

## ROZDZIAŁ 12. KOMPRESJA FRAKTALNA

Spośród wielu technik kompresji, schemat koderów fraktalnych wyróżnia się przede wszystkim oryginalnością podstaw teoretycznych, jak również wyjątkowo szeroką gamą możliwości realizacyjnych poszczególnych elementów składowych procesu kompresji. Kompresja fraktalna dotyczy zasadniczo obrazów i jest w pewnym stopniu przeniesieniem idei wektorowej kwantyzacji na grunt pojedynczego obrazu poprzez przybliżanie kolejnych fragmentów innymi fragmentami tego samego obrazu. W algorytmie kompresji formuje się wektory danych (bloki), które są sukcesywnie aproksymowane wektorami podobnymi, generowanymi z obrazu oryginalnego, według określonej metody, będącymi składowymi globalnego modelu całego obrazu. Parametry tych przybliżeń tworzą model całego obrazu, sporządzany w trakcie procesu kompresji i stanowią wyjściową reprezentację kodowanego zbioru danych o charakterze obrazowym. W rozdziale tym przedstawiono jedynie wybrane elementy teorii budowy struktur fraktalnych, która mogą pomóc zrozumieć ogólną koncepcję leżącą u podstaw działania koderów fraktalnych. Podstawowa charakterystyka algorytmów kompresji fraktalnej została uszczegółowiona schematem blokowym realizacji praktycznej przykładowego kodera i uzupełniona wybranymi aspektami zagadnienia optymalizacji takiego schematu.

### 12.1. Wprowadzenie

Obiekty występujące w przyrodzie mają pewne cechy samopodobieństwa, stąd też naturalne obrazy zawierają dość liczne fragmenty samopodobne i wzajemnie podobne. Rozwinięta w ostatnich latach geometria fraktalna [1] dostarczyła precyzyjne narzędzia do modelowania takich obiektów, co stało się podstawą idei kompresji fraktalnej (ang. fractal compression - FC) [2], konsekwentnie udoskonalanej w pracach Barnsleya, Jacquina i Beaumonta [3][4][5]. Opiera się na przejściu od tradycyjnych w przypadku teorii fraktali obiektów samopodobnych do poszukiwania różnych fragmentów obrazu, które są w przybliżeniu wzajemnie podobne. Określanie wzajemnego podobieństwa wybranych fragmentów obrazu z pewną skończoną dokładnością, a następnie zastępowanie fragmentów oryginalnych - przybliżonymi jest formą kwantyzacji i stanowi o stratności tej metody kompresji.

Podobieństwo to jest wykrywane nie tylko w relacji fragment - podfragment, ale też w relacji fragment - inny fragment. Wymaga to uogólnienia koncepcji iteracyjnego systemu funkcji (IFS), który modeluje obiekty samopodobne, a więc uwzględnia relację podobieństwa typu fragment - podfragment tak, by objąć także podobieństwo fragment - inny fragment i jednocześnie nie utracić dobrych cech algorytmicznych. Takim uogólnieniem jest lokalny iteracyjny system funkcji (LIFS), czy też przedstawiona przez Skarbka koncepcja paczłorku [6].

Kompresja fraktalna danego obrazu polega na konstrukcji operatora zwężającego  $F$  (tj. operatora fraktalnego) działającego w przestrzeni obrazów (przestrzeń Banacha), którego iteracja od dowolnego początkowego obrazu zbiega szybko do obrazu oryginalnego. Operator fraktalny jest sumą wszystkich wyznaczonych operatorów lokalnych, działających na przestrzeni kompresowanego obrazu. Dla uzyskania wiernej rekonstrukcji obrazu oryginalnego  $I$  należy zapewnić dobrą aproksymację obrazu przez  $F(I)$ , wówczas atraktor  $I^*$  operatora  $F$  jest dobrym przybliżeniem obrazu  $I$ .

Aby wyznaczyć operator fraktalny dla danego obrazu należy najpierw zdekomponować cały ten obraz na fragmenty zwane płatkami (jest to etap podziału obrazu) o

rozłącznych dziedzinach. Płatki to najczęściej bloki, przy czym będziemy je nazywać blokami przeciwdziedziny. Tworzy się także inny zbiór płatków (etap pokrycia obrazu), które zwane są blokami dziedziny. Pokrywają one pole obrazu w możliwie gęsty sposób przy zachowaniu opłacalności czasowej. Bloki te nie są rozłączne. Następnie szuka się podobieństwa pomiędzy płatkami pokrywającymi, traktowanymi jako płatki źródłowe, a płatkami docelowymi ze zbioru płatków utworzonych w procesie podziału obrazu. Podobieństwo rozważane jest względem pewnej ustalonej klasy transformacji. Docelowy płatek jest aproksymowany przez wynik pewnej transformacji z tej klasy, dokonanej na wybranym płatku źródłowym, przy czym wynik tej transformacji powinien być płatkami o dziedzinie identycznej z dziedziną płatka docelowego. Przekształcenie to najczęściej jest separowalne, tj. składa się z dwu niezależnych przekształceń: geometrycznego (dziedziny) oraz wartości (skali szarości), przy czym przekształcenie geometryczne jest przekształceniem afinicznym, a przekształcenia wartości poszukuje się przeważnie w postaci wielomianowej.

Błąd aproksymacji (dopasowania, kwantyzacji) płatka docelowego przez płatek będący transformacją płatka źródłowego określany jest przy użyciu różnych norm (takich jak supremum, błąd średniokwadratowy, mediana, itp.), przy czym minimalna wartość tego błędu świadczy o optymalnej transformacji płatkowej i najlepszym dopasowaniu danych dwóch płatków. Jeśli więc odnajdziemy taki płatek źródłowy, który z całego zbioru płatków pokrycia daje najmniejszy błąd dopasowania z rozpatrywanym płatkami docelowym, staje się on jego przybliżeniem w rekonstruowanym obrazie, a skutecznie zakodowane parametry przekształcenia stanowią skompresowaną postać oryginalnego obrazu.

Optymalizacja procesu kompresji przy pomocy przekształceń fraktalnych może dotyczyć zarówno odpowiedniego sposobu realizacji podziału i pokrycia obrazu, jak również wyboru właściwej normy i metody wyznaczenia optymalnych przekształceń, a także efektywnej techniki kodowania parametrów przekształceń lokalnych.

Zwiększenie stopnia kompresji można uzyskać poprzez zmniejszenie liczby elementów w podziale obrazu, czyli zwiększenie rozmiarów płatków, dodatkową kwantyzację niektórych parametrów przekształcenia, czy też złagodzenie kryterium podobieństwa płatków. Wiąże się to zwykle z redukcją kosztów pamięciowo-obliczeniowych oraz ze zwiększeniem błędu rekonstrukcji. Można także pewne elementy w algorytmie tworzenia operatora fraktalnego uczynić niezależnymi od kompresowanego obrazu (sposób pokrycia, parametry przekształcenia wartości płatków, kolejność kodowania płatków docelowych), a inne dobierać dla danego typu obrazu w zależności od potrzeb aplikacji.

Optymalizuje się także algorytmy kompresji pod kątem szybkości wykonania, gdyż jedną z podstawowych wad technik fraktalnych jest duża czasochłonność obliczeń koniecznych przy kompresji obrazu, podczas gdy czas dekompresji w najlepszych rozwiązaniach jest dziesięciokrotnie krótszy [6]. Usprawnienie to polega np. na ograniczeniu obszaru poszukiwań płatka podobnego jedynie do sąsiedztwa o pewnym promieniu wokół płatka docelowego. Może to jednak prowadzić w niektórych przypadkach do znacznych zniekształceń w obrazie rekonstruowanym (nie zawsze fragmenty podobne występują blisko siebie w obrazie).

Technika FC, jakkolwiek stworzona na podbudowie innej teorii, w zasadach praktycznej realizacji jest zbliżona do VQ. W obu tych metodach szuka się przybliżeń poszczególnych fragmentów obrazu według pewnego kryterium podobieństwa (choć w różny sposób), a wyjściowym kodem są wskaźniki do tych przybliżeń. Efekty są zbliżone, przy czym algorytmy kompresji VQ przy tej samej złożoności są zazwyczaj szybsze, ale ze względu na często używaną stałą (dla danej klasy obrazów) księgę kodów w VQ, algorytmy

FC mogą okazać się skuteczniejsze w przypadku aplikacji wymagających kompresji obrazów o dużej różnorodności.

## 12.2. Matematyczne podstawy kompresji fraktalnej

Fraktalami w opisie mniej formalnym nazywane są samopodobne struktury, określone przez pewien algorytm rekurencyjny, których wymiar nie jest liczbą całkowitą. Bardziej formalnie fraktal jest elementem pewnej przestrzeni, tzw. przestrzeni fraktali, która jest nazywana także przestrzenią Hausdorffa. W rzeczy samej jest ona przykładem przestrzeni Banacha.

Niech  $(X, d)$  będzie dowolną przestrzenią metryczną zupełną (tj. taką, w której ciągi Cauchy'ego mają granice). Jest nią przykładowo przestrzeń  $R^2$  z metryką euklidesową. Oznaczmy przez  $H(X)$  przestrzeń, której elementami są zwarte i niepuste podzbiory przestrzeni  $X$ . Suma dwóch elementów tej przestrzeni również należy do przestrzeni  $H(X)$ . Zdefiniujemy w tej przestrzeni tzw. metrykę Hausdorffa w sposób następujący:

**Definicja 12.1.** Wyrażenie

$$h(A, B) = \max\{d(A, B), d(B, A)\} \quad (12.1)$$

jest nazywane **metryką Hausdorffa**.

Przestrzenią fraktali jest  $H(X)$ , w której wprowadzono metrykę  $h$  czyli przestrzeń  $(H(X), h)$ . Jest to przestrzeń metryczna zupełna, do której należą fraktale i na której mogą działać różnego typu operatory i odwzorowania generujące struktury fraktalne.

Aby lepiej zrozumieć ideę tych odwzorowań wprowadzone zostaną pojęcia punktu stałego przekształcenia i odwzorowania zwężającego, by wreszcie sformułować twierdzenie o odwzorowaniach zwężających.

**Definicja 12.2.** Niech  $w : X \rightarrow X$  będzie przekształceniem na przestrzeni  $X$ . Punkt  $x_w \in X$  taki, że  $w(x_w) = x_w$  nazywamy **punktem stałym przekształcenia**  $w$ .

**Definicja 12.3.** Przekształcenie  $w : X \rightarrow X$  na przestrzeni metrycznej  $(X, d)$  zwane jest **odwzorowaniem zwężającym**, jeśli istnieje stała  $0 \leq s < 1$  taka, że

$$d(w(x), w(y)) \leq s \cdot d(x, y), \quad \forall x, y \in X \quad (12.2)$$

Nierówność ta nazywana jest warunkiem Lipschitza, a stała  $s$  - stałą Lipschitza lub inaczej współczynnikiem zwężania danego odwzorowania.

### **Twierdzenie 12.1. O odwzorowaniach zwężających**

Niech  $w : X \rightarrow X$  będzie odwzorowaniem zwężającym na przestrzeni metrycznej zupełnej  $(X, d)$ . Wtedy  $w$  posiada dokładnie jeden punkt stały  $x_w \in X$ , a ponadto dla każdego punktu  $x \in X$  ciąg  $\{x_n\}$  taki, że  $x_1 = x$ ,  $x_{i+1} = w(x_i)$ ,  $i = 1, 2, \dots, n-1$ , zbiega do punktu stałego, czyli

$$\lim_{n \rightarrow \infty} \{x_n\} = x_w, \quad \forall x \in X \quad (12.3)$$

Twierdzenie to zawiera fundamentalną zasadę leżącą u podstaw algorytmów tworzenia fraktali, która została wykorzystana również w metodach fraktalnej kompresji. Wystarczy zdefiniować odpowiednie przekształcenie zwężające, którego punktem stałym jest jakiś obiekt przestrzeni  $X$  (płaszczyzny), który chcemy zakodować. Wówczas zaczynając

dekompresję od dowolnego elementu tej przestrzeni po kilku iteracjach uzyskamy obraz tej struktury. Koder szuka właściwego odwzorowania, którego punktem stałym jest wejściowy obiekt, a nową, skompresowaną reprezentacją są parametry tegoż przekształcenia. Pomysł ten oczywiście jest zbyt skromny do opisu złożonej informacji na płaszczyźnie, czyli obrazu. Trzeba więc to narzędzie nieco rozbudować.

### Generacja fraktali przy pomocy iteracyjnego systemu funkcji

Fraktale są zazwyczaj złożonymi strukturami, które są generowane jako atraktor kilku przekształceń zwężających. Zbiór tych przekształceń nazywany jest układem iterowanych odwzorowań (ang. iterated function system - IFS).

**Definicja 12.4.** Układ iterowanych odwzorowań IFS składa się z przestrzeni metrycznej zupełnej  $(X, d)$  wraz ze skończonym zbiorem odwzorowań zwężających  $w_n : X \rightarrow X$ , ze współczynnikami zwężania równymi odpowiednio  $s_n$  dla  $n = 1, 2, \dots, N$ . IFS zapisujemy więc jako  $\{X; w_1, w_2, \dots, w_n\}$  o współczynniku zwężania  $s = \max\{s_1, s_2, \dots, s_n\}$ .

Następujące twierdzenie pozwala określić sposób budowania odwzorowania będącego złożeniem odwzorowań prostszych z wykorzystaniem pojęcia IFS.

#### **Twierdzenie 12.2. O odwzorowaniu według IFS**

Niech  $\{X; w_1, w_2, \dots, w_n\}$  będzie układem iterowanych odwzorowań ze współczynnikiem zwężania  $s$ . Wtedy przekształcenie  $W : H(X) \rightarrow H(X)$  zdefiniowane jako

$$W(A) = \bigcup_{i=1}^n w_i(A) \quad (12.4)$$

dla wszystkich  $A \in H(X)$  jest odwzorowaniem zwężającym w przestrzeni fraktali  $(H(X), h)$  ze współczynnikiem zwężania równym  $s$ . Zatem

$$h(W(A), W(B)) \leq s \cdot h(A, B) \quad (12.5)$$

dla wszystkich  $A, B \in H(X)$ .

Ponieważ odwzorowanie  $W$  jest zwężające, dla dowolnego  $B_0 \in H(X)$  ciąg  $\{B_0, B_1, \dots, B_k, \dots\}$  określony zależnością rekurencyjną  $B_{k+1} = W(B_k)$  jest zbieżny do granicy - punktu stałego  $A_W \in H(X)$ , który jest jedynym rozwiązaniem równania:

$$A_W = W(A_W) = \bigcup_{i=1}^n w_i(A_W). \quad (12.6)$$

Punkt stały  $A_W$ , będący granicą ciągu określonego przez układ iterowanych odwzorowań zwężających nazywany jest atraktorem lub deterministycznym fraktalem.

Przekształcenie  $W$  jest odwzorowaniem zwężającym mogącym wygenerować jako atraktor dużo bardziej złożone obiekty. W praktyce IFS definiowany jest najczęściej przy pomocy przekształceń afinicznych.

#### *IFS przekształceń afinicznych*

Wiele fraktali można wygenerować przy pomocy przekształceń afinicznych na obrazie lub jego części. Przykładowo trójkąt Sierpińskiego o wierzchołkach w punktach  $(0,0)$ ,  $(200,0)$  i  $(40,148)$  można wygenerować za pomocą trzech operatorów działających na całym obrazie:

$$\begin{aligned}
w_1\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
w_2\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix}, \\
w_3\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 20 \\ 74 \end{bmatrix}.
\end{aligned} \tag{12.5}$$

Przekształcenia geometryczne służące do generacji fraktala można zapisać ogólnym wzorem operatora działającego na obrazie:

$$w_i\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}. \tag{12.6}$$

Aby generator fraktali był bardziej użyteczny, należy dodać do tak skonstruowanego operatora jeszcze przekształcenie skali szarości. Przekształcenie to może przyjmować bardzo złożone postaci, jednak w praktycznych zastosowaniach kosztownych obliczeniowo algorytmów fraktalnej kompresji wykorzystywane są proste rozwiązania funkcji wielomianowej  $\Sigma: R \rightarrow R$  działającej na dyskretnym zbiorze wartości funkcji jasności danego obrazu  $S$  w sposób następujący:  $\Sigma(s) = a_n s^n + \dots + a_1 s + a_0$ . Można korzystając z typowego modelu z kontrastem i jaskrawością zdefiniować funkcję  $\Sigma$  jako:  $\Sigma(s) = ks + j$ , gdzie  $k$  - kontrast, a  $j$  - jaskrawość.

Globalny macierzowy zapis operatora z przekształceniami: geometrycznym i skali szarości będzie się wówczas wyrażał następująco:

$$w_i\left(\begin{bmatrix} x \\ y \\ s \end{bmatrix}\right) = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & k_i \end{bmatrix} \begin{bmatrix} x \\ y \\ s \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ j_i \end{bmatrix}. \tag{12.7}$$

### 12.3. Konstrukcja dowolnego obrazu przy pomocy lokalnego IFS

Tak prosty sposób określenia generatorów obrazów fraktalnych może być łatwo wykorzystany w praktyce, jednak nie do celów kompresji. Trudno bowiem wyobrazić sobie, że uda nam się tak zdefiniować przekształcenia fraktalne poprzez IFS, że przy jego pomocy w kolejnych iteracjach uzyskamy coraz wierniejszą postać kompresowanego obrazu. W przypadku bardzo prostego obrazu byłoby to może jeszcze możliwe, jednak zbudowanie IFS generującego dowolny obraz naturalny jako atraktor przekształcenia zwiężającego jest niemożliwe. Musiałby on mieć bowiem wyraźne cechy samopodobieństwa.

#### *Lokalny iteracyjny system funkcji*

Aby skonstruować kompresor fraktalny potrzeba dokonać pewnego rozszerzenia pojęcia IFS. Skoro nie sposób skonstruować obiektu samopodobnego, który będzie przybliżał dany obraz z dowolną dokładnością, to trzeba zrealizować koncepcję przybliżenia odpowiednio prostych fragmentów obrazu innymi podobnymi fragmentami tego samego

obrazu, które występują niemal w każdym obrazie. Idea samopodobieństwa obiektu fraktalnego musi zostać rozszerzona o ideą wzajemnego podobieństwa fragmentów obrazu. Taki pomysł realizuje lokalny iteracyjny system funkcji (ang. local iterated function system - LIFS), który w kolejnych iteracjach przybliża dany obraz poprzez pobranie fragmentów obrazu i przekształcenie ich na nowe fragmenty tego samego obrazu przy pomocy zdefiniowanych odwzorowań. W ten sposób można z mniejszą lub większą skutecznością opisać każdy obraz, a parametry tego opisu stanowią jednoznacznie dekodowalną reprezentację obrazu, na podstawie której można zrekonstruować obraz z bardzo dużą dokładnością.

**Definicja 12.5.** Niech  $(X, d)$  będzie zwartą przestrzenią metryczną, a  $A_i, i = 1, \dots, n$  niepustymi podzbiórmi  $X$ . Niech  $w_i : A_i \rightarrow X$  będą lokalnymi przekształceniami w  $(X, d)$  ze współczynnikami zwężania  $s_i$ . Wtedy

$$\{w_i : A_i \rightarrow X, i = 1, \dots, n\} \quad (12.8)$$

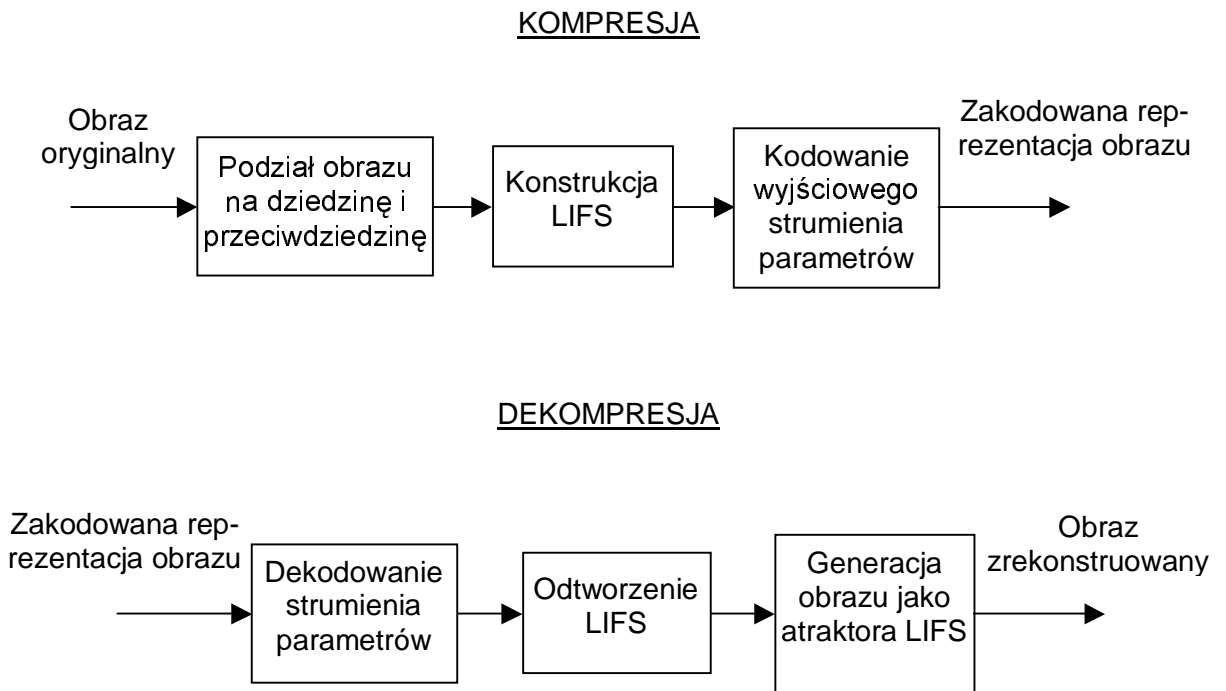
zwane jest lokalnym IFS, a liczba jest współczynnikiem zwężania LIFS.

Dotychczas dziedziną prezentowanych przez nas przekształceń była cała przestrzeń, na którym określony został zbiór wartości funkcji jasności. Działy więc one na całym obrazie. W definiowaniu lokalnego iteracyjnego systemu funkcji zachodzi jedna podstawowa zmiana - dziedziną poszczególnych przekształceń, z których złożony jest LIFS, nie musi być cały obraz. Są one definiowane lokalnie. Mniejszy nośnik poszczególnych odwzorowań zwężających (zamiast całego obrazu) sprawia, że do pełnego zdefiniowania LIFS potrzeba obok określenia wspomnianych parametrów przekształceń, także ich nośnika.

Lokalny IFS może nie mieć atraktora, bądź może mieć wiele różnych atraktorów. Jeżeli  $A$  i  $B$  są atraktorami, to także  $A \cup B$  jest atraktorem co oznacza, że jeśli istnieje atraktor, to istnieje największy, tj. taki, który zawiera wszystkie pozostałe. Nie wchodząc w większe szczegóły teoretyczne wydaje się, że LIFS jako narzędzie modelowania obrazów naturalnych ma wszystkie niezbędne cechy, by na jego podstawie konstruować praktyczne algorytmy kompresji.

## 12.4. Algorytmy praktyczne

Z przedstawiony wyżej rozważań wypływa zasadnicza koncepcja budowy kodera wykorzystującego przekształcenia fraktalne do dekompozycji oryginalnego zbioru danych obrazowych. Elementarny schemat fraktalnego kompresora i dekompresora przedstawia rysunek 12.1. Jak widać z tego rysunku, koder fraktalny jest silnie niesymetryczny. Proces kompresji jest dużo bardziej złożony, przede wszystkim przez czasochłonny algorytm wyznaczania LIFS, polegający na określeniu najbardziej podobnego bloku dziedziny dla każdego z bloków przeciwdziedziny, a następnie określeniu parametrów odwzorowania afinicznego definiującego przekształcenie jednego bloku na drugi. W czasie dekompresji po odczytaniu parametrów LIFS pozostaje jedynie podać dowolny obraz początkowy działaniu LIFS, co po kilku lub kilkunastu iteracjach daje obraz zrekonstruowany. Wierność rekonstrukcji zależy przede wszystkim od przyjętego kryterium podobieństwa bloków oraz procedury poszukiwań bloków najbardziej podobnych. Schemat ten może być oczywiście modyfikowany na wiele różnych sposobów, niektóre z nich zostaną zasygnalizowane w kolejnym podrozdziale.



Rys. 12.1. Podstawowy schemat kompresji i dekompresji fraktalnej obrazów.

Przy konstruowaniu LIFS bardzo istotnym zagadnieniem jest zapewnienie warunku zwężania dla uzyskanego zbioru odwzorowań. Wróćmy do afinicznej postaci przekształcenia  $w_i$ , zdefiniowanego dla  $i$ -tego bloku przeciwdziedziny według równania (12.6) i oznaczmy przez  $G_i$  macierz przekształceń geometrycznych  $G_i = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}$ , a przez  $\eta$  normę miary podobieństwa bloków dziedziny i przeciwdziedziny ( $\eta = 2$  dla błędu średniokwadratowego). Ponadto, należy określić maksymalną liczbę pokryć punktu przestrzeni obrazu  $\pi$  w przyjętym algorytmie podziału obrazu na bloki dziedziny. W podziale przedstawionym na rys. 12.2 wartość  $\pi$  wynosi 4. Warunek dostateczny zwężania wyrażony jest następującą zależnością:

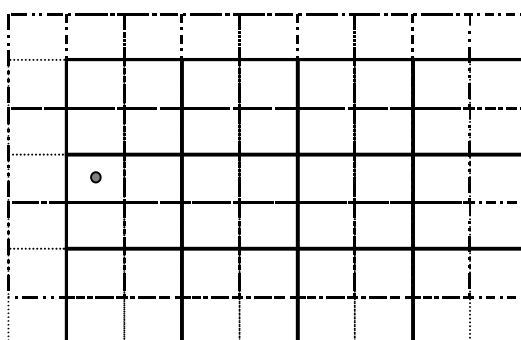
$$k \cdot (g \cdot \pi)^{1/\eta} < 1, \quad (12.9)$$

gdzie  $k = \max_i |k_i|$ ,  $g = \max_i |\det G_i|$ .

Warunek ten jest intuicyjnie zrozumiały, gdyż jest złożeniem kilku oczywistych warunków: na wyznacznik przekształcenia geometrycznego (skalowanie) mniejszy od jedności, kontrast mniejszy od jedności oraz konieczność uwzględnienia nadmiarowości pokrycia dziedziny. Przekształcając warunek zwężania do równoważnej postaci  $k^\eta \cdot d \cdot \pi < 1$ , staje się on jeszcze bardziej zrozumiały.

Podstawiając typowe wartości parametrów z zależności (12.9) jako:  $\pi = 4$ ,  $\eta = 2$ ,  $d=0.5$  otrzymujemy następujący warunek na kontrast lokalnych odwzorowań  $k < \frac{1}{\sqrt{2}}$ . W

praktycznych rozwiązaniach przyjmuje się często stałą wartość kontrastu  $k$  z zakresu  $k \in [0.5; 0.7]$  lub też dobiera się kontrast dla poszczególnych odwzorowań pamiętając o nie przekraczaniu wartości maksymalnej wynikającej z warunku zwięzania. Uzyskana w ten sposób poprawa dopasowania bloków może nie być skompensowana przez konieczność przekazania dodatkowego strumienia danych z wartościami  $k_i$  do dekompresji. W praktycznych rozwiązaniach upraszcza się również kodowanie obrotów definiując pewien zbiór typowych wartości obrotów, lustrzanych odbić, itp. i umieszczając w strumieniu wyjściowym jedynie indeksy tych geometrycznych przekształceń izometrycznych. Przekazane do kodowanego zbioru wyznaczone wartości translacji i jaskrawości dla każdego bloku przeciwdziedziny uzupełniają informację zawartą w wartości kontrastu i indeksie przekształceń geometrycznych, służąc do definiowania LIFS w dekodерze.



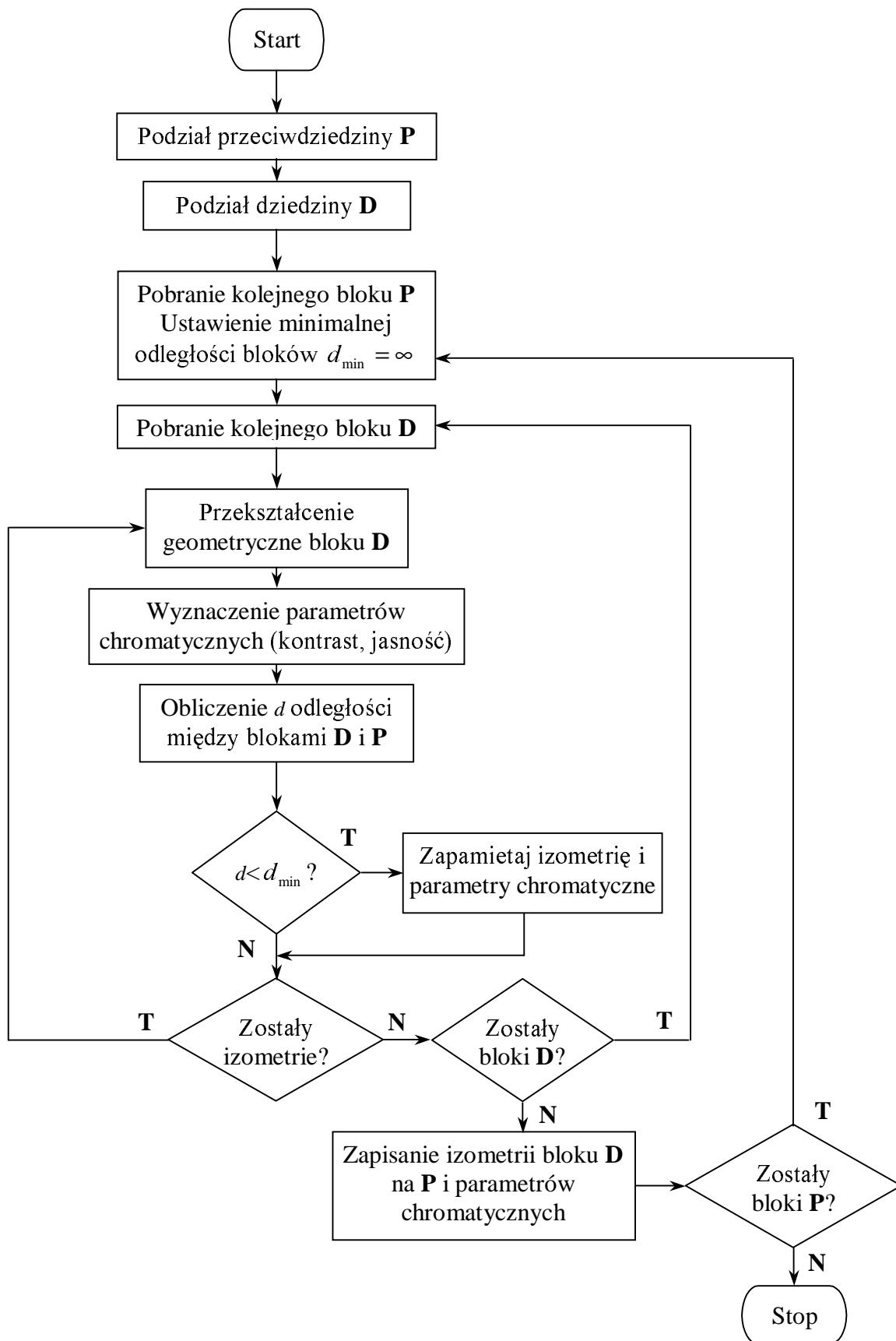
Rys.12.2. Przykład pokrycia dziedziny z zachodzącymi na siebie blokami rozpiętymi przy pomocy czterech siatek przesuniętych względem siebie o połowę szerokości, wysokości lub ukośnie – połowę szerokości i wysokości bloku. Przykładowy punkt oznaczony szarym kółkiem należy do czterech bloków pokrycia (od podstawowej siatki zaznaczonej grubą linią ciągłą oraz od siatek rysowanych cieńszą linią kropkowaną, kreska-kropka i kreska –dwie kropki).

Przykład praktycznego algorytmu kompresora fraktalnego pozwalającego uzyskać znaczącą efektywność kompresji przedstawia rys. 12.3, a schemat algorytmu dekompresji, odtwarzającego iteracyjnie obraz kompresowany przedstawia rysunek 12.4.

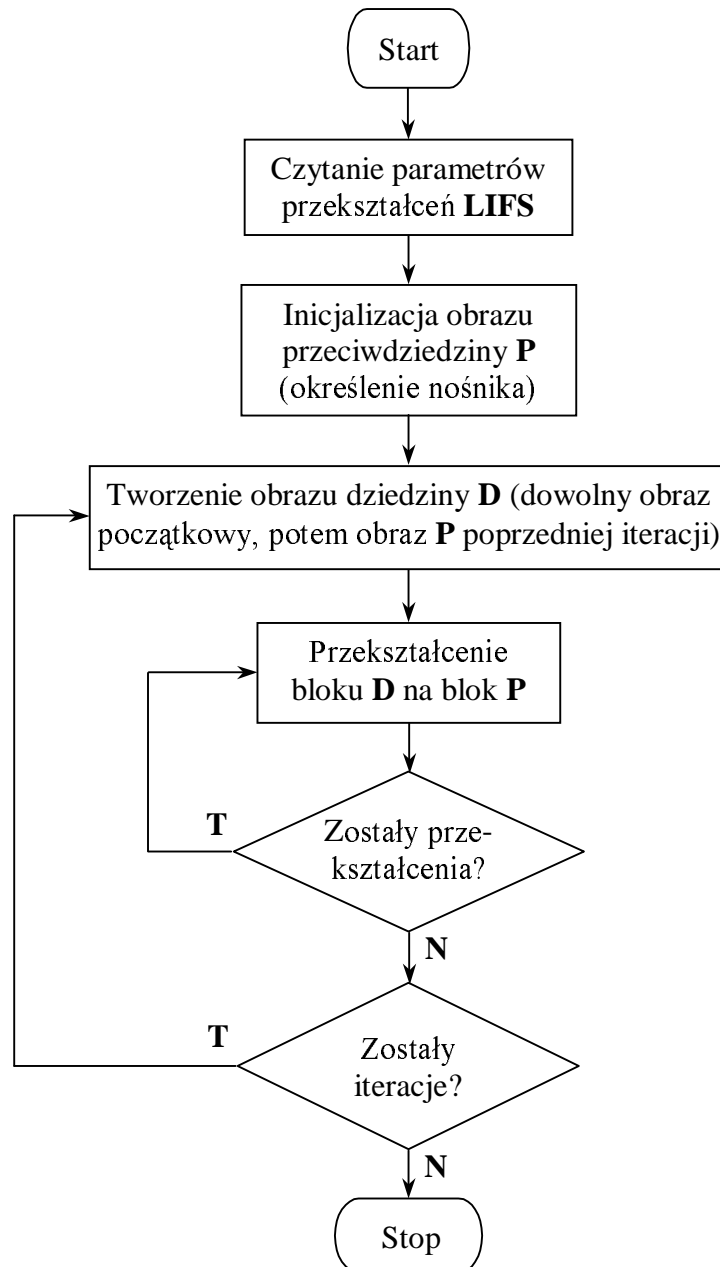
## 12.5. Optymalizacja algorytmu kompresji fraktalnej

Podstawowy schemat kompresji fraktalnej jest czasochłonny. Wyznaczenie LIFS przybliżającego obraz z określoną dokładnością pociąga za sobą konieczność wielokrotnego przeszukiwania bloków dziedziny, obracania tych bloków i określania optymalnego parametru jasności, a następnie określania odległości od bloku przeciwdziedziny, w poszukiwaniu najlepszej aproksymacji każdego bloku. Przykładowo kompresja średniej wielkości obrazu przy pomocy algorytmu transformacyjnego z DCT trwała około sekundy, podczas gdy metodą fraktalną, na tej samej maszynie, kilka minut. Ponadto uzyskiwania skuteczność kodowania nie jest przekonująca w większości zastosowań. Wymaga więc ten algorytm optymalizacji, zarówno w sensie R-D jak i czasu kompresji. Kilka przykładowych modyfikacji zostało przedstawionych w dalszej części tego podrozdziału.





Rys. 12.3 Schemat blokowy fraktalnej metody kompresji.



Rys. 12.4. Schemat blokowy algorytmu dekompresji metody fraktalnej.

### **Dobór wielkości parametrów**

Aby zwiększyć efektywność całego algorytmu należy po pierwsze dobrać odpowiednie wartości parametrów schematu, takich jak kontrast, wielkość bloków podziału i pokrycia, poziom kwantyzacji wartości jasności w stosunku do liczby izometrii, wielkości bloków i kontrastu. Istotny jest także wybór normy do określania odległości pomiędzy blokami oraz algorytmu poszukiwania najbardziej podobnego bloku dziedziny.

Wszystkie te specyfikacje algorytmu fraktalnego wpływają na jakość rekonstruowanego obrazu i długość kodu wyjściowego, jak też na czasochłonność procesu kompresji. Nie bez znaczenia jest także liczba koniecznych iteracji do rekonstrukcji najwierniejszej wersji obrazu oryginalnego przy zadanych parametrach. Jest to więc problem

optymalizacji wieloparametrycznej, praktycznie nierozwiązywalny w sposób globalny. Najczęściej ustala się więc większość parametrów w koderze/dekoderze na stałe regulując stopień kompresji np. poziomem kwantyzacji wartości funkcji jasności. Minimalizuje się w tym przypadku długość strumienia wyjściowego kosztem jakości przybliżenia obrazu oryginalnego, uzyskując ponadto redukcję czasu kompresji. Powstały także algorytmy realizujące tendencje odwrotne. Wprowadza się w nich do strumienia wyjściowego dokładne wartości izometrii i kontrastu dla każdego bloku, adaptacyjnie dobiera się rozmiar a nawet kształt bloku na podstawie analizy statystycznej lokalnych własności obrazu, stosuje lokalnie dobrane różne kryteria podobieństwa bloków, itp. Efektem jest wizualnie bezstratna kompresja obrazu kosztem ogromnej złożoności i czasochłonności algorytmu i najczęściej bardzo ograniczonej skuteczności kompresji w sensie R-D.

Nieznaczną poprawę skuteczności kompresji można także uzyskać poprzez kodowanie wartości parametrów LIFS w oddzielnych strumieniach lub też odpowiednio przemieszanych według ich statystyki. Dobór algorytmu bezstratnego kodowania sprowadza się najczęściej do ustalenia rzędu i postaci modelu statystycznego w koderze arytmetycznym.

### **Podział nośnika**

Postać zarówno podziału, jak i pokrycia wpływa znacząco zarówno na skuteczność kompresji, jak i jej czasochłonność. Kształt płatków, ich wielkość jak i gęstość pokrycia jest kompromisem pomiędzy dokładnością przybliżenia, a możliwością oszczędnego opisanie wybranych przekształceń afinicznych. Obok najczęściej stosowanych kwadratowych bloków, używane są też pokrycia trójkątami czy trapezami. Obok regularnych pokryć i podziałów, stosowana jest też wstępna segmentacja obrazu, np. metodą drzewa czwórkowego, dokonując podziału obrazu na bloki o różnej wielkości na podstawie analizy wierności aproksymacji bloków danej wielkości (mniejsze bloki o uboższej treści można przybliżyć dokładniej), a następnie w zależności od wielkości bloku w podziale obrazu dobierany jest najbardziej podobny blok dziedziny z pokrycia odpowiednio dużymi blokami.

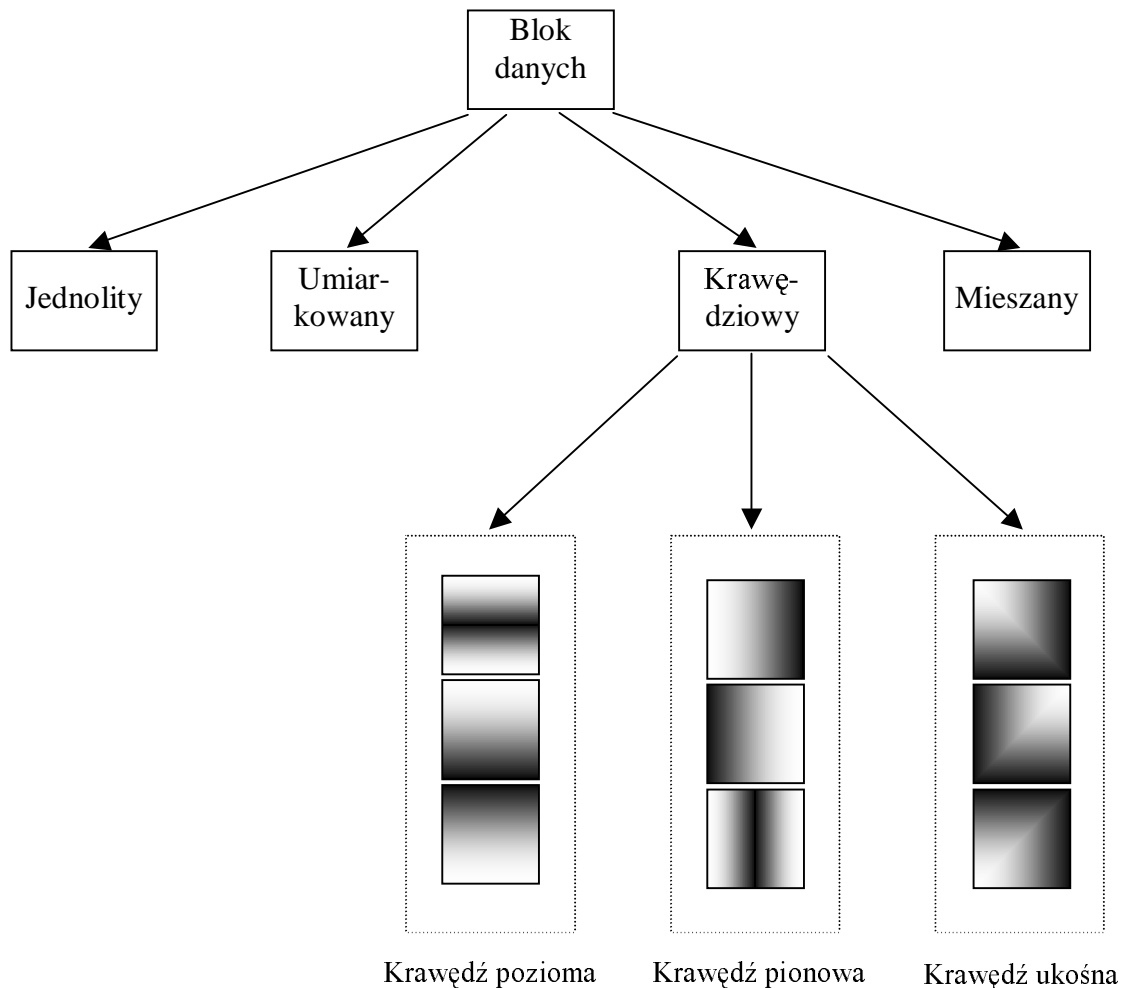
### **Akceleracja algorytmu kompresji**

Najprostszym rozwiązaniem przyspieszającym kompresję jest zmiana warunku w poszukiwaniu najbardziej podobnego bloku dziedziny. Otóż zamiast określania w każdym przypadku odległości minimalnej wystarczy jedynie założyć wstępnie pewną dopuszczalną wartość błędu przybliżenia, czyli dopuszczalną odległość pomiędzy blokami. Jeśli w poszukiwaniu odpowiedniego bloku znajdziemy taki blok, który nie jest odległy bardziej niż założony błąd, wtedy kończymy proces wyznaczania podobnego bloku dziedziny.

Innym sposobem jest zmniejszenie obszaru poszukiwań z całej dziedziny obrazu na pewien obszar wokół położenia bloku przeciwdziedziny. W przypadku, gdy najbardziej podobne fragmenty nie leżą blisko siebie, wpłynie to zapewne na pogorszenie jakości rekonstrukcji. Warto jednak zauważyć, że ilość bitów przeznaczona do zapisu wektorów translacji może być mniejsza, a więc generalnie wpływ na efektywność kompresji w sensie R-D nie musi być negatywny. Oszczędności czasowe mogą być natomiast znaczne.

Trzecia kategoria metod przyspieszania kompresji fraktalnej to różne techniki klasyfikacji bloków. Na rys. 12.5. przedstawiono przykład klasyfikacji krawędziowej [7]. Określono cztery klasy bloków w zależności od obecności i rodzaju krawędzi. Bloki *jednolite* nie mają znaczącego gradientu funkcji jasności, bloki *umiarkowane* posiadają średnią wartość gradientu, natomiast wyraźna krawędź jest charakterystyczna dla trzeciej klasy bloków - *krawędziowych*. Wyróżniono także bloki mieszane, w których występuje znaczący gradient,

ale trudno doszukać się ciągłej krawędzi o rozmiarach istotnych z punktu widzenia wielkości bloku. Ponadto w grupie bloków krawędziowych wyszczególniono także podklasy o krawędziach poziomych, pionowych i ukośnych.

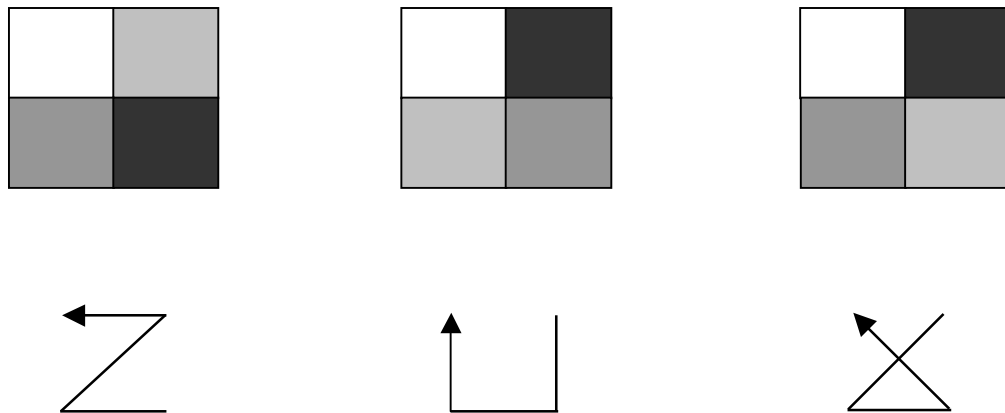


Rys. 12.5. Przykład klasyfikacji krawędziowej przyspieszającej algorytm kompresji fraktalnej.

Inną metodą klasyfikacji, dającą nieco lepsze wyniki, jest klasyfikacja normalna. Każdy blok dzielony jest na cztery równe części, a następnie obliczana jest średnia wartość jasności każdej części. Istotny staje się rozkład zmiany jasności (gradientu) w bloku, czy przebiega z dołu do góry czy z lewej na prawą, w sposób jednolity albo wymieszany. Wyróżnia się tu trzy klasy bloków, jak na rys. 12.6.

Klasyfikacja bloków dziedziny i przeciwdziedziny umożliwia poszukiwanie najbardziej podobnych bloków jedynie w odpowiedniej klasie czy podklasie, co znacznie zmniejsza czasochłonność wyznaczania optymalnych przekształceń. Ponadto w zależności od kategorii można stosować różne metody kwantyzacji informacji zawartej w blokach. Wiadomo bowiem, że bloki o silnych krawędziach są bardzo istotne z punktu widzenia jakości rekonstrukcji, podczas gdy jednolite bloki można rekonstruować z mniejszą dokładnością. Oczywiście klasyfikacja wymaga także pewnych nakładów czasowych, ale

zasadniczo techniki klasyfikacji bloków powodują zmniejszenie czasu kompresji przy zachowaniu jakości rekonstrukcji.



Rys. 12.6. Klasyfikacja normalna, dzieląca bloki na klasy w zależności od kierunku gradientu średniej funkcji jasności, liczonej w czterech częściach bloków. Czarny kolor oznacza najmniejszą wartość średniej, biały największą. Strzałki pokazują kierunek gradientu. Oczywiście taki algorytm klasyfikacji jest niezmienniczy w funkcji obrotu bloku.

### Bibliografia:

1. M.F. Barnsley, *Fractals Everywhere*, Nowy York: Academic Press, 1988.
2. A.E. Jacquin, *A Novel Fractal block-coding technique for digital images*, IEEE Int. Conf. on ASSP, 1990.
3. J.M. Beaumont, *Image Data Compression Using Fractal Techniques*, BT Technology Journal 9(4): 93-108, 1991.
4. A.E. Jacquin, *Image Coding based on Fractal Theory of Iterated contractive images*, 1(1):18-30, IEEE Trans. on Image Proc., 1992.
5. M.F. Barnsley, and M.P. Hurd *Fractal Image compression*, Wellesley: AK Peters Ltd., 1993.
6. W. Skarbek, *Metody reprezentacji obrazów cyfrowych*, Akademicka Oficyna Wydawnicza, 1993.
7. B. Ramamurthi, A. Gersho, *Classified Vector Quantization of Images*, 34(11): 1105-1115, 1986.