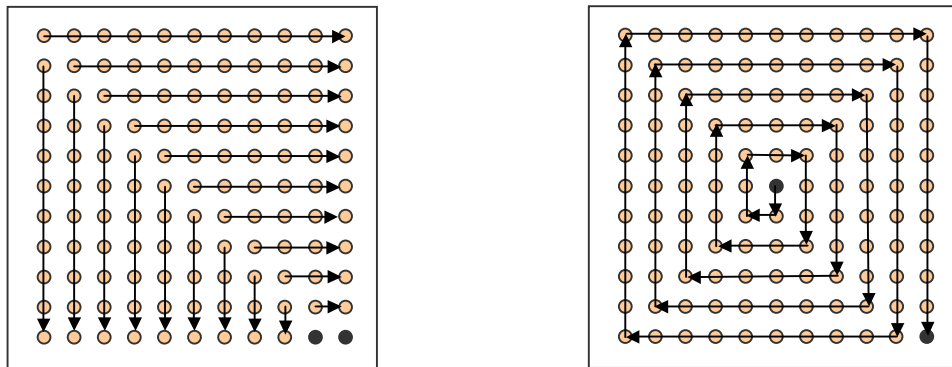


## ROZDZIAŁ 7. BEZSTRATNA KOMPRESJA OBRAZÓW

Ze względu na odmienny charakter danych obrazowych, oprócz opisanych w poprzednich rozdziałach metod bezstratnej kompresji danych stosowane są dodatkowe metody redukcji nadmiarowości przestrzennej, mającej charakter dwuwymiarowy. Nadmiarowość ta występuje zazwyczaj na poziomie istotnym w stosunku do całej redundancji reprezentacji danych obrazowych. Potrzebne są tutaj bardziej wyrafinowane metody predykcji, czy też bardziej rozbudowane wielowymiarowe modele statystyczne metod entropijnych, aby opisać rzeczywiste zależności występujące pomiędzy wartościami pikseli w przestrzeni obrazu. Zostaną one przedstawione w tym rozdziale.

### 7.1. Przeglądanie danych

Jedną z najprostszych technik poprawy efektywności metod kompresji obrazów jest odpowiednie przeglądanie danych w przestrzeni obrazu. Celem jest takie ustawienie danych ze zbioru kodowanego, aby kolejno występowały po sobie zbliżone wartości danych z obszarów obrazu treściowo zależnych (piksele z krawędzi, płaszczyzny o jednorodnej jasności itp.). Tworzony jest w ten sposób nowy strumień wejściowy dla bezstratnego kodera danych (np. entropijnego, słownikowego) o większej podatności na kompresję. Rozważmy przykładowo zagadnienie skutecznej archiwizacji obrazu o dominujących pasach pionowych w całej jego przestrzeni. Ustawiając wartości pikseli w jeden strumień poprzez dopisywanie danych kolejno wierszami obrazu uzyskamy znacznie gorszą skuteczność niż w przypadku kierunku przeglądania według kolumn. Znajdą się wtedy obok siebie silnie skorelowane (podobne) wartości pikseli zwiększając znacznie efektywność procesu kodowania.



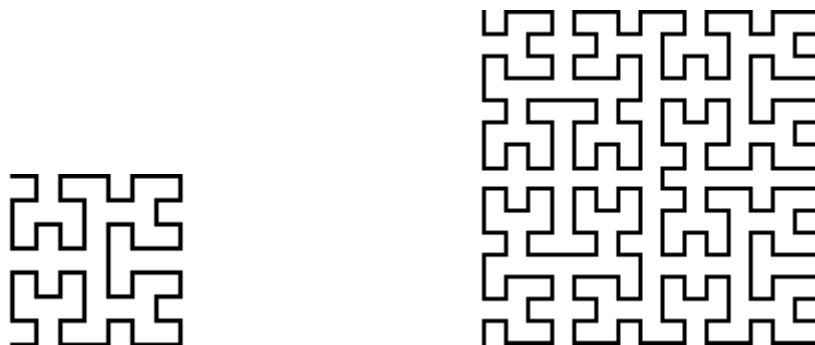
Rys. 7.1. Sposoby przeglądania danych obrazowych: naprzemienne (na lewo), po okręgach (na prawo)

Obok dwu narzucających się sposobów skanowania pikseli po wierszach i po kolumnach można zastosować bardziej złożone algorytmy przeglądania danych. Przykładowo nie chcąc wyróżniać żadnego z kierunków, można przeglądać dane naprzemienne, tj. wiersz, potem krótsza o jeden element kolumna, krótszy o jeden wiersz, krótsza o 2 kolumna itd. Obrazuje to rysunek 7.1. Uzasadnionym także wydaje się przeglądanie pikseli po coraz większych okręgach zataczanych wokół centralnej części obrazu (rys. 7.1), szczególnie przy kompresji obrazów naturalnych o najbardziej istotnej treści umieszczonej w centrum obrazu, np. telekonferencyjne obrazy twarzy. Koliste kształty prezentowanych obiektów są na tyle

powszechnie w różnego typu obrazach, że poszukiwanie rozwiązań uwzględniających częste zmiany kierunku przeglądania zdaje się w pełni uzasadnione. Jedną z takich metod została przedstawiona poniżej.

### Krzywa Hilberta

Przeoglądanie obrazu po krzywej Hilberta, zwanej niekiedy w literaturze także krzywą Peano [1], wydaje się uwzględniać różnorodność kształtów obiektów występujących w obrazie nie wyróżniając żadnego z kierunków, nawet lokalnie. Przykładową postać krzywej Hilberta dla bloków obrazu o rozdzielczości  $8 \times 8$  i  $16 \times 16$  przedstawia rys. 7.2.



Rys. 7.2. Krzywa Hilberta dla bloku obrazu o wymiarach  $8 \times 8$  i  $16 \times 16$ .

Aby zbudować algorytm przeglądania po tak złożonej krzywej można wykorzystać operator zdefiniowany poniżej.

Niech  $\{O_{x,y}; 0 \leq x, y \leq N-1\}$  oznacza kwadratowy obraz rozumiany jako dwuwymiarowa realizacja pola losowego. Przekształcenie tego pola w jednowymiarowy ciąg wartości tegoż pola  $\{H_n; 0 \leq n \leq N^2 - 1\}$  opisane jest następująco:

$$H_0 = O_{p_0} \quad \text{oraz} \quad H_n = O_{p_{n-1} + \Delta p_n} \quad \text{dla} \quad 1 \leq n \leq N^2 - 1, \quad (7.1)$$

gdzie  $p_n$  i  $\Delta p_n$  są wektorami definiowanymi poniżej jako funkcje współrzędnych obrazu:  $p_n = f_n(x, y)$ ,  $0 \leq n \leq N^2 - 1$  oraz  $\Delta p_n = g_n(x, y)$ ,  $1 \leq n \leq N^2 - 1$ , przy czym  $0 \leq x, y \leq N - 1$ . I tak  $p_n$  dla  $0 \leq n \leq N^2 - 1$  jest zbiorem wektorów, które definiują aktualną pozycję w rekursywnym przeglądzie pikseli obrazu, a  $\Delta p_n$  oznacza przyrost położenia w każdym kroku algorytmu. Można zapisać:

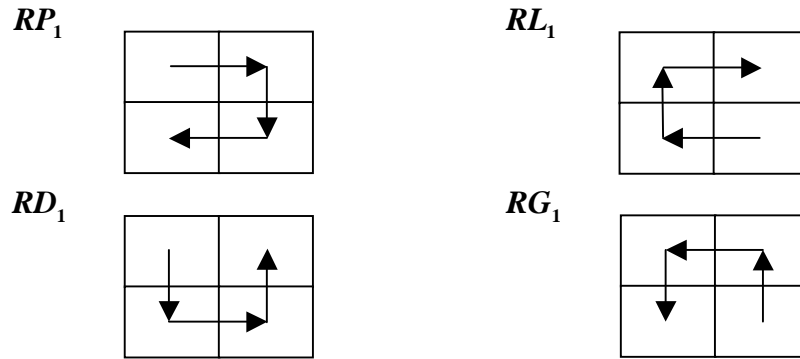
$$p_n = p_{n-1} + \Delta p_n = p_0 + \sum_{i=1}^n \Delta p_i. \quad (7.2)$$

Jeśli założymy początkową pozycję w algorytmie przeglądania  $p_0$  w punkcie o współrzędnych  $(0,0)$ , wtedy  $p_n = \sum_{i=1}^n \Delta p_i$ . Alfabet możliwych przyrostów położenia  $\Delta p_n$  jest następujący:

$$\Delta p_n = \begin{cases} \Delta p^P = [1,0] & \text{- w prawo} \\ \Delta p^L = [-1,0] & \text{- w lewo} \\ \Delta p^D = [0,1] & \text{- w dół} \\ \Delta p^G = [0,-1] & \text{- w górę} \end{cases}, \quad 1 \leq n \leq N^2 - 1. \quad (7.3)$$

Na podstawie tych czterech ruchów podstawowych można zdefiniować cztery analogiczne, bardziej złożone (wyższego rzędu) ruchy jak niżej (patrz rys. 7.3):

$$\begin{aligned} RP_1 &= \Delta p^P + \Delta p^D + \Delta p^L, & (\text{w prawo}) \\ RL_1 &= \Delta p^L + \Delta p^G + \Delta p^P, & (\text{w lewo}) \\ RD_1 &= \Delta p^D + \Delta p^P + \Delta p^G, & (\text{w dół}) \\ RG_1 &= \Delta p^G + \Delta p^L + \Delta p^D. & (\text{w górę}) \end{aligned} \quad (7.4)$$



Rys.7.3. Cztery podstawowe ruchy pierwszego rzędu (prawo, lewo, dół, góra) według równań (7.4).

Operacje dowolnego rzędu  $n$  ( $n=2^q$ ) w czterech elementarnych kierunkach określone są rekursywnie poprzez następujące równania:

$$\begin{aligned} RP_n &= Q_{n,q1} RD_{n/2} + Q_{n,q2} RP_{n/2} + Q_{n,q3} RG_{n/2} + Q_{n,q4} RP_{n/2} \\ RL_n &= Q_{n,q1} RL_{n/2} + Q_{n,q2} RD_{n/2} + Q_{n,q3} RL_{n/2} + Q_{n,q4} RG_{n/2} \\ RD_n &= Q_{n,q1} RP_{n/2} + Q_{n,q2} RL_{n/2} + Q_{n,q3} RD_{n/2} + Q_{n,q4} RD_{n/2} \\ RG_n &= Q_{n,q1} RG_{n/2} + Q_{n,q2} RG_{n/2} + Q_{n,q3} RP_{n/2} + Q_{n,q4} RL_{n/2}, \end{aligned} \quad (7.5)$$

gdzie  $Q_{n,qi}$  dla  $i=1, 2, 3, 4$  oznacza macierze  $n \times n$  składające się z czterech kwadrantów określających region operacji. I tak przykładowo pierwszy kwadrant odpowiada regionowi 'lewo-góra' i wygląda następująco:

$$Q_{n,q1} = \begin{bmatrix} 1\dots 1 & 0\dots 0 \\ \dots & \dots \\ 1\dots 1 & 0\dots 0 \\ 0\dots 0 & 0\dots 0 \\ \dots & \dots \\ 0\dots 0 & 0\dots 0 \end{bmatrix}. \quad (7.6)$$

Kwadrant drugi  $Q_{n,q2}$  określający region 'prawo-góra', trzeci  $Q_{n,q3}$  - 'lewo-dół', a czwarty  $Q_{n,q4}$  - 'prawo-dół' zdefiniowane są analogicznie.

Ostatecznie więc krzywa Hilberta jest jednoznacznie zdefiniowana przez wektor pozycji  $P$  określony jak niżej:

$$P = [p_1 \dots p_n] = \begin{cases} RP_n & \text{dla nieparzystych } q = \log_2 n \\ RD_n & \text{dla parzystych } q \end{cases} \quad (7.7)$$

Sposób generacji krzywej Hilberta, także dla obrazów o macierzy niekwadratowej, można znaleźć u Skarbka [2].

Często stosuje się też operator różnicowy Hilberta -  $HD$  poprawiający wyraźnie skuteczność kompresji metod słownikowych. Polega on na przeglądaniu obrazu według krzywej Hilberta i liczeniu współczynników  $HD$  obrazu jako różnicy wartości aktualnej i poprzedniej. Operator  $HD$  jest więc zdefiniowany następująco:

$$HD_0 = O_{p_0} \quad \text{oraz} \quad HD_n = O_{p_{n-1} + \Delta p_n} - O_{p_{n-1}} \quad \text{dla} \quad 1 \leq n \leq N^2 - 1 \quad (7.8)$$

Przekształcenie  $HD$  pozwala zmniejszyć korelację przestrzenną pomiędzy danymi obrazu, przez co umożliwia efektywniejszą kompresję. Lempel i Ziv [3] wykazali, że entropia dowolnego skończonego obrazu może być asymptotycznie osiągnięta przez algorytm jednowymiarowej kompresji słownikowej poprzedzonej skanowaniem po krzywej Hilberta danych obrazowych. Aby przedstawić korzyści wynikające z zastosowania różnych sposobów przeglądania danych w algorytmach kompresji wykonano eksperyment porównawczy. Opisuje go przykład 7.1

**PRZYKŁAD 7.1.** Kodowanie obrazu z wykorzystaniem różnych metod przeglądania danych.

Grupę standardowych obrazów naturalnych oraz obraz medyczny (patrz rys. 7.4) poddano kompresji bezstratnej przy pomocy kodera arytmetycznego rzędu 1 z zastosowaniem różnych technik przeglądania danych we wstępnej fazie algorytmu. Wyniki porównań efektywności tak skonstruowanych metod kompresji przedstawiono w tabeli 7.1.

Tabela 7.1. Porównanie skuteczności kompresji obrazów kodera arytmetycznego rzędu 1, dla którego strumień wejściowy był kształtowany przy pomocy różnych algorytmów przeglądania obrazu. Tabela zawiera wartości średnich bitowych w bitach na piksel - bpp.

Technika przeglądania\Obraz	Lena	Barbara	Goldhill	Obraz CT
Wiersz-kolumna	5,28	6,40	5,50	1,98
Kolumna-wiersz	4,87	5,86	5,45	1,90
Naprzemiennie	5,03	6,22	5,42	1,98
Po okręgach	4,97	6,04	5,38	<b>1,77</b>
Hilberta	5,06	6,17	5,47	1,93
Operator różnicowy HD	<b>4,79</b>	<b>5,79</b>	<b>5,31</b>	1,89

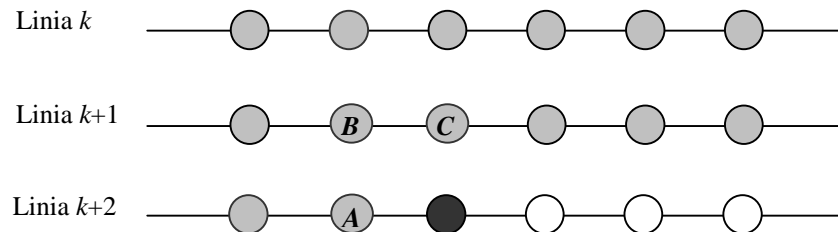
Na podstawie otrzymanych wyników można stwierdzić, iż sposób przeglądania danych w niektórych przypadkach zauważalnie wpływa na efektywność kompresji całego algorytmu. W stosunku do typowego przeglądania obrazu wiersz po wierszu uzyskano zmniejszenie średniej bitowej na wyjściu nawet ponad 10% w niektórych przypadkach. Wynika to najczęściej ze zdominowania scen testowanych obrazów przez struktury o charakterze pionowym bądź kolistym. Techniki nie wyróżniające żadnego z kierunków zwiększają nieco skuteczność kompresji, oczywiście jedynie w przypadkach kiedy dominacja żadnego z nich nie jest wyraźna. Prosty operator różnicowy, będący namiastką predykcji, skojarzony z algorytmem przeglądania pikseli wzdłuż krzywej Hilberta pozwolił dodatkowo zmniejszyć wartość średniej bitowej reprezentacji wyjściowej w niewielkim stopniu.



Rys. 7.4. Obrazy testowe (Lena: góra-lewo, Barbara: góra-prawo, Goldhill: dół-lewo, CT: dół-prawo) użyte w badaniach efektywności kompresji różnych metod bezstratnych. Wszystkie obrazy są jednobajtowe, o wymiarach 512×512 pikseli.

## 7.2. Metody predykcyjne

Predykcja 2-D (dwuwymiarowa) daje zazwyczaj lepsze rezultaty przy kompresji obrazów w stosunku do metod 1-D (jednowymiarowych), gdyż pozwala usunąć często dość silne zależności pomiędzy wartościami sąsiednich pikseli. Wykorzystując korelacje w zdefiniowanych lokalnie kontekstach techniki predyktywnego kodowania próbują określić domniemaną wartość danego piksela na podstawie wartości otaczających go pikseli. W predykcji liniowej wykorzystuje się najbliższe sąsiedztwo piksela, przy czym rząd predykcji jest równy liczbie pikseli, których wartości służą do wyznaczenia przybliżenia. Z przeprowadzonych w poprzednim rozdziale rozważań wynika zasadniczo, że wyższy rząd predykcji daje efektywniejszą kompresję. Okazuje się jednak, że w przypadku większości obrazów stosowanie modeli predykcyjnych wysokich rzędów nie daje oczekiwanych rezultatów. Z badań przeprowadzonych na obrazach ra diograficznych wynika, iż zwiększanie rzędu predykcji powyżej 3 daje znikome efekty. Potwierdzają to też rozważania teoretyczne przedstawione poniżej. Na rys. 7.5 zaprezentowano taki klasyczny kontekst trzeciego rzędu.



Rys. 7.5. Przedstawiono fragmenty trzech kolejnych linii obrazu. Kółka szare oznaczają piksele już zakodowane, białe - piksele jeszcze nie kodowane, a kółko czarne to piksel aktualnie kodowany. W jego bezpośrednim sąsiedztwie znajdują się oznaczone literami A, B, C piksele, które stanowią kontekst rzędu 3 dla modeli predykcji liniowej.

Przykładowe postacie prostych schematów predykcji liniowej DPCM wyglądają następująco ( $f_A, f_B, f_C$  są wartościami punktów sąsiednich w poziomie, po skosie i w pionie jak na rys. 7.5):

$$\begin{aligned}\hat{f} &= 0.50 \cdot f_A + 0.50 \cdot f_C \\ \hat{f} &= 0.90 \cdot f_A - 0.81 \cdot f_B + 0.90 \cdot f_C \\ \hat{f} &= 0.75 \cdot f_A - 0.50 \cdot f_B + 0.75 \cdot f_C\end{aligned}\quad (7.9)$$

Kształt kontekstu winien odzwierciedlać kształt struktur dominujących w obrazie, przy czym należy pamiętać o przestrzeganiu zasady przyczynowości. Poszczególne współczynniki modelu predykcji można wyznaczać stosując kryterium minimalnego błędu średniokwadratowego.

Zagadnienie predykcji 2-D zostanie przedstawione nieco bardziej formalnie. Oznaczmy przez  $F=F[f(x,y)]$  macierz obrazu o rozmiarach  $M \times N$ , gdzie element  $f(x,y)$  przedstawia wartość funkcji jasności znajdującą się w  $x$ -tej kolumnie i  $y$ -tym wierszu macierzy, przy czym  $x=0, \dots, M-1, y=0, \dots, N-1$ . W predykcji liniowej wartość  $f(x,y)$  jest przybliżana jako liniowa kombinacja wartości znajdujących się w pewnym sąsiedztwie  $\Omega$  (jest to kontekst modelu predykcji) w następujący sposób:

$$\hat{f}(m,n) = \sum_{p,q \in \Omega} \hat{\alpha}(m,n,p,q) \cdot f(p,q), \quad (7.10)$$

gdzie  $\hat{\alpha}(m,n,p,q)$  oznacza współczynniki predykcji, które mogą ulegać zmianie w funkcji położenia  $(x,y)$ , np. w algorytmach adaptacyjnych. Tworzona jest nowa macierz  $E = E[e(x,y)]$  zawierająca błędy predykcji odpowiednich wartości macierzy  $F$  wyznaczone z zależności:  $e(x,y) = f(x,y) - \hat{f}(x,y)$ , przy czym  $e(\cdot), f(\cdot), \hat{f}(\cdot) \in Z$ . Wyraźnie mniej skorelowane wartości macierzy  $E$  jako pośrednia postać oryginalnego zbioru danych są następnie kodowane.

Jeśli sąsiedztwo  $\Omega$  danej wartości macierzy  $F$  obejmuje jedynie wartości poprzedzające w strumieniu danych kodowanych, wówczas mamy do czynienia z modelem przyczynowym, koniecznym z punktu widzenia poprawności procesu dekompresji. W przykładach z rys. 7.5 oraz rys. 6.4 w poprzednim rozdziale widać kształtowanie takich modeli przyczynowych. Ich wadą, szczególnie widoczną w modelowaniu danych obrazowych, jest nie uwzględnienie znacznej części rzeczywistego sąsiedztwa punktu kodowanego, otaczającego punkt z 'drugiej strony', a co za tym idzie nie wykorzystanie silnych zależności pomiędzy danymi i słabsza redukcja nadmiarowości.

W przypadku predykcji przyczynowej oryginalne wartości są rekonstruowane na podstawie wartości macierzy  $E$  (po odtworzeniu wartości przewidywanej  $\hat{f}(x,y)$  ze zrekonstruowanych już wartości oryginalnych sąsiedztwa, przy pomocy identycznego jak w procesie kodowania modelu predykcji) według następującej zależności:  $f(x,y) = \hat{f}(x,y) + e(x,y)$ .

Współczynniki predykcji  $\hat{\alpha}(x,y,p,q)$  są najczęściej dobierane według kryterium minimalnego błędu średniokwadratowego. Załóżmy, że macierz  $F$  jest realizacją jednorodnego stochastycznego rozkładu (warunek stacjonarności) o wartości średniej  $\bar{f}$  z separowalną funkcją autokorelacji postaci  $R(m,n) = \bar{f}^2 + \sigma^2 \rho_{poz}^{|m|} \rho_{pion}^{|n|}$ , gdzie  $\sigma^2$  jest wariancją, zmienne  $m=x-p$  i  $n=y-q$  określają przemieszczenia poziome i pionowe, a  $\rho$  oznacza współczynnik korelacji ( $0 \leq \rho \leq 1$ ), opisujący statystycznie lokalny charakter danych w obrazie, odpowiednio w kierunku poziomym  $\rho_{poz}$  i pionowym  $\rho_{pion}$ . Taki model obrazu to często wykorzystywany w technikach kompresji model Markowa. Typowa wartość współczynnika korelacji dla obrazów naturalnych wynosi ponad 0.9 w obu kierunkach. Określmy także sąsiedztwo  $\Omega$  elementu  $f(x,y)$  jako zbiór zawierający cztery elementy sąsiednie:  $f(x-1,y)$  - punkt A na rys. 7.5,  $f(x-1,y-1)$  - punkt B,  $f(x,y-1)$  - punkt C - oraz  $f(x+1,y-1)$  jako kolejny punkt z wiersza leżącego powyżej- oznaczmy go przez D. Odejmijmy także wartość średnią obrazu oryginalnego od wartości każdego piksela ustalając  $\bar{f} = 0$  dla nowej reprezentacji i poszukajmy najlepszego modelu predykcji.

Okazuje się w takim przypadku, że optymalny z punktu widzenia minimalizacji błędu średniokwadratowego zbiór współczynników predykcji wygląda następująco:

$$\alpha_A = \rho_{poz}, \quad \alpha_B = -\rho_{poz} \rho_{pion}, \quad \alpha_C = \rho_{pion}, \quad \alpha_D = 0. \quad (7.11)$$

Widać, że punkt D kontekstu nie wnosi żadnej dodatkowej informacji o wartości kodowanej przy takim modelu obrazu. Oczywiście model ten jest pewnym przybliżeniem i nie zawsze opłacalnym jest pomijanie punktu D przy kształtowaniu optymalnej postaci kontekstu.

Postać modelu predykcji jest następująca:

$$\hat{f}(x, y) = \rho_{poz} f_A - \rho_{poz} \rho_{pion} f_B + \rho_{pion} f_C. \quad (7.12)$$

Pamiętając o warunku nakładanym często na współczynniki predykcji, że suma ich wynosi jeden (patrz równanie (6.12)). Upraszczając model obrazu założeniem o takim samym współczynniku korelacji w obu kierunkach  $\rho_{poz} = \rho_{pion} = \rho$ , uzyskujemy następującą postać funkcji przewidywania:

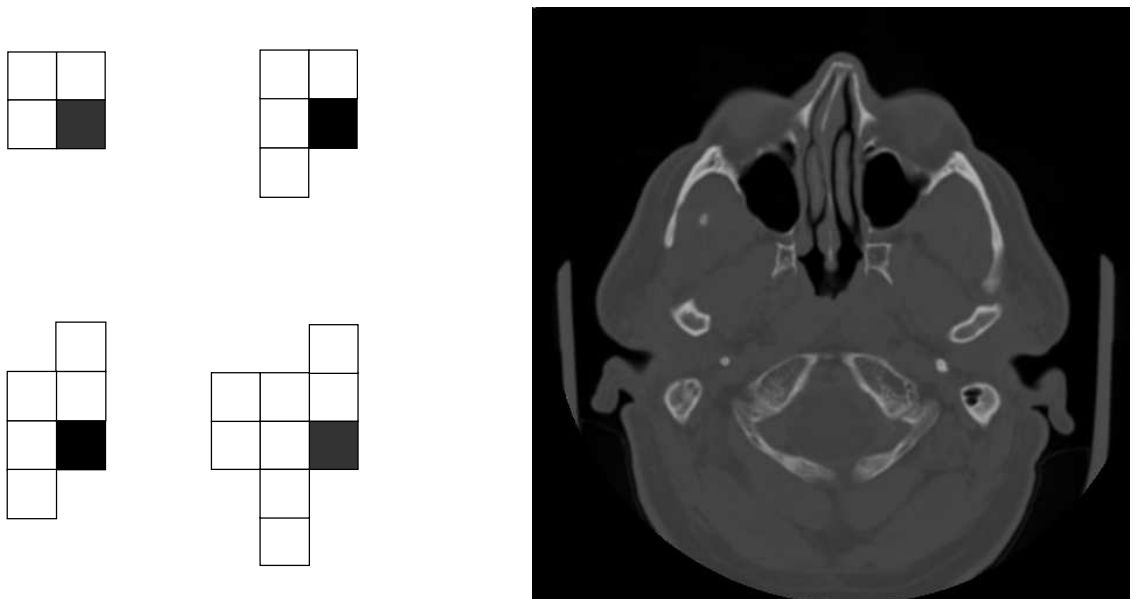
$$\hat{f}(x, y) = \frac{1}{\rho(2-\rho)} [\rho f(x, y-1) - \rho^2 f(x-1, y-1) + \rho f(x-1, y)]. \quad (7.13)$$

Z praktycznego punktu widzenia należy pamiętać, że wartości przewidywane, obliczane z powyższych równań, są zaokrąglane do postaci całkowitej dając następnie całkowite wartości błędu predykcji.

Aby zaobserwować wpływ kształtu i rozmiaru kontekstu w modelu predykcji na skuteczność kompresji obrazów wykonano testy prezentowane w przykładzie 7.2.

#### PRZYKŁAD 7.2. Kodowanie obrazu z wykorzystaniem różnych metod predykcji

Grupę obrazów jak w przykładzie 7.1 plus dodatkowy obraz testowy CT (rys. 7.6) poddano kompresji bezstratnej przy wykorzystaniu liniowej predykcji z kontekstu 2-D oraz kodera arytmetycznego rzędu 1 do kodowania reszt predykcyjnych. Skonstruowano 4 konteksty rzędu 3, 4, 5, 8 o kształcie jak na rys. 7.6, a współczynniki modeli predykcji liczone metodą regresji liniowej. Wyniki porównań efektywności tak skonstruowanych metod kompresji przedstawiono w tabeli 7.2.



Rys. 7.6. Po lewej stronie umieszczono konteksty rzędu odpowiednio 3, 4, 5 i 8, wykorzystane w predykcji wartości pikseli w przykładzie 7.2. Założony jest schemat przeglądania danych obrazowych kolumna po kolumnie ze względu na lepsze wyniki kompresji - patrz tab. 7.1. Kształt kontekstów uwzględnia piksele najbliższe położone w przestrzeni obrazu z zachowaniem przyczynowości modelu. Po prawej stronie - dodatkowy obraz testowy CT.



Tabela 7.2. Ocena skuteczności kompresji obrazów dla różnych kontekstów w modelu predykcji liniowej. Efektywność predykcji porównano z optymalną metodą przeglądania danych dla każdego obrazu. Tabela zawiera wartości średnich bitowych strumienia wyjściowego (bpp).

Metoda\Obraz	Lena	Barbara	Goldhill	CT1 (rys. 7.4)	CT2 (rys. 7.6)
Optymalny przegląd	4,87	5,86	5,38	1,77	1,84
Kontekst rzędu 3	4,55	5,28	4,98	2,00	1,51
Kontekst rzędu 4	4,48	5,23	4,95	2,02	1,48
Kontekst rzędu 5	4,48	5,17	4,95	2,12	1,51
Kontekst rzędu 8	4,49	5,69	5,13	2,14	1,45

Na podstawie przedstawionych wyników można generalnie stwierdzić, że wykorzystanie predykcji w technikach bezstratnej kompresji obrazów pozwala w większości przypadków bardziej zwiększyć skuteczność kompresji, niż dobór odpowiedniej metody przeglądania danych. Dla kolejnych obrazów uzyskano zmniejszenie średniej bitowej w stosunku do najlepszego schematu przeglądania odpowiednio o: 8% (Lena), 11% (Barbara), 8% (Goldhill) oraz 21% (CT2). Natomiast dla innego obrazu CT (CT1) predykcja dała gorszy o 13% rezultat niż optymalne skanowanie.

Duża różnorodność testowanych obrazów, dająca się zauważyć także w wyrażnie różnych, co do wartości współczynników, które odzwierciedlają poziom korelacji danych, modelach predykcji budowanych przy danym kontekście, powoduje odmienną skuteczność predykcji poszczególnych modeli. Bardziej skorelowane wartości pikseli obrazu CT2, o mniejszej dynamice zmian, są znacznie bardziej podatne na predykcję niż słabiej skorelowane piksele obrazu CT1. Już przy prostych modelach predykcji dla obrazu CT2 poprawa efektywności kompresji jest wyraźna w stosunku do metod z optymalizacją skanowania danych, podobnie zresztą jak w przypadku obrazów naturalnych.

Przykładowo, w tabeli 7.3 podano optymalne średniokwadratowo współczynniki modelu rzędu 3 dla każdego z kompresowanych obrazów.






Tabela 7.3. Współczynniki modelu predykcji obrazów rzędu 3 obliczone metodą minimalizacji błędu średniokwadratowego. Oznaczenia punktów kontekstu jak na rys.7.5.

Obraz	Lena			Barbara			Goldhill			CT1			CT2		
Współczynniki modelu rzędu 3	$\alpha_A$	$\alpha_B$	$\alpha_C$	$\alpha_A$	$\alpha_B$	$\alpha_C$	$\alpha_A$	$\alpha_B$	$\alpha_C$	$\alpha_A$	$\alpha_B$	$\alpha_C$	$\alpha_A$	$\alpha_B$	$\alpha_C$
	.7	-.5	.8	.25	-.06	.8	.81	-.53	.72	.53	-.11	.57	.79	-.65	.87

Korzyść ze zwiększania rzędu powyżej 3 jest raczej znikoma, co potwierdza słuszność teoretycznych rozważań opartych na modelu Markowa, charakteryzującym zbiór danych obrazowych. Największa poprawa, dla obrazu CT2, sięga 4% (dla modelu rzędu 8), przy czym nie ma charakteru monotonicznego. W innych przypadkach model rzędu 8 daje pogorszenie skuteczności kompresji, a optymalny rząd modelu wydaje się gdzieś pomiędzy rzędem 4 i 5. Dla obrazu CT1, kiedy to predykcja jest w ogóle nieskuteczna, zwiększenie rzędu modelu nie pozwala na uzyskanie poprawy efektywności, a wręcz przeciwnie.

### 7.3. Metody interpolacyjne






Metody hierarchicznej interpolacji (HINT) są próbą zbudowania kontekstu predykcji otaczającego punkt obrazu z każdej strony (360°). Przewidywanie wartości danego piksela, kiedy w kontekście modelu predykcji występują nie tylko piksele wcześniejsze, przy założonym porządku przeglądania danych, będziemy nazywać interpolacją. Konieczność zachowania przyczynowości modelu predykcji powoduje, że musi się to odbyć kosztem pewnej sztucznej hierarchicznej struktury nałożonej na obraz. Najczęściej jest to hierarchia kolejnych planów obrazu o rosnącej rozdzielczości. Mało efektywna prosta predykcja danych z grubej siatki daleko-odległych pikseli kontekstu 'z jednej strony' (często pomijana, czyli kodowane są jedynie kolejne wartości pikseli) jest następnie poprawiana dla kolejnych planów punktów obrazu leżących coraz bliżej siebie. Można bowiem dla nich zbudować bardziej doskonały kontekst 360°, składający się z punktów leżących bliżej w przestrzeni obrazu. Istotną staje się proporcja liczby pikseli z pierwszych porcji strumienia wyjściowego kodera o mało efektywnym kontekście do liczby pikseli kodowanych w dalszej kolejności, mających znacznie skuteczniejszy kontekst przewidywania.

Pomysł kompresji obrazu przez hierarchiczną interpolację wyjaśniony został na rysunku 7.7. Widać tam pięć grup pikseli (poziomów) wybranego bloku 9×9 obrazu. W pierwszej kolejności kodowane są piksele oznaczone jako  poprzez prostą predykcję 2-D DPCM i kodowanie Huffmana lub arytmetyczne. Oczywiście, tylko w przypadku występowania dużych, jednorodnych obiektów w danym fragmencie obrazu ta predykcja da zadawalające rezultaty, gdyż wartość piksela jest przewidywana jedynie z pikseli leżących z góry i z lewego boku i to w dodatku w dużej odległości (czterech kratak obrazujących podstawowe elementy obrazu - pisele). Dlatego też nierzadko opłaca się kodować te wartości bez żadnej predykcji. Warto przy tym zauważyć, że liczba tak nieefektywnie kodowanych pikseli jest stosunkowo mała - 9 z 81 pikseli bloku 9×9. Kolejna grupa kodowanych pikseli poziomu drugiego  w liczbie 4 z bloku ma również stosunkowo daleko-odległy kontekst predykcji (2 kratki po skosie), ale już otaczający wybrany piksel z czterech stron. Poprawy modelu dla grupy drugiej nie udaje się skonsumować efektywnie ze względu na małą licznosc tej grupy. Kodowanie pikseli oznaczonych przez  jest jeszcze bardziej efektywne, gdyż można wykonać interpolacje ich wartości na podstawie kodowanych wcześniej obu grup pikseli, z bliższej odległości (2 kratki, góra, dół, lewo, prawo). Tych pikseli jest 12. Ostatnie dwie grupy pikseli są liczniejsze i można w ich przypadku zbudować jeszcze korzystniejszy kontekst do interpolacji ich wartości. Dla pikseli  w liczbie szesnastu odległość pikseli użytych do przewidywania wynosi jedna kratka po skosie, a dla 40 pikseli  kontekst jest budowany z czterech stron, z najbliższej odległości. Wynika stąd, że prawie połowa pikseli w rozważanym bloku jest kodowana z dużą efektywnością, gdyż można dla nich zbudować przyczynowy kontekst najbliższych pikseli otaczających je z każdej strony (kontekst 360°).

Modele interpolacyjne zastosowane dla każdej grupy pikseli mają rząd równy 4. Wadą metody HINT jest słaba efektywność kompresji danych z pierwszych poziomów, kiedy to użyte schematy interpolacji są mało skuteczne za względu na odległy kontekst. Kluczowym zagadnieniem jest tutaj przesunięcie możliwie dużej liczby pikseli na poziomy wyższe, z lepszym kontekstem.

Niewątpliwą zaletą algorytmu hierarchicznej interpolacji jest naturalnie uzyskiwany progresywny charakter tworzonego strumienia danych, umożliwiający przesyłanie niskorozdzielczej wersji obrazu w pierwszej kolejności, a następnie dookreślanie obrazu z rosnącą rozdzielczością przy pomocy kolejnych porcji informacji dosyłanych z kodera.

+	♥	■	♥	+	♥	■	♥	+
♥	☾	♥	☾	♥	☾	♥	☾	♥
■	♥	😊	♥	■	♥	😊	♥	■
♥	☾	♥	☾	♥	☾	♥	☾	♥
+	♥	■	♥	+	♥	■	♥	+
♥	☾	♥	☾	♥	☾	♥	☾	♥
■	♥	😊	♥	■	♥	😊	♥	■
♥	☾	♥	☾	♥	☾	♥	☾	♥
+	♥	■	♥	+	♥	■	♥	+

Rys. 7.7. Ilustracja kodowania interpolacyjnego. Oznaczenia , , , ,  dotyczą kodowanych kolejno grup pikseli pierwszego, drugiego, trzeciego, czwartego i piątego poziomu..

Przedstawiony model hierarchicznej interpolacji może być oczywiście poprawiany i usprawniany. Ciekawy pomysł modyfikacji tego schematu stanowi zasadniczą koncepcję techniki CALIC [4], uznawanej obecnie za *state-of-art* w dziedzinie bezstratnej kompresji obrazów.

#### 7.4. Dwuwymiarowe modele statystyczne

Dwuwymiarowe modele statystyczne mogą być budowane na bazie analogicznych kontekstów, jak w przypadku metod predykcyjnych. Zasadniczy schemat kodera arytmetycznego się nie zmienia, gdyż znajdują zastosowanie te same modele prawdopodobieństw warunkowych wyższych rzędów, jak w opisanych w rozdz. 4 technikach kodowania arytmetycznego. W tym przypadku jednak kontekst modelu budowany jest w sposób bardziej złożony. W modelu rzędu  $m$  kontekstu nie tworzy  $m$  wartości danych bezpośrednio poprzedzających kodowany symbol w strumieniu wejściowym, ale  $m$  wartości sąsiednich w przestrzeni obrazu. Wartości pikseli kontekstu muszą być więc często pobierane również z wcześniejszych partii strumienia, np. z poprzedniego wiersza przy kodowaniu pikseli wiersz po wierszu. Konieczne jest więc poprawne ustalenie kontekstu wystąpienia danego piksela w kodowanym obrazie. Można wyobrazić sobie również interpolacyjną metodę ustalenia kontekstu, narzucającą odpowiednią kolejność kodowania pikseli tak, aby wszystkie wartości wykorzystane do określenia kontekstu były pobierane przy zachowaniu warunku przyczynowości.

Najpoważniejsze problemy związane z konstrukcją modeli statystycznych wyższych rzędów w koderach arytmetycznych dotyczą zagadnienia rozrzedzenia kontekstu (patrz rozdz. 4.3). Przedstawione zostały tutaj niektóre sposoby poprawy skuteczności kompresji obrazów

z wykorzystaniem modeli wyższych rzędów poprzez modelowanie kontekstu i jego kwantyzację.

Często stosowanym sposobem zmniejszenia kontekstu koderu arytmetycznego, tzw. kwantyzacji kontekstu, jest liniowa kombinacja elementów kontekstu modelowanego, który ma realny wpływ na kształtowanie modelu statystycznego. Współczynniki liniowego operatora kwantyzacji kontekstu są dobierane metodą regresji liniowej. W takim rozwiązaniu zredukowany jest rozmiar kontekstu, czyli rząd modelu statystycznego sterującego koderem.

Dodatkowy efekt zmniejszenia dynamiki (alfabetu) wartości dookreślających model prawdopodobieństw warunkowych w procesie kwantyzacji alfabetu powoduje szybsze wypełnienie modelu i co za tym idzie, większą jego wiarygodność, a także szybszą adaptację do zmian lokalnej statystyki strumienia wejściowego. Taki prosty algorytm, wspomniany w rozdz. 4.3, może znacznie zwiększyć skuteczność bezstratnej kompresji obrazów. Mówi o tym następujący przykład.

**PRZYKŁAD 7.3.** Kwantyzacja kontekstu w arytmetycznym kodowaniu obrazów.

Zbiór czterech obrazów testowych: Lena, Barbara, Goldhill i CT2 z poprzednich przykładów tego rozdziału poddano kodowaniu arytmetycznemu z wykorzystaniem kontekstu modelującego o różnych rozmiarach. Kontekst ten jest kwantowany do kontekstu pierwszego rzędu przy pomocy operatora liniowego z wagami dobranymi metodą regresji. Skorzystano także z możliwości poprawy skuteczności kodowania arytmetycznego poprzez dodatkową kwantyzację alfabetu elementów kontekstu, co daje następującą postać procesu modelowania i kwantyzacji kontekstu rzędu 1  $x'_{k-1}$  w punkcie  $f_k$  (według ustalonej kolejności przeglądania danych):

$$x'_{k-1} = \text{round} \left( \frac{\sum_{i=1}^m \alpha_i x_{k-i}}{q} \right) \quad (7.14)$$

gdzie  $m$ - rząd kontekstu modelowanego,  $i$ - indeks po elementach kontekstu 2-D, a  $q \geq 1$  to współczynnik kwantyzacji alfabetu. Wyniki zebrano w tabeli 7.4.

Tabela 7.4. Porównanie skuteczności kompresji obrazów dla różnych kontekstów modelujących (jak w przykładzie 7.2), kwantowanych do kontekstu 1-go rzędu według zależności (7.14). Tabela zawiera wartości średnich bitowych kodu. Wartość współczynnika  $q$  dobrana jest optymalnie dla każdego z obrazów.

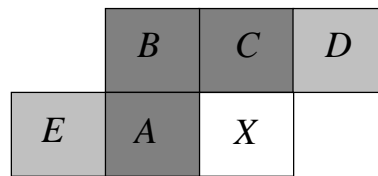
Kontekst modelujący	Lena $q=4$	Barbara $q=6$	Goldhill $q=5$	CT2 $q=1$
Rząd 3	4,61	5,56	4,98	1,49
Rząd 4	4,56	5,51	4,95	1,48
Rząd 5	4,55	5,39	4,95	1,41
Rząd 8	4,53	5,37	4,94	1,36

W porównaniu z przykładem 7.2 uzyskano zbliżoną do metod predykcyjnych efektywność kompresji. Zwiększanie rozmiaru kontekstu w przypadku każdego obrazu dało poprawę skuteczności kompresji. Potwierdza to ogólnie przyjmowaną zasadę, że w statystycznym modelowaniu stosuje się większy kontekst niż w predykcji, oczywiście pod warunkiem skutecznych rozwiązań anty-rozrzedzeniowych. Model statystyczny pozwala lepiej opisać i spożytkować rzeczywiste korelacje pomiędzy elementami obejmowanymi przez zdekomponowany

kontekst. Nie bez znaczenia jest też adaptacyjny schemat modyfikacji modelu koder (tj. stałe uwiarygodnianie prawdopodobieństw warunkowych) w przeciwieństwie do statycznych schematów predykcji stosowanych w przykładzie 7.2. Poprawa efektywności kompresji poprzez zastosowanie wyższych rzędów kontekstu jest stąd większa od rezultatów uzyskanych przy predykcji i wynosi blisko 9% w najlepszym przypadku (skuteczność kontekstu rzędu 8 w porównaniu z kontekstem rzędu 3). Ciekawym eksperymentem jest połączenie obu metod kompresji w jednym schemacie, kiedy to wartości różnicowe po predykcji są kodowane arytmetycznie przy wykorzystaniu dwuwymiarowych modeli statystycznych. Proste złożenie tych algorytmów, przy czym statystyczne modelowanie jest wykorzystane do kodowania wartości bezwzględnej, a znak kodowany jest oddzielnie binarnym koderem arytmetycznym, daje dodatkowe zwiększenie skuteczności kodowania w granicach kilku procent. Taki właśnie schemat jest podstawą najskuteczniejszych obecnie technik kompresji bezstratnej obrazów [4,5].

### JPEG-LS

Inny sposób doboru kontekstu zastosowano w standardzie kompresji bezstratnej JPEG-LS [5,6]. W pierwszym etapie predykcji wykorzystano prosty kontekst trójelementowy. Chcąc natomiast objąć modelem statystycznym koder szerszy obszar potencjalnych zależności pomiędzy danymi zdefiniowano kontekst modelujący jak na rys. 7.8.:



Rys. 7.8. Kontekst w predykcji nieliniowej JPEG-LS oraz modelowaniu statystycznym koder binarnego; ciemnoszare kwadraty oznaczają elementy kontekstu rzędu 3 dla predykcji, a pola jasnoszare uzupełniają ten kontekst do większego kontekstu rzędu 5 w modelowaniu statystycznym.

Model predykcji jest nieliniowy, określony dla piksela  $X$  poprzez zależność:

$$\hat{X} = \begin{cases} \min(A, C) & \text{if } B \geq \max(A, C) \\ \max(A, C) & \text{if } B \leq \min(A, C) \\ A + C - B & \text{w p.p.} \end{cases} \quad (7.15)$$

Następnie, w kodowaniu binarnym na podstawie kontekstu rzędu 5 z rys. 7.8 zdefiniowano skwantowany kontekst czwartego rzędu składający się z następujących wartości różnicowych:

$$g_1 = D - C, \quad g_2 = C - B, \quad g_3 = B - A, \quad g_4 = A - E. \quad (7.16)$$

Model probabilistyczny konstruowany na bazie prawdopodobieństw warunkowych na bazie takiego kontekstu jest postaci:

$$P(X | g_1, g_2, g_3, g_4). \quad (7.17)$$

Alfabet każdego z tych elementów sprowadzono przy pomocy operatora kwantyzacji  $Q(\cdot)$  do 9 symboli zbioru  $\{-4, -3, \dots, 3, 4\}$ , co daje  $9^4 = 6561$  stanów modelu (każda kombinacja czterech punktów kontekstu o dziewięcioelementowym alfabecie). Stan jest określony przez

kombinację  $C^{(4)} = [g_1, g_2, g_3, g_4]$ . Ponadto, dla zmniejszenia liczby stanów w modelowaniu statystycznym wykorzystano oddzielne ustalanie znaku stanu  $C$  w ten sposób, że jeśli pierwszy niezerowy element stanu (zaczynając od  $g_1$ ) jest ujemny to odwracane są znaki wszystkich niezerowych elementów stanu, czyli przechodzi on w stan  $-C$ . Wymaga to oczywiście włączenia znaku do określenia stanu modelu. Pozwala to zmniejszyć liczbę stanów do  $(9^4 + 1)/2 = 3281$ . To także okazała się zbyt duża liczba stanów i w kolejnych wersjach standardu pojawił się już stan określony tylko przez trzy elementy  $C^{(3)} = [g_1, g_2, g_3]$ , co przy tym samym algorytmie kwantyzacji alfabetu dało jedynie 365 stany dla modelu statystycznego użytego do kodowania obrazu.

Aby w pełni zdefiniować proces kwantyzacji kontekstu potrzeba jeszcze określić sposób kwantyzacji alfabetu elementów takiego kontekstu do alfabetu o 9-ciu symbolach, czyli zdefiniować jak działa operator  $Q(\cdot)$ . Nie musi to być wcale kwantyzacja równomierna i najczęściej nie jest. W schemacie JPEG-LS wybrano następujące regiony kwantyzacji dla wartości  $x$  kwantowanych elementów kontekstu, należących do 8-mio bitowego alfabetu:  $\{0\}, \pm\{1,2\}, \pm\{3,4,5,6\}, \pm\{7,8,\dots,20\}, \pm\{x > 20\}$ . Oczywiście okazuje się, że dla mniejszych zbiorów danych, np. obrazów o rozmiarach  $64 \times 64$ , lepiej jest użyć jeszcze silniejszej kwantyzacji kontekstów, np. do 63 stanów trójelementowego kontekstu z alfabetem składającym się z pięciu symboli zamiast dziewięciu. Daje to poprawę efektywności kompresji, bo model o mniejszej liczbie stanów może być tutaj lepiej określony.

Do kodowania binarnego stosowano na etapie rozwoju standardu koder Huffmana z prawdopodobieństwami warunkowymi, a także algorytm konstrukcji kodów Golomba. Skuteczność koderu JPEG-LS jest jednak ograniczona, co pokazują rezultaty testów przytoczone w kolejnym podrozdziale.

## 7.5. CALIC – efektywny algorytm bezstratnej kompresji obrazów

Metoda przedstawiona w tym punkcie jest uznawana obecnie za najefektywniejszą metodę bezstratnej kompresji obrazów. Pozwala uzyskać bardzo małe wartości średnich bitowych przy stosunkowo niewielkim czasie kompresji i małych wymaganiach pamięciowych. Potwierdzają to wyniki zamieszczone na końcu tego podrozdziału.

Technika CALIC (ang. Context-based Adaptive Lossless Image Codec) wykorzystuje adaptacyjne kodowanie predykcyjne z kontekstem wyższych rzędów, a także podobną do metod interpolacyjnych sekwencyjną analizę obrazu na coraz wyższym poziomie rozdzielczości (coraz mniejszej skali). Dużą efektywność kodowania zawdzięcza nieliniowemu, odpowiednio dobranemu kontekstowemu modelowi predykcji, który wykorzystuje dobrane na podstawie eksperymentów wartości współczynników predykcji. Niskie koszty aplikacji osiągane są z kolei poprzez ciekawy pomysł kwantyzacji kontekstu, który pozwala uzyskać dużą skuteczność kodowania przy niewielkim rozmiarze tablic prawdopodobieństw warunkowych. Algorytm modulowania kontekstu oraz predykcji jest prosty, zrealizowany w większości w arytmetyce liczb całkowitych, z prostą logiką. Tak więc czas wykonania predykcji i modelowania (w koderze i dekoderze jest identyczny) dla obrazu  $512 \times 512$  wynosi 1.52 sekundy na komputerze SUN SPARC10.

Do realizacji ostatniej fazy algorytmu CALIC można użyć entropijnego koderu dowolnego rodzaju (Huffmana, arytmetyczny), a cechy całego schematu kompresji zdają się pozwalać, przy pewnych modyfikacjach, na realizację tej metody w wersji z progresją zorientowaną na rozdzielczość do zastosowań transmisyjnych.

Twórcy techniki CALIC zdawali sobie dobrze sprawę z wymagań stawianych skutecznej metodzie odwracalnej kompresji. Nic dziwnego, skoro zgłosili swoją metodę do komitetu ISO jako propozycję nowego standardu bezstratnej kompresji. Zasadniczym elementem ich schematu kompresji jest modelowanie kontekstu. Kryteria stawiane algorytmom modelowania kontekstu, które pozwolą uzyskać dużą efektywność bezstratnego kodowania obrazu w znaczeniu maksymalnej kompresji oraz praktycznej przydatności, są następujące:

- a) silna dekorrelacja - przyjęty model obrazu pozwala skutecznie określić i zredukować statystyczną nadmiarowość źródła;
- b) uniwersalność - model obrazu może szybko przystosować się (adoptować) do nieznannej statystyki źródła;
- c) niski koszt obliczeniowy - same konteksty oraz modelowanie obrazu i predykcja oparte na tych kontekstach mogą być obliczone (wyznaczone) stosunkowo niskim kosztem;
- d) niskie wymagania pamięciowe - przestrzeń pamięci przeznaczona do gromadzenia kontekstów i informacji związanych z tymi kontekstami, używana w modelowaniu i predykcji nie powinna być nadmierna.

Zaproponowano rozwiązanie, które w zadawalający sposób spełnia te kryteria, przez co uznawane jest powszechnie jako *state-of-art* współczesnych technik odwracalnej kompresji. Krótka charakterystyka techniki CALIC, napisana na podstawie [4], przedstawia się następująco.

### **Trzy-stopniowy schemat kodowania predykcyjnego z przeplotem**

Niech obraz oryginalny, wielopoziomowy ze skalą szarości, o szerokości  $M$  i wysokości  $N$  będzie opisany funkcją jasności  $f(x, y)$ ,  $0 \leq x < M$ ,  $0 \leq y < N$ . Kodowanie predykcyjne realizowane jest w trzech kolejnych etapach, schematycznie pokazanych na rys. 7.9. Na każdym etapie przeprowadzane jest przeplatane próbkowanie obrazu oryginalnego. Celem realizacji modelu sekwencyjnej predykcji jest stworzenie możliwości przewidywania kodowanych wartości przy pomocy najlepszego kontekstu. Już w drugim etapie udaje się zbudować otoczenie bliskie kontekstowi  $360^\circ$ , przy czym jest to jednak otoczenie dość odległe. Najlepsza sytuacja występuje w trzecim etapie, kiedy model predykcji obejmuje otoczenie z każdej strony i jednocześnie jest to kontekst bardzo bliski. Warto przy tym zauważyć, że w trzecim etapie predykcji, przy tak dobrym kontekście, kodowana jest połowa pikseli obrazu. Jest to jeden z ważnych czynników pozwalających uzyskać dużą skuteczność kompresji całej metody.

#### *Pierwszy etap*

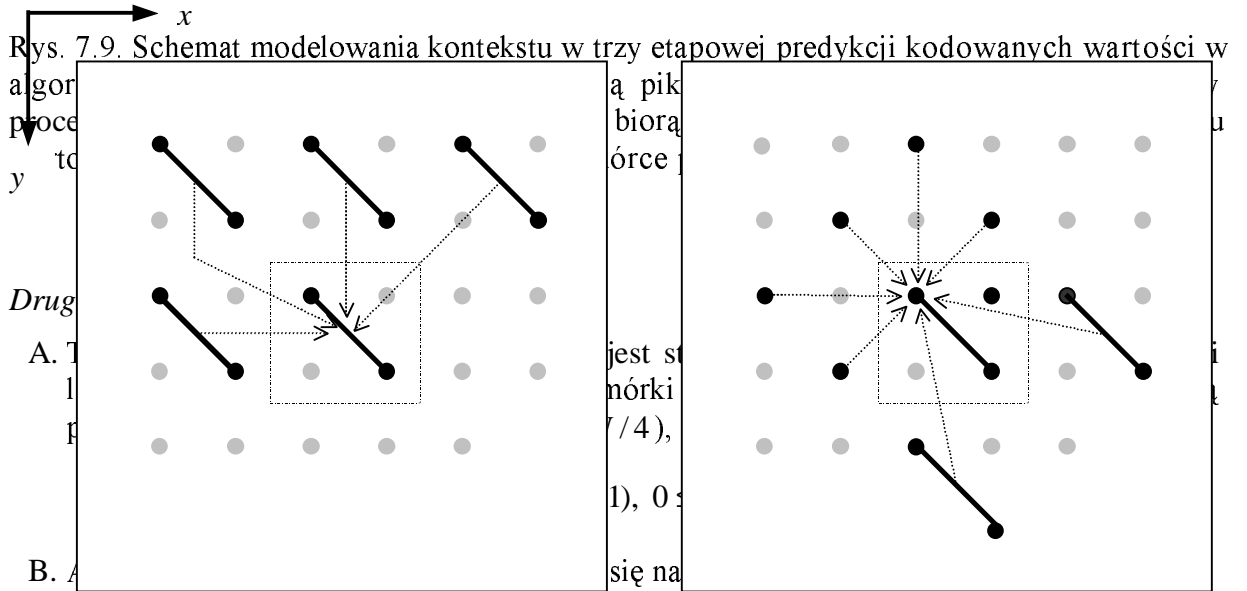
- A. Tworzony jest podobraz o dwukrotnie zmniejszonej rozdzielczości w obu kierunkach  $M/2 \times N/2$  w sposób następujący:

$$\mu(x, y) = \left\lfloor \frac{f(2x, 2y) + f(2x+1, 2y+1)}{2} \right\rfloor, \quad 0 \leq x < M/2, \quad 0 \leq y < N/2. \quad (7.18)$$

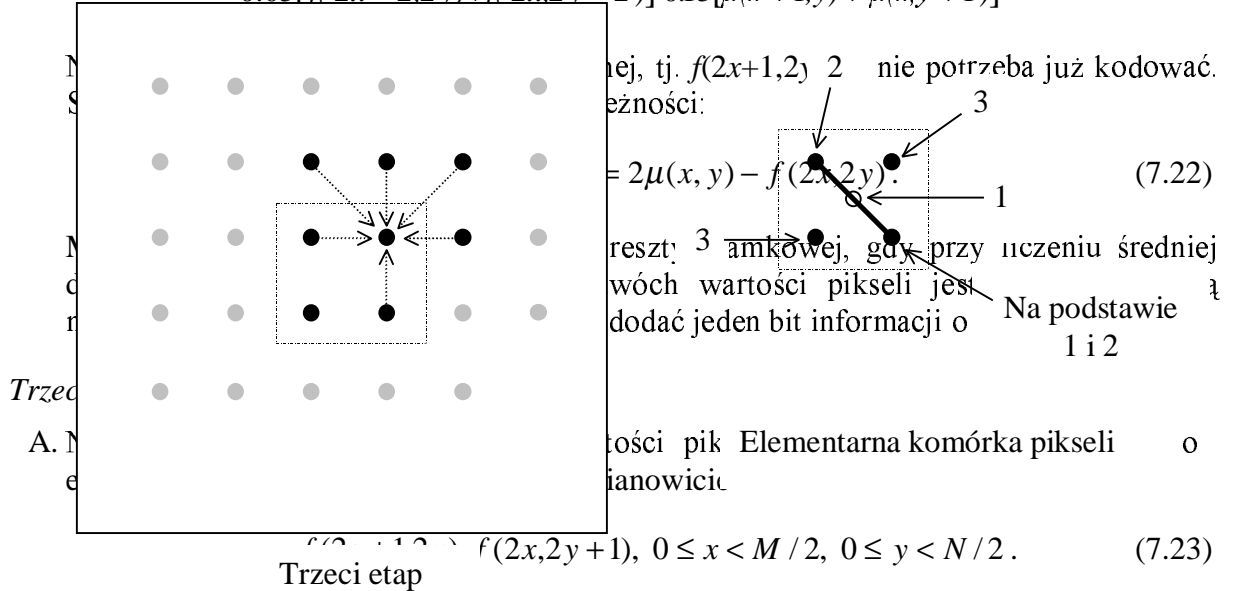
- B. Podobraz  $\mu$  kodowany jest z wykorzystaniem kontekstu  $180^\circ$ , jak w typowym schemacie dla DPCM, tj.

$$\hat{\mu}(x, y) = \frac{\mu(x, y-1) + \mu(x-1, y)}{2} + \frac{\mu(x+1, y-1) - \mu(x-1, y-1)}{4} \dots \quad (7.19)$$

Współczynniki predykcji zostały wyznaczone metodą regresji liniowej na dużym zbiorze obrazów testowych. Zaokrąglone zostały do potęgi dwójki ze względu na prostotę obliczeń.



$$\hat{f} = \frac{f(2x-1, 2y-1) + f(2x-1, 2y) + f(2x, 2y-1) + f(2x, 2y)}{6} - 0.05[f(2x-2, 2y) + f(2x, 2y-2)] - 0.15[\mu(x+1, y) + \mu(x, y+1)] \quad (7.21)$$



Dla zwiększenia skuteczności predykcji wykorzystano kontekst 360° składający się z czterech pikseli najbliższego sąsiedztwa w rastrze cztero-połączeniowym oraz dwóch pikseli z sąsiedztwa ośmio-połączeniowego. Wszystkie one zostały już wcześniej zakodowane w koderze oraz zdekodowane w dekodерze, a więc są dostępne w procesie rekonstrukcji.

B. Zastosowano następujący model predykcji:



$$\hat{f} = \frac{3}{8} [f(x-1, y) + f(x, y-1) + f(x+1, y) + f(x, y+1)] - \frac{f(x-1, y-1) + f(x+1, y-1)}{4}. \quad (7.24)$$

### Modelowanie i kwantyzacja kontekstu.

Dotąd do modelowania kontekstów na trzech etapach predykcji wartości poszczególnych pikseli  $f$  wykorzystywano  $m$  wartości sąsiednich pikseli  $f_1, f_2, \dots, f_m$ , przy czym  $m = 4, 9, 6$  odpowiednio na 1, 2 i 3 etapie. Aby przy entropijnym kodowaniu uzyskać minimalną długość kodu wyjściowego dla zbioru wartości różnicowych  $e = f - \hat{f}$  (gdzie  $\hat{f}$  wyznaczono dla każdej wartości  $f$  w trójetapowej predykcji) potrzeba maksymalizować prawdopodobieństwa warunkowe  $p(e / f_1, \dots, f_m)$ . Używając standardowych metod np. kodowania arytmetycznego z założonym modelem Markowa  $m$ -tego rzędu potrzeba wówczas określić w sposób istotny i statystycznie model o  $2^{mW}$  stanach ( $W$ -liczba poziomów szarości), co jest problemem niebagatelnym. Do przechowania parametrów takiego modelu statystycznego potrzeba bardzo dużo pamięci, a nabranie przez ten model statystycznej wiarygodności wymaga bardzo dużego zbioru danych (problem rozrzedzenia kontekstu). Rozwiązanie to jawi się więc jako mało praktyczne.

Zastosowano wobec tego następujący model kwantyzacji kontekstu, w celu efektywniejszego określenia zredukowanej liczby prawdopodobieństw warunkowych oraz korekcji przewidywanej wartości w pętli sprzężenia zwrotnego. Pomysł przedstawia się następująco:

- A. Aby ułatwić modelowanie kontekstu błędów predykcji DPCM, kwantyzuje się kontekst  $f_1, f_2, \dots, f_m$  związany z przewidywaną wartością  $\hat{f}$  do liczby o postaci binarnej  $t = t_m \dots t_1$ , składającej się z  $m$  bitów tak, że:

$$t_i = \begin{cases} 0 & \text{if } f_i \geq \hat{f} \\ 1 & \text{if } f_i < \hat{f} \end{cases}, \quad 1 \leq i \leq m. \quad (7.25)$$

Liczba  $t$  reprezentuje wyższego rzędu przestrzenną strukturę modelowanego kontekstu, czyli cechę obrazu, która może wskazywać na zachowanie się błędu predykcji DPCM lub inaczej - ma wyrażać korelację otoczenia z błędem predykcji. Można w ten sposób uwzględnić krawędzie, rogi struktur, tekstury, które nie mogły być wychwycone przez DPCM (którego współczynniki są globalne, wspólne dla całego obrazu).

- B. Obliczana jest także wartość wielkości zwanej dyskryminatorem mocy błędu, która charakteryzuje zmienność wartości kontekstu (jego gładkość), w sposób następujący:

$$\Delta = \sum_{i=1}^m w_i |f_i - \hat{f}|. \quad (7.26)$$

Na jej podstawie można estymować prawdopodobieństwo warunkowe błędu predykcji jako  $p(e/\Delta)$  zamiast  $p(e / f_1, \dots, f_m)$ . Aby dodatkowo zlikwidować problem rozrzedzenia kontekstu przy estymacji  $p(e/\Delta)$  dokonywana jest kwantyzacja wartości  $\Delta$  na  $L$  poziomów (w praktyce najczęściej  $L=8$ , a dokładniej sposób kwantyzacji zostanie opisany nieco później). Łącząc teraz, jako iloczyn kartezjański  $L \times 2^m$ , dyskryminator  $\Delta$  kwantowany do  $L$  poziomów oraz  $2^m$  kwantowanych wzorców tekstury kontekstu  $t_i$  otrzymujemy ostatecznie kwantyzację  $2^{mW}$  stanów źródła początkowego do znacznie

zredukowanej liczby  $L \cdot 2^m$  stanów nowego modelu kontekstu. Tak uproszczony kwantowaniem kontekst oznaczony jest następująco:  $C(d, t)$ ,  $0 \leq d < L$ ,  $0 \leq t < 2^m$ . Pozostaje jeszcze ważny problem - jak dobrać wartości współczynników  $w_i$  w wyrażeniu (7.26). Wartość  $\Delta$  jest ustalana jako średniokwadratowa estymata  $|e|$  (z minimalnym błędem średniokwadratowym naśladuje wartości  $e = f - \hat{f}$ ). Określa się więc wartości współczynników  $w_i$  stosując standardową metodę regresji liniowej. Wobec tego, mając przykładowy predyktor DPCM:  $\hat{f} = \sum_{i=1}^m \alpha_i x_i$  jednego z trzech etapów predykcji, wyznaczany jest treningowy zbiór  $U$  wartości  $|e| = |f - \hat{f}|$  oraz  $|f_i - \hat{f}|$ ,  $1 \leq i \leq m$ .

Następnie wartości  $w_i$ ,  $1 \leq i \leq m$  są obliczane metodą regresji liniowej tak, aby zminimalizować wartość:

$$\sum_U \{|e| - \Delta\} = \sum_U \left\{ |f - \hat{f}| - \sum_{i=1}^m w_i |f_i - \hat{f}| \right\} \quad (7.27)$$

na treningowym zbiorze danych.

### **Adaptacyjne, oparte na kontekście modelowanie błędu (wraz z drugą fazą kwantyzacji)**

Jednak  $L \cdot 2^m$  stanów kontekstu w danym modelu to jednak dalej zbyt wiele do uzyskania szybkiej i skutecznej estymacji prawdopodobieństw warunkowych, w tym przypadku  $p(e|C(d, t))$ . Wobec tego następuje:

- A. Estymowanie warunkowej wartości oczekiwanej  $E\{e|C(d, t)\}$  poprzez wyznaczenie odpowiedniej średniej próbek  $\bar{e}(d, t)$  dla różnych kwantowanych kontekstów. Intuicyjnie wiadomo, znacznie mniej próbek potrzeba do dobrej estymacji warunkowej wartości oczekiwanej niż prawdopodobieństwa (dokładniej  $O(n^{-0.5})$  versus  $O(n^{-0.4})$ ). Ponieważ średnia warunkowa  $\bar{e}(d, t)$  lepiej estymuje błąd predykcji DPCM w kwantowanym kontekście  $C(d, t)$ , można skompensować błąd DPCM poprzez modyfikację predykcji wartości  $f$  z  $\hat{f}$  na predykcję wartości  $f$  z wartości zmiennej, określonej następująco:  $\dot{f} = \hat{f} + \bar{e}(d, t)$ . Nowy predyktor  $\dot{f}$  jest adaptacyjny, nieliniowy i realizuje mechanizm sprzężenia zwrotnego błędu predykcji z opóźnieniem jednego kroku. Wykorzystując nowy predyktor mamy teraz nowy błąd predykcji:  $\varepsilon = f - \dot{f}$  dla entropijnego kodowania błędu. Jednak, aby zamknąć pętlę sprzężenia zwrotnego należy zmodyfikować postać nowego predyktora na  $\dot{f} = \hat{f} + \bar{\varepsilon}(d, t)$ , gdzie  $\bar{\varepsilon}(d, t)$  jest średnią próbek  $\varepsilon$  warunkowanych skwantowanym kontekstem  $C(d, t)$ . Znaczy to, że kontekstowe modelowanie błędu predykcji będzie się odbywać na podstawie wartości  $\varepsilon$ . Rozkład wartości błędu predykcji zachowuje w tym przypadku postać rozkładu Laplace'a, ale jest wyraźnie wyostrzony, co powinno poprawić skuteczność entropijnego kodowania.
- B. Optymalizacja kwantyzatora  $Q$  wartości  $\Delta$ . Kryterium kwantyzacji jest minimalizacja warunkowej entropii wartości błędów zależnej od  $p(\varepsilon|Q(\Delta))$ . Ze zbioru obrazów treningowych obliczamy zbiór wartości par  $(\varepsilon, \Delta)$  i używamy standardowej dynamicznej techniki wyboru wartości  $0 = q_0 < q_1 < \dots < q_{L-1} < q_L = \infty$  dzielących przedział wartości  $\Delta$  na  $L$  podprzedziałów:  $U_d = \{\varepsilon | q_d \leq \Delta < q_{d+1}\}$ , takich że wrazenie:

$$-\sum_{\varepsilon} p(\varepsilon) \log p(\varepsilon | \varepsilon \in U_d) \quad (7.28)$$

osiąga minimum.

Dwa procesy poszukiwania optymalnego schematu kwantyzacji, tj. wyznaczanie współczynników w procesie minimalizacji wyrażenia (7.27) i podział przedziału wartości  $\Delta$  na  $L$  podprzedziałów według rozwiązania dającego minimum zależności (7.28), mogą być przeprowadzane również w sposób łączny (kwantyzacja łączna).

- C. Entropijne kodowanie wartości  $\varepsilon$  - entropijne kodowanie adaptacyjne (najlepiej arytmetyczne) z użyciem tylko  $L$  prawdopodobieństw warunkowych  $p(\varepsilon | Q(\Delta)=d)$ ,  $0 \leq d < L$  dla poszczególnych wartości  $\varepsilon$ .

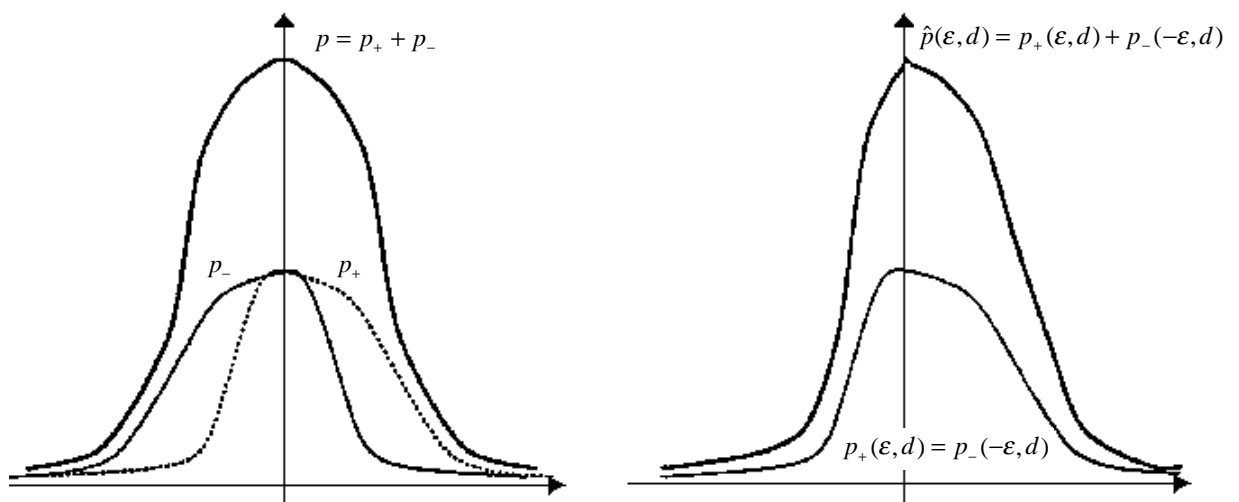
## Przerzucanie znaku błędu predykcji

Znaki warunkowych średnich próbek  $\bar{\varepsilon}(d, t)$  mogą być użyte do wyostrenia prawdopodobieństw warunkowych  $p(\varepsilon|Q(\Delta)=d)$ , a więc redukcji warunkowej entropii. Dla dwóch różnych kontekstów  $C(d, t_1)$  i  $C(d, t_2)$ , wartości warunkowych średnich próbek  $\bar{\varepsilon}(d, t_1)$  i  $\bar{\varepsilon}(d, t_2)$  mogą mieć przeciwne znaki i odpowiednio różne wartości  $p(\varepsilon|C(d, t_1))$  i  $p(\varepsilon|C(d, t_2))$ . Dla ustalonej wartości  $0 \leq d < L$  można rozdzielić prawdopodobieństwo  $p(\varepsilon|Q(\Delta)=d)$  na dwa:

$$p_+(\varepsilon|d) = p(\varepsilon|d, \bar{\varepsilon}(d, t) \geq 0), \quad p_-(\varepsilon|d) = p(\varepsilon|d, \bar{\varepsilon}(d, t) < 0). \quad (7.29)$$

Oczywistym jest, że obydwa prawdopodobieństwa warunkowe  $p_+$  i  $p_-$  dają niższą entropię niż  $p(\varepsilon|Q(\Delta)=d)$  (bardziej wyróżniona jest statystyka błędu). Wydaje się jednak, że wówczas podwajają się: zużycie pamięci oraz rozrzedzenie kontekstów. Poniższe rozwiązanie zapobiega tym ograniczeniom.

Można zaobserwować, że  $p_+(\varepsilon|Q(\Delta)=d)$  oraz  $p_-(\varepsilon|Q(\Delta)=d)$  są w przybliżeniu swoim lustrzanym odbiciem względem wartości oczekiwanej, równej zero, rozkładu  $p = p_+ + p_-$ . Można więc przetrzucić  $p_-$  symetrycznie względem osi zerowej ( $p_-(-\varepsilon|d)$ ) i nałożyć na  $p_+$  otrzymując obciążony estymator prawdopodobieństwa  $\hat{p}(\varepsilon|Q(\Delta)=d)$  o mniejszej wariancji w stosunku do  $p(\varepsilon|Q(\Delta)=d)$ , co widać wyraźnie na rys. 7.10 ( $\hat{p}(\varepsilon|Q(\Delta)=d)$  jest bardziej wyostreny). Zastosowanie tego estymatora pozwala zmniejszyć średnią bitową kodu bez dodatkowej pamięci oraz bez rozrzedzenia kontekstu. Podsumowując, algorytm przetrzucania znaku błędu jest następujący: przed kodowaniem wartości  $\varepsilon = x - \hat{x}$ , koder sprawdza, czy  $\bar{\varepsilon}(d, t) < 0$  na podstawie kontekstu  $C(d, t)$ . Jeśli błąd predykcji jest ujemny, wówczas kodowana jest wartość  $-\varepsilon$ , a w pozostałych przypadkach  $\varepsilon$ . Ponieważ dekodek także zna  $C(d, t)$  i  $\bar{\varepsilon}(d, t)$  może, jeśli to konieczne, odwrócić znak w celu poprawnej rekonstrukcji wartości  $\varepsilon$ . Mamy więc tutaj do czynienia z przewidywaniem znaku błędu predykcji.



Rys. 7.10. Wyostrenie prawdopodobieństwa warunkowego  $p = p_+ + p_-$  poprzez przetrzucanie  $p_-$  i tworzenie bardziej obciążonego estymatora  $\hat{p}(\varepsilon, d)$ .

## Ogólny opis algorytmu

Algorytm składa się z trzech głównych elementów:

- adaptacyjna predykcja,
- modelowanie kontekstu,
- kodowanie z entropią warunkową,

przy czym modelowanie kontekstu wpływa zarówno na predykcję jak i entropijne kodowanie. Przedstawiony niżej algorytm pokazuje w sposób skondensowany zasadniczy pomysł realizacji CALIC-a.

### **Algorytm 7.1. Kompresja obrazu metodą CALIC**

Dla każdego z trzech etapów następuje:

1. Inicjalizacja:  $N(d,t)=1, S(d,t)=0, 0 \leq d < L, 0 \leq t < 2^m$ .
2. Parametry użytych modeli: wyznaczone wcześniej optymalne wartości  $\alpha_i, w_i, 1 \leq i \leq m$ .
3. Dla wartości funkcji jasności  $f$  kolejnych pikseli według porządku skanowania danego etapu:

a).  $\hat{f} = \sum_{i=1}^m \alpha_i f_i$  dla kontekstu predykcji  $f_1, f_2, \dots, f_m$ ;

b).  $\Delta = \sum_{i=1}^m w_i (f_i - \hat{f})$ ;

c).  $d = Q(\Delta)$ ;

d). wyznacz wzorzec tekstury  $t = t_m \dots t_1$ :

$$\text{if } (f_i < \hat{f}) \ t_i = 0 \ \text{else } \ t_i = 1, \ 1 \leq i \leq m;$$

e).  $\bar{\varepsilon} = S(d,t) / N(d,t)$ ;

f).  $\dot{f} = \hat{f} + \bar{\varepsilon}$ ;

g).  $\varepsilon = f - \dot{f}$ ;

h).  $S(d,t) = S(d,t) + \varepsilon; N(d,t) = N(d,t) + 1$ ;

i). if  $N(d,t) \geq 128$

$$S(d,t) = S(d,t) / 2; N(d,t) = N(d,t) / 2;$$

j). if  $S(d,t) < 0$  koduj  $(-\varepsilon, d)$  else koduj  $(\varepsilon, d)$ ;

Do najważniejszych zalet algorytmu techniki CALIC można zaliczyć złagodzenie problemu rozrzedzenia kontekstu poprzez:

- rozróżnienie kontekstu  $C(d,t)$  dla modelowania (adaptacyjnej korekcji) błędu predykcji oraz kontekstu  $Q(\Delta)$  wykorzystywanego do warunkowego entropijnego kodowania wartości błędu predykcji;
- estymatę pojedynczego parametru wartości oczekiwanej zamiast szeregu prawdopodobieństw warunkowych dla danego kontekstu.

Pozwoliło to uzyskać dużą liczbę kontekstów przy modelowaniu błędu, aby uchwycić bardziej złożone struktury błędu (złożone korelacje) w celu zwiększenia efektywności kodowania.

## Ocena efektywności techniki CALIC

W tabeli 7.5 przedstawiono porównanie skuteczności kompresji metody CALIC z innymi, najbardziej znanymi i efektywnymi technikami kompresji odwracalnej. Wytłuszczono najmniejszą średnią bitową dla każdego z obrazów oraz najmniejszą średnią.

Tabela 7.5. Ocena efektywności kompresji kodera CALIC. Podano wartości średnich bitowych w bpp. Wyniki przytoczono za [7].

Obraz	UCM	ECECOW	S+P	JPEG-LS	Sunset	CALIC
Balloon	2.81	2.86	2.97	2.90	2.89	<b>2.78</b>
Barb 1	4.44	4.34	4.53	4.69	4.68	<b>4.33</b>
Zelda	3.80	<b>3.71</b>	3.84	3.89	3.81	3.72
Hotel	4.28	4.38	4.53	4.38	4.46	<b>4.22</b>
Barb 2	4.57	4.57	4.71	4.69	4.77	<b>4.49</b>
Board	3.57	3.62	3.82	3.68	3.73	<b>3.50</b>
Girl	3.81	3.81	3.96	3.93	3.91	<b>3.71</b>
Gold	4.45	4.42	4.56	4.48	4.56	<b>4.38</b>
Boats	3.85	3.86	4.03	3.93	4.01	<b>3.77</b>
Lena	4.17	4.09	4.16	4.24	4.21	<b>4.04</b>
Średnio	3.97	3.98	4.12	4.08	4.10	<b>3.89</b>

Techniki wymienione w tabeli 7.5 to:

- UCM (ang. Universal Context Modelling algorithm) - uniwersalny algorytm modelowania kontekstu opracowany przez Weinbergera [8], który ze względu na duży stopień złożoności i duże koszty czasowo-pamięciowe nie nadaje się właściwie do praktycznych zastosowań.
- ECECOW (ang. Embedded Conditional Entropy Coding of Wavelet Coefficients) - metoda zagnieżdżonego kodowania współczynników falkowej transformaty z wykorzystaniem entropii warunkowej i modelowaniem kontekstu wysokiego rzędu opracowana przez X. Wu [7]. Technika ta w wersji bezstratnej wykorzystuje odwracalną transformatę falkową (z całkowitymi współczynnikami), prosty zagnieżdżony algorytm sukcesywnej kwantyzacji, a koncentruje się na modelowaniu prawdopodobieństw warunkowych przy kodowaniu strumienia współczynników. Wykorzystuje tutaj podobny pomysł jak w metodzie CALIC: modelowanie kontekstu wysokiego rzędu i jego kwantyzacja, przy czym szczególną uwagę zwraca na oddzielne kontekstowe kodowanie znaku współczynników.
- S+P - bezstratna wersja techniki kompresji Saida i Pearlmana, której zasadniczy pomysł polega na partycjonowaniu podzbiorów hierarchicznego drzewa współczynników, tzw. drzewa zer, któremu towarzyszy zwiększenie efektywności kodowania tychże współczynników [9].
- JPEG-LS - standard bezstratnej kompresji [6] o znacznie zwiększonej efektywności w stosunku do poprzedniej wersji odwracalnego JPEG-a.
- Sunset - odwracalny koder Langdona i Haidinyaka, oparty zasadniczo na kształtowaniu i parametryzacji zależnego od kontekstu rozkładu wartości kodowanych danych [10].

Przedstawione wyniki potwierdzają bardzo dobrą skuteczność techniki CALIC. Prawie dla każdego z dziesięciu obrazów testowych ze standardu JPEG metoda ta dała najmniejszą średnią bitową skompresowanej reprezentacji.

## 7.6. Bezstratna kompresja sekwencji obrazów

Obok korelacji przestrzennych w pojedynczych obrazach coraz bardziej palącym problemem staje się szybkie przesyłanie czy zapis długich sekwencji obrazów. Nadmiarowość czasowa, wynikająca z podobieństwa kolejnych obrazów w sekwencji jest często na znaczącym poziomie, jak chociażby w przypadku telekonferencji. Na długich seriach kolejnych klatek widnieje wtedy ta sama osoba wykonująca niezbyt gwałtowne ruchy, mówiąca i gestykulująca, czy też innym razem widoczna jest jedynie twarz ze zmieniającą się mimiką. Także występujące w większości obrazowych badań medycznych sekwencje obrazów skorelowanych czasowo (np. zapis pracy serca w echokardiografii) i przestrzennie (np. kolejne warstwy mózgu w badaniach CT) mogą być kompresowane skuteczniej przy pomocy metod, które eliminują tę międzyobrazową nadmiarowość. Wykorzystuje się tutaj techniki kodowania międzyobrazowego (ang. *interframe*) kodujące jednocześnie całą grupę obrazów, bazując na podobieństwie kolejnych ramek danej sekwencji. Okazują się one bardziej efektywne od technik jednoobrazowych (ang. *intraframe*) niezależnej, oddzielnej kompresji każdego z obrazów w sekwencji.

Techniki międzyobrazowego kodowania sekwencji obrazów stosowane są częściej w wersji stratnej (standard MPEG), ale także jako bezstratne, szczególnie w przypadku obrazów medycznych, kiedy nie można sobie pozwolić na utratę jakiegokolwiek informacji z obrazów oryginalnych. Metody te jednak nie dają zazwyczaj (oprócz przypadków trywialnych) znaczącej poprawy skuteczności kompresji w stosunku do metod jednoobrazowych, są jednak godne odnotowania. Przyjrzyjmy się niektórym rozwiązaniom.

W pracy Roosa i Viergevera [11] przeprowadzono obszerną ocenę różnych technik międzyobrazowego kodowania na znaczącym zbiorze obrazów testowych, przy czym badano redukcję nadmiarowości czasowej. Kompresowano metodą bezstratną angiogramy rentgenowskie (serię 15 obrazów  $512 \times 512 \times 9b$ , zarejestrowanych z szybkością 30 obrazów na sekundę, 14 obrazów  $1512 \times 1512 \times 8b$  przy rejestracji 25 obrazów/s, 18 obrazów  $256 \times 256 \times 8b$  przy rejestracji 25 obrazów/s oraz 18 obrazów  $256 \times 256 \times 8b$  przy rejestracji 12.5 obrazów/s) oraz scyntygramy wątroby (40 obrazów  $128 \times 128 \times 8b$  przy rejestracji 1 obrazu na minutę). Stosowano różne metody dekorelacji czasowej, takie jak nierejestrującą dekorelację (ang. *unregistered decorrelation*) bez kompensacji ruchu obiektów występujących w obrazie, blokową dekorelację z kompensacją ruchu (ang. *block motion-compensated decorrelation*) oraz lokalną dekorelację z kompensacją ruchu (ang. *local motion-compensated decorrelation*) z wykorzystaniem metod zaproponowanych przez Netraliego i Robbinsa oraz wienerowskich. Metody te były realizowane w wersji ekstrapolacyjnej (estymacja każdego obrazu jedynie na podstawie obrazów poprzednich) oraz interpolacyjnej (estymacja obrazu przy pomocy obrazów zarejestrowanych zarówno wcześniej jak i później). Po fazie dekorelacji czasowej obrazy były następnie kompresowane przy pomocy techniki HINT.

Schemat dekorelacji nierejestrującej zarówno w wersji ekstrapolacyjnej jak i interpolacyjnej jest bardzo prosty. Bazuje on na porównaniu pojedynczych pikseli z kadrów sąsiednich w sekwencji. Jeśli przez  $f(\mathbf{x}, t)$ ,  $t = 0, 1, 2, \dots$  oznaczymy czasową sekwencję obrazów ( $\mathbf{x}$  jest przestrzenią określoności funkcji jasność), a przez  $\hat{f}(\mathbf{x}, t)$  ich estymatę, to przy ekstrapolacji mamy:

$$\hat{f}(\mathbf{x}, t) = f(\mathbf{x}, t - 1), \quad t = 1, 2, 3, \dots \quad (7.30)$$

Interpolacja uwzględnia informacje o danym kadrze zawartą w sąsiednich obrazach z dwu stron w kolejności czasowej, a więc:

$$\hat{f}(\mathbf{x}, 2^k(2t+1)) = \text{round} \left\{ \frac{1}{2} [f(\mathbf{x}, 2^{k+1}t) + f(\mathbf{x}, 2^{k+1}(t+1))] \right\} \quad k, t = 0, 1, 2, 3, \dots \quad (7.31)$$

gdzie  $k$  oznacza poziom rozdzielczości czasowej, przy czym najwyższy poziom (wszystkie kadry sekwencji oryginalnej) występuje dla  $k=0$ .

Po wyznaczeniu wartości przewidywanych kodowana jest wartość błędu predykcji:  $e(\mathbf{x}, t) = f(\mathbf{x}, t) - \hat{f}(\mathbf{x}, t)$  w przypadku każdego kolejnego kadru sekwencji.

W metodach blokowych bada się już podobieństwo bloków pikseli w obrazach sąsiednich wybierając najbardziej podobne w sensie przyjętej miary. Estymacja ruchu jest jednak prosta, sprowadzająca się jedynie do wyznaczenia parametrów translacji najbardziej podobnych bloków z sąsiednich kadrów na siebie. Bardziej złożona estymacja ruchu występuje w trzeciej kategorii metod lokalnej kompensacji ruchu, kiedy to wyznaczany jest wektor przemieszczenia dla każdego piksela. Uzyskuje się w ten sposób pole wektorowe, gdzie wszystkie geometryczne przekształcenia w płaszczyźnie obrazu są opisane przez lokalne translacje. Są to schematy bardzo złożone i czasochłonne.

Uzyskane rezultaty pokazują, że poprawa skuteczności kompresji poprzez kodowanie międzyobrazowe jest niewielka w porównaniu z niezależną kompresją każdego z obrazów (bez dekorelacji czasowej) przy pomocy techniki HINT. W niektórych przypadkach wyniki są porównywalne, przy czym oczywiście koszty czasowe technik z kompensacją ruchu są większe. Okazuje się, że interpolacyjne algorytmy kompresji są nieco skuteczniejsze od ekstrapolacyjnych (uśredniający charakter interpolacyjnej estymacji silniej tłumi szumy czasowe) w stosunku 5% do 3% zmniejszenia długości kodu w stosunku do niezależnego HINT. Spośród wszystkich stosowanych metod dekorelacji czasowej najefektywniejsza była metoda wienerowskiej interpolacji. Metoda nierejestrującej dekorelacji daje zbliżone stopnie kompresji w stosunku do bardziej złożonych metod kodowania międzyobrazowego, przy znaczących czasowych oszczędnościach.

Powodem nieskuteczności metod dekorelacji czasowej przy kompresji sekwencji obrazów medycznych może być z jednej strony niski poziom korelacji czasowej w stosunku do silnej korelacji przestrzennej występującej w każdym z obrazów (silny szum czasowy) lub też bardziej prawdopodobna słaba estymacja przesunięć pikseli czy grup pikseli w kolejnych obrazach (trudne do uwzględnienia takie zjawiska, jak zmiany średniej jasności, ruch w kierunku prostopadłym do obrazu, bardziej złożony ruch w płaszczyźnie obrazu, który nie da się dokładnie opisać przy pomocy liniowej translacji). Omawiane sposoby dekorelacji mogą też znaleźć zastosowanie w kompresji sekwencji obrazów skorelowanych przestrzennie, np. kolejnych warstw dużego organu uzyskanych w tomografii.

Metody kodowania międzyobrazowego są zazwyczaj znacznie skuteczniejsze w przypadku kompresji sekwencji obrazów video, telekonferencji czy nawet naturalnych obrazów filmowych, ale tam z racji potencjalnych możliwości silnej selekcji informacji stosuje się z dużym powodzeniem techniki stratne. Wspomniane techniki w zastosowaniu do wybranej sekwencji wideo dały blisko 13% poprawę skuteczności kompresji w stosunku do metod niezależnych (jednoobrazowych).



## Bibliografia:

1. Y.-Q. Zhang, M.H. Loew, R.L. Pickholtz, *A methodology for modeling the distributions of medical images and their stochastic properties*, IEEE Trans. Medical Imaging, 9(4), pp. 376-383, 1990.
2. W. Skarbek - redakcja, *Multimedia - Algorytmy i Standardy Kompresji*, Akademicka Oficyna Wydawnicza PLJ, Warszawa, rozdz. 2, 1998.
3. A. Lempel., J. Ziv, *Compression of two-dimensional data*, IEEE Trans. Information Theory, 32(3):2-8, 1986.
4. X. Wu, *Lossless compression of continuous -tone images via context selection and quantization*, IEEE Trans. Image Process., 5(6), pp. 656-664, 1996.
5. M. Weinberger, G. Seroussi, G. Sapiro, *The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS*, November 1998.
6. ISO/IEC JTC 1/SC 29/WG 1, *JPEG LS image coding system*, ISO Working Document ISO/IEC JTC 1/SC 29/WG 1 N399 - WD14495, June 1996.
7. X. Wu, *High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression*, preprint, 1997.
8. M. J. Weinberger, J. Rissanen, R. B. Arps, *On universal context modeling for lossless compression of the gray-scale images*, preprint, 1994.
9. A. Said, W. Pearlman, *An image multiresolution representation for lossless and lossy compression*, IEEE Trans. Image Process., 5(9), pp. 1303-1310, 1996.
10. G. Langdon, C. Haidinyak, *Context-dependent distribution shaping and parametrization for lossless image compression*, Applications of Digital Image Processing XVII, SPIE, pp. 62-70, 1994.
11. P. Roos, M. A. Viergever, *Reversible Interframe Compression of Medical Images: A Comparison of Decorrelation Methods*, IEEE Trans. Medical Imaging, 10(4):538-547, 1991.