i wideo), CIE XYZ, CIE L*U*V*, CIE L*a*b* (dopasowywanie kolorów, cyfrowa animacja, produkty multimedialne).

Literatura:

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

Agoston, M. K., Computer Graphics and Geometric Modelling, Implementation & Algorithms. Springer-Verlag London, (2005)

Ibraheem, N.A.; Hasan, M.M.; Khan, R.Z.; Mishra, P.K. Understanding color models: A review. ARPN J. Sci. Technol. (2012), 2, 265–275.

Binaryzacja Obrazu³

1. Wstęp

Proces binaryzacji obrazu zazwyczaj rozumiany jest jako przekształcenie obrazu w skali szarości do obrazu dwu kolorowego (czarno - biały). Zakładamy istnienie pewnej funkcji, która przypisywać będzie wartość 0 lub 1 (255) w zależności od wartości progu T (dlatego często binaryzacje obrazu nazywa się progowaniem obrazu – ang. *image thresholding*):

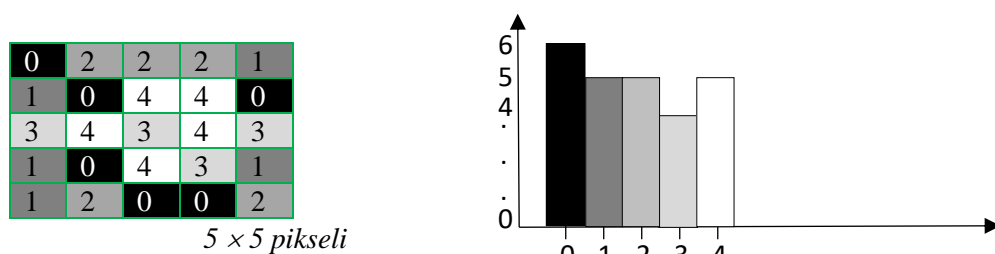
$$I(x, y) = \begin{cases} 0, & \text{jeśli } I(x, y) < T \\ 1 \text{ lub } 255, & \text{jeśli } I(x, y) \geq T \end{cases}$$

Zakładamy, $I(x, y) \in \mathbb{N}^d$, gdzie $d = 1$ (właściwie, niekoniecznie musimy rozważać wartości w zakresie $\{0, \dots, 255\}$, możemy pracować np. na wybranej składowej innego modelu barw). Przyjęło się również, że kolor czarny jest dla tła, natomiast biały dla obiektów znajdujących się na obrazie – celem binaryzacji przeważnie jest wyodrębnienie obiektów od tła (segmentacja obrazów poprzez progowanie).

Aby zrozumieć stosowane algorytmy binaryzacji – np. na przykładzie algorytmu Otsu, należy wstępnie zrozumieć czym jest histogram obrazu oraz wprowadzić pewną interpretację obrazu w kategoriach probabilistycznych.

– Histogram obrazu

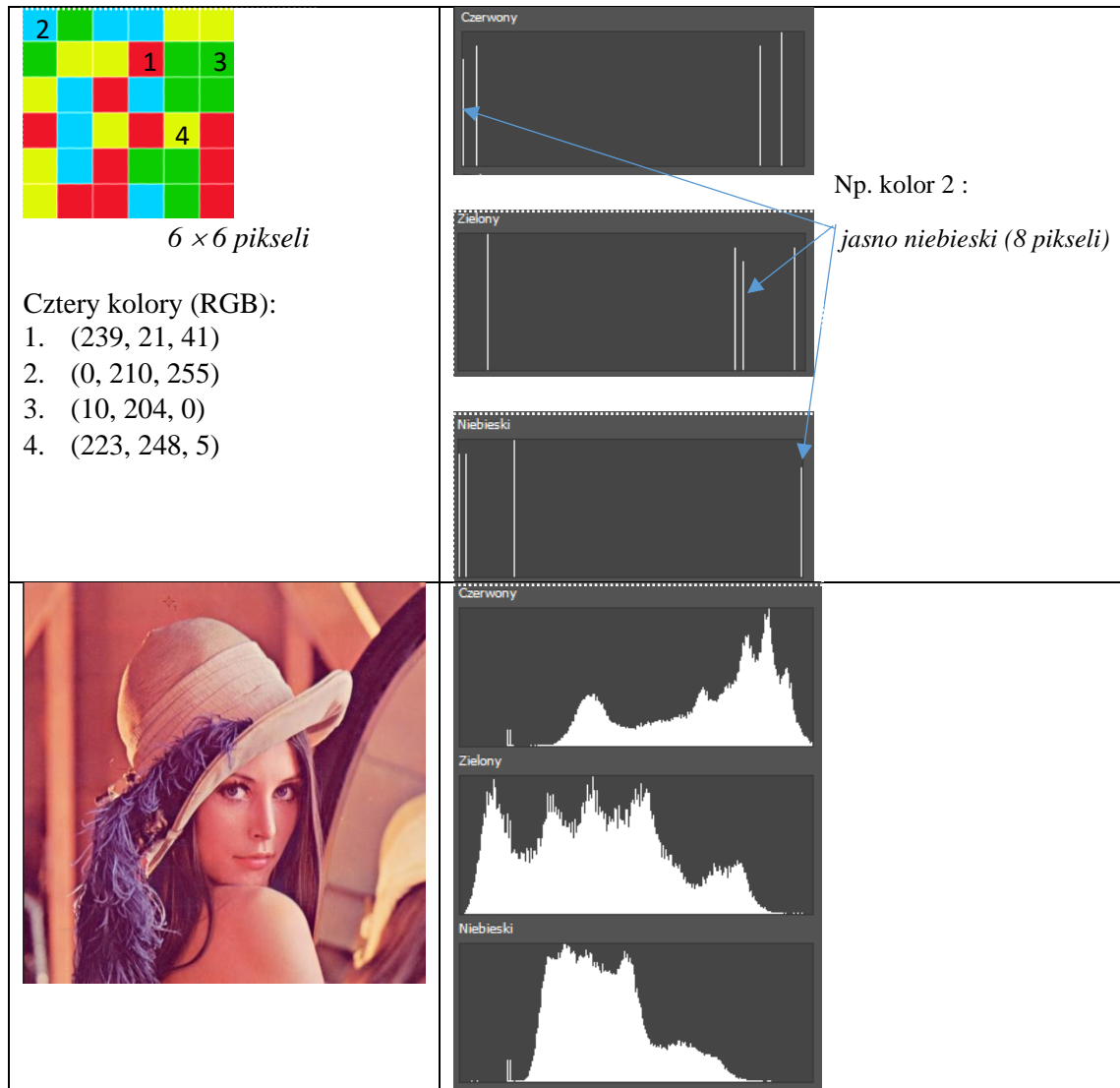
Histogram obrazu daje nam informacje o tym jaka jest częstotliwość występowania danej wartości piksela (koloru – czyli, np. w modelu RGB wartości poszczególnych kanałów koloru lub wartości odcienia szarości dla obrazów monochromatycznych). Dla przykładu, poniżej przedstawiono histogram hipotetycznego obrazu (zob. rys. 1).



Rys. 1. Histogram przykładowego obrazu

Poniżej, histogram obrazu w modelu RGB (obraz modelowy i rzeczywisty) – wizualizacja poszczególnych składowych.

³ Wykorzystane obrazy w opracowaniu mogą być przedmiotem praw autorskich.



– Interpretacja probabilistyczna

W rzeczywistości, korzystamy ze znormalizowanych histogramów (normalizacja – podział wartości częstości przez liczbę wszystkich pikseli na obrazie), które interpretowane są jako rozkłady prawdopodobieństwa realizacji pewnej dyskretnej zmiennej losowej. Oczywiście, w tym celu należy określić prawdopodobieństwo wylosowania danej wartości koloru na obrazie. Prawdopodobieństwo to określane jest jako:

$$P(A) =_{df} \frac{\text{Liczba pikseli o zadanej wartości koloru}}{\text{Liczba wszystkich pikseli na obrazie}},$$

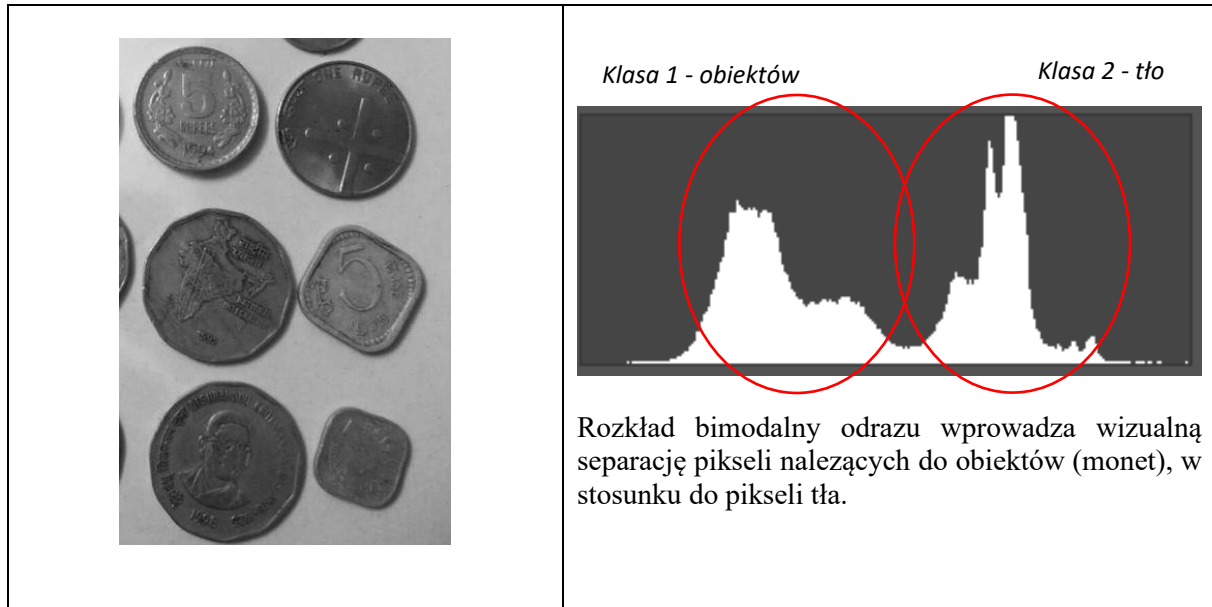
gdzie A oznacza zdarzenie wylosowania piksela o określonym kolorze, zadana wartość koloru w przypadku obrazów kolorowych oznacza zadaną wartością rozważanej składowej koloru.

Tym samym, znormalizowany histogram możemy interpretować jako rozkład prawdopodobieństwa realizacji pewnej dyskretnej zmiennej losowej - rozkład ten mówi nam o prawdopodobieństwie przyjęcia przez zmienną losową określoną wartość koloru.

Dzięki takiej interpretacji, możemy dalej wprowadzać takie pojęcia jak np. *wariancja*, mając prawdopodobieństwo możemy określić *entropię* obrazu itd.

To co dla nas jest ciekawe i dotyczy zastosowania algorytmu binaryzacji Otsu, związane jest z żadaną formą histogramu (rozkładu) do wykonania binaryzacji oraz założenie metody Otsu, badającej tzw. wariancje między klasowe lub wewnątrz klasowe.

Najlepszym wejściem dla algorytmu Otsu, jako że metoda ta polega na znalezieniu optymalnego progu dla binaryzacji obrazu, jest to obraz monochromatyczny o rozkładzie bimodalnym (zob. rys. 2).



Rys. 2. Odpowiednie wejście dla algorytmu binaryzacji Otsu

Tego typu histogramy często można spotkać np. w praktyce laboratoryjnej w mikrobiologii lub np. teksty pisane itd.

2. Algorytm Otsu

Algorytm Otsu minimalizuje ważoną wariancję wewnątrz klasową (zob. rys. 2) wyrażoną w sposób następujący:

$$\sigma_w^2(t) =_{df} q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t),$$

gdzie t jest wybranym progiem binaryzacji.

A więc, szukamy wartość najmniejszą dla powyższego równania. Próg optymalny znajdziemy poprzez przetestowanie wszystkich możliwych wartości progowych, gdzie:

$$q_1(t) = \sum_{i=1}^t P(i); \quad q_2(t) = \sum_{i=t+1}^I P(i)$$

$P(i)$ oznacza prawdopodobieństwo wystąpienia wartości szarości i (już wiadomo jak się go oblicza), a I oznacza ostatnią wartość odcienia szarości w obrazie na osi odciętych histogramu, oraz:

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)}, \quad \mu_2(t) = \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)},$$

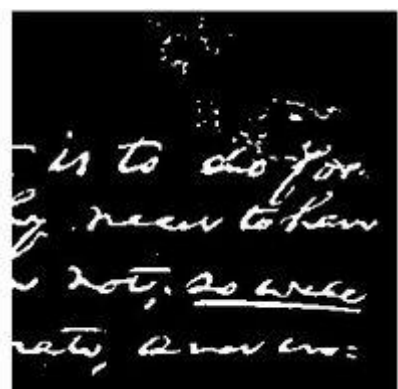
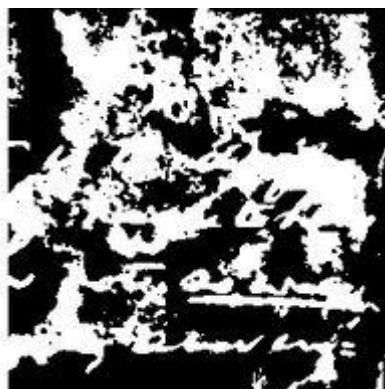
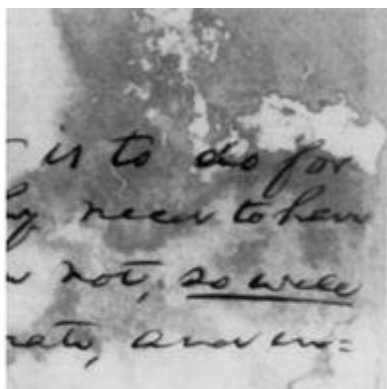
Ostatecznie, wyznaczamy wartości wariancji wewnątrz klasowych:

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} ; \quad \sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

Poniżej, różne wyniki działania algorytmu Otsu:



Często w praktyce mamy do czynienia z nierównym tłem – np. nierównomiernie oświetlone podłoże na którym znajdują się obiekty do segmentacji przez binaryzację. W tym przypadku, ciekawym rozwiązaniem jest zastosowanie metody Otsu przy wykorzystaniu okna przesuwającego o określonej wielkości – tzw. lokalne działanie algorytmu Otsu:



Oraz wejściowy

Otsu progowanie 'globalne'

Otsu progowanie 'lokalne'

Literatura:

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

Tadeusiewicz, R., Korohoda, P. Komputerowa analiza i przetwarzanie obrazów. Kraków : Wydaw. Fundacji Postępu Telekomunikacji, (1997).

Malina W, Smiatacz M., Cyfrowe przetwarzanie obrazów. Akademicka Oficyna Wydawnicza EXIT, (2008)

Otsu, N., A Threshold Selection Method from Gray-Level Histograms. „IEEE Transactions on Systems, Man, and Cybernetics”, 9 (1), 1979, s. 62–66

Filtracja obrazów cyfrowych⁴

1. Wstęp

Celem laboratorium jest zapoznanie się z zasadą filtracji obrazów cyfrowych poprzez tzw. konwolucję (splot) obrazu i filtru. Filtracja obrazów stanowi operację kontekstową, w której wartość piksela wyjściowego zależy od wartości pikseli z jego sąsiedztwa oraz wartości samego filtru. Filtracja wykonywana jest z różnych powodów – celem mogą być: usuwanie szumu z obrazu, wzmacnianie niektórych elementów obrazu (np. krawędzi), poprawy jakości wizualnej obrazu itd. Kluczowe w kontekście filtrów konwolucyjnych stanowią dwa zagadnienia – splot (konwolucja) dwóch funkcji oraz interpretacja obrazu cyfrowego w kategorii funkcji określonej na dziedzinie wartości współrzędnych pikseli oraz przeciwdziałanie wartości jasności lub składowych kolorów w zależności od zastosowanego modelu barw.

- Splot funkcji

Konwolucja dwóch funkcji (f i g) to operacja matematyczna, której wynikiem stanowi trzecia funkcja, która określa sposób modyfikacji kształtu jednej funkcji poprzez drugą. Wartości konwolucji (splotu) funkcji związane są ze stopniem pokrycia powierzchni pod funkcjami.

Definicja:

Za symbol operatora konwolucji często przyjmuje się: $*$ ($f * g$) lub \otimes ($f \otimes g$), my przyjmiemy zapis $[conv]$.

$$(f [conv] g)(t) =_{df} \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$

A więc, konwolucja dwóch funkcji to całka iloczynu dwóch funkcji, z których jedna (g) jest odbita względem osi rzędnych i przesunięta w kierunku drugiej (f).

Pomimo użycia symbolu t powyżej, nawiązującego do dziedziny czasu, ogólnie nie musi tak być. Jednak w kontekście takiej interpretacji, intuicja powyższego działania matematycznego jest następująca: konwolucja (splot) dwóch funkcji określa pole powierzchni pod funkcją $f(\tau)$ ważoną przez wartości funkcji $g(-\tau)$ przesuniętą w kierunku funkcji f o wartość t . Gdy t się zmienia, funkcja wagi $g(t - \tau)$ wskazuje różne części funkcji wejściowej $f(\tau)$ (zob. rys. 1).

W przypadku, gdy funkcje określone są tylko na wartościach dodatnich ($f, g : [0, \infty) \rightarrow \mathbb{R}$), mamy:

$$(f [conv] g)(t) =_{df} \int_0^t f(\tau)g(t - \tau) d\tau$$

Wybrane własności operatora konwolucji:

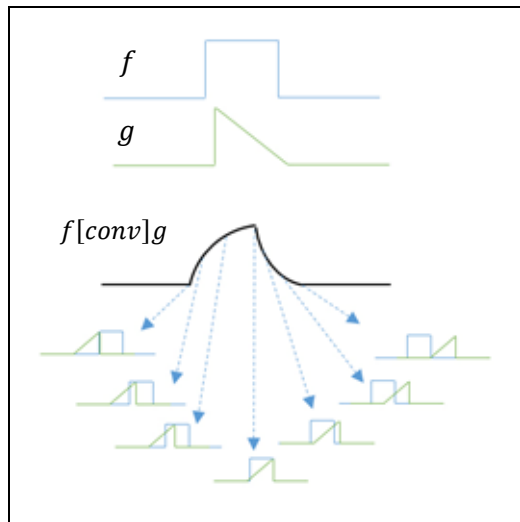
1) *przemienność*:

$$(f [conv] g) = (g [conv] f),$$

2) *łączność*:

$$(f [conv] (g [conv] h)) = ((f [conv] g) [conv] h).$$

⁴ Wykorzystane obrazy w opracowaniu mogą być przedmiotem praw autorskich.



Rys. 1. Wartości funkcji będącej konwolucją f i g ($f[\text{conv}]g$).
Obicie funkcji g i przesunięcie w kierunku funkcji f .

- Interpretacja obrazu

W kontekście filtracji, obraz należy rozumieć jako pewną funkcję określoną na dziedzinie wartości współrzędnych i przeciwdziedzinie – zbiorze wartości jasności lub kolorów. W przypadku rozważania poszczególnych wierszy kolorów, możemy określić funkcję jednej zmiennej, natomiast w przypadku ujęcia obrazu jako całość w naturalny sposób możemy określić obraz jako funkcja dwóch zmiennych. Dodatkowo, w technice cyfrowej po zastosowaniu przez odpowiednie urządzenia akwizycji obrazów procesów próbkowania oraz kwantyzacji, mowa o obrazie (I) jako o funkcji dyskretnej ($X, Y \neq \emptyset; X, Y \subseteq \mathbb{N}$):

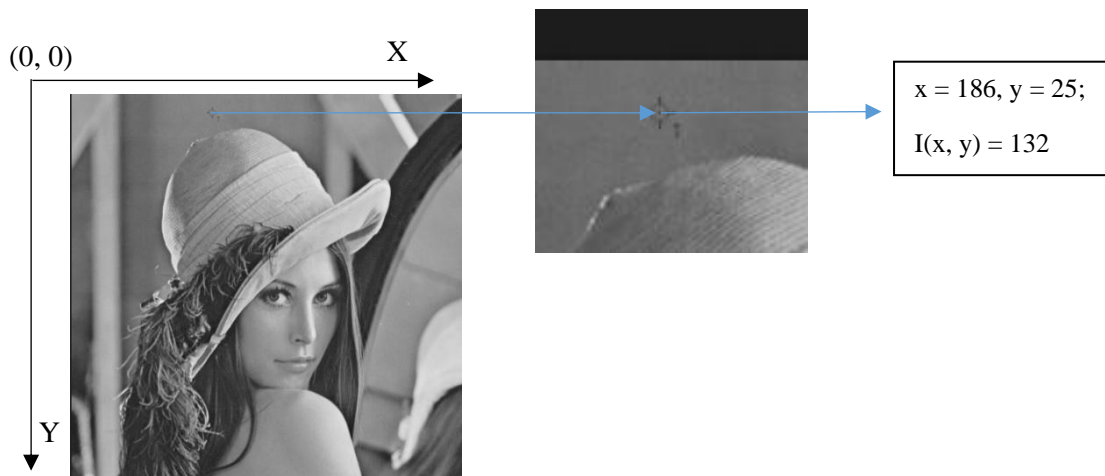
$$I: X \times Y \rightarrow \{0, \dots, 255\},$$

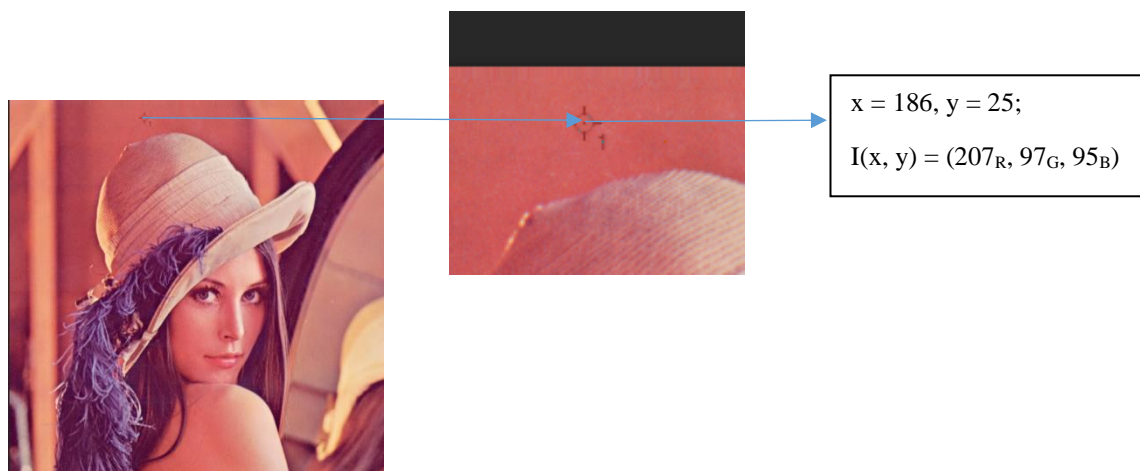
lub (obraz kolorowy, w klasycznym modelu RGB):

$$I: X \times Y \rightarrow \mathbb{N}^d$$

gdzie $I(x, y) \in \mathbb{N}^d$ ($d = 3$ dla modelu RGB, gdzie $R, G, B = \{0, \dots, 255\}$), zakładając 8 bitów na reprezentację jasności lub na poszczególne kanały koloru w wybranym modelu barw.

Intuicyjnie, obraz cyfrowy (dyskretny) można przedstawić jako macierz punktów (pikseli) o określonych wartościach jasności lub koloru (zob. rys. 2).





Rys. 2. Obraz jako funkcja dyskretna dwóch zmiennych.

2. Konwolucja funkcji dyskretnych

Operację konwolucji dwóch funkcji dyskretnych w 2D możemy określić w sposób następujący:

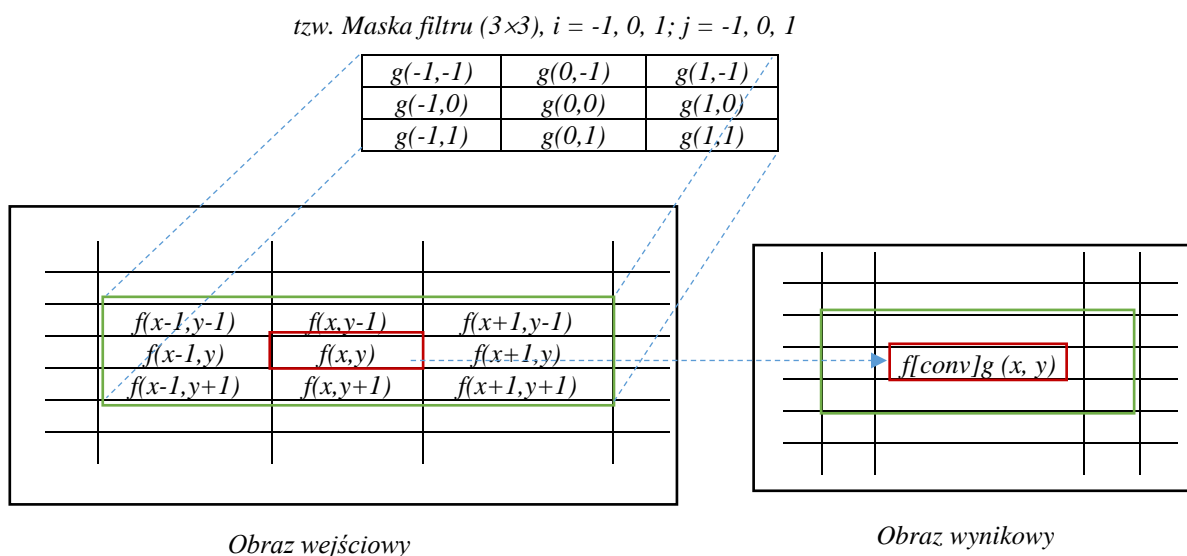
$$(f [conv] g)(x, y) =_{df} \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j) g(x - i, y - j)$$

lub stosując własność przemienności:

$$(f [conv] g)(x, y) =_{df} \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(x - i, y - j) g(i, j) .$$

W praktyce, i i j iterowane są w zależności od wielkości filtru g (wymiarów macierzy reprezentującej funkcję filtru).

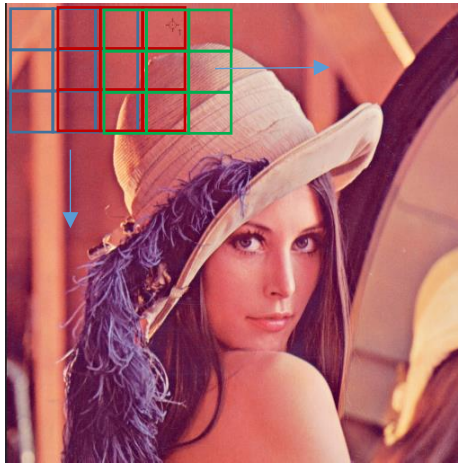
A więc, intuicyjnie filtracja obrazu dla danego punktu obrazu (x, y) stanowi wartość funkcji konwolucji (splotu) funkcji obrazu oraz filtru. Uzyskanie obrazu przefiltrowanego określonym filtrem oznacza przesunięcie filtru po całym obrazie. Wynikiem takiego działania staje się nowy obraz, wartości którego są wartościami splotu funkcji obrazu wejściowego i filtru (zob. rys. 3).



Rys. 3. Splot obrazu i filtru w punkcie (x, y) .

Analiza i przetwarzanie obrazów i wideo


Filtr przesuwany jest po całym obrazie z lewej do prawej oraz z góry na dół (rys. 4, dla filtru 3×3).




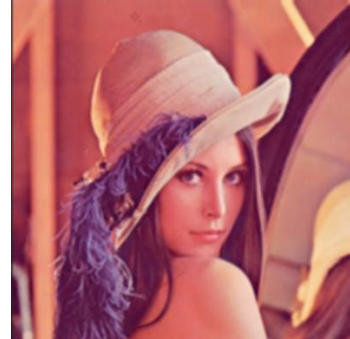

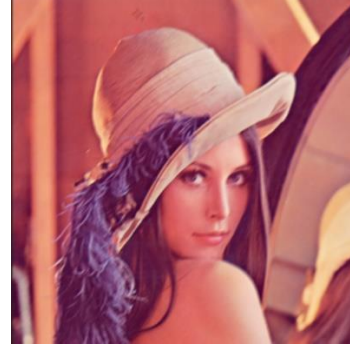
Rys. 4. Filtracja obrazu



Poniżej, wyniki działania wybranych filtrów (zakładamy zastosowanie normalizacji filtru, czyli podział wartości filtru przez ich sumę). Dodatkowo, należy pamiętać, że w przypadku obrazów kolorowych (model barw RGB) trzeba przefiltrować każdą składową koloru osobno.

	<table><tr><td>0</td><td>-1</td><td>0</td></tr><tr><td>-1</td><td>5</td><td>-1</td></tr><tr><td>0</td><td>-1</td><td>0</td></tr></table> <p>Maska filtru: <i>filtr wyostrzający</i> (tzw. górnoprzepustowy) wartość norm. = 1</p>	0	-1	0	-1	5	-1	0	-1	0	
0	-1	0									
-1	5	-1									
0	-1	0									
	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <p>Maska filtru: <i>filtr uśredniający</i> (tzw. dolnoprzepustowy) wartość norm. = 9</p>	1	1	1	1	1	1	1	1	1	
1	1	1									
1	1	1									
1	1	1									
<i>Obraz wejściowy</i>		<i>Obraz wynikowy</i>									
<i>Obraz wejściowy</i>		<i>Obraz wynikowy</i>									

	<table border="1" data-bbox="646 280 853 376"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>-8</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <p data-bbox="654 409 845 562"> Maska filtru: filtr detekcji krawędzi wartość norm. przyjmujemy = 1 </p>	1	1	1	1	-8	1	1	1	1	
1	1	1									
1	-8	1									
1	1	1									
<p><i>Obraz wejściowy</i></p>		<p><i>Obraz wynikowy</i></p>									

Najczęściej filtracja odbywa się kwadratową maską filtru 3×3 , aczkolwiek można uwzględnić większy kontekst działania filtru – maski: 5×5 , 7×7 itd. co automatycznie wzmocni efekt. Rozumiejąc już zasadę filtracji obrazu, można wykorzystać w zależności od potrzeb również inny kształt maski niż kwadratowy o ile zostanie zachowana możliwość wyboru piksela referencyjnego i ma sens w kontekście stawionego celu (rys. 4 – na przykładzie filtracji uśredniającej).

	<table border="1" data-bbox="609 947 960 1108"> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table> <p data-bbox="646 1142 928 1265"> Maska filtru: filtr uśredniający (większe otoczenie: 5×5) wartość norm. = 25 </p>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1																							
1	1	1	1	1																							
1	1	1	1	1																							
1	1	1	1	1																							
1	1	1	1	1																							
<p><i>Obraz wejściowy</i></p>		<p><i>Obraz wynikowy</i></p>																									
	<table border="1" data-bbox="609 1339 960 1500"> <tr><td></td><td></td><td></td><td>1</td><td>1</td></tr> <tr><td></td><td></td><td>1</td><td>1</td><td>1</td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>1</td><td></td><td></td></tr> <tr><td>1</td><td>1</td><td></td><td></td><td></td></tr> </table> <p data-bbox="646 1563 922 1686"> Maska filtru: filtr uśredniający (kształt otoczenia: romb) wartość norm. = 13 </p>				1	1			1	1	1		1	1	1		1	1	1			1	1				
			1	1																							
		1	1	1																							
	1	1	1																								
1	1	1																									
1	1																										
<p><i>Obraz wejściowy</i></p>		<p><i>Obraz wynikowy</i></p>																									

 <p><i>Obraz wejściowy</i></p>	<table border="1" style="margin: 0 auto; border-collapse: collapse;"> <tr><td>1</td><td></td><td>1</td><td></td><td>1</td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td style="color: red;">1</td><td>1</td><td>1</td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td></td><td>1</td><td></td><td>1</td></tr> </table> <p>Maska filtru: <i>filtr uśredniający</i> <i>(kształt otoczenia: gwiazda)</i> <i>wartość norm. = 17</i></p>	1		1		1		1	1	1		1	1	1	1	1		1	1	1		1		1		1	 <p><i>Obraz wynikowy</i></p>
1		1		1																							
	1	1	1																								
1	1	1	1	1																							
	1	1	1																								
1		1		1																							

Poniżej maski często wykorzystywanych filtrów:

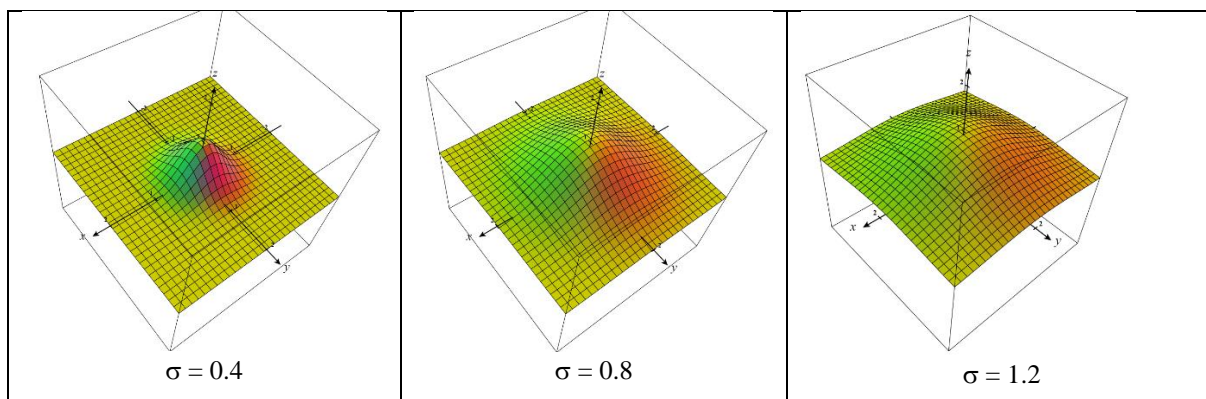
Filtry uśredniające (dolnoprzepustowe) – często stosowane do usuwania szumu na obrazie	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	1	1	1	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	2	1	1	1	1
	1	1	1																	
	1	1	1																	
	1	1	1																	
	1	1	1																	
	1	2	1																	
1	1	1																		
<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>4</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	4	1	1	1	1	<table><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>2</td><td>4</td><td>2</td></tr><tr><td>1</td><td>2</td><td>1</td></tr></table>	1	2	1	2	4	2	1	2	1	
1	1	1																		
1	4	1																		
1	1	1																		
1	2	1																		
2	4	2																		
1	2	1																		
Filtry wyostrzające (górnoprzepustowe)	<table><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>-1</td><td>9</td><td>-1</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table>	-1	-1	-1	-1	9	-1	-1	-1	-1	<table><tr><td>0</td><td>-1</td><td>0</td></tr><tr><td>-1</td><td>5</td><td>-1</td></tr><tr><td>0</td><td>-1</td><td>0</td></tr></table>	0	-1	0	-1	5	-1	0	-1	0
	-1	-1	-1																	
	-1	9	-1																	
	-1	-1	-1																	
	0	-1	0																	
	-1	5	-1																	
0	-1	0																		
<table><tr><td>1</td><td>-2</td><td>1</td></tr><tr><td>-2</td><td>5</td><td>-2</td></tr><tr><td>1</td><td>-2</td><td>1</td></tr></table>	1	-2	1	-2	5	-2	1	-2	1	<table><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>-1</td><td>14</td><td>-1</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table>	-1	-1	-1	-1	14	-1	-1	-1	-1	
1	-2	1																		
-2	5	-2																		
1	-2	1																		
-1	-1	-1																		
-1	14	-1																		
-1	-1	-1																		
Filtry wykrywające krawędzie	<table><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>-2</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table>	0	1	0	0	-2	0	0	1	0	<table><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>-2</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	1	-2	1	0	0	0
	0	1	0																	
	0	-2	0																	
	0	1	0																	
	0	0	0																	
	1	-2	1																	
	0	0	0																	
	<table><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>-2</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr></table>	0	0	1	0	-2	0	1	0	0	<table><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>-2</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td></tr></table>	1	0	0	0	-2	0	0	0	1
	0	0	1																	
	0	-2	0																	
	1	0	0																	
	1	0	0																	
0	-2	0																		
0	0	1																		
<table><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>-4</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table>	0	1	0	1	-4	1	0	1	0	<table><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>-4</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr></table>	1	0	1	0	-4	0	1	0	1	
0	1	0																		
1	-4	1																		
0	1	0																		
1	0	1																		
0	-4	0																		
1	0	1																		
<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>-8</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	-8	1	1	1	1											
1	1	1																		
1	-8	1																		
1	1	1																		

3. Filtry Gaussowskie

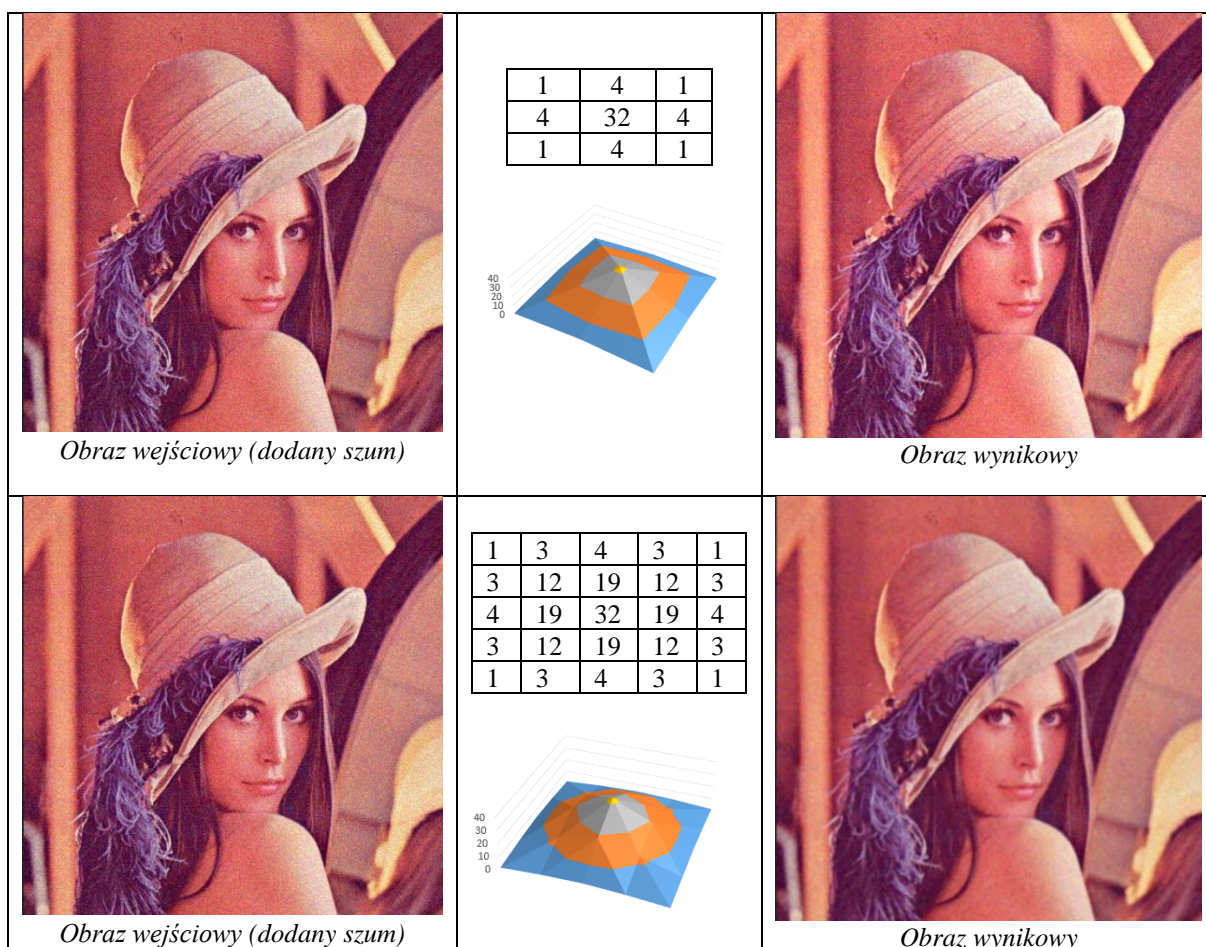
Ciekawym rozwiązaniem stosowanym w celu redukcji szumu na obrazie stanowią filtry Gaussowskie. Ich współczynniki aproksymują dwuwymiarową funkcję Gaussa:

$$f(x, y) \sim e^{-\frac{(x^2+y^2)}{2\sigma^2}}, \text{ zakładając } \mu_x = \mu_y = 0 \text{ oraz } \sigma = \sigma_x = \sigma_y.$$

Poniżej⁵ kilka przykładów o różnych wartościach σ , w zakresie $-2 \leq x, y \leq 2$.



W sposób naturalny, w filtrach Gaussowskich największą wagę otrzymuje element centralny, natomiast pozostałe współczynniki mają tym mniejszą wartość im dalej znajdują się od elementu centralnego. Poniżej kilka przykładów filtrów Gaussa oraz ich efekt na obrazie.



⁵ Grafiki wykonano w programie CalcPlot3D (<https://c3d.libretexts.org/CalcPlot3D/index.html>)



Obraz wejściowy (dodany szum)



Obraz wynikowy (filtr Gaussa 5×5)



Obraz wynikowy (filtr uśredniający 5×5)
Widać większą degradację szczegółów.

4. Inne filtry

Ciekawym podejściem do filtracji obrazów stanowią filtry statystyczne dotyczące wartości tendencji centralnych, takie jak *średnia*, czy też *mediana*. O ile średnia została już ujęta w kontekście filtru uśredniającego o wartościach wszystkich wag = 1, to mediana stanowi również ciekawy wybór, zwłaszcza w przypadku próby redukcji szumu na obrazie, ponieważ wartości odstające mają na nią mniejszy wpływ.

Zasada działania filtru mediany jest bardzo prosta:

- 1) Wybierz otoczenie (kształt maski filtru) – w zależności od celu, może to być klasyczna maska kwadratowa, ale nie musi. Ciekawym podejściem np. stanowi określenie otoczenia jako okręgu o określonym promieniu, względem analizowanego piksela.

- 2) Posortuj wartości pikseli z otoczenia.
- 3) Wybierz wartość mediany jako wartość dla piksela referencyjnego.

Poniżej przykład:



W podobny sposób można określić również filtry minimalny i maksymalny, poprzez wybór wartości *min* i *max* z rozważanego otoczenia piksela referencyjnego:

		
<i>Obraz wejściowy</i>	Maksymalny (otoczenie: okrąg, promień = 1)	Maksymalny (otoczenie: okrąg, promień = 3)
		
<i>Obraz wejściowy</i>	Minimalny (otoczenie: okrąg, promień = 1)	Minimalny (otoczenie: okrąg, promień = 3)

Literatura:

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

Tadeusiewicz, R., Korohoda, P. Komputerowa analiza i przetwarzanie obrazów. Kraków : Wydaw. Fundacji Postępu Telekomunikacji, (1997).

Malina W, Smiatacz M., Cyfrowe przetwarzanie obrazów. Akademicka Oficyna Wydawnicza EXIT, (2008)