

# Algorytmy przetwarzania obrazów i sygnałów cyfrowych, kryteria oceny

Dobrosław Kowalski

Wrocław, 2017

**Streszczenie** Dokument opisuje wybrane zagadnienia z dziedziny przetwarzania sygnałów oraz przegląd wybranych algorytmów.

---

**Spis treści**

1	Sygnal - wiadomości ogólne . . . . .	3
1.1	Sygnal - definicja . . . . .	3
1.2	Możliwości sygnału . . . . .	3
1.3	Podział sygnału . . . . .	3
2	Dyskretyzacja sygnału ciągłego w czasie . . . . .	4
2.1	Transformacja Fouriera - podstawowa analiza sygnału . . . . .	4
3	Wybrane algorytmy przetwarzania sygnałów cyfrowych . . . . .	5
3.1	Dyskretna transformata Fouriera . . . . .	5
3.2	Dyskretna transformata cosinusowa - kompresja jpeg obrazu . . . . .	7
3.3	Filtry cyfrowe - typy i charakterystyki . . . . .	10
3.4	Przekształcenia obrazu - operacje geometryczne . . . . .	12
3.5	Interpolacja . . . . .	15
3.6	Filtrowanie obrazów . . . . .	17

## 1 Sygnał - wiadomości ogólne

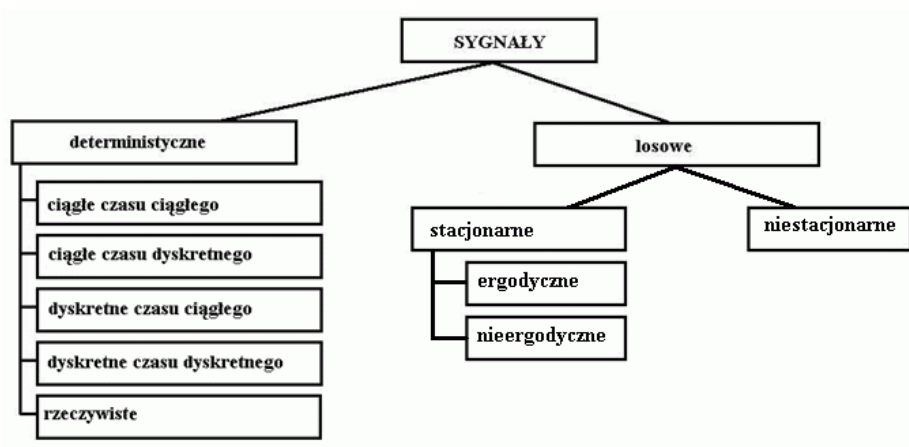
### 1.1 Sygnał - definicja

Sygnałem nazywamy abstrakcyjny model dowolnej mierzalnej wielkości zmieniającej się w czasie, generowanej przez zjawiska fizyczne lub systemy. Może być opisany za pomocą aparatu matematycznego, np. poprzez podanie pewnej funkcji zależnej od czasu. Mówi się wtedy, że sygnał niesie informację lub też umożliwia przepływ strumienia informacji.

### 1.2 Możliwości sygnału

1. badanie stanu i zachowania się fizycznych lub systemów
2. mierzenie badanych wielkości
3. przekazywanie informacji w czasie i przestrzeni
4. sterowanie wybranymi zjawiskami lub systemami

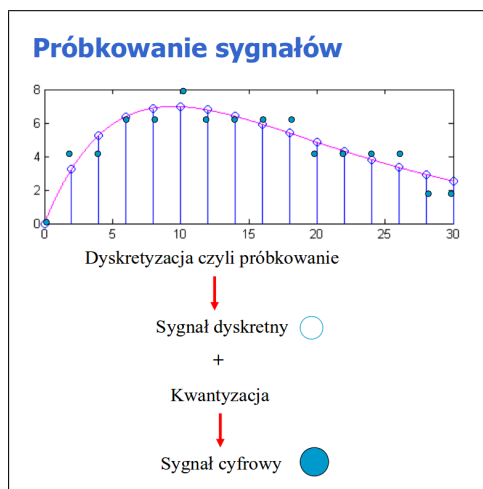
### 1.3 Podział sygnału



Rysunek 1 Podstawowy podział sygnałów

## 2 Dyskretyzacja sygnału ciągłego w czasie

Przetwarzanie ciągłego sygnału analogowego na sygnał cyfrowy polega na dyskretyzacji sygnału w czasie czyli jego **próbkowaniu**, dyskretyzacji wartości sygnału czyli **kwantowaniu** oraz na **kodowaniu** uzyskanego sygnału dyskretnego. Próbkowanie następuje przez kolejne pobieranie próbek wartości sygnału w pewnych odstępach czasu, w taki sposób, aby ich ciąg umożliwił jak najwierniejsze odtworzenie całego przebiegu funkcji. Kwantowanie polega na przyporządkowaniu każdej próbce skończonej liczby poziomów amplitudy, odpowiadającym dyskretnym wartościom od zera do pełnego zakresu. Za proces dyskretyzacji odpowiedzialny jest przetwornik analogowo-cyfrowy. Zamiana sygnału analogowego na cyfrowy powinna przebiegać z możliwie małą utratą informacji.



Rysunek 2 Schemat dyskretyzacji sygnału

### 2.1 Transformacja Fouriera - podstawowa analiza sygnału

Transformacja Fouriera jest podstawowym narzędziem analizy częstotliwościowej sygnałów. Analiza sygnałów najczęściej sprowadza się do ich analizy częstotliwościowej. Częstotliwości zawarte w sygnałach występujących samoistnie w przyrodzie wiele mówią o właściwościach lub stanach obiektu, który je wygenerował. Przykładem mogą być różne sygnały biometryczne, np. elektrokardiograficzne, świadczące o aktualnym stanie pacjenta.

W wielu urządzeniach technicznych, np. służących do transmisji sygnałów za pomocą gal elektromagnetycznych, człowiek sam dzieli dostępne pasmo częstotliwościowe na podpasma oraz generuje w nich sygnały użytkowe o różnych częstotliwościach, np. radiowe, telewizyjne i telefoniczne.

Prosta i odwrotna transformata Fouriera są zdefiniowane następującą parą równań:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt, \quad x(t) = \int_{-\infty}^{\infty} X(f)e^{-2\pi ft} df \quad (1)$$

Gdzie  $X(f)$  jest zespolonym widmem Fouriera sygnału  $x(t)$  i zawiera informację o jego zawartości częstotliwościowej. Wartość  $X(f)$  informuje o tym "ile" jest w sygnale zespolonej składowej harmonicznej  $e^{-j2\pi ft} = \cos(2\pi ft) + j\sin(2\pi ft)$  o konkretnej częstotliwości  $f$ .

Dla sygnałów dyskretnych  $x(n)$  para transformacji Fouriera (1) przyjmuje następującą postać ( $-f_{pr}/2 < f \leq f_{pr}/2$ ):

$$X(f) = \sum_{n=-\infty}^{+\infty} x(n\Delta t)e^{-j2\pi(f/f_{pr})n} \quad (2)$$

$$x(n\Delta t) = \frac{1}{f_{pr}} \int_{-f_{pr}/2}^{+f_{pr}/2} X(f)e^{j2\pi(f/f_{pr})n} df \quad (3)$$

Sygnał  $x(n)$  lub  $x(n\Delta t)$  powstaje w wyniku pobierania "próbek" z sygnału  $x(t)$  odległych o czas  $\Delta t = 1/f_{pr}$  gdzie  $f_{pr}$  oznacza częstotliwość próbkowania, wyrażoną w hercach. Matematyczny zapis próbki sygnału pobranej z sygnału ciągłego reprezentuje równanie:

$$x_{\delta}(t) = x(n) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - n\Delta t) \quad (4)$$

gdzie  $\delta(t)$  jest impulsem Diraca. **Zgodnie z twierdzeniem o próbkowaniu częstotliwość próbkowania musi być co najmniej dwa razy większa niż maksymalna częstotliwość sygnału.** Widmo częstotliwościowe  $X_{\delta}(j\omega)$  sygnału spróbowanego  $x_{\delta}(t)$  reprezentuje równanie:

$$X_{\delta}(j\omega) = \frac{1}{2\pi} [X(j\omega) \otimes \omega_p \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_p)] = \frac{\omega_p}{2\pi} \sum_{k=-\infty}^{\infty} X(j(\omega - k\omega_p)) \quad (5)$$

gdzie  $\omega_p = 2\pi f_p$ . Widmo częstotliwościowe jest więc sumą poprzesuwanych w częstotliwości "kopii" widma sygnału ciągłego  $X(j\omega)$ . Dysponując jedną z tych kopii oraz znając pasmo częstotliwościowe oryginału, jesteśmy w stanie zsyntetyzować z niej ciągły sygnał oryginalny  $x(t)$ . W tym celu należy wyznaczyć odwrotne przekształcenie Fouriera „kopii” i zmodulować częstotliwościowo uzyskany wynik, tzn. wymnożyć go z odpowiednim sygnałem harmonicznym  $\exp(jk\omega_p t)$ .

### 3 Wybrane algorytmy przetwarzania sygnałów cyfrowych

#### 3.1 Dyskretna transformata Fouriera

**DFT** przekształca skończony ciąg próbek sygnału ( $a_0, a_1, a_2, \dots, a_{N-1}$ ) w ciąg harmonicznymi ( $A_0, A_1, A_2, \dots, A_{N-1}$ ),  $A_i \in C$ . Inaczej mówiąc, transformata ta jest zespolona i zawiera informację o amplitudzie i fazie przetwarzanego sygnału. Jest też symetryczna względem indeksu Nyquista (równego  $N/2$ , gdzie  $N$  jest całkowitą

liczbą próbek), przy czym amplituda transformaty jest symetryczna a faza antysymetryczna. Oznacza to, że próbki od  $N/2 + 1$  do  $N - 1$  mogą zostać obliczone na podstawie próbek z zakresu od 0 do  $N/2$ . Jeśli dodatkowo, liczba próbek  $N$  będzie parzysta, to rozkład harmonicznych (dodatnich i ujemnych) będzie symetryczny indeksu Nyquista. Podstawowym parametrem związanym z dyskretną transformatą jest tzw. rozdzielczość częstotliwościowa  $df$  określana wzorem  $df = F_s/N = 1/(Ndt)$  gdzie:  $F_s$  oznacza częstotliwość próbkowania a  $dt$  okres próbkowania. Zwiększenie rozdzielczości częstotliwościowej jest możliwe na dwa sposoby:

- przez zwiększenie liczby pobranych próbek  $N$  (dla  $F_s = const$  oznacza to wydłużenie czasu rejestracji)
- przez zmniejszenie częstotliwości próbkowania  $F_s$  (dla  $N = const$  oznacza to zmniejszenie pasma analizy).

Jedną z *metod oceny* jakości przeprowadzonej transformaty Fouriera - twierdzenie Parsewala - jest sprawdzenie równości mocy w sygnale przetwarzanym (dziedzina czasu) i przetworzonym (dziedzina częstotliwości).

Z dyskretną transformatą Fouriera powiązanych jest kilka problemów. Podstawowym założeniem transformaty Fouriera jest nieskończenie długo trwający (lub reprezentowany przez nieskończoną liczbę próbek) sygnał poddawany analizie widmowej. W praktyce to założenie nie może być zrealizowane i możliwe są dwa przypadki:

- czas trwania sygnału zakończy się przed pobraniem ustalonej liczby próbek  $N$  (co wraz z  $F_s$  - częstotliwością próbkowania tworzy tzw. okno czasowe  $T = N * F_s$  określające czas w jakim pobierane są próbki)
- czas generowania próbek  $T$  zostanie zatrzymany zanim analizowany sygnał przestanie być generowany

Dodatkowo, dla sygnałów okresowo zmiennych, pojawia się problem „synchronizacji” okna pomiarowego z okresem transformowanego sygnału. Jeżeli w oknie czasowym nie wystąpi całkowita wielokrotność okresów przebiegu badanego sygnału to widmo w takim wypadku ulegnie poszerzeniu (w stosunku do sygnału o okresie mieszczącym się całkowitą liczbą razy) i pojawi się „efekt przecieku” (energia zawarta w jednej składowej częstotliwości przecieka do sąsiadujących). Taki efekt jest szczególnie widoczny przy detekcji sygnałów o małej amplitudzie leżących blisko sygnału wysoko amplitudowego.

Sposobem na ograniczenie przecieku widma jest stosowanie okienkowania sygnału w celu zmniejszenia nieciągłości sygnału na krańcach obszaru wyznaczonego przez okno pomiarowe. Okienkowanie jest wykonaniem splotu na sygnale i funkcji zastosowanego okna. Dla okna prostokątnego uzyskujemy efekt ograniczenia analizy (dalej transformaty Fouriera) do czasu równego szerokości okna, inne okna np. Hanninga, exponentialne, Kaisera-Bessela, itp. służą do zmniejszenia stopnia nieciągłości sygnału na jego końcach przez zastosowanie funkcji wagowych zmierzających w gładki sposób do zera. Działanie okna najlepiej obserwować po stronie widmowej. Zastosowanie okien widoczne jest tam w formie zmniejszania rozdzielczości częstotliwościowej (dla całkowitej liczby okresów analizowanego sygnału) i zmniejszeniem efektu przecieku widma (dla niecałkowitej liczby okresów analizowanego sygnału, w porównaniu do okna prostokątnego). Przeciek widma najlepiej obserwować przez porównanie zachowania się szerokości listka głównego i bocznych widma analizowanych sygnałów. Dla

sygnałów przejściowych (rozpoczynają się i kończą amplitudami zerowymi w oknie pomiarowym oraz całkowicie się w nim mieszczą). Stosowanie okien prostokątnych powoduje utratę rozdzielczości częstotliwościowej i deformację kształtu sygnału na końcach. Ostatecznie **DFT** opisuje para równań:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, 2, \dots, N - 1 \quad (6)$$

$$x(n) = \frac{1}{2\pi} \sum_{k=0}^{N-1} X(k)e^{j\frac{2\pi}{N}kn}, \quad n = 0, 1, 2, \dots, N - 1 \quad (7)$$

Obliczanie tych sum za pomocą powyższego wzoru zajęłoby  $O(N^2)$  operacji. Istnieją algorytmy szybszego obliczania DFT tzw. procedury **szybkiej transformacji Fouriera - FFT** np. algorytm Cooleya-Tukeya. Bazują na metodzie *dziel i zwyciężaj*, rekurencyjnie dzieląc transformatę wielkości  $N = N_1N_2$  na transformaty  $N_1$  i  $N_2$  z wykorzystaniem  $O(n)$  operacji mnożenia.

Najpopularniejszą wersją FFT jest FFT o podstawie 2. To bardzo efektywna operacja, jednak wektor próbek wejściowych (spróbkowany sygnał) musi mieć długość  $N = 2^k$ , gdzie  $k$  to pewna liczba naturalna. Wynik otrzymuje się na drodze schematycznych przekształceń, opartych o tak zwane *struktury motylkowe*.

Złożoność obliczeniowa Szybkiej transformacji Fouriera wynosi  $O(N\log_2N)$  zamiast  $O(N^2)$  algorytmu wynikającego wprost ze wzoru określającego dyskretną transformatę Fouriera. Dzięki istnieniu takiego algorytmu praktycznie możliwe stało się cyfrowe przetwarzanie sygnałów (DSP), a także zastosowanie dyskretnych transformat kosinusowych (DCT) (JPEG, MP3 itd.) do kompresji.

### 3.2 Dyskretna transformata cosinusowa - kompresja jpeg obrazu

JPEG jest algorytmem stratnej kompresji grafiki rastrowej. Może być używany do kompresji pojedynczych obrazów oraz ich sekwencji (video).

W standardzie JPEG kompresowany obraz jest dzielony na bloki pikseli o wymiarach  $8 \times 8$ . Każdy z nich jest dalej do pewnego momentu przetwarzany osobno. Na początku jest obliczana **dwuwymiarowa transformata kosinusowa** dla  $N = M = 8$  opisana wzorem (wzór dla postaci dwuwymiarowej).

$$X_{DCT}(k, l) = \sum_{m=0}^{M-1} \left[ \sum_{n=0}^{N-1} x(m, n) * \beta(l)\cos\left(\frac{\pi l}{N}(n + 1/2)\right) * \alpha(k)\cos\left(\frac{\pi k}{M}(m + 1/2)\right) \right] \quad (8)$$

gdzie:

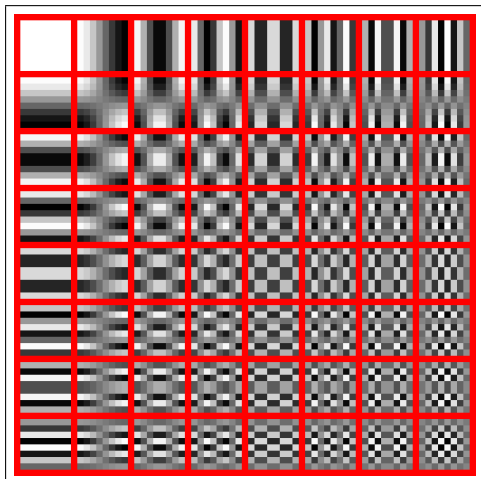
$$\alpha(k) = \begin{cases} \sqrt{1/M}, & k = 0 \\ \sqrt{2/M}, & k = 1, \dots, M - 1 \end{cases} \quad (9)$$

$$\beta(l) = \begin{cases} \sqrt{1/N}, & l = 0 \\ \sqrt{2/N}, & l = 1, \dots, N - 1 \end{cases} \quad (10)$$

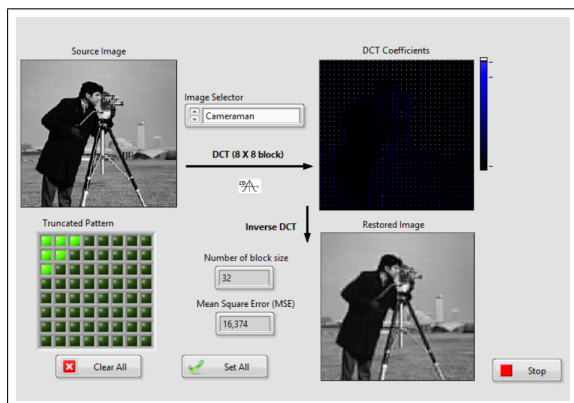
W praktyce przy pomocy **DCT** można rozłożyć dowolny obraz na sumę składowych, zależnych od częstotliwości funkcji bazowych - kosinusów. Składowe te odpowiadają

za poziom szczegółowości obrazu. Ten etap nie odpowiada jeszcze za docelową kompresję i jest całkowicie odwracalny.

Poniższy rysunek obrazuje wszystkie funkcje bazowe transformacji DCT dla bloku 8 x 8 pikseli. Obraz (blok 8 x 8) zrekonstruowany z jego reprezentacji częstotliwościowej DCT będzie ważoną sumą wszystkich 64 obrazków.

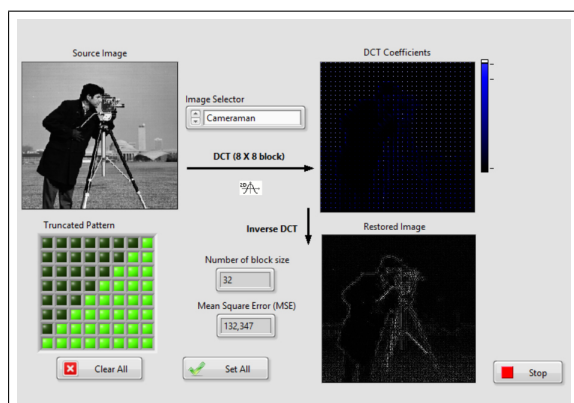


Poniższy obrazek prezentuje porównanie obrazu oryginalnego z obrazem utworzonym z harmonicznych o *niskich* częstotliwościach. Jak widać większość energii obrazu zachowana jest właśnie w niskich składowych harmonicznych.



Kolejny obrazek prezentuje porównanie obrazu oryginalnego z obrazem utworzonym z harmonicznych o *wysokich* częstotliwościach. Jak widać energia zgromadzona w tych sygnałach odpowiada za szczegóły oraz szum obrazu.





Eksperyment wykonano w środowisku LabView. Klikając na odpowiednie kwadraty „maski” można aktywować lub deaktywować wpływ wybranych składowych harmonicznych na obraz wynikowy.

Po przeprowadzeniu transformaty algorytm dokonuje kwantyzacji współczynników  $X_{DCT}(k, l)$  z wykorzystaniem przyporządkowanych im wartości  $Q(k, l)$  z tablic kwantyzacji  $Q$ :

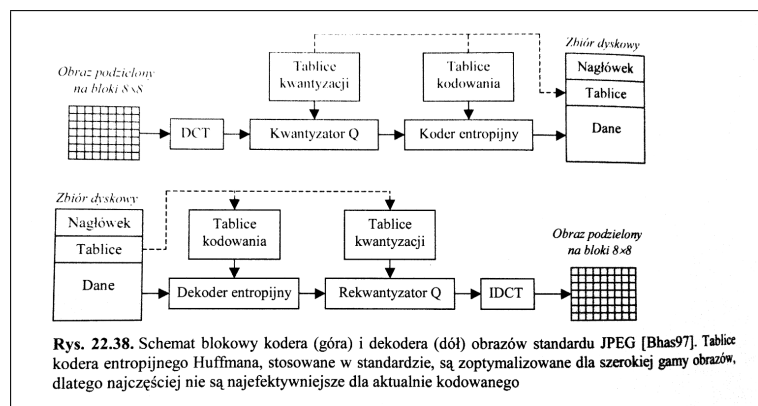
$$Z(k, l) = \text{round}\left(\frac{Y_{DCT}(k, l)}{Q(k, l)}\right) = \lfloor \frac{Y_{DCT}(k, l) \pm Q(k, l)/2}{Q(k, l)} \rfloor, \quad k, l = 0, 1, \dots, 7 \quad (11)$$

W wyniku tej operacji współczynniki transformaty o małych wartościach zostają wyzerowane. Warto pamiętać, że norma JPEG zaleca używanie kilku tablic kwantyzacji np. osobna tablica dla składowej luminancji i osobne dla tablic chrominancji.

Na tym etapie w wyniku zastosowanych operacji obraz reprezentowany jest przez bloki macierzy, które zawierają współczynniki energetyczne określające „wagę” czyli zawartość składowej harmonicznej w danym bloku obrazu - już po procesie kompresji (wyzerowane składowe o wysokich częstotliwościach). W tym momencie można już przekształcić liczby na ciągi bitowe, oraz zakodować je **entropijnym koderem Huffmana**.

Należy pamiętać, że **współczynniki DC** (lewy górny narożnik) bloków 8x8 pikseli są najczęściej duże i skorelowane (z powodu składowej stałej obrazu), dlatego są one traktowane specjalnie: współczynnik DC pierwszego bloku oraz różnice tych współczynników pomiędzy kolejnymi blokami są zapisywane w kodzie VLI (*Variable Length Integers*). Pozostałe **współczynniki tzw. AC**, każdego bloku są kodowane oddzielnie. Ich wartości skanuje się wzdłuż linii ukośnych prostopadłych do głównej diagonalnej macierzy współczynników transformaty, zaczynając od lewego górnego rogu, czyli porządkuje od najniższych do najwyższych częstotliwości. Ponieważ w wyniku kwantyzacji większość współczynników AC jest wyzerowanych, uzyskuje się długie sekwencje zer, poprzedzielane nielicznymi wartościami niezerowymi.

Schemat JPEG zaprezentowano na poniższym rysunku.



Warto w tym miejscu wspomnieć o działaniu dekodera JPEG czyli o powrocie obrazu do postaci macierzy liczb. W dekoderyze wszystkie operacje są przeprowadzane w odwrotnej kolejności, a rekwantyzacja jest przeprowadzana według wzoru:

$$\hat{Y}_{DCT}(k,l) = Q(k,l) * \hat{Z}(k,l), \quad k,l = 0, 1, \dots, 7 \quad (12)$$

gdzie  $\hat{Z}(k,l)$  jest wynikiem dekodowania entropijnego.

### 3.3 Filtry cyfrowe - typy i charakterystyki

Filtracja wykorzystująca dyskretną transformatę Fouriera (DFT) jest związana z procesem przetwarzania sygnału w dziedzinie czasu w celu uzyskania pożądanych zmian w widmie sygnału oryginalnego. Filtr cyfrowy może być dedykowanym układem cyfrowym lub programowalnym procesorem albo programem komputerowym. Rozwiązania takie, w przeciwieństwie do filtrów analogowych, są stosunkowo łatwe do zbudowania i implementacji. Nie zmieniają charakterystyk w funkcji temperatury ani wilgotności, nie starzeją się ani nie zależą od producenta ani od serii i daty produkcji, są też dużo tańsze w produkcji.

Rozróżnia się dwa podstawowe typy filtrów cyfrowych:

- **FIR (Finite Impulse Response)** - filtry o skończonej odpowiedzi impulsowej znane też jako filtry MA (Moving Average) o ruchomej średniej, lub filtry nierekurencyjne. Odpowiedź tego filtra zależy od bieżącej i poprzednich wartości wejściowych.
- **IIR (Infinite Impulse Response)** - filtry o nieskończonej odpowiedzi impulsowej znane też jako filtry ARMA (Autoregressive Moving Average) autoregresyjne z ruchomą średnią, lub filtry rekurencyjne. Odpowiedź filtra zależy od bieżącej i poprzednich wartości wejściowych - jak dla FIR - oraz bieżących i poprzednich wartości wyjściowych.

Podstawowe różnice między filtrami:

- Filtry FIR są: łatwiejsze do projektowania i zastosowania, są z natury stabilne i odporne na błędy zaokrążeń (odpowieź filtra po skończonym czasie osiąga wartość zero), mają liniową fazę, działają jednak wolniej, wymagają większej pamięci i większej liczby współczynników
- Filtry IIR są: trudniejsze do projektowania i zastosowania, mogą być niestabilne a błędy zaokrążeń mogą się kumulować, przesunięcie fazowe jest nieliniowe, działają jednak znacznie szybciej i wymagają mniej współczynników (niższy rząd filtrów)

W kontekście matematycznym filtry cyfrowe można opisać za pomocą liniowego równania różnicowego:

$$\sum_{k=0}^N a_k y[n-k] = \sum_{l=0}^M b_l x[n-l] \quad (13)$$

przy czym

- dla  $N = 0$  równanie opisuje filtr nierekursywny (FIR)
- dla  $N > 0$  równanie opisuje filtr rekursywny (IIR)

Idealny filtr charakteryzuje się jednostkowym wzmocnieniem ( $0dB$ ) w paśmie przepustowym i wzmocnieniem zero  $-\infty dB$  w paśmie zaporowym. Rzeczywisty filtr posiada jednak skończone tłumienie a jego charakterystyka może zawierać zafalowania w pasmach przepustowych i zaporowych. Zwykle, jeśli jest to filtr pasmowy, dąży się do tego aby zafalowania były symetryczne. Wartość tłumienia można określić korzystając z następującego wzoru:

$$dB = 20 \log(A * o(f) / A * i(f)) \quad (14)$$

gdzie  $Ai(f)$  amplituda określonej harmonicznej (częstotliwości) przed filtracją a  $Ao(f)$  po filtracji (na wyjściu filtra). Tłumienie i kształt charakterystyk filtrów zależy od rodzaju filtru i rzędu filtru (taps - liczby współczynników). Zwykle im wyższy rząd filtra tym jego charakterystyka posiada bardziej strome zbocza, mniejsze zafalowania i wyższy współczynnik tłumienia. Projektowanie filtrów cyfrowych jest poszukiwaniem jak najprostszego filtru spełniającego przyjęte założenia. Generalnie, filtry FIR stosuje się, gdy wymagana jest liniowa faza, filtry IIR: Butterwortha gdy poszukiwana jest płaska charakterystyka bez zafalowań dopuszczająca niezbyt dobrą selektywność (łagodne przejście charakterystyki między pasmem przewodzenia a zaporowym), Czebyszewa gdy wymagana jest większa selektywność i dopuszczalne są symetryczne zafalowania poza pasmem przepustowym, eliptyczne gdy wymagane jest wąskie pasmo przepustowe i dopuszczalne są zafalowania, a filtry Bessela gdy poszukiwany jest filtr o możliwie płaskiej odpowiedzi (dla amplitudy i fazy) i prawie liniowej fazie w paśmie przepustowym.

Znanych jest wiele metod projektowania filtrów cyfrowych. Filtry FIR można projektować korzystając np. z metody okienkowej zwanej też klasyczną. Zadanie polega na takim doborze kształtu funkcji wycinającej (okienkowej) aby dla zadanej charakterystyki częstotliwościowej filtra uzyskać jak najmniejszą liczbę współczynników (taps - rząd filtra). Nie jest to metoda najlepsza, lepsze wyniki (niższy rząd filtra) można znaleźć korzystając np. z metody Parksa i McClellana opartej o algorytm Remez. Algorytm ten śledzi, na bieżąco, różnicę między żadaną a aktualną charakterystyką w całym zakresie częstotliwościowym filtra, dając w wyniku filtr o stałym poziomie tętnień w paśmie przepustowym i paśmie zaporowym. W porównaniu do metody

okienkowej wymagany jest długi czas projektowania, złożoność obliczeniowa i brak gwarancji zbieżności algorytmu.

Projektowanie filtra typu IIR jest zadaniem o wiele bardziej skomplikowanym niż projektowanie FIR (w filtrach IIR amplituda odpowiedzi impulsowej narasta z czasem a w filtrach FIR dąży do zera). Wyznaczenie charakterystyki częstotliwościowej filtra IIR na podstawie znanych (przyjętych) wartości nie jest problemem (zadanie wprost). Problem pojawia się gdy należy wyznaczyć współczynniki filtra na podstawie charakterystyki częstotliwościowej (zadanie odwrotne). Nie jest znana metoda takiego rozwiązania i stosowane są metody przybliżone np. metoda niezmienności odpowiedzi impulsowej, zakładająca istnienie, odpowiadającego projektowanemu filtrowi cyfrowemu, filtra analogowego o takiej samej odpowiedzi impulsowej. Projektowanie sprowadza się do: określenia kształtu pożądanej odpowiedzi częstotliwościowej, zaprojektowania filtra analogowego, wyznaczenia jego odpowiedzi, wykonania próbkowania tej odpowiedzi a próbki te decydują o poszukiwanych współczynnikach filtra. Należy tutaj pamiętać o *aliasingu* zniekształcającym takie działanie. Badanie stabilności filtra może być realizowane na wykresie biegunowym (Zero-Pole-Gain) tworzonym z przestrzeni częstotliwościowej  $-\infty < f > +\infty$  za pomocą transformacji dwuliniowej. Funkcja odpowiedzi impulsowej filtra IIR będzie zanikała (gdy bieguny będą wewnątrz koła jednostkowego), będzie ustabilizowana (bieguny w kole jednostkowym) lub rozbieżna (bieguny na zewnątrz koła). Wadą wykresu biegunowego (w porównaniu do wykresu 'moduł-faza') jest brak odwzorowania charakterystyk przejściowych filtrów.

### 3.4 Przekształcenia obrazu - operacje geometryczne

Operacje geometryczne, w których położenie piksela  $(i, j)$  zmieniane jest zgodnie zadaną relacją matematyczną, a jego intensywność nie ulega zmianie. Przekształcenia tego typu są stosowane do przestrzennego dopasowania obrazów uzyskanych w różnym czasie lub z różnych źródeł. Brak odpowiedniego dopasowania może skutkować w pogorszeniu ich jakości ze względu na utratę ważnych lub interesujących informacji. Do podstawowych operacji geometrycznych zaliczamy:

- **przesunięcie**
- **skalowanie**
- **obrót**
- **odbijanie symetryczne**

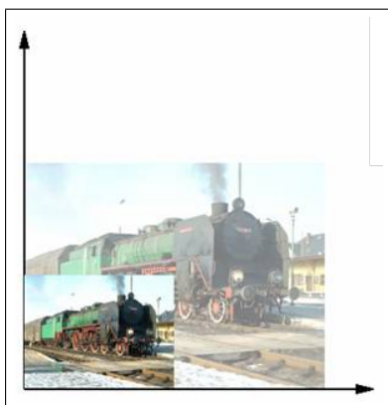
#### *Skalowanie obrazu*

Obraz o wymiarach  $x_{oryg}, y_{oryg}$  ma być przeskalowany w kierunku osi  $x$  ze współczynnikiem  $x_{wsp}$ , a w kierunku osi  $y$  ze współczynnikiem  $y_{wsp}$ . Nowe wymiary obrazu to  $(x_{nowe}, y_{nowe})$ .

$$x_{nowe} = x_{oryg} * x_{wsp} \quad (15)$$

$$y_{nowe} = y_{oryg} * y_{wsp} \quad (16)$$

Jeśli współczynniki  $x_{wsp}$  oraz  $y_{wsp}$  pozostaną równe to takie przekształcenia nazywamy jednorodnym i obraz przekształcony zachowa proporcje. Przy wyznaczaniu nowych współrzędnych pikseli podczas zwiększania wymiaru obrazu, mogą powstać

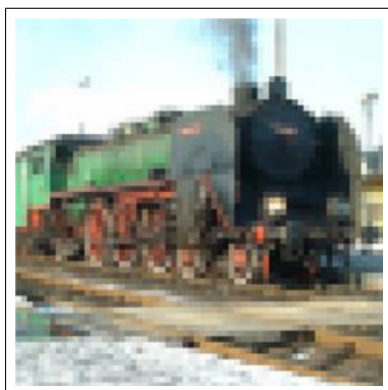


nieciągłości między pikselami. Pewnym rozwiązaniem tego problemu jest **próbko-**  
**wanie obrazu wejściowego** . W tym celu korzysta się z równań:

$$x_{oryg} = x_{nowe} / x_{wsp} \quad (17)$$

$$y_{oryg} = y_{nowe} / y_{wsp} \quad (18)$$

Pozycje starego piksela są obliczane w funkcji nowego piksela. Dla każdego piksela obrazu wyjściowego obliczana jest intensywność na podstawie obrazu oryginalnego, czyli oryginalny obraz jest próbkowany. Wadą tego podejścia jest fakt, że po przeskalowaniu, obraz będzie wyglądał jakby składał się z bloków.



Wyjściem jest **interpolacja** .

#### Obrót

Obrót jest znacznie bardziej złożonym przekształceniem niż skalowanie. Wykorzystuje bowiem interpolację i korekcję współczynnika kształtu obrazu. Poniżej podane są równania reprezentujące obrót wokół początku układu współrzędnych o kąt  $\gamma$ .

$$x_{nowe} = x_{oryg} \cos(\gamma) - y_{oryg} \sin(\gamma) \quad (19)$$

$$y_{nowe} = x_{oryg} \sin(\gamma) + y_{oryg} \cos(\gamma) \quad (20)$$

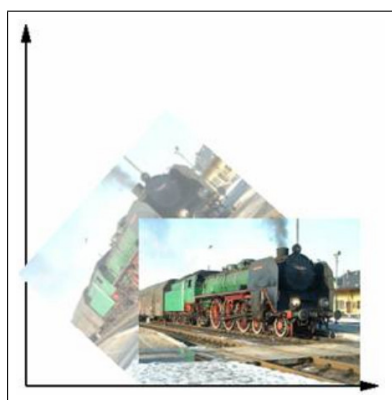
gdzie  $(x_{nowe}, y_{nowe})$  oznaczają nowe współrzędne piksela obróconego obrazu, zaś  $(x_{oryg}, y_{oryg})$  - współrzędne piksela w oryginalnym obrazie.

Ogólna postać tych wzorów pozwalająca na obrót obrazu wokół dowolnego punktu  $(A, B)$ :

$$x_{nowe} = A + (x_{oryg} - A) \cos(\gamma) - (y_{oryg} - B) \sin(\gamma) \quad (21)$$

$$y_{nowe} = B + (x_{oryg} - A) \sin(\gamma) + (y_{oryg} - B) \cos(\gamma) \quad (22)$$

Analogicznie jak przy skalowaniu po obrocie obrazu korzystamy z równań odwrotnych. Unikamy w ten sposób występowania nieciągłości w obrazie. Możemy również stosować *interpolację*, dzięki czemu uzyskujemy obraz gładki bez widocznych bloków.



### *Odbicie symetryczne*

Odbijanie symetryczne obrazu to nic innego jak zmiana przyporządkowania pikseli. Wzory dla odbicia w poziomie i pionie wyglądają następująco:

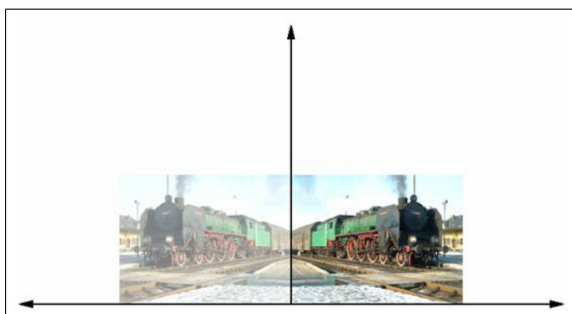
$$x_{nowe} = (x_{max} - 1) - x_{oryg} \quad (23)$$

$$y_{nowe} = (y_{max} - 1) - y_{oryg} \quad (24)$$

gdzie:

$x_{nowe}, y_{nowe}$  - współrzędne piksela po odbiciu,

$x_{max}, y_{max}$  - rozdzielczość obrazu wzdłuż osi  $x$  i  $y$



### 3.5 Interpolacja

#### Interpolacja obrazu przy zmianie rozdzielczości.

Podczas zwiększania rozdzielczości obrazu sąsiednie piksele są od siebie odsuwane. Tak powstałe luki należy zappełnić pikselami o odpowiedniej intensywności.

Interpolacja jest procesem polegającym na wyznaczeniu wartości na podstawie wartości sąsiednich. Istnieje kilka podstawowych algorytmów:

- metoda najbliższego sąsiedztwa
- interpolacja Bi-linearna
- interpolacja Bikubiczna
- Sinc

Metody interpolacji różnią się między sobą sposobem wyznaczania „brakujących” wartości. Jednakże bez względu na metodę interpolacji, wynikowy obraz zawsze będzie w pewnym stopniu rozmyty. *Metoda najbliższego sąsiedztwa*



Jest najprostszą, a zarazem dającą najgorsze rezultaty metodą uzupełniania nieciągłości (np. na skutek skalowania). Piksele są kopiowane w puste miejsca, jakie

powstały podczas rozciągania obrazu. Wygląda to jakby piksele były powiększane. Obraz poddany takiej operacji cechuje się dużymi jednobarwnymi blokami ostrymi krawędziami.

### Interpolacja Bi-linearna, bikubiczna

Interpolacja ta wyznacza wartość danego piksela na podstawie średniej wyznaczonej z otaczających go pikseli. Jest rozszerzeniem interpolacji liniowej i pozwala na interpolację funkcji dwóch zmiennych. Intuicyjnie interpolacja dwuliniowa jest złożeniem dwóch interpolacji liniowych. W celu przeprowadzenia interpolacji dwuliniowej przeprowadza się dwie interpolacje liniowe dla jednego kierunku (np. wzdłuż osi OX w układzie kartezjańskim), a następnie dla tak uzyskanych wartości przeprowadza się interpolację liniową dla drugiego kierunku (osi OY).

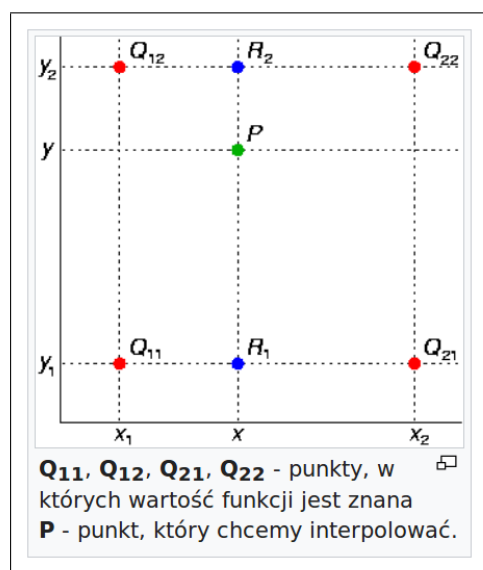
Najpierw przeprowadzana jest interpolacja liniowa wzdłuż osi OX, otrzymujemy:

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \quad \text{gdzie } R_1 = (x, y_1) \quad (25)$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad \text{gdzie } R_2 = (x, y_2) \quad (26)$$

Następnie przeprowadzana jest interpolacja wzdłuż osi OY:

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \quad (27)$$



Jeśli przyjmie się system współrzędnych, w którym znane wartości funkcji  $f$  znajdują się w punktach o współrzędnych  $Q_{11}(0,0)Q_{12}(0,1)Q_{21}(1,0)Q_{22}(1,1)$  wtedy wzór na interpolację upraszcza się do postaci:

$$f(x, y) \approx f(0,0)(1-x)(1-y) + f(1,0)x(1-y) + f(0,1)(1-x)y + f(1,1)xy \quad (28)$$



Metoda ta daje znacznie lepsze rezultaty niż metoda najbliższego sąsiedztwa. Obraz interpolowany jest nieco rozmazany. Interpolacja bikubiczna oparta jest na tym samym mechanizmie, co bilinearna z tą różnicą, że wykorzystuje piksele z większego otoczenia. Daje to nieco lepsze wyniki, okupione jednak zwiększonym zapotrzebowaniem na moc obliczeniową.

### 3.6 Filtrowanie obrazów

Zastosowanie filtrów w przetwarzaniu obrazów oznacza, że do obliczenia nowej wartości punktu brane są pod uwagę wartości punktów z jego otoczenia. Każdy piksel z otoczenia wnosi swój wkład - wagę podczas przeprowadzania obliczeń. Wagi te zapisywane są w postaci maski. Typowe rozmiary masek to 3 x 3, 5 x 5 bądź 7 x 7. Rozmiary masek są z reguły nieparzyste ponieważ piksel na środku reprezentuje piksel dla którego wykonywana jest operacja przekształcania filtrem. Przeanalizujmy filtrację na podstawie filtra o masce 3 x 3.

$f_{-1,-1}$	$f_{0,-1}$	$f_{1,-1}$
$f_{-1,0}$	$f_{0,0}$	$f_{1,0}$
$f_{-1,1}$	$f_{0,1}$	$f_{1,1}$

Wówczas nową wartość składowej punktu  $a$  o współrzędnych  $(i, j)$  obliczymy według następującego wzoru. Najpierw obliczymy sumę ważoną składowej punktu i wszystkich sąsiadów zgodnie z wagami wskazanymi przez maskę filtra.

$$s = f_{-1,-1} \cdot a_{i-1,j-1} + f_{0,-1} \cdot a_{i,j-1} + f_{1,-1} \cdot a_{i+1,j-1} + f_{-1,0} \cdot a_{i-1,j} + f_{0,0} \cdot a_{i,j} + f_{1,0} \cdot a_{i+1,j} + f_{-1,1} \cdot a_{i-1,j+1} + f_{0,1} \cdot a_{i,j+1} + f_{1,1} \cdot a_{i+1,j+1} \quad (29)$$

Tak otrzymaną sumę należy podzielić przez sumę wszystkich wag maski, jeżeli jest ona różna od 0. Ten proces normalizacji wartości składowej punktu zapobiegne zmianie jasności przetwarzanego obrazu.

$$a'_{i,j} = \frac{s}{f_{-1,-1} + f_{0,-1} + f_{1,-1} + f_{-1,0} + f_{0,0} + f_{1,0} + f_{-1,1} + f_{0,1} + f_{1,1}} \quad (30)$$

Filtrację przeprowadza się osobno dla każdej składowej obrazu. Zatem jeżeli mamy obraz reprezentowany w modelu RGB, to wówczas będziemy wykonywać oddzielne przekształcenia dla składowej R, G oraz B. Jak łatwo zauważyć próba zastosowania filtracji dla punktów położonych na krawędzi obrazu, prowadzi do sytuacji, w której maska "wystaje" poza przetwarzany obraz. Istnieje kilka sposobów obejścia tego problemu. Jednym z nich jest pominięcie procesu filtracji dla takich punktów, innym jest zmniejszenie obrazu po filtracji o punkty, dla których proces ten nie mógł być wykonany. Kolejnym sposobem jest dodanie do filtrowanego obrazu zduplikowanych pikseli znajdujących się na jego brzegu.

Filtry dolnoprzepustowe (*ang. low-pass*) przepuszczają elementy obrazu o małej częstotliwości. Elementy o wysokiej częstotliwości (szumy, drobne szczegóły) są natomiast tłumione bądź wręcz blokowane. Wynikiem działania takich filtrów jest zredukowanie szumów, w szczególności gdy jest on jedno, dwupikselowy ale również wygładzenie i rozmycie obrazu. Poniżej znajduje się przykładowy obraz przed (po



lewej) i po zastosowaniu filtra dolnoprzestupowego: Filtr uśredniający - jest podstawowym filtrem dolnoprzestupowym, jego wynikiem jest uśrednienie każdego piksła razem ze swoimi ośmioma sąsiadami.

1	1	1
1	1	1
1	1	1

### Literatura

1. T. Zieliński, Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań, WKŁ, Warszawa, 2007