

Bartłomiej Mróz:

Histogram - czyli wykres częstości występowania kolejnych wartości pikseli obrazu. Histogram pokazuje, jak liczne są w obrazie punkty o różnych wartościach jasności. Przyjmuje się, że pierwszy element histogramu ma numer 0, a ostatni Z_{\max} , gdzie Z_{\max} - zakres maksymalny. Zakładamy, że $Z_{\max} = 2^n - 1$, jeżeli barwa jest reprezentowana przez „n” bitów

Obliczenia:

1. Ustalenie zakresu jasności lub przyjęcie domyślnego dla danej liczby bitów na piksel;
2. Określenie liczby przedziałów;
3. Wyznaczenie szerokości przedziałów poprzez podzielenie zakresu przez ich liczbę;
4. Obliczenie liczby pikseli o wartościach jasności należących do poszczególnych przedziałów.

Parametry liczone z histogramu:

W oparciu o obliczony histogram obrazu możliwe jest wygenerowanie szeregu parametrów opisujących różne własności histogramu, i obrazu. Zdefiniowano szereg parametrów wykorzystujących obliczony już wcześniej histogram $H(b)$: -

$$\text{Średnia: } \bar{x} = \sum_{b=0}^{L-1} \left(p - \frac{p}{2b} \right) b H(b)$$

$$\text{Dyspersja: } \sigma_b^2 = \sum_{b=0}^{L-1} \left[\left(p - \frac{p}{2b} \right) b - \bar{x} \right]^2 H(b)$$

$$\text{Współczynnik asymetrii } b_s = \frac{1}{\sigma_b^3} \sum_{b=0}^{L-1} \left[\left(p - \frac{p}{2b} \right) - \bar{x} \right]^3 H(b)$$

$$\text{Współczynnik ekscesu } b_k = \frac{1}{\sigma_b^4} \sum_{b=0}^{L-1} \left[\left(p - \frac{p}{2b} \right) - \bar{x} \right]^4 H(b) - 3$$

$$\text{Energia } b_N = \sum_{b=0}^{L-1} [H(b)]^2$$

$$\text{Entropia } b_E = - \sum_{b=0}^{L-1} H(b) \log_2 [H(b)]$$

Gdzie:

H - histogram

p - szerokość przedziału

L - liczba przedziałów

Normalizacja histogramu:

Normalizacja histogramu jest prostą operacją punktową stosowaną w celu poprawy obrazów o złym kontraście.

Załóżmy, iż histogram $H(b)$ posiada niezerowe wartości jedynie w pewnym przedziale $[a, b]$ będącym podzakresem przedziału $[0, 255]$, czyli $H(b) = 0$ dla $0 \leq b < a$ oraz $b < b \leq 255$.

Efektom działania operacji normalizacji jest rozszerzenie przedziału $[a, b]$ na pełen zakres odcienie szarości.

Normalizacja jest opisywana funkcją liniową F o wartościach wzrastających od 0 do 255 w przedziale $[a,b]$. $F = 0$ dla $b < a$ oraz $F = 255$ dla $b > a$.

Z liczba elementów

$$H_n(b) = \frac{H(b)}{\text{LiczbaElementowObrazu}}$$

Z liczbą elementów i liczbą przedziałów

$$H_n(b) = \frac{\text{LiczbaPrzedzialow} H(b)}{\text{LiczbaElementowObrazu}}$$

Normalizacja obrazu ma za zadanie ściągnąć cały zakres do przedziału $\langle 0;255 \rangle$

$$X_{norm} = 255 \frac{X - \min(X)}{\max(X) - \min(X)}$$

X - macierz danych
Xnorm - znormalizowana macierz danych

Dysponując histogramem obrazu możemy określić pozycję i szerokość zakresu w którym znajduje się większość danych. W tym celu wprowadza się próg określający minimalną częstość występowania elementów z lewej i prawej strony histogramu.

$$X_{norm} = 255 \frac{X - X_L}{X_P - X_L}$$

XL - lewa granica
XP - prawa granica

Rozciągnięcie histogramu wzdłuż krzywej:

Rozciągnięcie histogramu wzdłuż zadanej krzywej zmienia rozkład jasności pikseli poprzez zmianę ich przyporządkowania do przedziałów histogramu. Przekłada to się na zmianę szerokości przedziałów histogramu:

$$de = f'(b)db$$

gdzie

b – jasność piksela przed rozciągnięciem histogramu;

e – jasność piksela po rozciągnięciu histogramu;

$f(b)$ – funkcja rozciągnięcia histogramu.

Tangens kąta nachylenia stycznej funkcji $f(b)$ jest współczynnikiem zmiany szerokości przedziału.

Chodzi oto, że nową wartości dla piksela liczymy na podstawie równanie (np $y=x+b$ lub $y=x^a$), przy czym trzeba pamiętać aby później zrobić normalizację obrazka do przedziału $\langle 0;255 \rangle$

Wyrównywanie histogramu (ang. histogram equalization) ma na celu takie dobranie wartości aby wykres był możliwie "płaski". W praktyce wyrównywanie histogramu sprowadza się do wykonania przekształcenia obrazu przy pomocy odpowiednio przygotowanej tablicy normalizacyjnej. Operacja wyrównywania histogramu pozwala na uwypuklenie tych szczegółów w obrazie, które z uwagi na niewielki kontrast są mało widoczne.

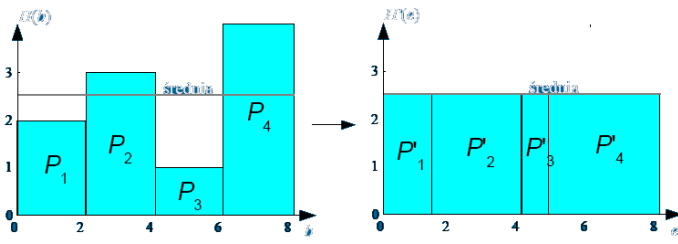
$$sr_H = \frac{\sum_{b=1}^N H(b)}{N}$$

- obliczenie średniej wysokości słupków

N - liczba przedziałów
H - histogram

- Obliczenie nowej szerokości przedziałów

- Obliczanie granic przedziałów i zmiana wartości jasności tak aby znalazły się w odpowiednich przedziałach

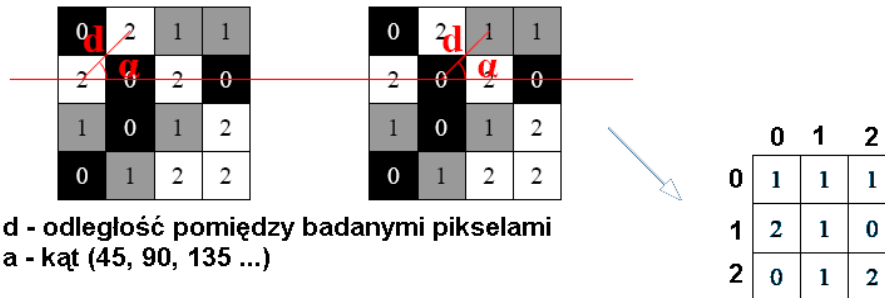


$P_i = P'_i$ Z tego: $H(b)sz\text{er} = sr'_H \text{sz\text{er}'(e)}$

Czyli: $\text{sz\text{er}'(e)} = \frac{H(b)sz\text{er}}{sr'_H}$

Histogram dwuwymiarowy

Histogram dwuwymiarowy służy do badania statystycznych zależności między sąsiednimi pikselami. Jest szczególnie przydatny w analizie tekstur.



d - odległość pomiędzy badanymi pikselami
a - kąt (45, 90, 135 ...)

Łukasz Koreń:

Negatyw

$$P'(x, y) = 255 - P(x, y)$$

gdzie

$P(x, y)$ – jasność oryginalnego piksela obrazka;

$P'(x, y)$ – jasność nowego piksela obrazka.

Progowanie binarne

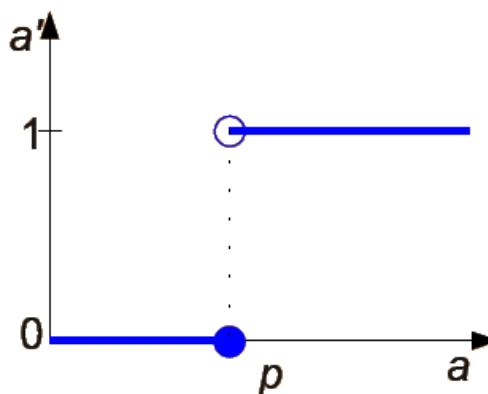
$$a' = \begin{cases} 0 & \text{dla } a \leq p \\ 1 & \text{dla } a > p \end{cases}$$

gdzie

a – jasność oryginalnego piksela obrazka;

a' – jasność nowego piksela obrazka;

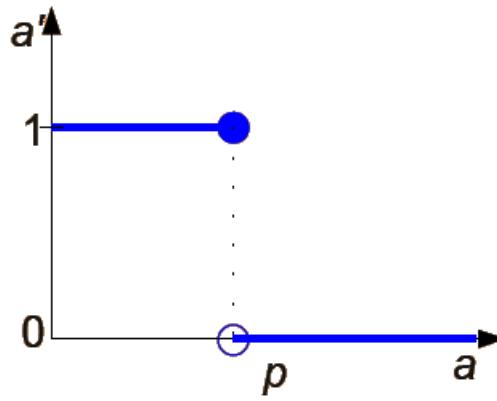
p – ustalony próg.



Progowanie binarne odwrotne

$$a' = \begin{cases} 0 & \text{dla } a > p \\ 1 & \text{dla } a \leq p \end{cases}$$

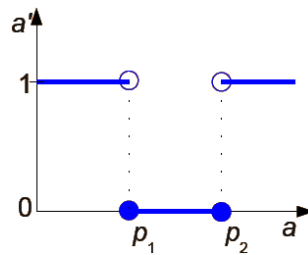
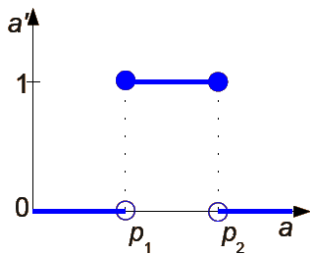
gdzie
 a – jasność oryginalnego piksela obrazka;
 a' – jasność nowego piksela obrazka;
 p – ustalony próg.



Progowanie binarne przedziałowe

$$a' = \begin{cases} 0 & \text{dla } a < p_1 \vee a > p_2 \\ 1 & \text{dla } p_1 \leq a \leq p_2 \end{cases} \quad \text{lub}$$

$$a' = \begin{cases} 0 & \text{dla } p_1 \leq a \leq p_2 \\ 1 & \text{dla } a < p_1 \vee a > p_2 \end{cases}$$

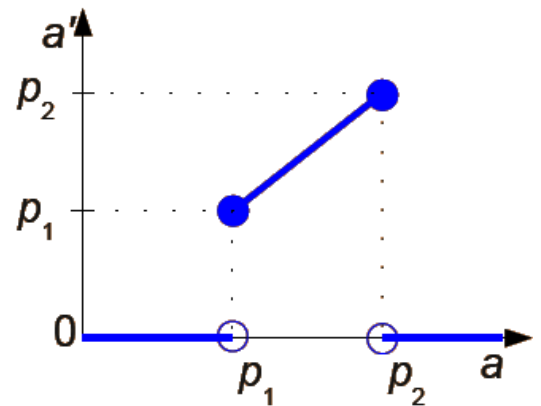


gdzie
 a – jasność oryginalnego piksela obrazka;
 a' – jasność nowego piksela obrazka;
 p_1, p_2 – ustalone progi.

Progowanie z zachowaniem poziomów szarości

$$a' = \begin{cases} 0 & \text{dla } a < p_1 \vee a > p_2 \\ a & \text{dla } p_1 \leq a \leq p_2 \end{cases}$$

gdzie
 a – jasność oryginalnego piksela obrazka;
 a' – jasność nowego piksela obrazka;
 p_1, p_2 – ustalone progi.



Progowanie obrazu kolorowego z zachowaniem poziomów jasności

$$A' = \begin{cases} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \text{dla } a_r < p_{r1} \vee a_r > p_{r2} \vee a_g < p_{g1} \vee a_g > p_{g2} \vee a_b < p_{b1} \vee a_b > p_{b2} \\ A & \text{dla } p_{r1} \leq a_r \leq p_{r2} \wedge p_{g1} \leq a_g \leq p_{g2} \wedge p_{b1} \leq a_b \leq p_{b2} \end{cases}$$

gdzie

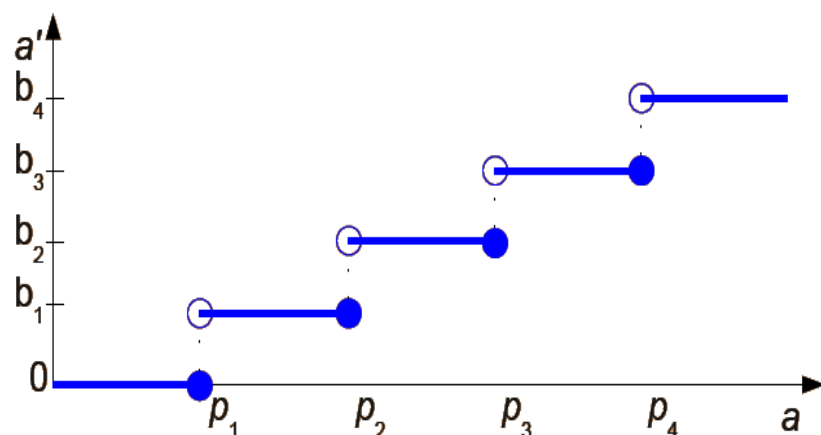
A – wektor składowych koloru piksela obrazka przed progowaniem;

A' – wektor składowych koloru piksela obrazka po progowaniu;

$p_{r1}, p_{r2}, p_{g1}, p_{g2}, p_{b1}, p_{b2}$ – ustalone progi dla poszczególnych kolorów.

Progowanie wielopoziomowe

$$a' = \begin{cases} 0 & \text{dla } a \leq p_1 \\ b_1 & \text{dla } p_1 < a \leq p_2 \\ b_2 & \text{dla } p_2 < a \leq p_3 \\ \vdots & \\ b_{N-1} & \text{dla } p_{N-1} < a \leq p_N \end{cases}$$



gdzie

a – jasność oryginalnego piksela obrazka;

a' – jasność nowego piksela obrazka;

b_1, b_2, \dots, b_N – ustalone poziomy szarości;

p_1, p_2, \dots, p_N – ustalone progi.

Dodawanie obrazów

Dwa lub więcej obrazów możemy połączyć wykonując prostą operację sumowania macierzy:

$$O' = \frac{O_1}{w_1} + \frac{O_2}{w_2} + \dots + \frac{O_N}{w_N}$$

gdzie

O_1, O_2, \dots, O_N – obrazki sumowane;

O' – obrazek powstały w wyniku zsumowania;

w_1, w_2, \dots, w_N – współczynniki normalizujące (wagi).

Na ogół współczynniki normalizujące powinny spełniać warunek:

$$\frac{1}{w_1} + \frac{1}{w_2} + \dots + \frac{1}{w_N} = 1$$

Splot

Splot dwóch funkcji f i g można zapisać następująco:

$$h(x) = \int_{-\infty}^{\infty} f(x') g(x - x') dx'$$

Splot liniowy

Splotem liniowym $y[n]$ dwóch sygnałów czasu dyskretnego $x_1[n]$ i $x_2[n]$ nazywamy sumę:

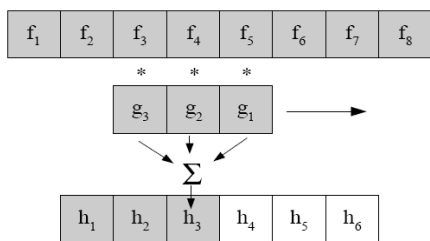
$$y[n] = \sum_{k=0}^n x_1[k] x_2[n-k] = \sum_{k=0}^n x_1[n-k] x_2[k]$$

Splot kołowy

Dla dwóch ciągów $x[n]$ i $h[n]$ o identycznej długości N splot kołowy definiujemy jako:

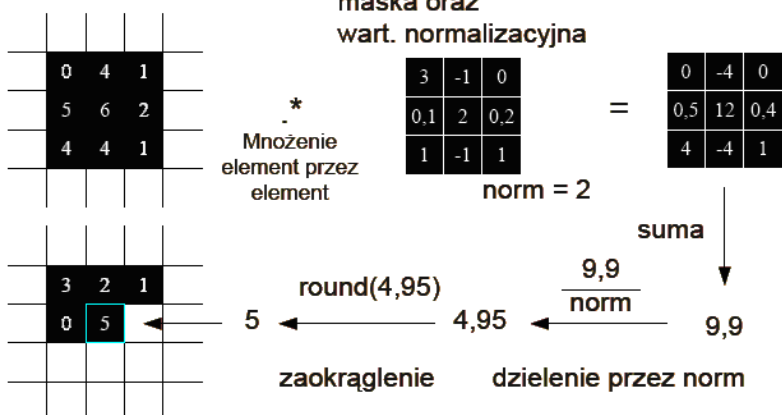
$$h[n] \otimes x[n] = \sum_{k=0}^{N-1} h[k] x[(n-k) \bmod N] \quad n = 0, 1, \dots, N-1$$

Splot sektorowy



Dwuwymiarowa filtracja splotowa

obraz



Problem elementów skrajnych

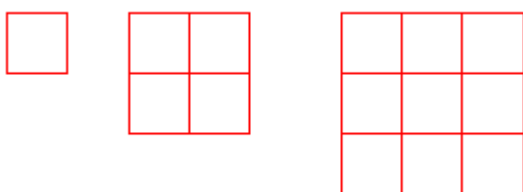
1. Pominięcie elementów skrajnych

0	4	1
5	6	2
4	4	1

2. Powielenie elementów skrajnych

0	0	4	1
0	0	4	1
5	5	6	2
4	4	4	1

3. Dynamiczny rozmiar maski



Filtr splotowy uśredniający (dolnoprzepustowy)

W filtrze tym wartość piksela wyznaczana jest na podstawie uśrednienia jego najbliższego otoczenia. Stopień uśrednienia a zarazem pewnego rozmycia obrazu zależy od wielkości analizowanego otoczenia. W przypadku gdy konieczne jest osłabienie działania filtru elementom centralnym można nadać wartości większe od zera. Wartość normalizacyjna jest sumą wszystkich elementów maski. Efektem działania filtru uśredniającego jest wygładzenie obrazu i usunięcie szumu o niewielkiej amplitudzie. Maski:

1	1	1
1	1	1
1	1	1

norm = 9

1	1	1
1	2	1
1	1	1

norm = 10

1	2	1
2	4	2
1	2	1

norm = 16

Filtr splotowy wyostrzający (górnoprzepustowy)

Filtr górnoprzepustowy wykorzystywany jest do wzmacniania szczegółów o dużej częstotliwości występujących w obrazie. W filtrach tych środkowe elementy maski są zazwyczaj bardzo duże, a pozostałe są niewielkimi liczbami ujemnymi lub zerami. Po filtracji zwiększa się ostrość i kontrast obrazu, ale ujemnym efektem jest wzmocnienie również szumu. Często filtry górnoprzepustowe stosuje się po silnej filtracji uśredniającej, aby przywrócić ostrość obrazu. Maski:

0	-1	0
-1	5	-1
0	-1	0

norm = 1

-1	-1	-1
-1	9	-1
-1	-1	-1

norm = 1

1	-2	1
-2	5	-2
1	-2	1

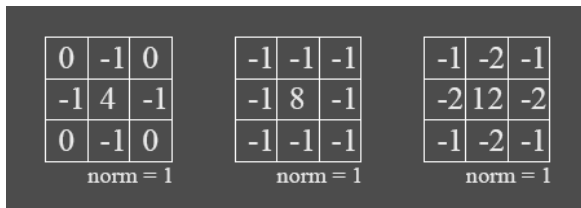
norm = 1

Filtr splotowy konturyzacyjny

Po filtracji obszary o jednolitym kolorze mają wartość zero, a obszary gdzie jasność zmieniała się mocno (czyli głównie brzegi obiektów) mają bardzo duże wartości (dodatnie lub ujemne)

Filtr ten służy do przekształcania obrazów do postaci wektorowej.

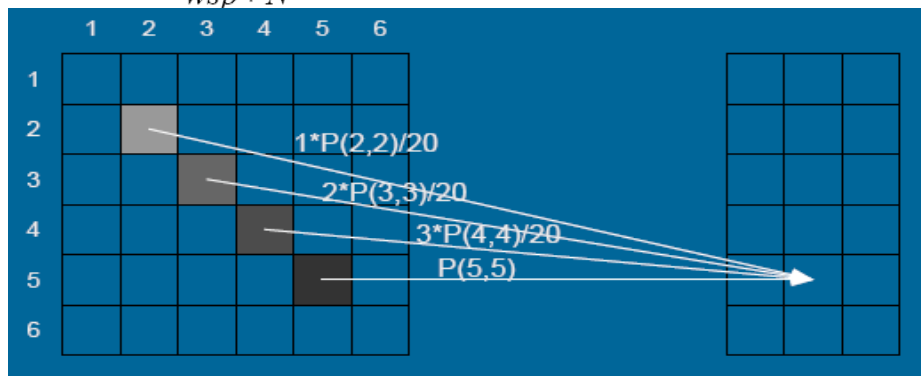
przykładowe maski:



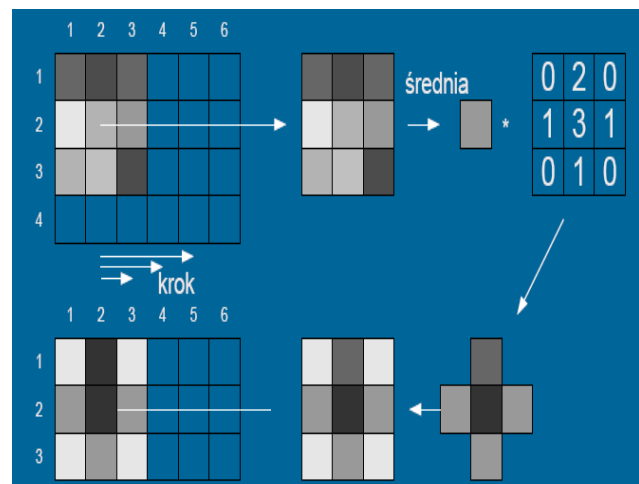
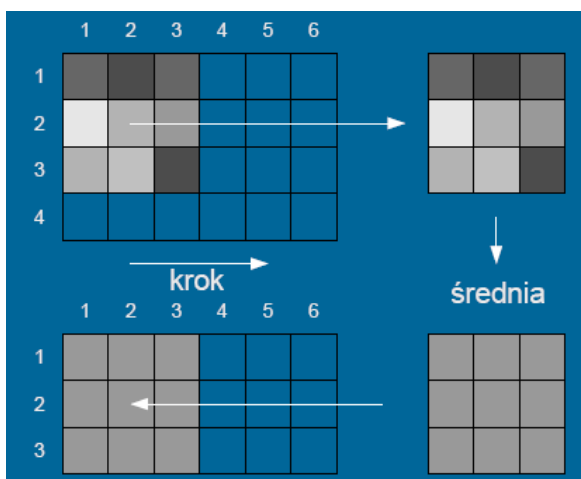
Modyfikacje filtru splotowego – efekt wiatru- pozwala ona na rozmycie obrazu w wybranym kierunku i daje wrażenie jakby zdjęcie było zrobione w ruchu lub ze zbyt krótkim czasem naświetlania. Efekt wiatru uzyskujemy przekształcając obraz za pomocą wzoru:

$$P'(x, y) = P(x, y) + \frac{\sum_{i=1}^N (N+1-i) P(x-i, y-i)}{wsp * N}$$

gdzie: N – liczba powtórzeń,
 wsp – współczynnik.

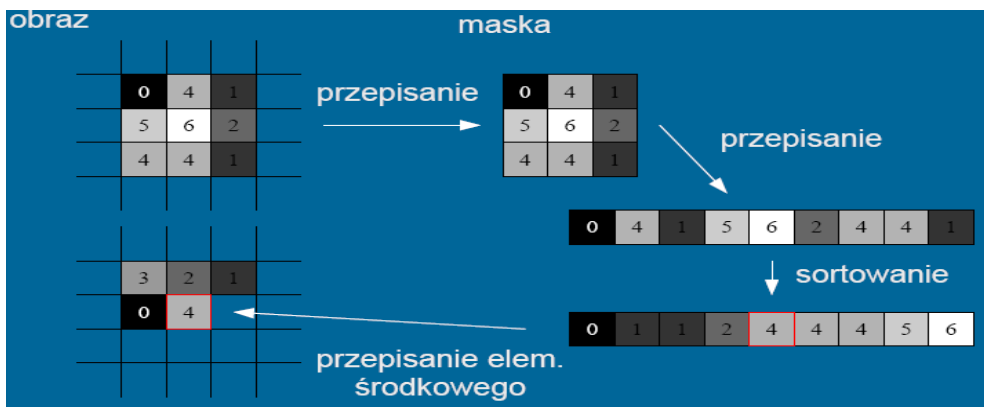


Modyfikacje filtru splotowego – pikselizacja

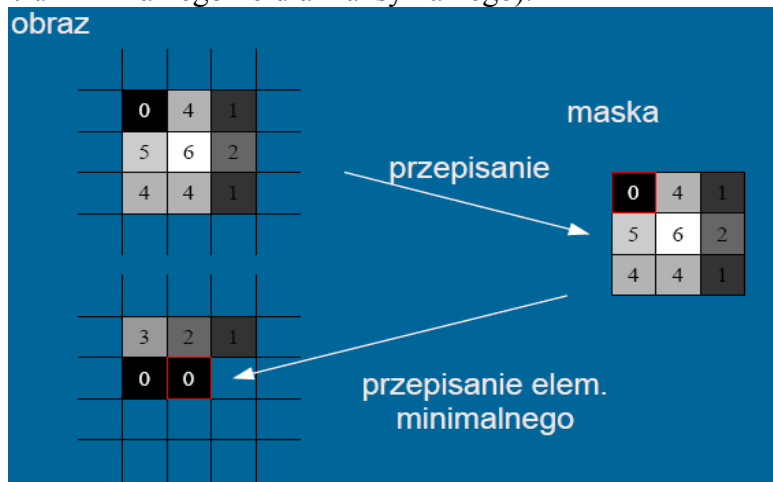


Filtry statyczne- Służą przede wszystkim do usuwania zakłóceń losowych (szumów) na obrazie, nie zmieniają one przy tym użytecznych informacji.

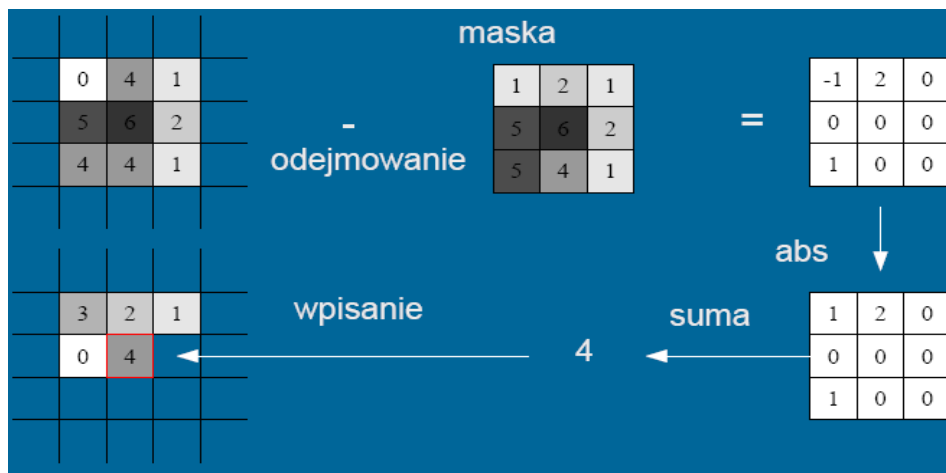
- **filtr medianowy**-przepisujemy określoną liczbę pikseli(wymiar maski) i sortujemy rosnąco, a następnie zapisujemy w wyjściowym obrazie wartość środkową naszej listy. Istnieją 2 wersje: mocna- gdy obliczony piksel zastępuje nam wejściowy, i bierze udział w obliczaniu sąsiednich pikseli, oraz wersja słabsza, kiedy obliczone piksele są zapisywane w innym miejscu.



• **filtr minimalny i maksymalny:** w zależności od wielkości maski, badamy wartość jasności w pikseli w obrębie naszej wybranej kropki, i zastępujemy ją największą lub najmniejszą wartością (w tym przypadku 0 dla filtru minimalnego i 6 dla maksymalnego).



• **filtr różnicowy-** w zależności o wielkości maski, od wartości pikseli w badanym obszarze odejmujemy wartości maski, obliczamy wartość bezwzględna i sumujemy- wynik zapisujemy w miejsce badanego piksela.



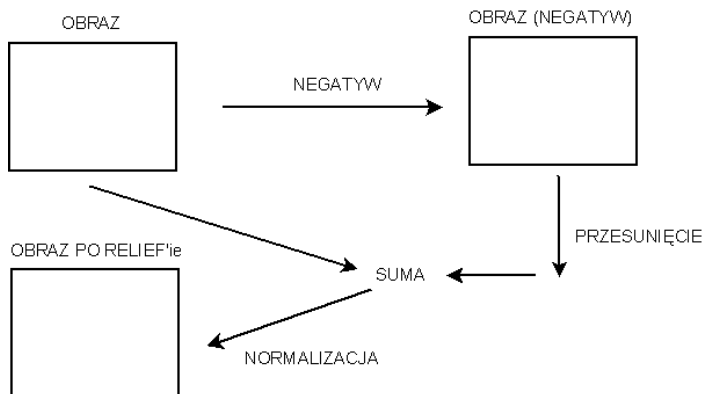
Tomasz Waligóra:

KONTURYZACJA

Relief

Mamy obraz . Robimy z niego negatyw, a następnie robimy przesunięcie tego negatywu (np. o 1 piksel, o 2 piksele itd.). Następnie sumujemy obraz oryginalny z obrazem negatywu po przesunięciu. Otrzymany obraz

normalizujemy.



Konturyzacyjny filtr korelacyjny

Konturyzacyjny filtr korelacyjny definiuje maskę filtru splotowego:

$$M = \begin{bmatrix} \rho_C \rho_R & -\rho_C(1 + \rho_R^2) & \rho_C \rho_R \\ -\rho_R(1 + \rho_C^2) & (1 + \rho_C^2)(1 + \rho_R^2) & -\rho_R(1 + \rho_C^2) \\ \rho_C \rho_R & -\rho_C(1 + \rho_R^2) & \rho_C \rho_R \end{bmatrix}$$

Gdzie :

ρ_C i ρ_R – współczynniki korelacji między punktami obrazu;

M – maska filtru splotowego.

Współczynniki ρ_C i ρ_R obieramy sami, wtedy np. dla

$\rho_C=0$ i $\rho_R=0$ mamy maskę :

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

a dla $\rho_C=1$ i $\rho_R=1$, maskę

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Filtr konturyzacyjny Gaussa

Funkcja Macleoda:

$$M(x, y) = \exp\left\{-\frac{1}{2}\left(\frac{y}{t}\right)^2\right\} \left[\exp\left\{-\frac{1}{2}\left(\frac{x-p}{p}\right)^2\right\} - \exp\left\{-\frac{1}{2}\left(\frac{x+p}{p}\right)^2\right\} \right]$$

Gdzie:

p i t – współczynniki podające wpływ elementów obrazu na wyliczaną wartość;

n i m – rozmiary maski.

$$x \in \left\langle -\frac{n}{2}; \frac{n}{2} \right\rangle \quad y \in \left\langle -\frac{m}{2}; \frac{m}{2} \right\rangle$$

Filtr Robertsza

Do realizacji tego filtru wystarczy jedynie skorzystać ze wzoru :

$$o_w(j, k) = \sqrt{[o(j, k) - o(j+1, k+1)]^2 + [o(j, k+1) - o(j+1, k)]^2}$$

lub :

$$o_w(j, k) = |o(j, k) - o(j+1, k+1)| + |o(j, k+1) - o(j+1, k)|$$

gdzie

O_w – obraz wynikowy

O – obraz źródłowy

Filtr Sobela

W tym przypadku poruszamy się po obrazie podobnie jak w filtracji splotowej. Przesuwamy się kolejno od lewego górnego rogu do prawego dolnego i wycinamy z obrazu maski o rozmiarze 3x3. Maski takie przedstawiają się następująco:

A_0	A_1	A_2
A_7	$o(j,k)$	A_3
A_6	A_5	A_4

Następnie dokonujemy obliczeń :

$$X = (A_2 + 2A_3 + A_4) - (A_0 + 2A_7 + A_6)$$

$$Y = (A_0 + 2A_1 + A_2) - (A_6 + 2A_5 + A_4)$$

Po obliczeniu wartości X oraz Y podstawiamy je do obrazu wynikowego według wzoru :

$$o_w(j, k) = \sqrt{X^2 + Y^2}$$

Filtr Kirscha

Tutaj podobnie jak w filtrze Sobela wyznaczamy kolejne maski obrazu :

A_0	A_1	A_2
A_7	$o(j,k)$	A_3
A_6	A_5	A_4

Następnie trzeba obliczyć kolejne wartości S_i oraz T_i :

$$S_i = A_i + A_{i+1} + A_{i+2}$$

$$T_i = A_{i+3} + A_{i+4} + A_{i+5} + A_{i+6} + A_{i+7}$$

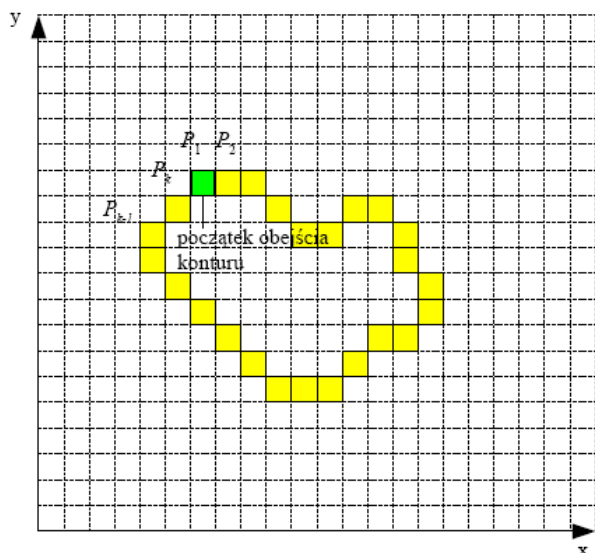
pamiętając przy tym, że

$i \in \langle 0; 7 \rangle$, a indeksy zmieniają się modulo 8.

Po obliczeniu wartości S_i oraz T_i , podstawiamy wszystko do obrazu wynikowego według wzoru :

$$o_w(j, k) = \max \left\{ 1, \max_{i \in \langle 0; 7 \rangle} |5S_i - 3T_i| \right\}$$

Wektoryzacja konturu rastrowego – metoda najbliższego sąsiada



Aktualnie znaleziony punkt zaznaczamy jako bieżący, usuwamy go z tablicy I, a następnie szukamy jego najbliższego sąsiada. Oznacza to, że wszystkie punkty o jednakowej odległości od punktu bieżącego, znajdują się na obrzeżu kwadratu, o środku w tym punkcie.

1. Start. Szukamy punktu początkowego, zapamiętujemy go i ustawiamy jako bieżący.
2. Ustalamy bok kwadratu na 3.
3. Sprawdzamy czy są jakieś punkt na obwodzie kwadratu.
4. Jeśli są to wybieramy ten o najmniejszej odległości

od punktu bieżącego. Jeśli nie ma to skaczemy do pkt. 3 i zwiększamy bok kwadratu o 2.

5. Zapamiętujemy punkt i ustawiamy go jako bieżący. Przechodzimy do pkt 3.

Cały proces powtarzamy do momentu aż nie przekroczymy ustalonego rozmiaru kwadratu.

Filip Przybysz

1. Aproksymacja obrazu wieloma obrazami

Obraz można przedstawić jako sumę ważoną innych obrazów (z błędem). Polega na podziale obrazu na bloki odpowiadające rozmiarom obrazów z bazy (nimi będziemy przybliżać) fragmenty skrajne mniejsze od obrazów bazy należy odrzucić.

współczynniki aproksymacji dla każdego fragmentu według wzoru:

wsp. aproksymacji (korelacji) mówią nam jak podobne są do siebie dwa obrazy, przyjmuje wart. Od -1 do 1, 0 oznacza max niepodobieństwo, wart. Skrajne to max podobieństwo.

$$\begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{bmatrix} = \begin{bmatrix} (X_1, X_1) & (X_2, X_1) & \dots & (X_n, X_1) \\ (X_1, X_2) & (X_2, X_2) & \dots & (X_n, X_2) \\ \dots & \dots & \dots & \dots \\ (X_1, X_n) & (X_2, X_n) & \dots & (X_n, X_n) \end{bmatrix}^{-1} \begin{bmatrix} (A, X_1) \\ (A, X_2) \\ \dots \\ (A, X_n) \end{bmatrix}$$

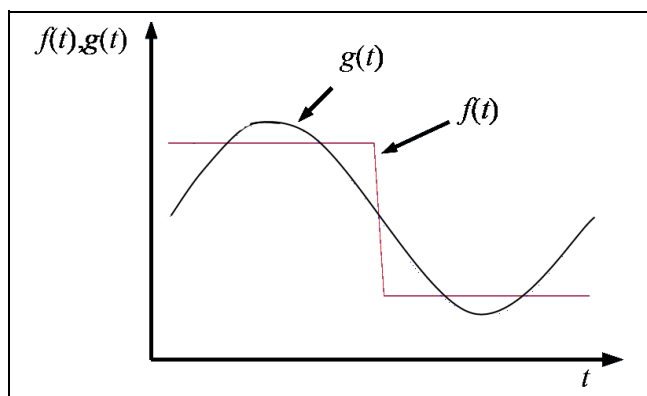
Po obliczeniu współczynników odtwarzamy każdy wg wzoru:

$$A = c_1 X_1 + c_2 X_2 + \dots + c_n X_n$$

$X_1 \dots X_n$ - obrazy bazowe

$c_1 \dots c_n$ - wsp. aproksymacji

4. Przybliżanie sygnału innym sygnałem



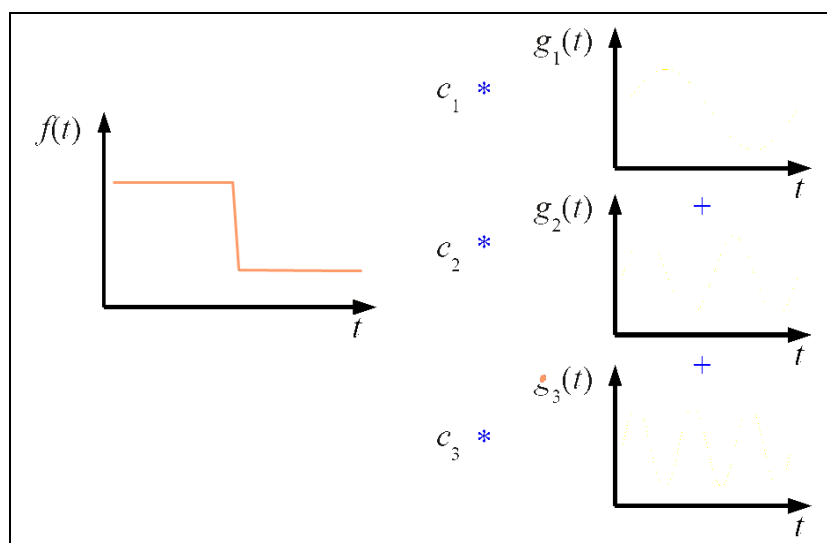
$f(t) = c \cdot g(t) + v_e(t)$ z tego:

$$c = \frac{\int_{t_1}^{t_2} f(t) * g(t) dt}{\int_{t_1}^{t_2} g^2(t) dt}$$

dla obrazu:

$$c = \frac{\sum f(t) * g(t)}{\sum g^2(t)}$$

5. Przybliżanie dla wielu sygnałów:



$$c_i = \frac{\int_{t_1}^{t_2} f_i(t) * g_i(t) dt}{\int_{t_1}^{t_2} g_i^2(t) dt}$$

Konrad Górczyca:

1) Porównywanie obrazów:

- Punkt po punkcie

Cała metoda polega na porównaniu kolejnych pikseli obrazów. Wzór:

$a = \frac{h * w * n}{100}$, gdzie a – ilość tych samych pikseli, h – wysokość obrazka, w – szerokość obrazka, n – wartość podobieństwa.

- średnia RGB z kwadratu

Występuje wzór jak wyżej oraz:

$$r = 256 - \frac{256 \cdot n}{100}, \quad r - \text{różnica}$$

Każdy obraz jest dzielony na mniejsze obrazy, przy pomocy siatki, następnie:

$$c = \frac{R + G + B}{3}, \quad \text{gdzie } c - \text{średnia koloru w danym kwadracie, RGB - wartości składowych danego piksela,}$$

2) Kompresja

Składa się z 4 etapów:

Transformata cosinusowa:

$$\begin{cases} c(k, l) = \frac{2}{64} \alpha(k) \alpha(l) \sum_{m=0}^7 \left(\sum_{n=0}^7 x(m, n) \cos\left(\frac{(2n+1)l\Pi}{16}\right) \right) \cos\left(\frac{(2n+1)k\Pi}{16}\right) \\ x(m, n) = 2 \sum_{m=0}^7 \left(\sum_{n=0}^7 \alpha(k) \alpha(l) c(k, l) \cos\left(\frac{(2n+1)l\Pi}{16}\right) \right) \cos\left(\frac{(2n+1)k\Pi}{16}\right) \end{cases}$$

Gdzie:

$x(m, n)$ – wartość piksela obrazu początkowego

$c(k, l)$ – współczynnik transformaty

$$\alpha(k) = \begin{cases} \frac{1}{\sqrt{2}}; k, l \equiv 0 \\ 1; k, l = 1, 2, \dots, 7 \end{cases}$$

Kwantowanie:

$$\begin{cases} \hat{C}(k, l) = C(k, l) / (k * l) \\ \hat{C}(k, l) = C(k, l) / V \end{cases}$$

Gdzie $V \gg 1$ lub $\equiv C(1, 1)$

Kodowanie obrazów:

Kod Hoffmana działa na zasadzie wyliczania częstości występowanie danej jasności

3) Składowa wektorów wzdłuż innych wektorów

$$A = c_1 X_1 + c_2 X_2 + \dots + c_n X_n$$

4) Przybliżanie obrazów innymi obrazami

Sprowadza się do znalezienia wektora c współczynników z ww. równania

5) Rozbarwienie

Dodanie do każdego X pewnej stałej, przesuwałającej kolor.

6) Usunięcie składowej

Usunięcie czyli zlikwidowanie jednego X z danego równania

7) Wykrywanie kierunków

Wiele metod detekcji krawędzi bazuje na pierwszej pochodnej luminancji - daje to nam stopień przyrostu (gradient) wartości oryginalnych danych. Wykorzystując tą informację możemy szukać ekstremów pierwszej pochodnej luminancji (gradientu luminancji).

Jeśli $I(x)$ odzwierciedla luminancję piksela x , a $I'(x)$ jest pierwszą pochodną (intensity gradient) piksela x , można zapisać :

$$I'(x) = -1/2 * I(x-1) + 0 * I(x) + 1/2 * I(x+1);$$

Andrzej Licznierski – Godziuk

Definicje

Grupa – zdefiniowane mnożenie łączne, element neutralny 1, element odwrotny

Grupa abelowa – mnożenie jest przemienne

Ciało – dodawanie, mnożenie, rozdzielność mnożenia względem dodawania

Przestrzeń wektorowa – zdefiniowane mnożenie przez skalar; jest grupa abelowa; jest rozdzielność i łączność

Przestrzeń prehilbertowska – zdefiniowany iloczyn skalarny (x,y)

Przestrzeń unitarna – prehilbertowska + zdefiniowano normę $\sqrt{(x,x)}$

Malowanie jednym pedzlem

Malowanie jednym pedzlem polega na odtworzeniu obrazu w odcieniach jednego koloru – np. piękne odcienie różowego. Mając dany wektor koloru różowego $C = (255, 128, 128)$ i piksel obrazka $A = (r, g, b)$ wyliczamy najpierw współczynnik skalowania c :

$c = \frac{(A,C)}{(C,C)}$, a następnie skalujemy wektor C o wartość c – czyli nowa wartość piksela $A' = c \cdot C$. Jako bonus, wzór

na iloczyn skalarny: $(A,C) = a_1c_1 + a_2c_2 + a_3c_3$.

Filtracja barw

Filtracja barw polega na zmieszaniu obrazu oryginalnego z obrazem „namalowanym jednym pedzlem”. Dodajemy obraz oryginalny (z wagą np. 0.8) do obrazu monochromatycznego (powiedzmy z wagą 0.2). Gotowe.

Laczenie obrazow za pomoca rzutowania kolorow

Jak malowanie jednym pedzlem, tylko kolor pedzla w kazdym pikselu bierzemy z drugiego obrazka.

Filtr Gabora

Filtr Gabora należy do filtrów spłotowych. Jego maska to $h(x, y) = s(x, y) g(x, y)$ - powstaje z pomnożenia dwóch filtrów składowych. Składowa s to zespolona składowa sinusoidalna i jej wzór to $s(x, y) = e^{-j2\pi(u_0x + v_0y)}$. Składowa g to gaussian evenlop (jakkolwiek głupio by to nie brzmiało po polsku) i jej wzór to

$$g(x, y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)}$$

Filtr ten służy m.in. do wykrywania składowych o różnych kątach nachylenia. Kąt który będziemy wykrywać

$$\omega_0 = \tan^{-1}\left(\frac{v_0}{u_0}\right)$$

można wyliczyć ze wzoru. Parametry σ_x, σ_y określają szerokość pasma przepustowego filtru (lub, jak kto woli, standardowe odchylenie). Można przyjąć, że ich miara są piksele. Tak więc, ustawiając wartości σ na odpowiednio 40 i 50 oraz kąt na -45° , wypadaloby ustalić rozmiar maski na 128x128!

Składowa wektora wzdłuż innych wektorów wzajemnie ortogonalnych

Wektor A wyrażamy jako kombinację liniową wzajemnie ortogonalnych wektorów $X_1..X_n$:

$$\bar{A} = c_1 \bar{X}_1 + c_2 \bar{X}_2 + \dots + c_n \bar{X}_n, \text{ gdzie } c_1..c_n - \text{skalary.}$$

- Wersja macierzowa obliczeń:

$$\begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{bmatrix} = \begin{bmatrix} (\bar{X}_1, \bar{X}_1) & 0 & \dots & 0 \\ 0 & (\bar{X}_2, \bar{X}_2) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & (\bar{X}_n, \bar{X}_n) \end{bmatrix}^{-1} \begin{bmatrix} (\bar{A}, \bar{X}_1) \\ (\bar{A}, \bar{X}_2) \\ \dots \\ (\bar{A}, \bar{X}_n) \end{bmatrix}$$

- Wersja iteracyjna obliczeń:

$$c_r = \frac{\bar{A} \cdot \bar{X}_r}{\bar{X}_r \cdot \bar{X}_r}$$

Ortonormalizacja

Wektory bazy są **ortonormalne** jeśli są wzajemnie ortogonalne (iloczyn skalarny wynosi 0) oraz unormowane do 1.

Ortonormalizacja – przekształcenie wektorów liniowo niezależnych w ortonormalne.

Proces ortonormalizacji **Grama-Schmidta**: dla wektorów wejściowych $X_1..X_n$, obliczamy wektory wyjściowe (zortonormalizowane) $Y_1..Y_n$.

Najpierw obliczamy Y_1 :

$$\bar{Y}_1 = \frac{\bar{X}_1}{|\bar{X}_1|}$$

Następnie kolejne wektory $Y_2..Y_n$:

$$\bar{Y}_k = \frac{\bar{X}_k - \bar{V}_k}{|\bar{X}_k - \bar{V}_k|} \quad \text{gdzie} \quad \bar{V}_k = \sum_{i=1}^{k-1} (\bar{Y}_i \cdot \bar{X}_k) \bar{Y}_i$$

Inny (bardziej czytelny) wzór na obliczanie Y_k to:

$$\bar{Y}_k = \bar{X}_k - \sum_{i=1}^{k-1} \frac{(\bar{X}_k, \bar{Y}_i)}{\|\bar{Y}_i\|^2} \bar{Y}_i, \text{ gdzie } (X, Y) - \text{iloczyn skalarny; } \|X\| - \text{norma}$$

Ortogonalność funkcji zespolonych

$$c_r = \frac{\int_{t_1}^{t_2} f(t) g_r^*(t) dt}{\int_{t_1}^{t_2} g_r(t) g_r^*(t) dt}$$

, gdzie $f(t)$, $g(t)$ to funkcje których ortogonalność badamy; $g^*(t)$ – sprzężenie wartości funkcji

Funkcje $f(t)$, $g(t)$ są ortogonalne jeśli wartość c jest zerowa.

Szereg trygonometryczny

Chcemy zapisać funkcję $f(t)$ w przedziale $(t_0 < t < t_0 + T)$ jako ciąg współczynników $a_0, a_1, a_2, \dots, b_1, b_2, \dots$

Aby „złożyć” wartość funkcji ze współczynników, dokonujemy operacji:

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 t + b_n \sin n\omega_0 t), \text{ gdzie } n = 1, 2, 3, \dots, \omega = \frac{2\pi}{T}$$

Współczynniki z funkcji $f(t)$ uzyskuje się następująco:

$$a_0 = \frac{1}{T} \int_{t_0}^{t_0+T} f(t) dt$$

$$a_n = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \cos n\omega_0 t dt$$

$$b_n = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \sin n\omega_0 t dt$$

Szereg wykładniczy

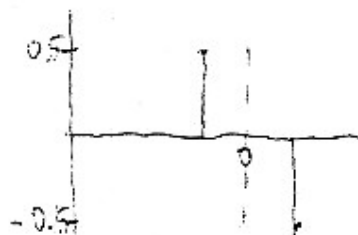
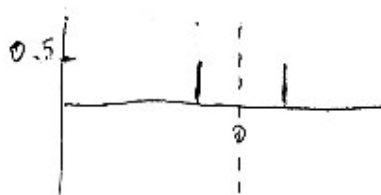
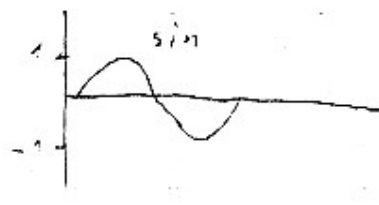
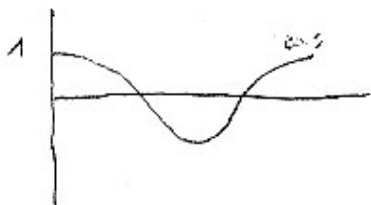
Przedstawiamy funkcję $f(t)$ jako ciąg współczynników F_n , gdzie $n = 0, \pm 1, \pm 2, \dots$

Operacja „złożenia” wartości funkcji ze współczynników:

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{jn\omega_0 t}$$

Liczenie współczynników F_n :

$$F_n = \frac{1}{T} \int_{t_0}^{t_0+T} f(t) e^{-jn\omega_0 t} dt$$



Odpowiedniki dla wersji **dwuwymiarowej**:

$$f(x, y) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} F(u, v) e^{j(ux\omega_{x0} + vy\omega_{y0})} \quad (x_0 < x < x_0 + T_x) \quad (y_0 < y < y_0 + T_y)$$

, gdzie $u = 0, \pm 1, \pm 2, \dots$ $v = 0, \pm 1, \pm 2, \dots$

$$F_{u,v} = \frac{1}{T_x T_y} \int_{y_0}^{y_0 + \frac{2\pi}{\omega_{y0}}} \int_{x_0}^{x_0 + \frac{2\pi}{\omega_{x0}}} f(x, y) e^{-j(ux\omega_{x0} + vy\omega_{y0})} dx dy$$

Transformata Fouriera

Uwaga: ω jest ciągłe, nie – jak w poprzednich transformatach – dyskretne!

$$f(t) = \int_{-\infty}^{\infty} F(\omega) e^{j2\pi\omega t} d\omega$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi\omega t} dt$$

Wersja **dwuwymiarowa**:

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv$$

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy$$