

Przetwarzanie obrazów 3

- Handout

SL 2024/2025

Wydział Matematyki, Fizyki i Informatyki



Dr.-Ing. A. Rowińska-Schwarzsteller
agnieszka.rowinska-schwarzsteller@ug.edu.pl

11. März 2025

Spis treści

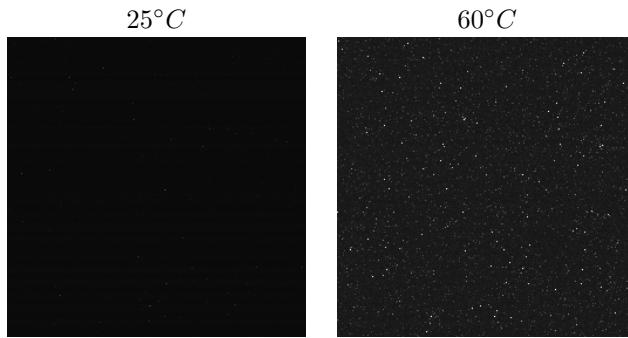
6 Obrazy i statystyka	2
6.1 Dlaczego statystyka?	2
6.2 Profil liniowy	3
6.3 Histogram	4
6.4 Operatory punktowe oparte na histogramie	5
6.4.1 Globalne metody progowania	6
6.4.2 Przesunięcie histogramu	9
6.4.3 Rozproszenie histogramu	10
6.4.4 Ograniczenie histogramu, clipping	10
6.4.5 Inwersja histogramu	11
6.4.6 Inne transformacje histogramu	11
6.5 Znormalizowany histogram	14
6.5.1 Centralne momenty obrazu	15
6.6 Skumulowany histogram, wyrównanie i hiperbolizacja histogramu	15
6.7 Kolor w celu zwiększenia kontrastu	17
7 Inne operatory punktowe	20
7.1 Arytmetyka obrazu	20
7.1.1 Dodawanie obrazów	21
7.1.2 Odejmowanie obrazów	23
7.1.3 Mnożenie obrazów	24
7.1.4 Korekta niejednorodnego oświetlenia - dzielenie obrazów	29
7.1.5 Ograniczenie zakresu wartości	31
7.2 Operacje logiczne na obrazach	33
7.2.1 Operacja \vee	33
7.2.2 Operacja \wedge	34
7.2.3 Operacja \neg	37
7.2.4 Chroma key	38

6

Obrazy i statystyka

6.1 Dlaczego statystyka?

- Wielkości statystyczne dotyczące obrazu
 - ↪ Histogramy wartości szarości, histogramy kolorów, entropia
 - ↪ Profile liniowe, zintegrowane profile liniowe
 - ↪ Centroida, wariancja, odchylenie standardowe, momenty centralne
- Możliwości wykorzystania tych wielkości do oceny obrazów i ich korekty
 - ↪ Prześwietlenie, niedoświetlenie, zakres dynamiczny, zdolność kompresji, lokalizacja błędów, ogniskowanie,...
- Statystyczne zjawiska szumu i ich kontrola
 - ↪ Szum fotonowy, szum termiczny (szum spowodowany termiczną generacją nośników w półprzewodniku → grafika 6.1) itp.

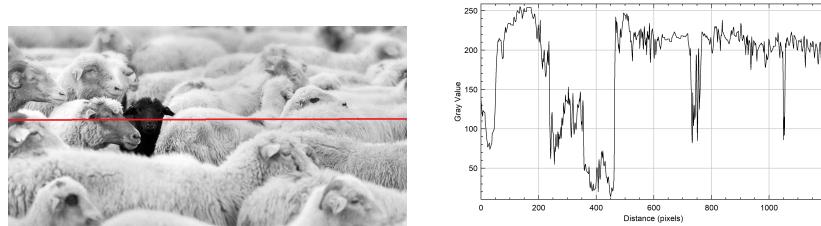


Grafika 6.1: Szum termiczny: sygnały ciemnego obrazu z wysokiej jakości czujnika z technologią CMOS (wzmocnionego 8–krotnie przy czasie ekspozycji wynoszącym 8s).
Zródło: wikibooks.org

6.2 Profil liniowy

Profil liniowy to przebieg wartości szarości obrazu wzdłuż dowolnej linii na obrazie.
→ grafika 6.2

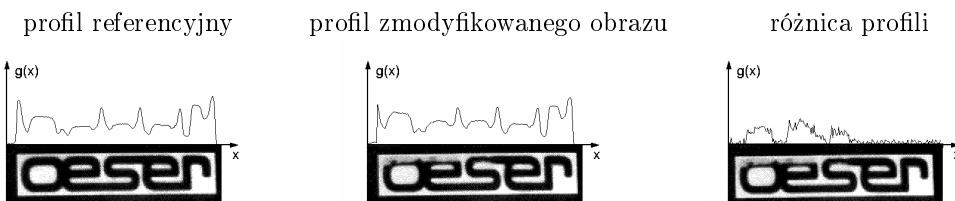
- Profile liniowe umożliwiają wykrywanie niewielkich wahań wartości szarości lub lokalnych błędów na obrazie.
- Ostre krawędzie obiektów tworzą strome profile, a obrazy o nieostrych krawędziach obiektów mają gładkie profile liniowe.



Grafika 6.2: Profil liniowy obrazu wzdłuż zaznaczonej linii

Zastosowanie profilu liniowego:

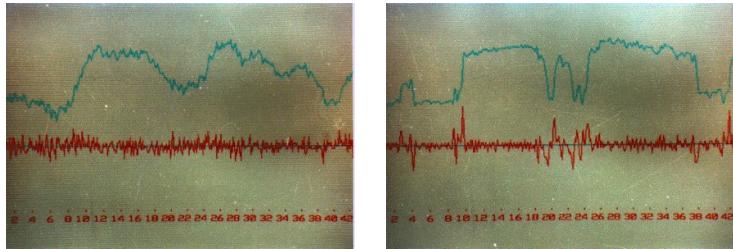
- lokalizacja błędów w obrazie poprzez porównanie z profilem referencyjnym:
- **zintegrowany profil wierszy** to profil liniowy zsumowany w kilku wierszach, analogicznie **zintegrowany profil kolumn**
 - ważne i przede wszystkim szybkie narzędzie porównania obrazów
 - umożliwiają lokalizację błędów poprzez porównanie z profilem referencyjnym (grafika 6.3)
 - bardzo wytrzymałe narzędzie ze względu na niezmienność przesunięcia
 - kombinacja profili wierszy i kolumn pozwala na dokładną lokalizację błędów w obrazie



Grafika 6.3: (Przeskalowana) różnica zintegrowanych profili wierszy obrazów. Źródło: Materiały do wykładu "Digitale Bildverarbeitung", B. Haasdonk

- precyzyjne ogniskowanie autofokusa aparatu fotograficznego/mikroskopu aż do znalezienia stromego profilu linii (tzn. największej pochodnej). → grafika 6.4

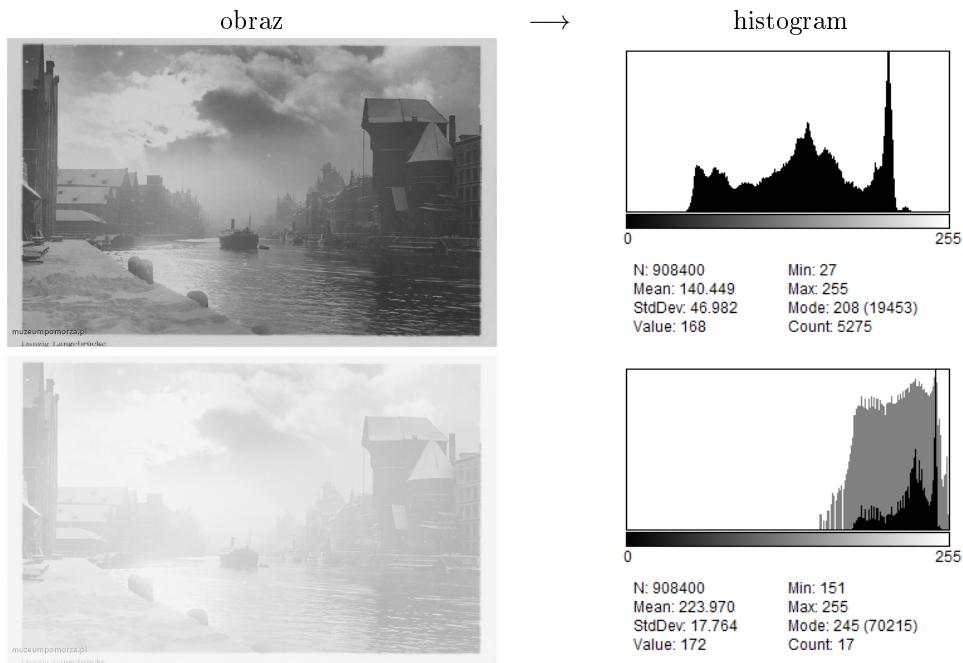
→ ...



Grafika 6.4: Precyzyjne ogniskowanie autofokusa mikroskopu aż do znalezienia stromego profilu linii (tzn. największej pochodnej). Źródło: Materiały do wykładu "Digitale Bildverarbeitung", B. Haasdonk

6.3 Histogram

→ grafika 6.5



Grafika 6.5: Histogram wartości szarości (wartości jasnoszare: histogram w skali logarytmicznej)

Wartości charakteryzujące obraz na histogramie:

1. **N**: liczba pikseli obrazu
2. **Max**: najwyższy poziom wartości szarości
3. **Min**: najniższy poziom wartości szarości

4. **Avg, Mean:** średnia wartość szarości

5. **StdDev:** odchylenie standardowe

Definicja histogramu $h(g)$:

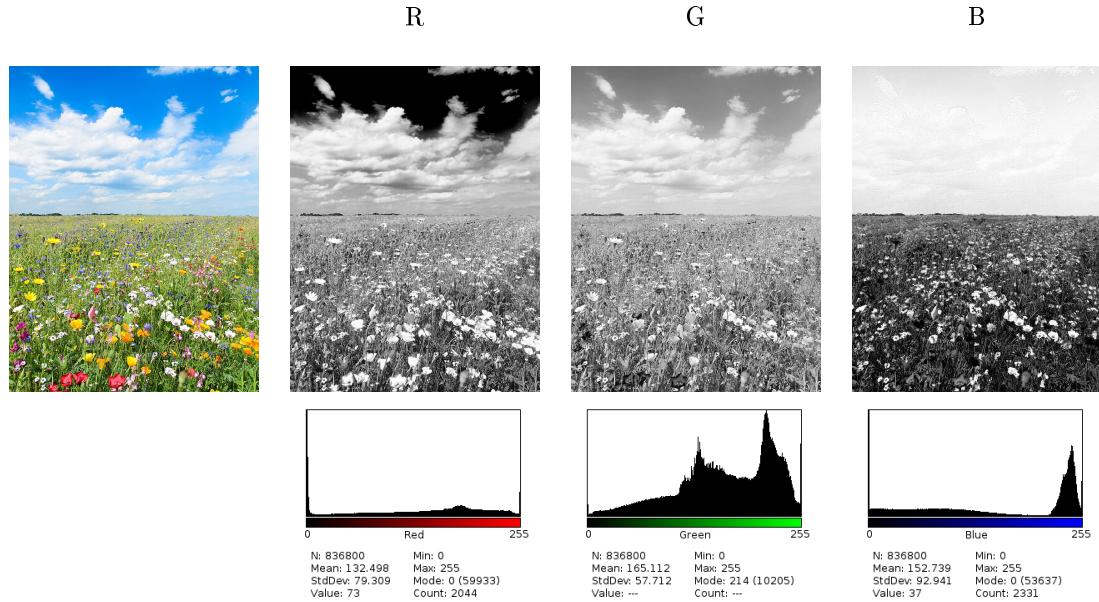
$h(g) = \text{liczba pikseli obrazu } I(\mathcal{M} \times \mathcal{N}) \text{ o wartości intensywności } g \in \mathcal{G} = \{0, \dots, G-1\}$:

$$h(g) = \sum_n^N \sum_m^M \delta(f(m, n) - g), \quad \text{gdzie } \delta(i) = \begin{cases} 1 & \text{dla } i = 0 \\ 0 & \text{dla } i \neq 0 \end{cases}$$

$$h(g) = \text{card} \{(m, n) | f(m, n) = g\}$$

Histogram kolorów RGB

Kanały kolorów są pokazane indywidualnie na histogramach → grafika 6.6



Grafika 6.6: Obraz i jego histogramy kanałów RGB.

6.4 Operatory punktowe oparte na histogramie

Operatory punktowe: klasa operacji przetwarzania obrazu.

We wszystkich procedurach tej klasy tworzona jest nowa wartość koloru lub szarości piksela wyłącznie w zależności od jego własnego poprzedniego koloru lub wartość szarości i jego pozycji bez względu na sąsiedztwo i/lub kontekst piksela (→ operatory Low-Level)

Transformacja obrazu

$$f_2(m, n) = T(f_1(m, n))$$

$$\begin{array}{ll} f_1(m, n) & \text{obraz oryginalny} \\ f_2(m, n) & \text{obraz przetworzony} \\ T & \text{transformacja} \end{array}$$

Alternatywnie:
transformacja wartości szarości (koloru)

$$g^* = T(g)$$

$$\begin{array}{ll} T : \{0, \dots, G-1\} \rightarrow \{0, \dots, G-1\} & \text{f. dyskretna} \\ T : [0, 1] \rightarrow [0, 1] & \text{f. ciągła} \end{array}$$

6.4.1 Globalne metody progowania

→ grafika 6.7

Binaryzacja obrazu

$$f_2(m, n) = \begin{cases} 0 : & f_1(m, n) \leq l \\ G-1 : & f_1(m, n) > l \end{cases} \quad T_{\text{thresh}}(g) = \begin{cases} 0 : & g \leq l \\ G-1 : & g > l \end{cases}$$

$$f_2(m, n) = \begin{cases} 0 : & f_1(m, n) < l_1 \\ G-1 : & l_1 \leq f_1(m, n) \leq l_2 \\ 0 : & f_1(m, n) > l_2 \end{cases}$$

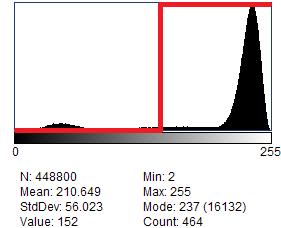
0 → czarny

G-1 → największa możliwa wartość szarości (biały)

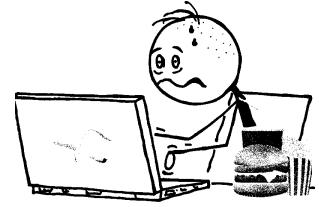
obraz oryginalny



histogram z krzywą transformacji
 $l = 154$



obraz przetworzony



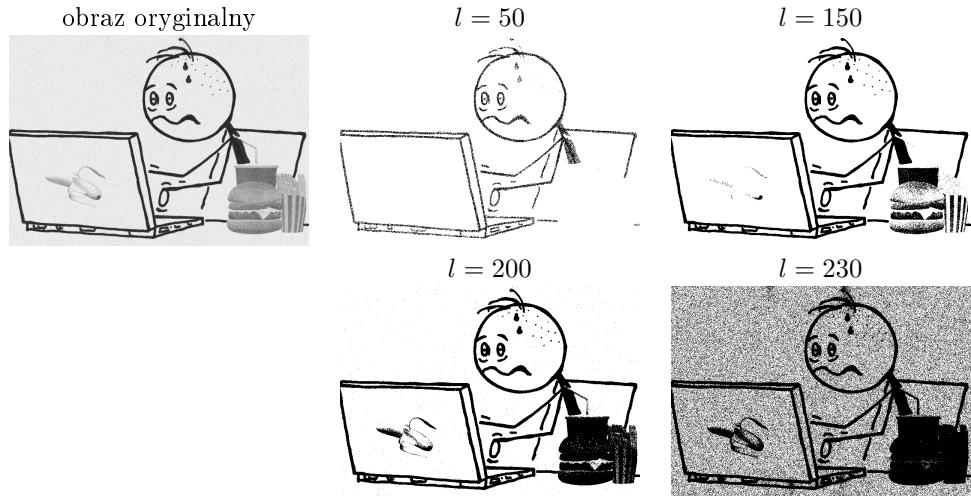
Grafika 6.7: Binaryzacja obrazu (wybór wartości progowej metodą Otsu)

Segmentacja objektu

→ grafika 6.8

Wybór wartości progowej wpływa nie tylko na jakość samej segmentacji, ale także na wielkość segmentowanych obszarów. Powodem wzrostu obszaru są piksele w obszarze krawędzi, które stopniowo pokrywają wartości szarości od bieli do czerni.

W przykładzie na grafice 6.8 próg $l = 230$ przypisuje niektóre piksele tła jako należące do obiektu. To znak, że został wybrany za duży. W przypadku segmentacji z



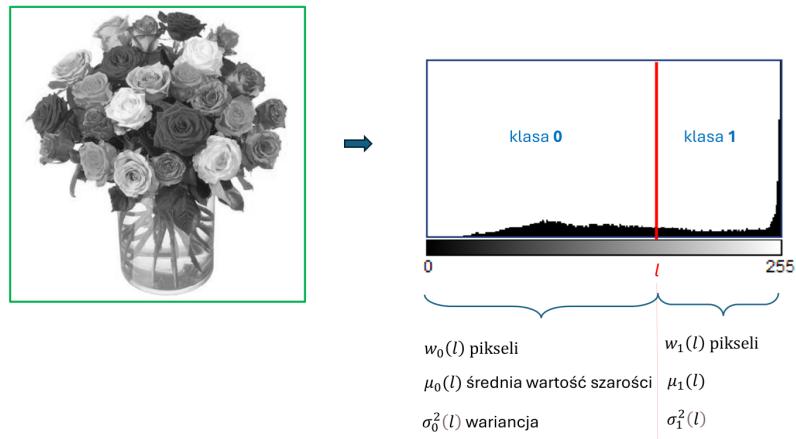
Grafika 6.8: Segmentacja obiektu.

wartością progową $l = 50$ niektóre piksele należące do obiektu zostały zabarwione na biało, czyli segmentowane w tło. Dlatego też wartość progowa została wybrana zbyt nisko.

Istnieją różne podejścia do wyboru optymalnej wartości progowej, z których najbardziej znaną spośród nietrywialnych jest metoda Otsu.

Wybór wartości progowej metodą Otsu

- Binaryzacja dzieli histogram obrazu na klasy 0 i 1 z wartością progową l
→ grafika 6.9

Grafika 6.9: Podział histogramu na klasy 0 i 1 wartością progową l .

- w klasie i jest $w_i(l)$ pikseli o średniej wartości szarości $\mu_i(l)$ i wariancji $\sigma_i^2(l)$

$$\begin{aligned}
 & \text{klasa 0} \\
 w_0(l) &= \sum_{i=0}^l h(i) \\
 \mu_0(l) &= \frac{1}{w_0(l)} \sum_{i=0}^l i h(i) \\
 \sigma_0^2(l) &= \frac{1}{w_0(l)} \sum_{i=0}^l (i - \mu_0(l))^2 h(i) \\
 \\
 & \text{klasa 1} \\
 w_1(l) &= \sum_{i=l+1}^{255} h(i) \\
 \mu_1(l) &= \frac{1}{w_1(l)} \sum_{i=l+1}^{255} i h(i) \\
 \sigma_1^2(l) &= \frac{1}{w_1(l)} \sum_{i=l+1}^{255} (i - \mu_1(l))^2 h(i)
 \end{aligned}$$

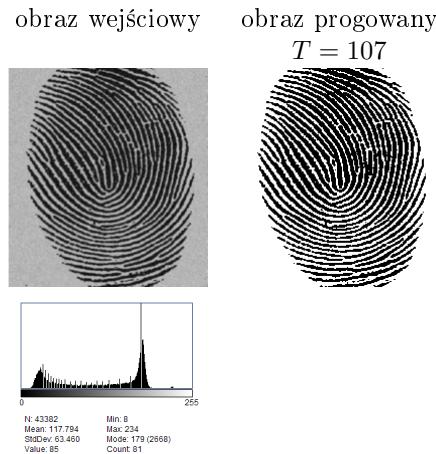
- Metoda Otsu oblicza próg l , który minimalizuje wariancję wewnętrzklasową $\sigma_w^2(l)$, zdefiniowaną jako średnia ważona wariancji każdej klasy

$$\sigma_w^2(l) = w_0(l)\sigma_0^2(l) + w_1(l)\sigma_1^2(l)$$

- wariancja wewnętrzklasowa $\sigma_w^2(l)$ obliczona zostaje dla wszystkich progów $l \in \{0, 1, \dots, 255\}$ i ten, który daje najniższą wartość dla $\sigma_w^2(l)$ jest zwracany:

$$T := \{k : \sigma_w^2(k) = \min, k \in \{0, 1, \dots, 255\}\}$$

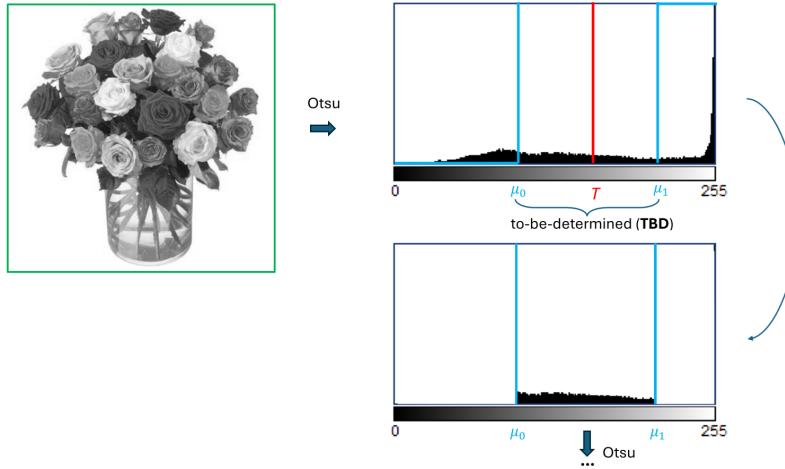
Przykład: grafika 6.10



Grafika 6.10: Przykład zastosowania metody Otsu do wyznaczania wartości progowej.

Iteracyjne progowanie trójklastowe w oparciu o metodę Otsu

→ grafika 6.11



Grafika 6.11: Iteracyjne progowanie trójklastowe w oparciu o metodę Otsu

Algorytm obliczania progu T metodą progowania trójklastowego

Dla obrazu $g(m, n)$

1. wyznaczenie

- wartości progowej $T^{[0]}$ metodą Otsu
- średniej wartości szarości $\mu_i^{[0]}, i = 0, 1$ klas 0 i 1

2. iteracja:

dla regionu $[\mu_0^{[k]}, \mu_1^{[k]}], k \geq 0$ (region "to-be-determined", TBD)
wyznaczenie

- wartości progowej $T^{[k+1]}$ metodą Otsu
- średniej wartości szarości $\mu_i^{[k+1]}, i = 0, 1$ klas 0 i 1

do osiągnięcia dostatecznie małej różnicy Δ

$$\Delta = T^{[k+1]} - T^{[k]}$$

3. $T := T^{[k+1]}$

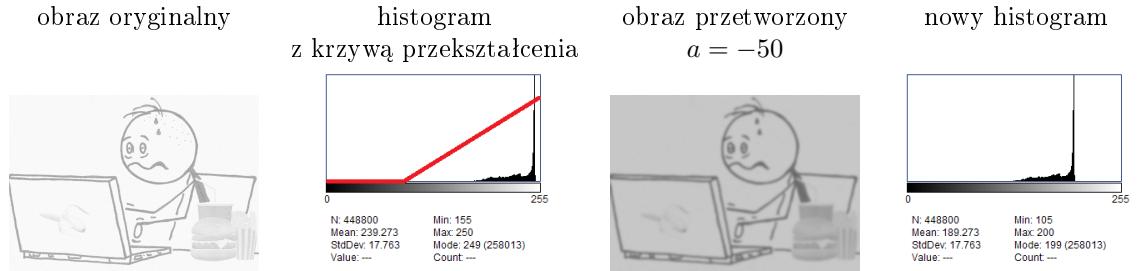
6.4.2 Przesunięcie histogramu

→ grafika 6.12

Regulacja jasności obrazu: wszystkie szare wartości g obrazu są przesuwane do jasnego lub ciemnego obszaru o stałą stałą a .

$$f_2(m, n) = \begin{cases} 0 : & f_1(m, n) + a < 0 \\ f_1(m, n) + a : & 0 \leq f_1(m, n) + a \leq G - 1 \\ G - 1 : & f_1(m, n) + a > G - 1 \end{cases} \quad \left| \quad T_{\text{trans}}(g) = \begin{cases} 0 : & g + a < 0 \\ g + a : & 0 \leq g + a \leq G - 1 \\ G - 1 : & g + a > G - 1 \end{cases} \right.$$

Przesunięcie histogramu zwykle nie jest odwracalne.



Grafika 6.12: Przesunięcie histogramu.

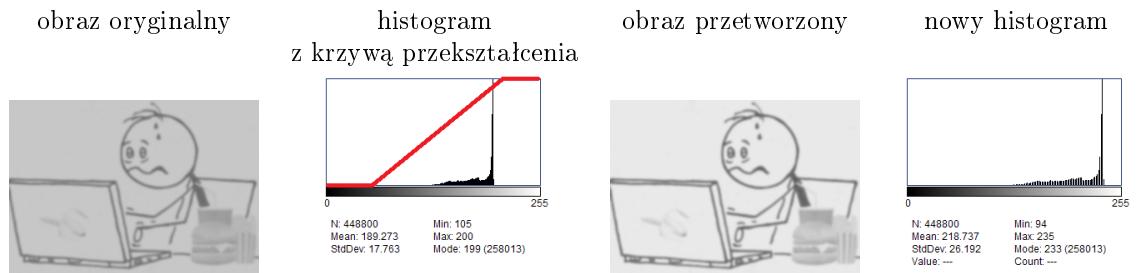
6.4.3 Rozproszenie histogramu

→ grafika 6.13

Skalowanie zakresu kontrastu: właściwości rozkładu jasności zostają przeniesione na większą skalę. Metoda jest często stosowana w celu zwiększenia kontrastu w obrazach w skali szarości o niskim kontraste.

$$T_{\text{stretch}}(g) = \left[(G - 1) \cdot \frac{g - g_{\min}}{g_{\max} - g_{\min}} \right]$$

$$T : \{g_{\min}, \dots, g_{\max}\} \rightarrow \{0, \dots, G - 1\}$$



Grafika 6.13: Rozproszenie histogramu.

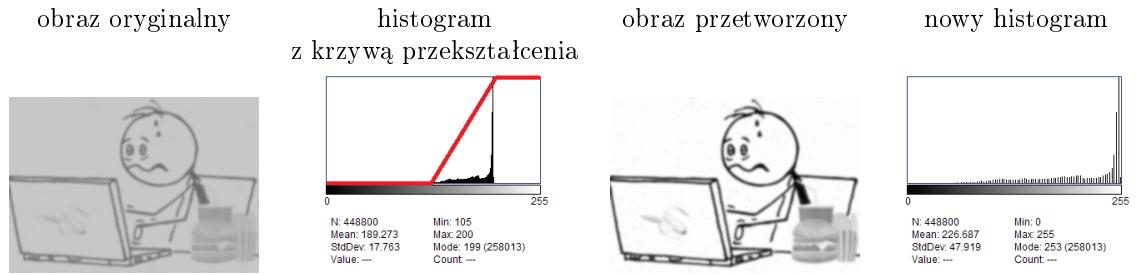
6.4.4 Ograniczenie histogramu, clipping

→ grafika 6.14

Eliminacja niedoświetlenia, prześwietlenia lub słabego kontrastu: jeśli pożądany jest różnicowany widok określonego zakresu wartości szarości, można zastosować ograniczenie histogramu. Szare wartości poniżej i powyżej tego obszaru ($\{0, \dots, g_{\min}\}$ i $\{g_{\max}, \dots, G - 1\}$) są „odecinane” poprzez mapowanie ich odpowiednio na 0 (czarny) i

$G - 1$ (biały). Pozostałe wartości szarości są następnie wzmacniane za pomocą odpowiedniej metody, zazwyczaj poprzez rozproszenie histogramu.

$$T_{\text{clip}}(g) = \begin{cases} 0 : & 0 \leq g < g_{\min} \\ T_{\text{stretch}}(g) : & g_{\min} \leq g \leq g_{\max} \\ G - 1 : & g_{\max} < g \leq G - 1 \end{cases}$$



Grafika 6.14: Ograniczenie histogramu.

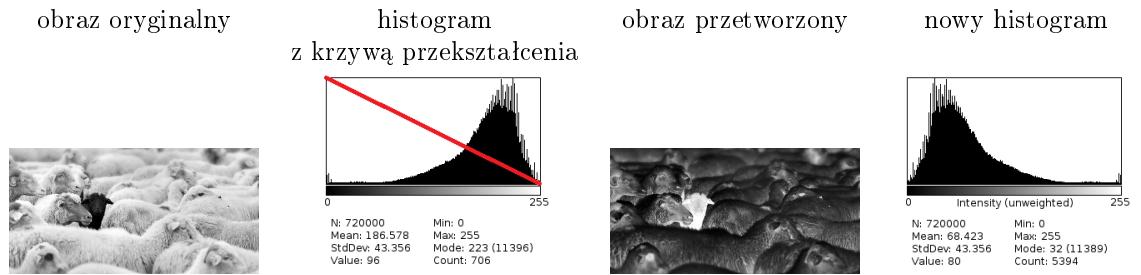
6.4.5 Inwersja histogramu

→ grafika 6.15

Ponieważ ludzki układ wzrokowy dostrzega w różnym stopniu drobne różnice między wartościami szarości w różnych zakresach jasności, negatywna transformacja może prowadzić do lepszego postrzegania drobnych struktur. Powstały obraz jest negatywem.

$$T_{\text{neg}}(g) = [(G - 1) - g]$$

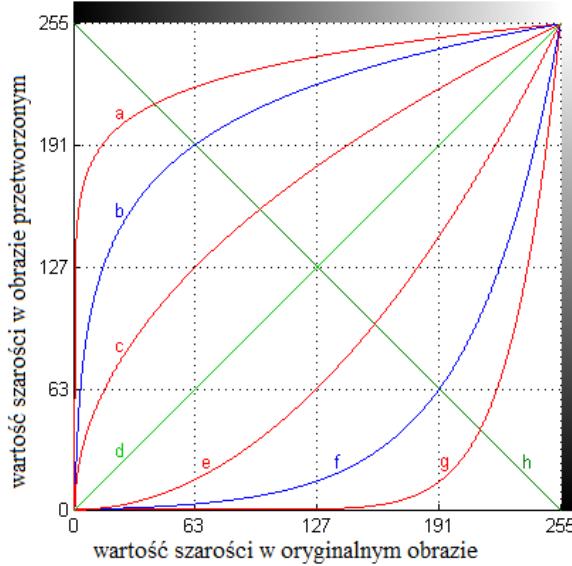
$$T : \{0, \dots, G - 1\} \rightarrow \{0, \dots, G - 1\}$$



Grafika 6.15: Inwersja histogramu.

6.4.6 Inne transformacje histogramu

→ grafika 6.16



Grafika 6.16: Transformacje histogramu. Źródło: de.wikipedia.org/wiki/Punktoperator

a. Korekcja gamma (transformacja potęgowa), $\gamma = 10$

Transformacja monotoniczna oparta na potęgowaniu. Dzięki tej metodzie można zmienić jasność obrazu, przeprowadzając nieliniowe rozproszenie części wartości szarości, podczas gdy pozostała część jest kompresowana nieliniowo.

$$T_{\text{Power}}(g) = \left[(G - 1) \cdot \left(\frac{g}{G - 1} \right)^{\gamma} \right]$$

$$T : \{0, \dots, G - 1\} \rightarrow \{0, \dots, G - 1\}$$

Transformacja gamma daje możliwość manipulowania wartością szarości lub charakterystyką kolorów urządzeń technicznych (np. aparatu cyfrowego lub monitora). Jest to konieczne, aby dostosować do siebie zapis lub odwzorowanie wartości szarości lub kolorów poszczególnych urządzeń. W ten sposób można także dostosować profil jasności monitora do logarytmicznej percepcji jasności przez ludzki układ wzrokowy.

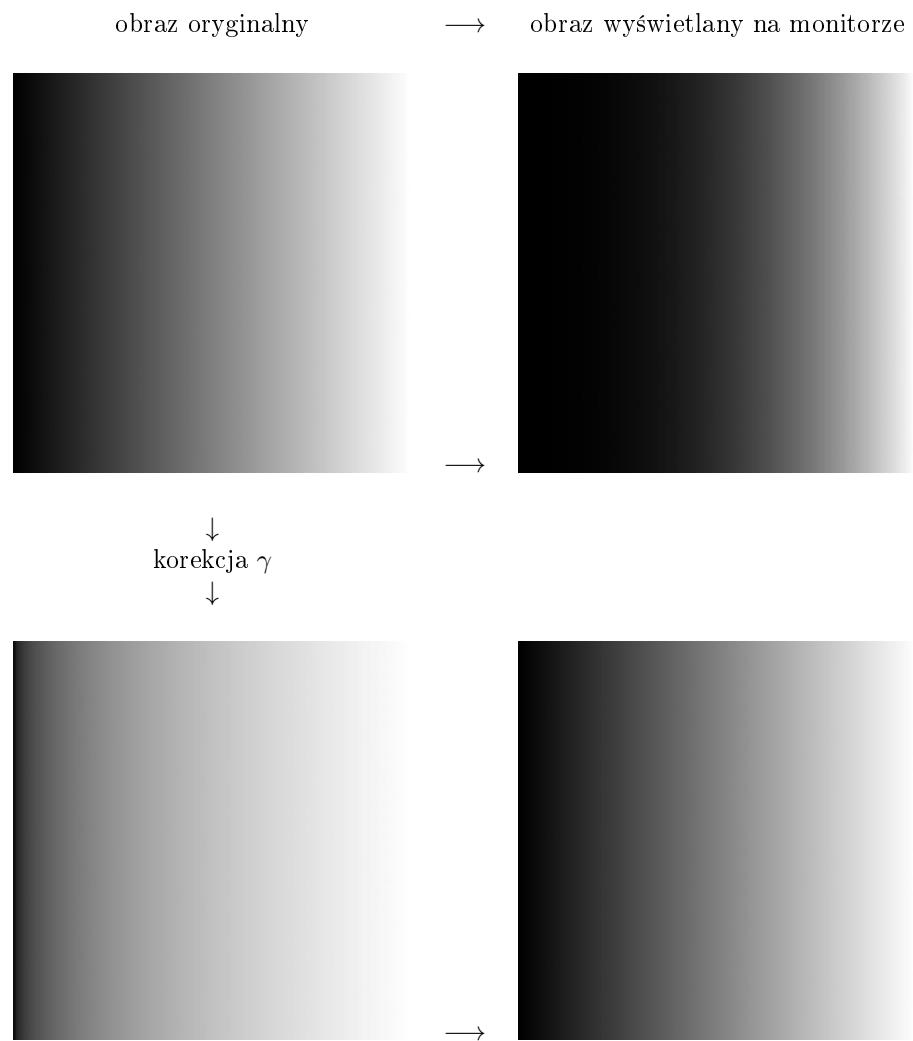
→ grafika 6.17

b. Transformacja logarytmiczna

Transformacja monotoniczna oparta na logarytmie. Odwzorowuje mały zakres wartości szarości w dolnej części skali w obrazie wejściowym na większy zakres wartości szarości w obrazie wynikowym, podczas gdy górne wartości szarości skali są kompresowane. Tym samym przetworzony obraz jest rozjaśniony.

$$T_{\log}(g) = \left[(G - 1) \cdot \frac{\log_2(g + 1)}{\log_2 G} \right] = [(G - 1) \cdot \log_G(g + 1)]$$

$$T : \{0, \dots, G - 1\} \rightarrow \{0, \dots, G - 1\}$$



Grafika 6.17: Korekcja gamma na ekranie komputera. Źródło: R. Gonzalez, Z. Faisal (2019). Digital Image Processing Second Edition

- c. Korekcja gamma, $\gamma = 2$
- d. Tożsamość
- e. Korekcja gamma, $\gamma = 0.5$
- f. Transformacja wykładnicza

Transformacja monotoniczna oparta na funkcji wykładniczej. Odwzorowuje mały zakres wartości szarości w górnej części skali w obrazie wejściowym na większy zakres wartości szarości w obrazie wynikowym, podczas gdy dolne wartości szarości skali są kompresowane. To ogólnie przyciemnia obraz.

$$T_{\text{exp}}(g) = T_{\text{exp}}^{-1}(g) = \left[G^{\frac{g}{G-1}} - 1 \right]$$

$$T : \{0, \dots, G-1\} \rightarrow \{0, \dots, G-1\}$$

- g. Korekcja gamma, $\gamma = 0.1$
- h. Inwersja histogramu

6.5 Znormalizowany histogram

Znormalizowany histogram $H_n(g)$ jest relatywną częstością wartości szarości g w obrazie:

- Normalizacja w zależności od liczby pikseli w obrazie (rozmiar $M \times N$)

$$H_n(g) = \frac{h(g)}{M \cdot N}$$

→ empiryczny rozkład prawdopodobieństwa wartości szarości g

- Zachodzi zależność:

$$\sum_{g=0}^{255} H_n(g) = 1$$

Średnia wartość szarości w obrazie

$$g_{\text{Mean}} = \sum_{g=0}^{255} g H_n(g)$$

Średnie odchylenie wartości szarości w obrazie

$$q_{\text{Mean}} = \sum_{g=0}^{255} (g - g_{\text{Mean}})^2 H_n(g)$$

6.5.1 Centralne momenty obrazu

→ średnie ważone (momenty) intensywności (odcieni szarości) g pikseli obrazu

$$M_n = \sum_{g=0}^{255} (g - g_{\text{Mean}})^n H_n(g), n > 1$$

- $M_1 = g_{\text{Mean}}$: średnia wartość intentywności
- $M_2 = q_{\text{Mean}}$: średnie odchylenie intensywności
- M_3 : skośność *Skewness*
Skośność wskazuje stopień asymetrii (określa odchylenie rozkładu histogramu od postaci normalnej Gaussa w prawo i w lewo).
- M_4 : kurtoza *Excess kurtosis*
Kurtoza wskazuje, jak bardzo rozkład odbiega w góre lub w dół od rozkładu normalnego Gaussa.

6.6 Skumulowany histogram, wyrównanie i hiperbolizacja histogramu

Cel: → grafika 6.18

Wyrównanie histogramu

→ grafika 6.20

Metoda poprawy kontrastu obrazów w skali szarości wykraczająca poza zwykłe zwiększenie kontrastu: z rozkładu wartości szarości na histogramie obliczany jest rozkład równomierny, dzięki czemu wykorzystywany jest optymalnie cały dostępny zakres wartości szarości.

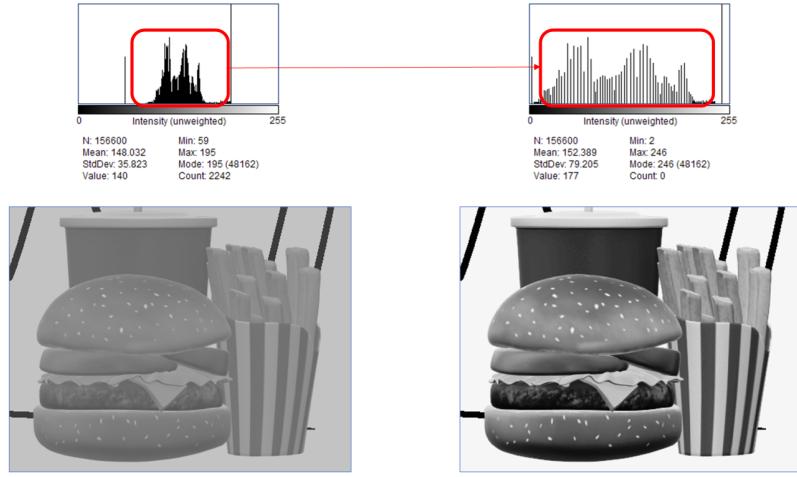
Metodę tę stosuje się szczególnie w przypadkach, gdy interesujące obszary obrazu stanowią stosunkowo dużą część obrazu (tzn. odpowiadające im wartości szarości $h(g)$ w histogramie występują często), ale są ograniczone do małego zakresu wartości szarości g .

Do określenia charakterystyki transformacji wykorzystuje się **histogram H_S skumulowanej wartości szarości obrazu**: każdej wartości szarości g zostaje przypisana suma wszystkich relatywnych częstości szarości $H_n(i)$ dla wartości szarości $i \in \{0, \dots, g\}$ w obrazie :

$$\begin{aligned} H_S(g) &= \sum_{i=0}^g H_n(i) \\ &= \frac{1}{M \cdot N} \sum_{i=0}^g h(i) \end{aligned}$$

Skumulowany histogram $H_S(g)$ jest ciągiem wartości w przedziale $[0, 1]$. Po wymnożeniu każdego elementu ciągu przez $G - 1$ (gdzie G to ilość możliwych wartości szarości), a następnie zaokrągleniu, otrzymuje się charakterystyczną krzywą transformacji z zakresem wartości $\{0, \dots, G - 1\}$:

$$H_{\text{equal}}(g) = [(G - 1) \cdot H_S(g)].$$

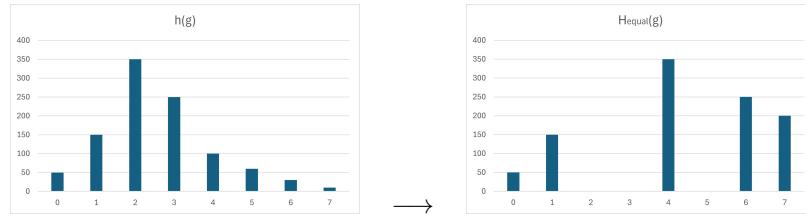


Grafika 6.18: Wyrównanie histogramu.

Przykład $M \cdot N = 1000$ pikseli $G = 8$ odcieni szarości $\rightarrow g \in \{0, \dots, 7\}$

g	0	1	2	3	4	5	6	7
$h(g)$	50	150	350	250	100	60	30	10
$\sum_{i=0}^g h(i)$	50	200	550	800	900	960	990	1000
$H_S(g)$	0.05	0.2	0.55	0.8	0.9	0.96	0.99	1
$H_{\text{equal}}(g)$	0	1	4	6	7	7	7	7

→ grafika 6.19



Grafika 6.19: Wyrównanie histogramu

Hiperbolizacja histogramu

→ grafika 6.20

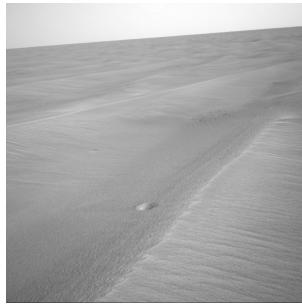
Hiperbolizacja histogramu to dostosowywanie do zmysłu widzenia człowieka: po wy-

równaniu histogramu wartości szarości na powstałym obrazie są równomiernie rozłożone, ale dla człowieka często wydają się być zbyt jasne, ponieważ postrzeganie jasności przez nasz system wzrokowy nie jest liniowe, ale logarytmiczne.

$$H_{\text{hyper}}(g) = \left[(G - 1) \cdot H_S^{\frac{1}{\alpha+1}}(g) \right], \text{ gdzie } \alpha \in] -1, 0].$$

Charakterystyka transformacji wyrównania przesunięta jest nieco w stronę rozkładu hiperbolicznego. Wartości ciemnoszare mają większe prawdopodobieństwo niż jasne, co oznacza, że obraz jest przyciemniony. Wartości przyjmowane dla α to z reguły $\alpha \in [-\frac{2}{3}, -\frac{1}{3}]$, dla $\alpha = 0$ hiperbolizacja odpowiada wyrównaniu histogramu.

obraz oryginalny



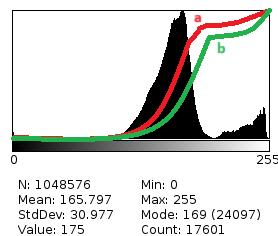
wyrównanie



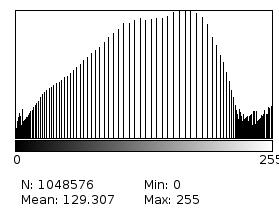
hiperbolizacja



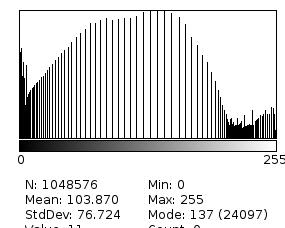
- histogram
- i krzywe przekształcenia
 - a. wyrównania
 - b. hiperbolizacji



histogram
po transformacji a



histogram
po transformacji b
 $\alpha = -\frac{1}{3}$



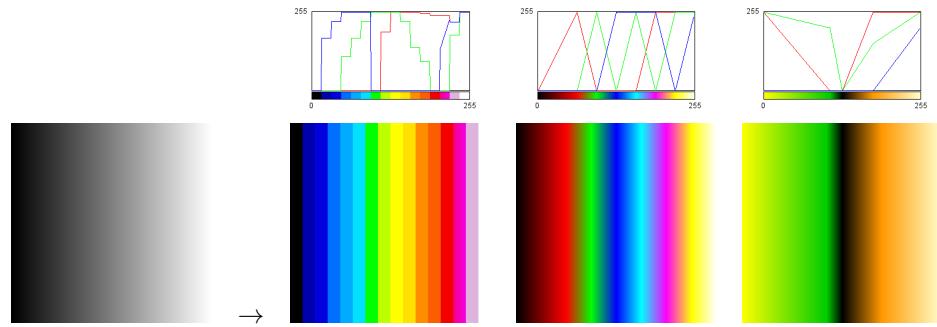
Grafika 6.20: Wyrównanie i hiperbolizacja histogramu. Zdjęcie przedstawia mały krater na Marsie (zdjęcie wykonane przez łazik Opportunity).

6.7 Kolor w celu zwiększenia kontrastu

→ grafika 6.21

- Człowiek wyróżnia znacznie więcej wartości kolorów niż wartości szarości.

- Zwiększenie kontrastu poprzez trzy nieliniorwe, niemonotoniczne funkcje mapowania wartości szarości: *czerwony(g)*, *zielony(g)*, *niebieski(g)*



Grafika 6.21: Przykłady funkcji mapowania wartości szarości poprzez kolory.

Przykład: zwiększenie kontrastu w mapach

→ grafika 6.22



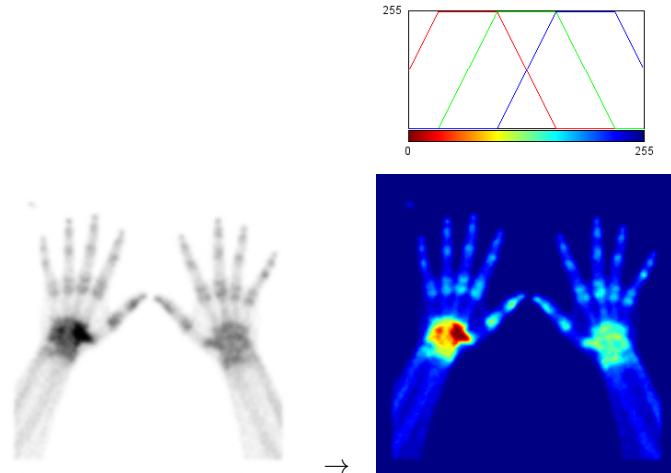
Grafika 6.22: Użycie koloru w celu zwiększenia kontrastu. Źródło: maps-of-europe

Przykład: obrazowanie medyczne

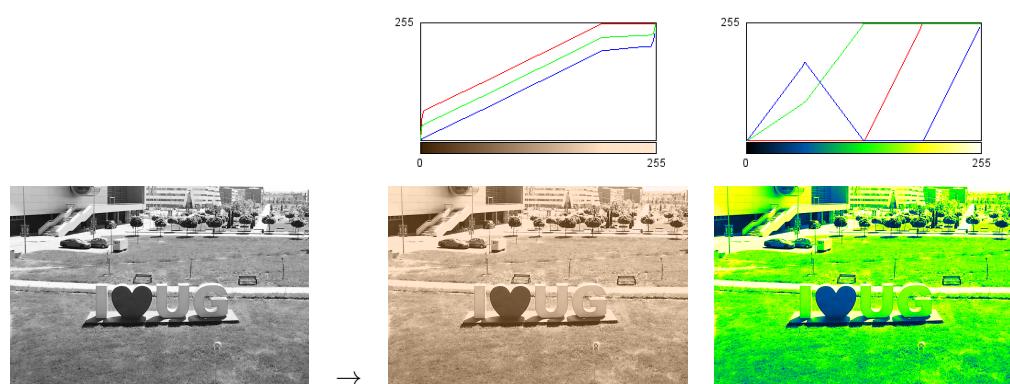
Wizualizacja obrazów w medycynie → grafika 6.23

Przykład: efekty artystyczne

→ grafika 6.24



Grafika 6.23: Colour lookup table (CLUT) w obrazowaniu medycznym (przykładowe mapowanie kolorów)



Grafika 6.24: Mapowanie kolorów

7

Inne operatory punktowe

Operacje arytmetyczne i logiczne na obrazach

Obrazy są macierzami, które można wykorzystać do obliczeń:

- nie wszystkie operacje macierzy są przydatne w przetwarzaniu obrazu, np. „mnożenie macierzy obrazów” nie ma sensu,
- możliwe są jednak ogólne operacje na obrazach punkt po punkcie,
- dwa lub więcej obrazów wejściowych jest połączonych z obrazem wyjściowym poprzez funkcję f
 - warunkiem koniecznym jest, aby wszystkie obrazy miały ten sam rozmiar
 - ImageJ → "Math", "Image Calculator"

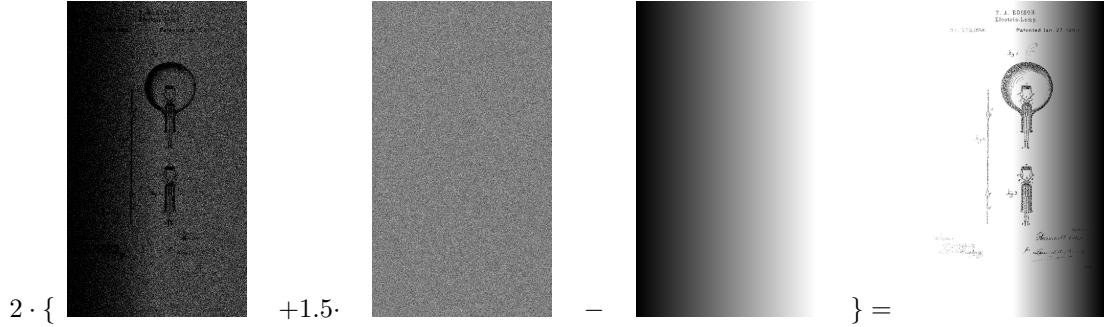
Arytmetyczne i logiczne operatory punktowe umożliwiają „obliczanie” za pomocą obrazów poprzez połączenie dwóch lub więcej obrazów wejściowych w jeden obraz wyjściowy. Operacje arytmetyczne i logiczne na obrazach to proste, ale potężne narzędzia umożliwiające

- ↪ łączenie części obrazów,
- ↪ ekstrakcję obiektów,
- ↪ segmentację obiektów,
- ↪ redukcję szumów,
- ↪ wykrywanie zmian na obrazach,
- ↪ ...

7.1 Arytmetyka obrazu

→ grafika 7.1

- $f^*(m, n) = f(m, n) + g(m, n)$



Grafika 7.1: Arytmetyka obrazów

- $f^*(m, n) = f(m, n) - g(m, n)$
- $f^*(m, n) = f(m, n) \cdot g(m, n)$
- $f^*(m, n) = \frac{f(m, n)}{g(m, n)}$

7.1.1 Dodawanie obrazów

$$f^*(m, n) = f(m, n) + g(m, n)$$

- Nakładanie obrazów: wzmacnianie nakładających się obszarów → grafika 7.2

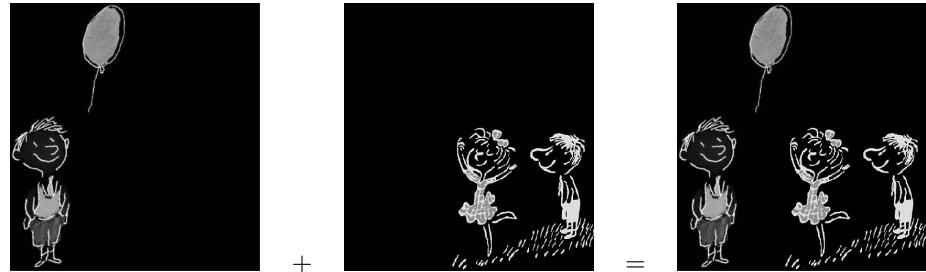


Grafika 7.2: Nakładanie obrazów

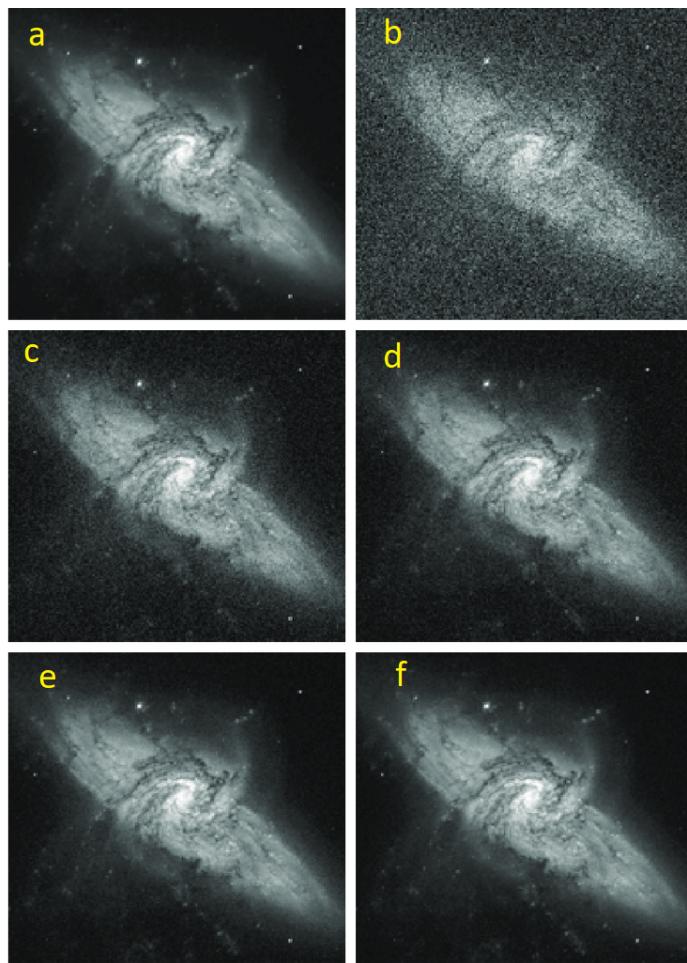
- Łączenie segmentowanych obiektów → grafika 7.3
Działa dobrze w przypadku obrazów z 0-wym tłem i nienakładającymi się obiek-tami.
- Uśrednianie obrazów → grafika 7.4

$$f^*(x, y) = \frac{1}{n} \sum_{i=1}^n f_i(x, y)$$

Z obrazów wykonywanych w krótkich odstępach czasu eliminowane są przypad-kowe zakłócenia, takie jak szum.



Grafika 7.3: Łączenie segmentowanych obiektów



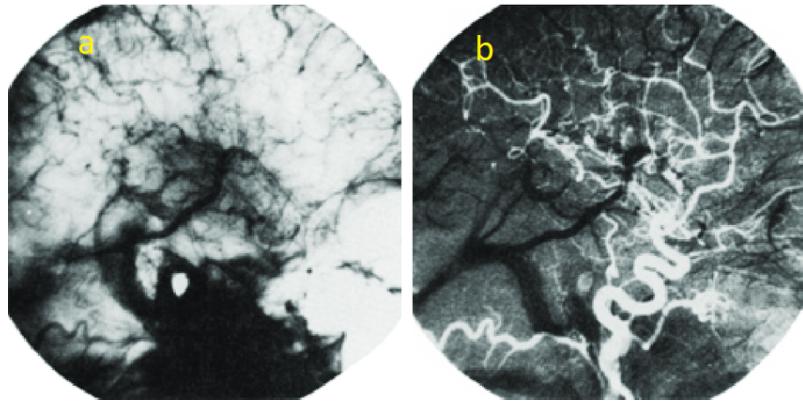
Grafika 7.4: Redukcja szumów poprzez uśrednianie wielu obrazów: (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)-(f) Results of averaging K=8, 16, 64, and 128 noisy images. (Original image courtesy of NASA.)

Źródło: R. Gonzalez, Z. Faisal (2019). Digital Image Processing Second Edition

7.1.2 Odejmowanie obrazów

$$f^*(m, n) = f(m, n) - g(m, n)$$

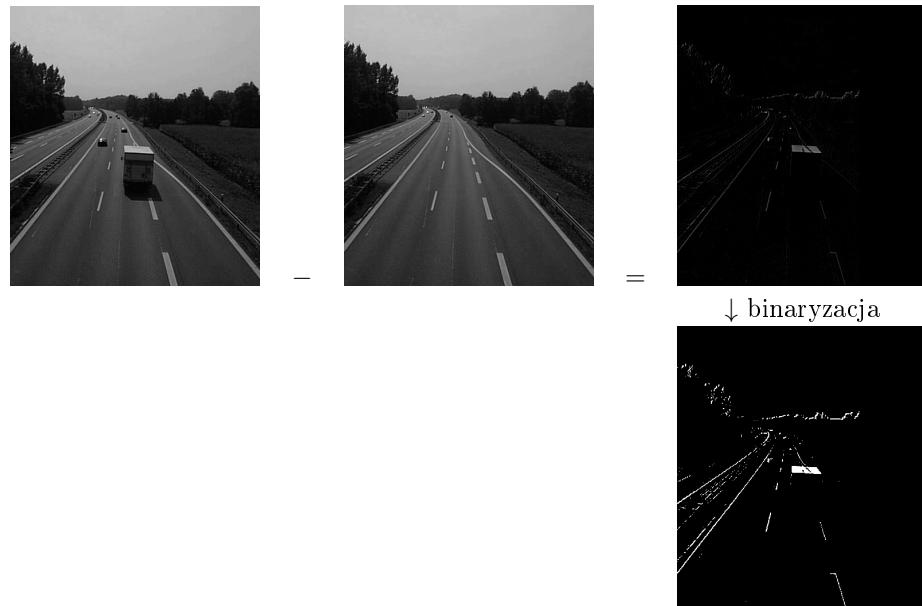
- Segmentacja obrazu → grafika 7.5



Grafika 7.5: Różnica między zdjęciami rentgenowskimi ze środkiem kontrastowym i bez niego: Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

Źródło: R. Gonzalez, Z. Faisal (2019). Digital Image Processing Second Edition

- Detekcja ruchu, detekcja zmian → grafika 7.6

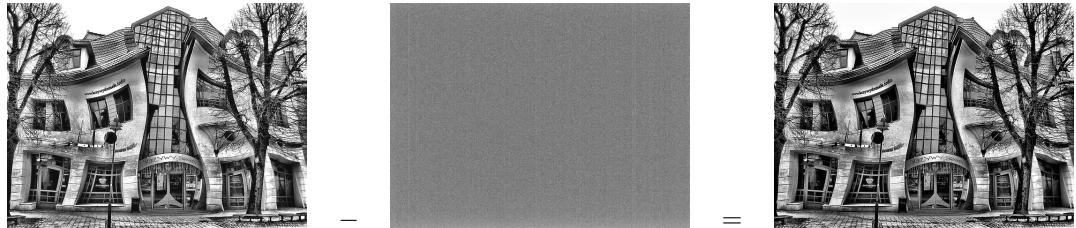


Grafika 7.6: Detekcja ruchu

Dark-frame subtraction

Dark-frame: obraz zarejestrowany sensorem w całkowitej ciemności \rightarrow obraz szumu wytwarzanego przez czujnik.

Korekta obrazu metodą Dark-frame subtraction: \rightarrow grafika 7.7



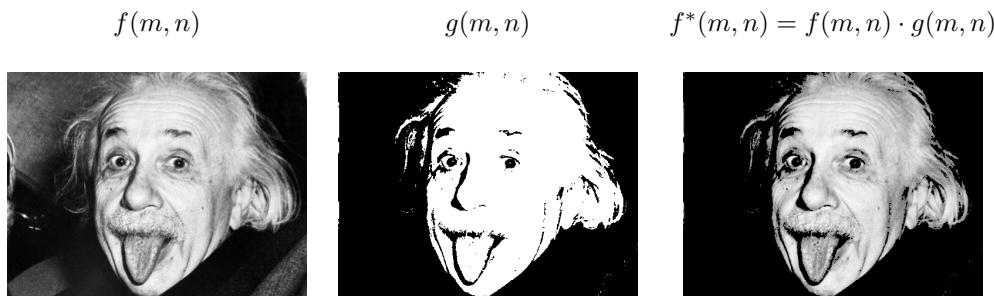
Grafika 7.7: Korekta obrazu metodą dark-frame subtraction (tu: Dark-Frame z podwyższonym kontrastem)

7.1.3 Mnożenie obrazów

$$f^*(m, n) = f(m, n) \cdot g(m, n)$$

W przeciwnieństwie do mnożenia macierzy, mnożenie między dwoma obrazami następuje punkt po punkcie.

- mnożenie obrazu przez maskę (obraz składający się z 0 i 1) pozwala na wycięcie części obrazu (maskowanie obszarów) \rightarrow grafika 7.8

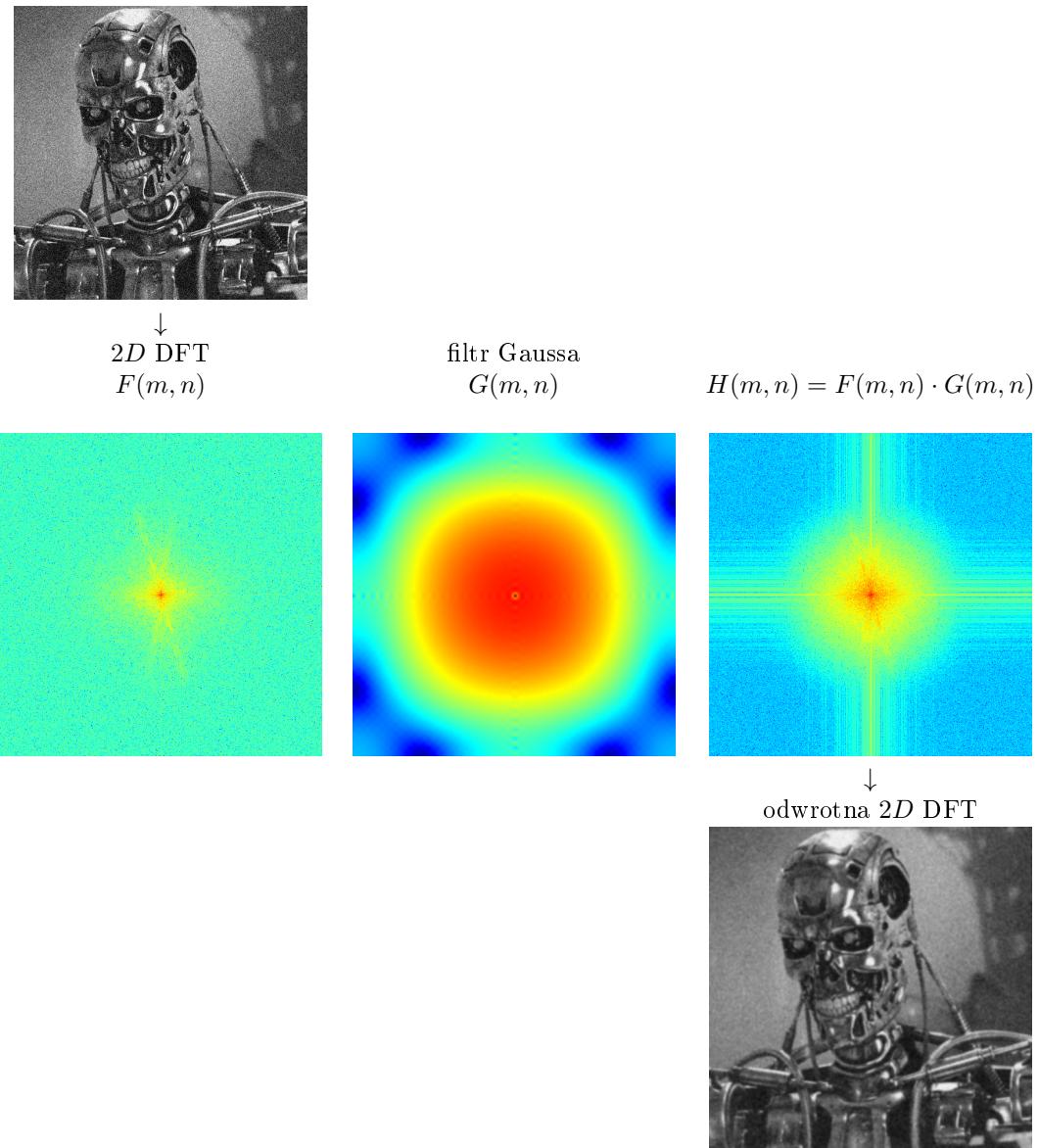


Grafika 7.8: Mnożenie obrazu przez maskę.

- Mnożenie (i dzielenie) obrazów wykorzystywane jest głównie do filtrowania w przestrzeni Fouriera \rightarrow grafika 7.9

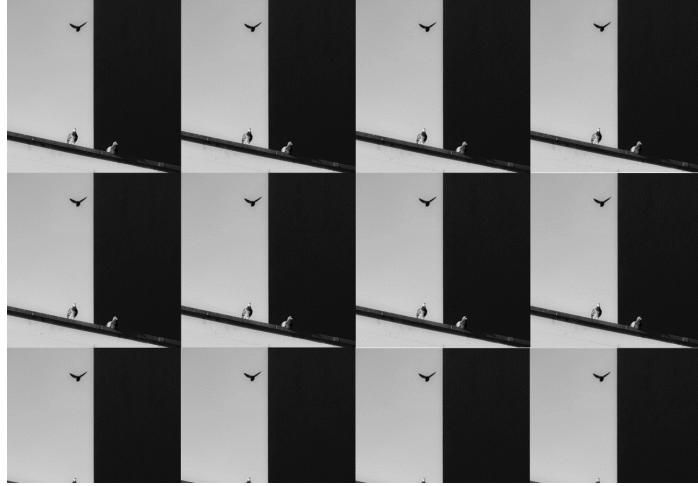
Okienkowanie obrazu

Aby obliczyć transformatę Fouriera obrazu, zakłada się, że jest on okresowy. Jeśli jednak lewa i prawa strona lub góra i dolna krawędź obrazu mają różną jasność lub strukturę, to okresowe powtarzanie prowadzi do nieciągłości na krawędziach obrazu. Są one widoczne w widmie jako duże gęstości widmowe wzdłuż osi. W przypadku



Grafika 7.9: Mnożenie obrazów w przestrzeni Fouriera (filtrowanie)

stosowania zaawansowanych operacji w przestrzeni Fouriera (np. splot z operatorem wygładzającym), te duże gęstości mogą znacząco przeszkadzać. (grafika 7.10)



Grafika 7.10: Obraz jako dwuwymiarowa funkcja okresowa

Zaburzającego efektu można uniknąć mnożąc obraz $g(m, n)$ przez odpowiednią **funkcję okienkową** $w(m, n)$, która przyjmuje wartości z przedziału $[0, 1]$ i maleje do 0 od środka w stronę krawędzi obrazu:

$$f^*(m, n) = w(m, n) \cdot g(m, n)$$

Mnożenie obrazu oznacza splot transformat Fouriera $W(k, l) * G(k, l)$ (rozdział "Filtры линиевые").

Przykład: okrągłe, symetryczne okno (prostokątne) o środku w punkcie (m_0, n_0) i promieniu r

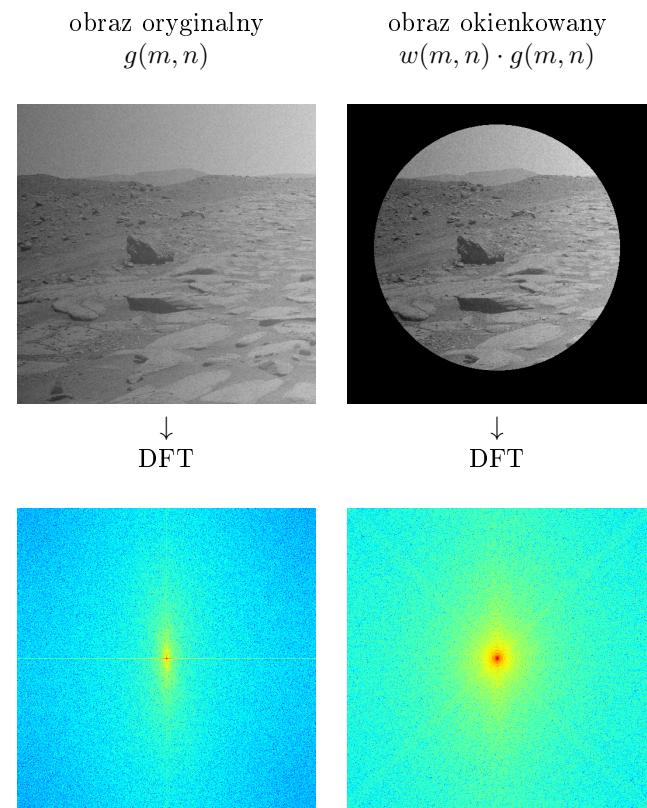
$$w(m, n) = \begin{cases} 1 & : (m - m_0)^2 + (n - n_0)^2 \leq r^2 \\ 0 & (m - m_0)^2 + (n - n_0)^2 > r^2 \end{cases}$$

→ grafika 7.11

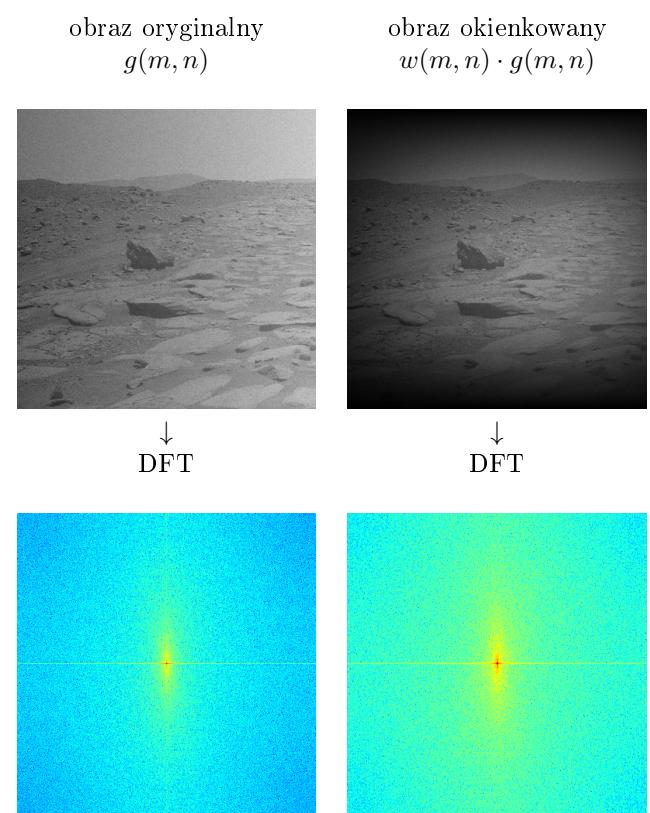
Przykład: okno sinusoidalne

$$w(m, n) = \sin\left(\pi \cdot \frac{m}{M}\right) \sin\left(\pi \cdot \frac{n}{N}\right)$$

→ grafika 7.12

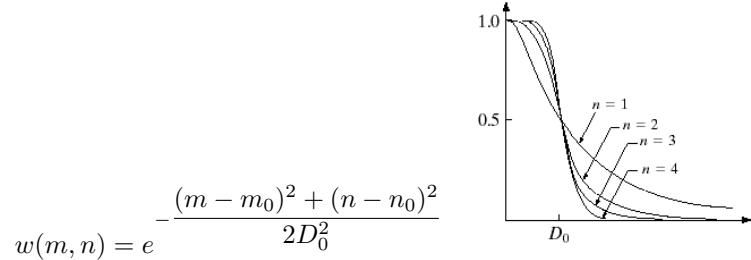


Grafika 7.11: Okienkowanie obrazu: okrągłe, symetryczne okno (prostokątne) $w(m, n)$

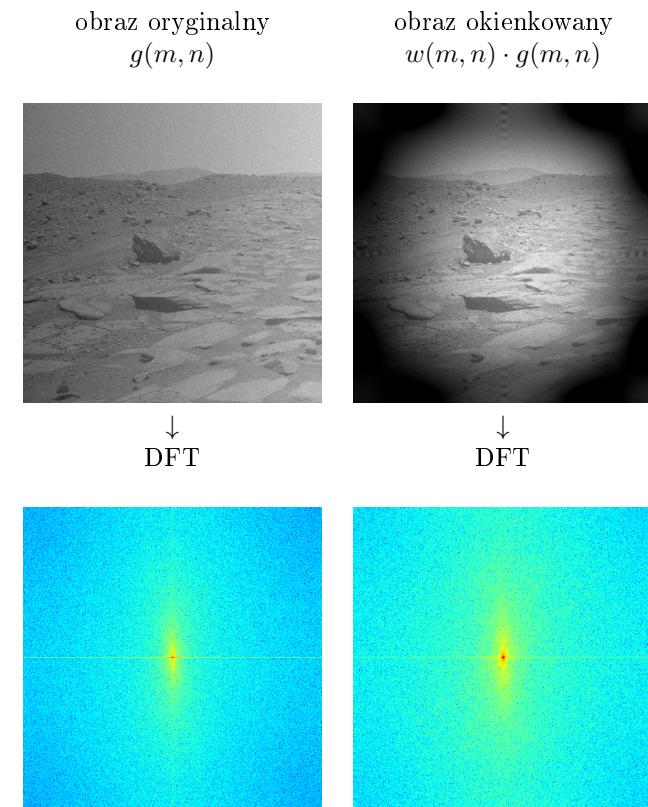


Grafika 7.12: Okienkowanie obrazu: okno sinusoidalne $w(m, n)$

Przykład: okno Gaussa o środku w punkcie (m_0, n_0) i parametrze D_0



→ grafika 7.13

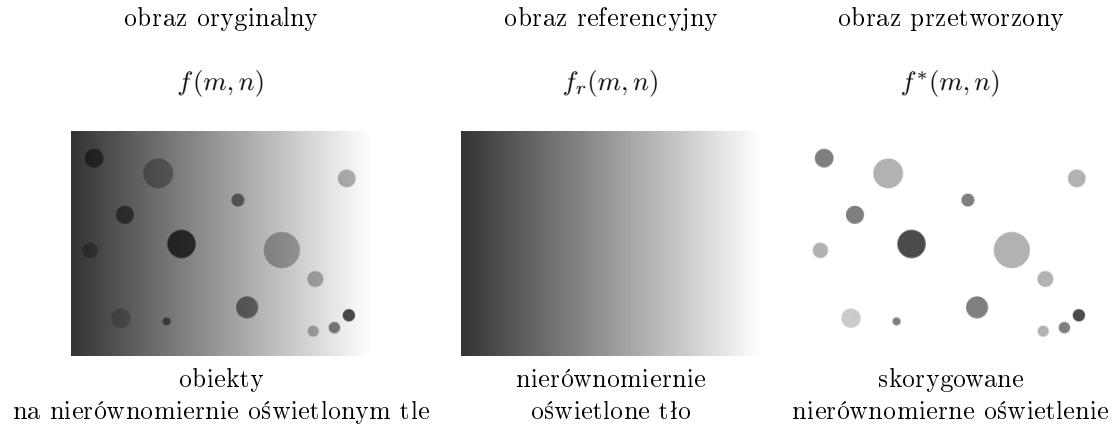


Grafika 7.13: Okienkowanie obrazu: okno Gaussa $w(m, n)$

7.1.4 Korekta niejednorodnego oświetlenia - dzielenie obrazów

$$f^*(m, n) = \frac{f(m, n)}{g(m, n)}$$

→ grafika 7.14



Grafika 7.14: Dzielenie obrazów. Źródło: de.wikipedia.org/wiki/Punktoperator

$$f^*(m, n) = T_{mn}(f(m, n)) = \left[(G - 1) \cdot \frac{f(m, n)}{f_r(m, n)} \right]$$

W idealnym przypadku obraz przetworzony ma białe tło o jednorodnej jasności.

Flat-field correction

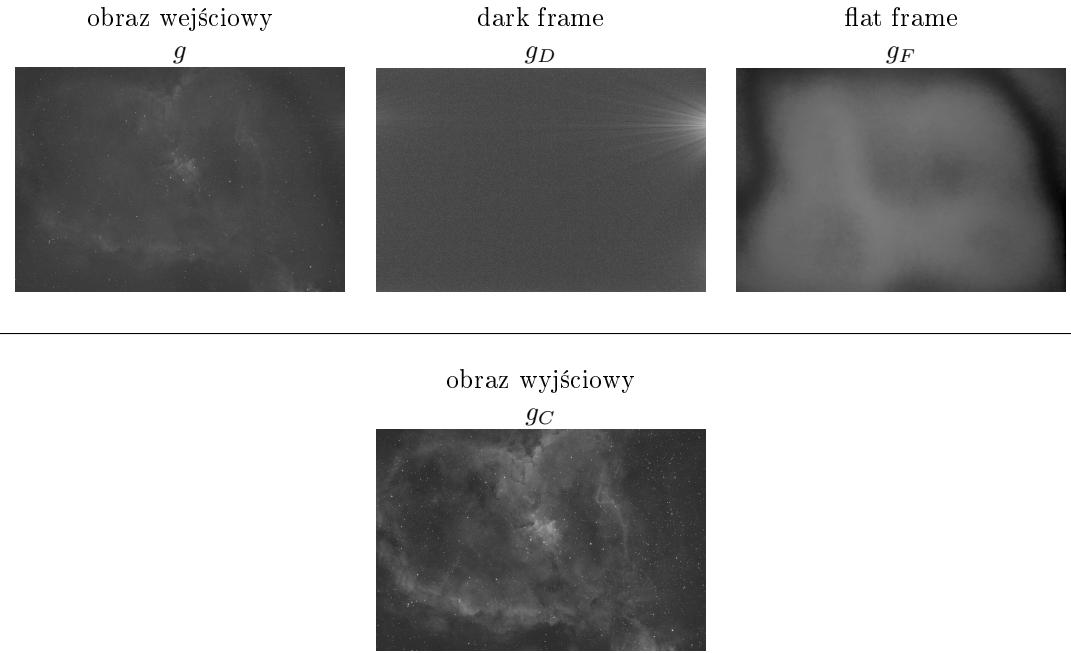
→ standardowa procedura kalibracji sensorów optycznych (od aparatów cyfrowych po duże teleskopy)

Flat-field: obraz równomiernie oświetlonej sceny (tła) → grafika 7.15



Grafika 7.15: Typowy kadr flat-field teleskopu. Źródło: <https://www.astroshop.eu/>

Przykład: kalibracja teleskopu → grafika 7.16



Grafika 7.16: Kalibracja teleskopu. Źródło: <https://www.macobservatory.com/>

Obliczenie skorygowanego obrazu g_C :

g_C obraz wyjściowy (skorygowany)

g obraz wejściowy

g_F flat frame

g_D dark frame

g_{Mean} średnia wartość jasności dla $(g_F - g_D)$

$$g_C = \frac{(g - g_D) \cdot g_{\text{Mean}}}{g_F - g_D}$$

→ grafika 7.17

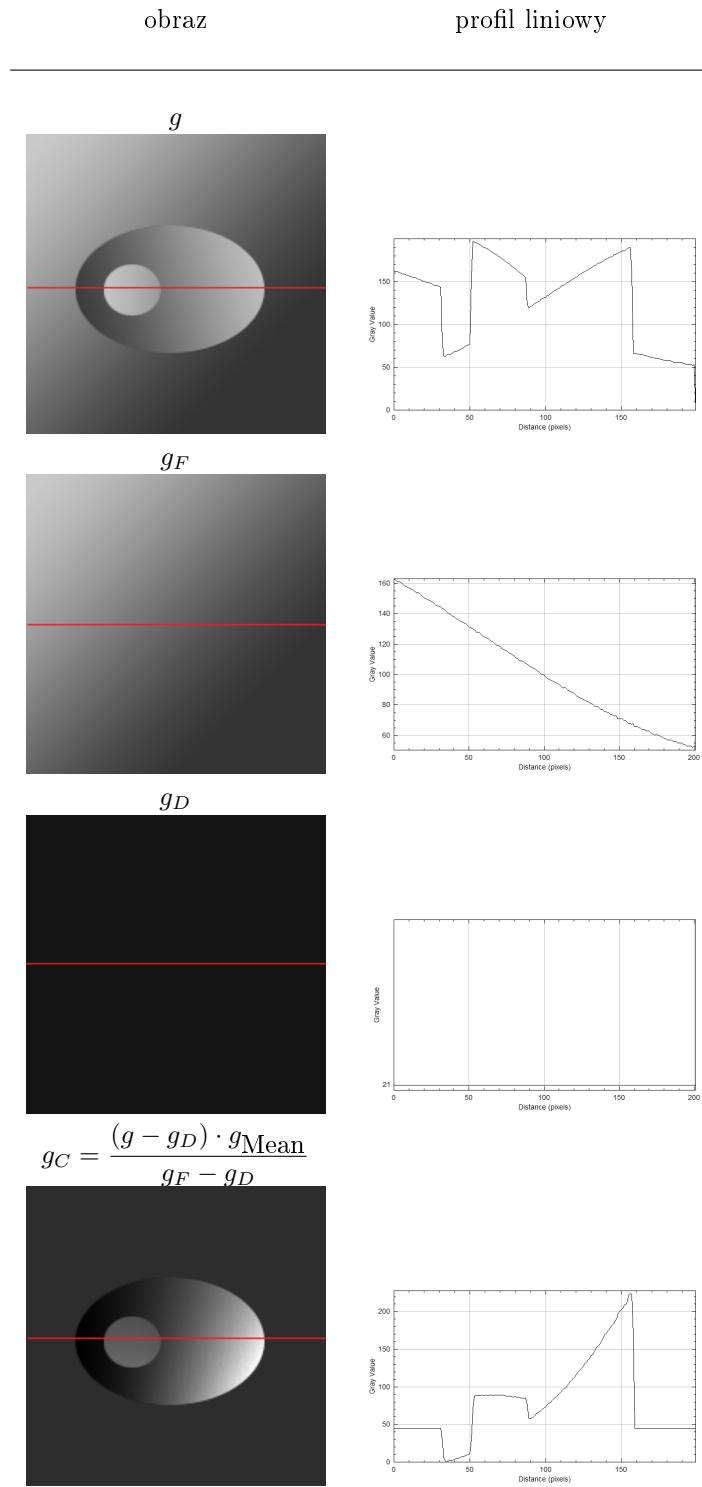
7.1.5 Ograniczenie zakresu wartości

Wartości pikseli nie są dowolnymi liczbami rzeczywistymi, lecz są

- nie negatywne,
- nie większe niż największa wartość szarości, jaką można wyświetlić,
- liczbami całkowitymi.

Dlatego często przy wykonywaniu operacji arytmetycznych na obrazie występuje

→ zaokrąglanie wyników niecałkowitych,



Grafika 7.17: Metoda Flat-field correction

- przycinanie zakresu w dół (clipping),
 - ↪ wyniki mniejsze niż 0 są ustawiane na 0
- przycinanie zakresu w górę (clipping),
 - ↪ wyniki powyżej maksymalnej wartości szarości są ustawiane na wartość maksymalną.
- głębia bitowa powstającego obrazu jest zwiększana, aby móc wyświetlać większe lub ujemne liczby.

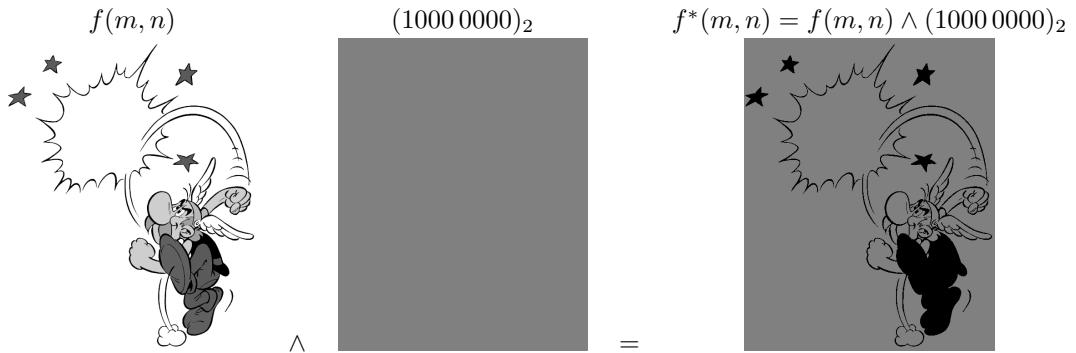
Każdy program przetwarzający obraz może działać inaczej!

7.2 Operacje logiczne na obrazach

Operatory logiczne AND, OR i NOT są funkcjonalnie kompletne, tzn. dowolny inny operator logiczny można zaimplementować, używając tylko tych trzech podstawowych operatorów:

- $f^*(m, n) = f(m, n) \vee g(m, n)$
- $f^*(m, n) = f(m, n) \wedge g(m, n)$
- $f^*(m, n) = \neg f(m, n)$

W przypadku operacji logicznych na obrazach w skali szarości wartości pikseli są przetwarzane jako ciągi liczb binarnych. → grafika 7.18



Grafika 7.18: Przykładowa operacja logiczna na obrazach

7.2.1 Operacja \vee

$$f^*(m, n) = f(m, n) \vee g(m, n)$$

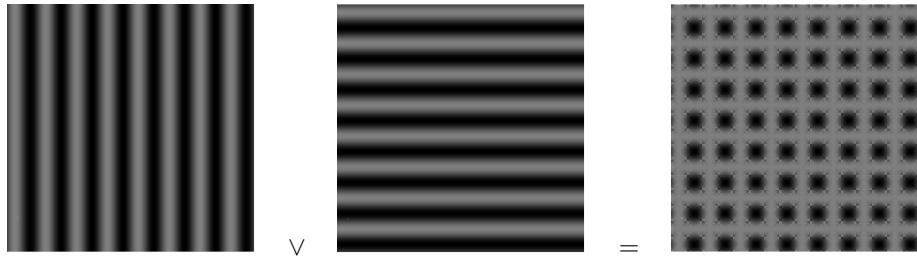
Operacja OR jest wykonywana na odpowiadających sobie pikselach interpretowanych jako ciągi liczb binarnych. Na każdym bicie wykonywana jest operacja \vee :

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Przykład: $(72)_{10} \vee (112)_{10}$

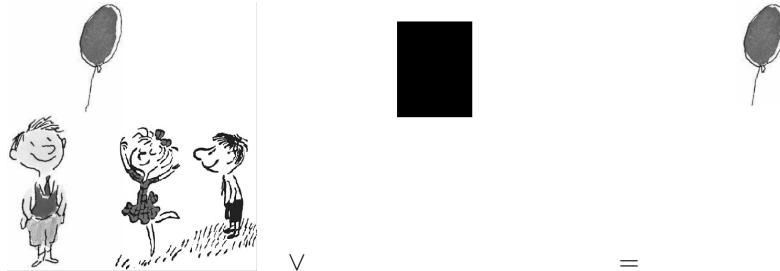
$$\begin{aligned}(72)_{10} \vee (112)_{10} &= (01001000)_2 \vee (01110000)_2 \\ &= (01111000)_2 \\ &= (120)_{10}\end{aligned}$$

- Łączenie obrazów bez „wzmocnienia” (por. dodawanie obrazów), \rightarrow grafika 7.19
Działa dobrze przy nakładających się obiektach!



Grafika 7.19: Łączenie obrazów bez ich wzmocnienia (por. grafika 7.2)

- Maskowanie obiektów \rightarrow grafika 7.20



Grafika 7.20: Maskowanie obiektów poprzez operację \vee

7.2.2 Operacja \wedge

$$f^*(m, n) = f(m, n) \wedge g(m, n)$$

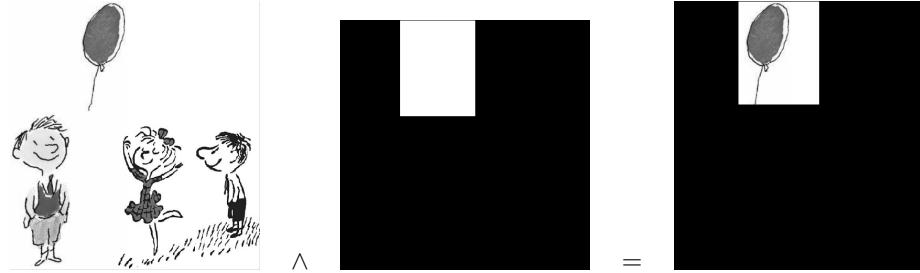
Operacja AND jest wykonywana na odpowiadających sobie pikselach interpretowanych jako ciągi liczb binarnych. Na każdym bicie wykonywana jest operacja \wedge :

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Przykład: $(72)_{10} \wedge (112)_{10}$

$$\begin{aligned}(72)_{10} \wedge (112)_{10} &= (01001000)_2 \wedge (01110000)_2 \\ &= (01000000)_2 \\ &= (64)_{10}\end{aligned}$$

- Maskowanie obiektów —> grafika 7.21



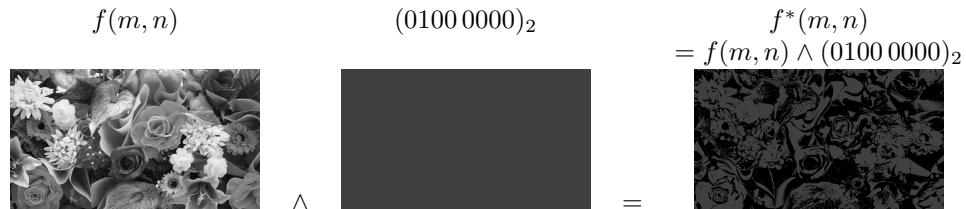
Grafika 7.21: Maskowanie obiektów poprzez operację \wedge

- Ekstrakcja płaszczyzn bitowych /steganografia

Obraz (w skali szarości, $G = 256$) można podzielić na osiem różnych poziomów (płaszczyzn) bitowych. Każda płaszczyzna bitowa obrazu otrzymywana jest poprzez operację \wedge z obrazem o wartości szarości każdego piksela równej $(2^n)_{10}$, $n = 0 \dots 7$. Na przykład czwartą płaszczyznę bitową otrzymuje się poprzez operację \wedge z obrazem o wartości szarości $(00001000)_2$, piątą poprzez operację \wedge z obrazem $(00010000)_2$ itd. —> grafika 7.22

Przykład: czwarta płaszczyzna bitowa dla wartości $(72)_{10}$

$$\begin{aligned} (72)_{10} \wedge (16)_{10} &= (01001000)_2 \wedge (00001000)_2 \\ &= (00001000)_2 \\ &= (8)_{10} \end{aligned}$$

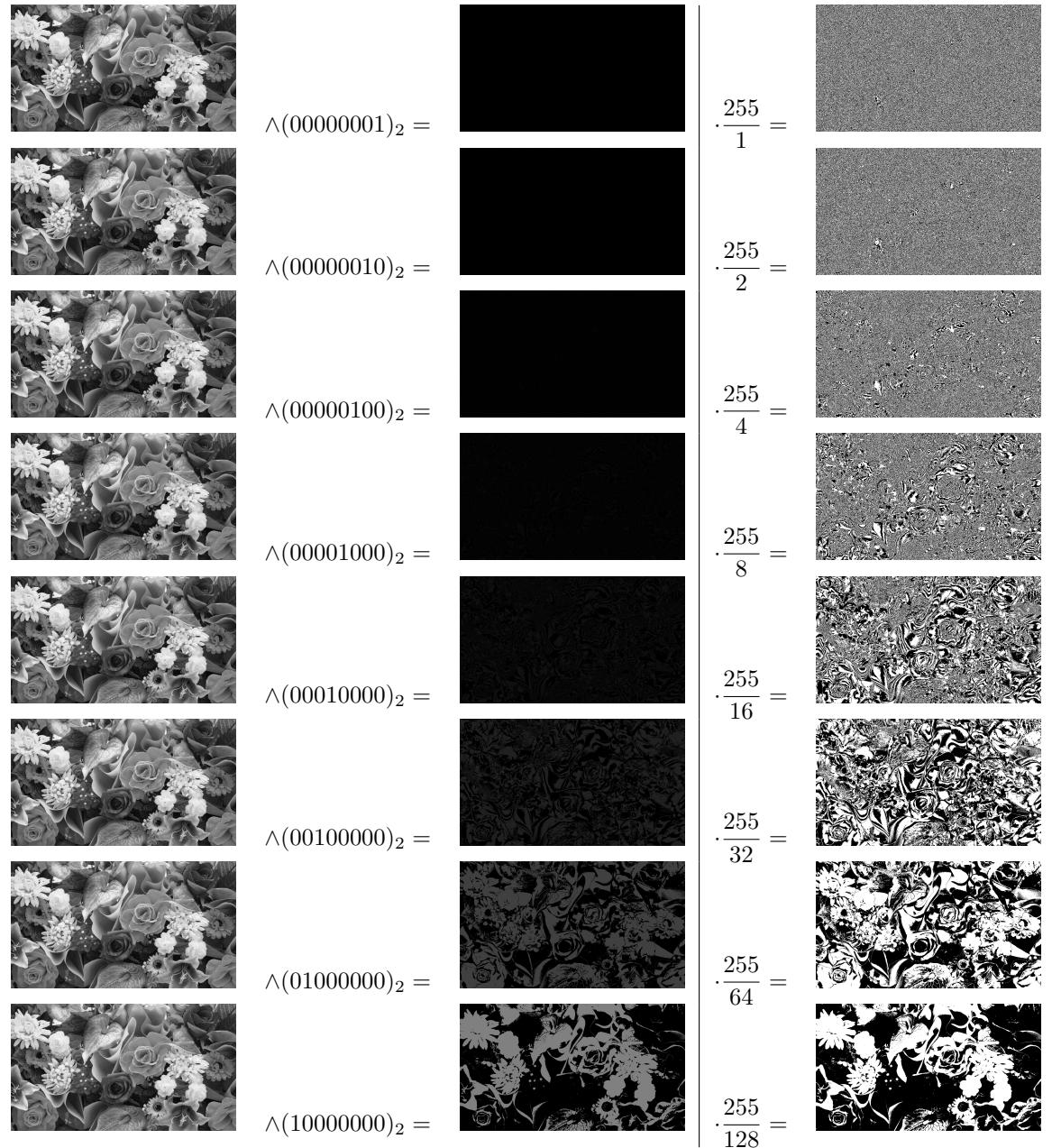


Grafika 7.22: Ekstrakcja 7-mej płaszczyzny bitowej poprzez operację \wedge z obrazem o wartości szarości wszystkich pikseli $(0100\ 0000)_2$

Przykład ekstrakcji płaszczyzn bitowych

—> grafika 7.23

Płaszczyzny bitów niższego rzędu na pierwszy rzut oka wydają się być całkowicie czarne, ponieważ ich maksymalna jasność wynosi tylko 1 (poziom pierwszego bitu), 2 (poziom drugiego bitu) lub 4 (poziom trzeciego bitu). Dlatego obrazy na grafice 7.23 zostały odpowiednio przeskalowane.

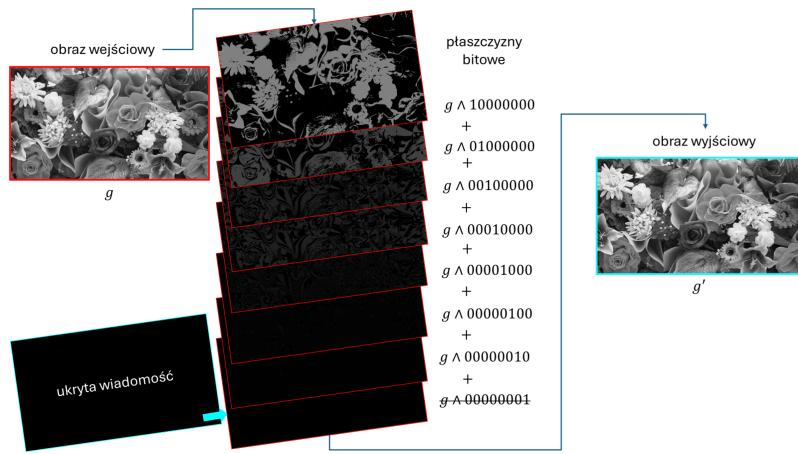


Grafika 7.23: Ekstrakcja płaszczyzn bitowych

Można zauważyc, że płaszczyzny bitów wyższego rzędu określają, czy obszar obrazu jest jasny, czy ciemny. Płaszczyzny bitów niższego rzędu wpływają tylko na drobne struktury.

Wpływ płaszczyzny pierwszego bitu nie jest zauważalny gołym okiem. Można to wykorzystać do ukrycia na obrazie informacji w taki sposób, aby ta nie była widoczna (zamienić pierwszy poziom bitowy obrazu na inny czarno-biały obraz) —→ **steganografia** (grafika 7.24). Wyodrębnienie pierwszej płaszczyzny bitowej ze zmienionego obrazu pozwala na odczytanie ukrytej informacji.

Steganografia nie jest jednak metodą szyfrowania, ponieważ nie jest tu używany żaden klucz. Każdy, kto zna procedurę, może ponownie wyodrębnić wiadomość/informację. Proces steganografii można wykorzystać na przykład do włączenia znaku wodnego do obrazu.



Grafika 7.24: Steganografia

7.2.3 Operacja \neg

$$f^*(m, n) = \neg f(m, n)$$

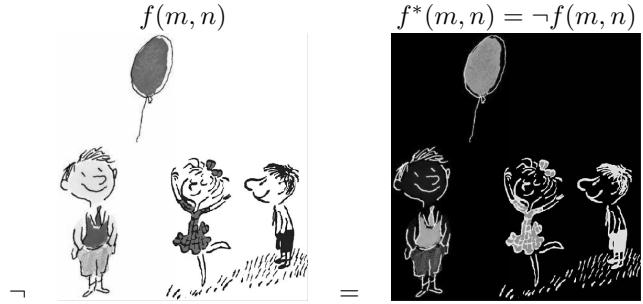
Operacja NOT jest wykonywana na odpowiadających sobie pikselach interpretowanych jako ciągi liczb binarnych. Na każdym bicie wykonywana jest operacja \neg :

A	$\neg A$
0	1
1	0

Wykonanie operacji NOT na czarnym, 8-bitowym pikselu (ciągu ośmiu 0) daje biały piksel (ciąg ośmiu 1). Wartości pośrednie są przetwarzane w ten sam sposób, tzn. wszystkie 1 zmieniają się na 0 i odwrotnie. —→ grafika 7.25

Przykład: $\neg(112)_{10}$

$$\begin{aligned} \neg(112)_{10} &= \neg(01110000)_2 \\ &= (10001111)_2 \\ &= (143)_{10} \end{aligned}$$



Grafika 7.25: Negacja obrazu

7.2.4 Chroma key

Blue box, bluescreen, greenscreen, colorkey → grafika 7.26

Izolowanie elementów obrazu od tła (zwykle w oparciu o kolor kluczowy, keying). Tworzona jest maska (kanał alfa), która definiuje przezroczystość każdego obszaru obrazu lub piksela.



Grafika 7.26: Green screen. Źródło: wikipedia

Maska chroma key (kanał alfa)

W najprostszym zastosowaniu wartość $\alpha(m, n)$ maski dla każdego piksela o wartości koloru $h(m, n)$ oblicza się za pomocą wzoru

$$\alpha(m, n) = \begin{cases} 1 : & \text{jeżeli } (h_{\text{key}} - t) < h(m, n) < (h_{\text{key}} + t) \\ 0 : & \text{w innym przypadku} \end{cases}$$

h_{key} wybrana wartość koloru klucza
 t tolerancja