

**Бомба А. Я., д.т.н., професор, Шпортько О. В., к.т.н.** (Рівненський державний гуманітарний університет)

## **ЗАСТОСУВАННЯ СИМЕТРИЧНИХ І АСИМЕТРИЧНИХ ПРЕДИКТОРІВ В ПРОЦЕСІ ПРОГРЕСУЮЧОГО СТИСНЕННЯ ЗОБРАЖЕНЬ БЕЗ ВТРАТ**

***Анотація.** Обґрунтована доцільність, наведений спосіб обходу пікселів та запропоновані предиктори для реалізації прогресуючого ієрархічного контекстно-незалежного стиснення зображень без втрат. Подано результати застосування запропонованих підходів для зменшення ентропії зображень набору АСТ в процесі попередніх перетворень.*

***Ключові слова:** прогресуюче стиснення зображень, стиснення без втрат, предиктори, ентропія.*

***Аннотация.** Обоснована целесообразность, приведен способ обхода пикселей и предложены предикторы для реализации прогрессирующего иерархического контекстно-независимого сжатия изображений без потерь. Поданы результаты применения предложенных подходов для уменьшения энтропии изображений набора АСТ в процессе предварительных преобразований.*

***Ключевые слова:** прогрессирующее сжатие изображений, сжатия без потерь, предикторы, энтропия.*

***Annotation.** The authors substantiated expedience; they produced the method of the pixels' bypass and proposed the predictors for realization of progressive hierarchical context-independent image lossless compression. The results of the offered approaches application for diminishing of the set of АСТ images entropy in the process of previous transformations are presented.*

***Keywords:** progressive image compression, lossless compression, predictors, entropy.*

**На сьогодні, у зв'язку** зі стрімким розвитком комунікаційних технологій та підвищенням інформаційних потреб суспільства, проблема стиснення зображень не втрачає своєї актуальності, адже компресія відповідних файлів дає змогу пропорційно підвищити швидкість обміну даними по мережі та зменшити обсяги використання дискового простору.

**Як відомо, кодування** пікселів зображень у поширених графічних форматах, що виконують стиснення без втрат, найчастіше здійснюється послідовно по рядках зверху вниз, а у кожному рядку – підряд зліва направо [1, с. 281]. Тому вивести стиснуте зображення у цих форматах можливо лише після декодування всіх пікселів. Як наслідок, час декомпресії зображень, кількість пікселів у яких становить сотні тисяч, при такому способі обходу може перевищувати десятки секунд незалежно від розміру області та роздільної здатності пристрою, на якому воно має бути відтворене.

При цьому одним з основних методів стиснення зображень є контекстно-незалежне кодування, яке ставить у відповідність елементам, які зустрічаються частіше, коди не більшої (найчастіше – меншої) довжини ніж для елементів, котрі трапляються рідше. Підвищити ефективність цього кодування в процесі стиснення без втрат намагаються за допомогою *предикторів*, які під час обходу прогнозують значення яскравості кожної компоненти чергового пікселя, використовуючи значення яскравостей тих самих компонентів опрацьованих раніше суміжних пікселів, оскільки вони з нею мають найбільшу степінь кореляції. В процесі використання цього підходу обчислюють і надалі кодують відхилення  $\Delta_{ij}$  значення яскравості чергової компоненти пікселя  $F_{ij}$  від прогнозованого обраним предиктором значення  $predict_{ij}$ , тобто  $\Delta_{ij} = F_{ij} - predict_{ij}$  ( $i$  та  $j$  пробігають відповідно по всіх рядках та стовпцях компонентів пікселів зображення). Суміжні піксели зображень найчастіше мають близькі кольори, а значить і близькі значення яскравостей відповідних компонентів, тому значення прогнозу часто збігається зі значенням яскравості чергової компоненти, найчастіше – є близьким до цього значення і рідко – значно відрізняється від нього. Тобто більшість значень  $\Delta_{ij}$  виявляються близькими до нуля. Тим самим застосування предикторів найчастіше збільшує нерівномірність розподілу ймовірностей значень яскравостей і, як наслідок, зменшує ентропію [1, с. 17]. Але під час послідовного обходу пікселів предиктори можуть використовувати для прогнозування лише значення яскравостей з попередніх рядків та зліва у черговому рядку, що зменшує ефективність їх застосування.

Поряд з цим, для прискорення виводу великих зображень у форматах компресії з втратами найчастіше застосовують прогресуюче (поступальне) ієрархічне стиснення [2, с. 176]. У процесі застосування цього підходу

кожне зображення опрацьовують пошарово, збільшуючи щоразу роздільну здатність, причому в процесі послідовної обробки даних чергового шару використовують дані попереднього шару. Зображення з пікселів чергового шару фактично є зменшеною у декілька разів (найчастіше – у чотири) копією зображення з пікселів наступного шару, а останній шар співпадає з вхідним зображенням. Тому під час прогресуючого ієрархічного декодування деталі зображення проявляються поступово. Зупинити таке декодування доцільно вже після декомпресії шару з кількістю пікселів, не меншою від області виводу по кожній з осей, не очікуючи відтворення всіх пікселів зображення.

**Отже, розробка методів і графічного формату компресії зображень без втрат з використанням принципів прогресуючого ієрархічного опрацювання, що є метою дослідження, є на сьогодні актуальним завданням.**

**Враховуючи зазначене,** ми пропонуємо застосовувати принципи прогресуючого ієрархічного обходу пікселів і в процесі стиснення зображень без втрат. Для цього на першому шарі послідовно обійдемо піксели зображення, починаючи з першого, по рядках зверху вниз, а у кожному рядку – підряд зліва направо з кроком  $h_1 = 2^k$ , де  $k$  визначається

$$\text{з умови } k = \left\lfloor \log_2 \left( \frac{\max(\min(\text{height}, \text{width}), 16) - 1}{15} \right) \right\rfloor, \quad \text{height} -$$

кількість рядків,  $\text{width}$  – кількість стовпців пікселів зображення. Цей крок забезпечує опрацювання на першому шарі принаймні 16 пікселів по кожній з осей. На наступних шарах ( $l = 2, k + 1$ ) обробимо проміжні піксели зображення в два проходи: на першому послідовно опрацюємо ті з них, які містяться на перетині діагоналей квадратів з вершинами у суміжних пікселях попереднього шару з кроком  $h_l = 2^{k+2-l}$  як по рядках, так і по стовпцях, а на другому обійдемо необроблені піксели між суміжними пікселями попереднього шару і першого проходу з тим самим кроком по стовпцях і з удвічі зменшеним – по рядках (рис. 1).

Запропонована послідовність обходу пікселів зображення дає змогу не лише прискорити декодування, коли розміри області виводу значно менші від розмірів зображення, а й застосовувати ієрархічні предиктори для прогнозування значення кожного елемента чергового пікселя (на рис. 2 позначено через  $X$ ) на всіх шарах, починаючи з другого. Для опису цих предикторів позначимо значення яскравостей аналогічних компонентів *найближчих* (суміжних) опрацьованих раніше пікселів з попереднього і чергового шару чи проходу за допомогою позначень  $a, b, c, d, ab, ad$  (рис. 2).



Рис. 1. Схема прогресуючого ієрархічного опрацювання пікселів зображення на шарах, починаючи з другого: П – пікселі попереднього шару; 1- пікселі першого проходу чергового шару; 2 - пікселі другого проходу чергового шару.

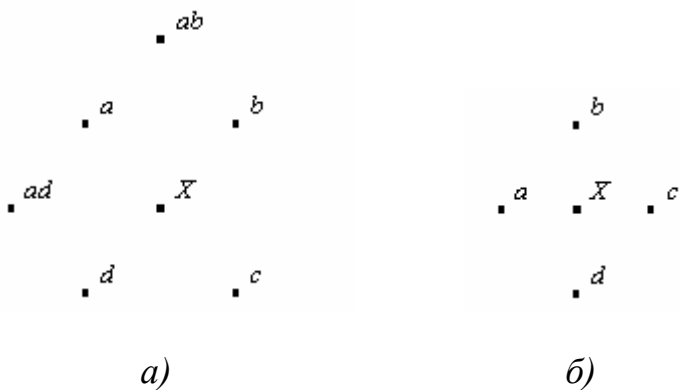


Рис. 2. Схеми розміщення суміжних опрацьованих елементів для елемента X на шарах, починаючи з другого: а) для першого проходу; б) для другого проходу

Використовуючи ці позначення, дослідимо принципи прогнозування двох основних симетричних ієрархічних предикторів, *орієнтованих на базове середнє арифметичне* (шумова складова) *тих двох протилежних елементів з найближчих чотирьох, які найменше між собою відрізняються* (трендова складова), тобто найімовірніше належать одному об'єкту на зображенні. У випадку, коли більший вплив на яскравості компонентів близьких пікселів мають слабкі фонові коливання, спрогнозуємо значення чергового елемента за

допомогою трендово-шумового предиктора *ProgresPredict 1*, який повертає базове середнє арифметичне. Якщо ж переважають сильні коливання яскравостей пікселів, то спрогнозуємо значення чергового елемента за допомогою шумово-трендового предиктора *ProgresPredict 2*. Цей предиктор серед чотирьох суміжних опрацьованих елементів  $a, b, c, d$  визначає і повертає найближче значення до базового середнього арифметичного, коли таке значення єдине. Якщо ж серед суміжних опрацьованих елементів є два найближчих рівновіддалених значення до базового середнього арифметичного, то серед них повертається те, яке частіше повторюється, а коли й кількість їх повторень однакова – то *менше* з цих двох значень. Загальна орієнтація на менші значення дає змогу змістити ненульові відхилення  $\Delta_{ij}$  в бік додатних значень і цим самим підвищити нерівномірність їх розподілу. Мовою C ці предиктори записуються так:

```

ubyte ProgresPredict1(ubyte a, ubyte b, ubyte c, ubyte d)
{if (abs(a-c)<=abs(b-d)) return (a+c)/2
 else return (b+d)/2; }

ubyte ProgresPredict2(ubyte a, ubyte b, ubyte c, ubyte d)
{ubyte absac, absbd, maxac, minac, maxbd, minbd, prognoz;
 if (a<=c) {absac=c-a; maxac=c; minac=a;}
 else {absac=a-c; maxac=a; minac=c; }
 if (b<=d) {absbd=d-b; maxbd=d; minbd=b; }
 else {absbd=b-d; maxbd=b; minbd=d; }
 if (absac<=absbd)
 {if (minbd>=minac && minbd<=maxac) return minbd;
  if (maxbd>=minac && maxbd<=maxac) return maxbd;
  return minac; }
 else
 {if (minac>=minbd && minac<=maxbd) return minac;
  if (maxac>=minbd && maxac<=maxbd) return maxac;
  return minbd; }}

```

Проаналізуємо результати застосування запропонованих предикторів ієрархічного обходу (табл. 1) для зменшення ентропії компонентів пікселів стандартного тестового набору Archive Comparison Test (ACT), який містить як синтезовані (№№ 1 (з шумами), 2, 7), так і фотореалістичні (всі інші) зображення (для порівняння у рядку *NonePredict* цієї таблиці наведена ентропія яскравостей компонентів пікселів без застосування предикторів, а нижче – ентропія цих же яскравостей після використання найпоширеніших предикторів послідовного обходу [3; 4, с. 315]). Завантажити TIFF-версії цих зображень можна, наприклад, з <http://www.compression.ru/arctest/act/act-tif.htm>.

Таблиця 1

Ентропія яскравостей компонентів пікселів зображень набору АСТ після застосування різних предикторів,  $brb$

Предиктор	Шар	Прохід	№ файла								Середня ентропія
			1	2	3	4	5	6	7	8	
NonePredict	1	1	7.54	4.65	7.75	7.50	7.66	7.32	5.99	7.66	7.01
LeftPredict	1	1	3.57	1.68	5.26	4.57	4.60	5.67	1.74	5.13	4.03
AbovePredict	1	1	3.68	2.23	4.92	4.60	4.72	5.97	1.86	4.94	4.12
AveragePredict	1	1	4.66	2.88	4.86	4.27	4.40	5.59	2.58	4.74	4.25
PaethPredict	1	1	1.90	1.51	4.90	4.26	4.29	5.50	1.35	4.61	3.54
MedPredict	1	1	1.90	1.54	4.84	4.15	4.20	5.43	1.37	4.50	3.49
ProgresPredict1	2	1	7.46	6.63	6.93	6.83	7.17	7.28	7.08	7.33	7.09
	2	2	7.05	5.80	6.19	6.43	6.07	6.96	6.11	7.10	6.46
	3	1	7.26	6.36	6.43	6.44	6.61	7.12	6.34	6.91	6.68
	3	2	6.70	5.66	5.80	6.02	5.69	6.81	5.27	6.53	6.06
	$k+1$	1	5.65	2.88	4.80	4.11	4.44	5.60	2.85	4.53	4.36
	$k+1$	2	3.99	1.89	4.45	3.61	3.83	4.93	1.73	3.87	3.54
	Разом		4.89	2.56	4.73	4.04	4.22	5.41	2.47	4.42	4.09
ProgresPredict2	2	1	7.40	5.68	6.99	6.89	7.21	7.34	6.39	7.32	6.90
	2	2	7.04	5.18	6.18	6.49	6.12	6.98	5.22	7.02	6.28
	3	1	7.00	5.36	6.54	6.56	6.71	7.24	5.59	6.94	6.49
	3	2	6.69	4.97	5.88	6.11	5.82	6.88	4.27	6.52	5.89
	$k+1$	1	4.73	2.07	5.03	4.49	4.65	5.84	1.90	4.95	4.21
	$k+1$	2	0.23	1.19	4.68	4.03	4.11	5.24	0.96	4.39	3.10
	Разом		2.75	1.81	4.94	4.41	4.50	5.66	1.64	4.83	3.82

Як і слід було чекати, застосування шумово-трендового предиктора виявилося ефективнішим для синтезованих і на деяких початкових проходах окремих (№№ 3, 8) фотореалістичних зображень, оскільки для них характерні різкі перепади яскравостей опрацьованих пікселів на межах зображених об'єктів (переважає вплив тренду), а трендово-шумового – загалом для фотореалістичних зображень, адже їх суміжні піксели найчастіше мають близькі, але неоднакові кольори (переважає вплив шуму). Використання симетричних предикторів істотно зменшує ентропію на останніх шарах у порівнянні з іншими відомими на сьогодні предикторами послідовного обходу, оскільки ці предиктори враховують вплив чотирьох рівновіддалених пікселів з різних боків, а не лише пікселів зліва і зверху, як у випадку послідовного обходу. При цьому кожен опрацьований піксел чергового шару в середині зображення використовується для обробки восьми пікселів наступного шару (а не для максимум трьох наступних пікселів, як для предикторів послідовного обходу).

Рівень кореляції яскравостей компонентів суміжних пікселів на останніх шарах істотно вищий від рівня кореляції цих яскравостей на перших шарах, особливо для синтезованих зображень. Ось чому застосування предикторів послідовного обходу (а інколи – і відмова від їх використання) забезпечують меншу ентропію від прогресуючих на початкових шарах, хоча ефективність останніх зі збільшенням номера шару зростає. Тому, щоб загалом не збільшувати ентропію, для синтезованого зображення № 2 доцільно відмовитися від застосування предикторів як на другому, так і на третьому шарі, а для зображення № 7 – на першому проході другого шару.

Крім цього, під час виконання оптимізованого варіанту шумово-трендового предиктора *ProgresPredict2* в середньому здійснюється 6.33 операцій порівняння, а трендово-шумового предиктора *ProgresPredict1* – лише 3 таких операції. Тобто використання трендово-шумового предиктора замість шумово-трендового зменшує кількість таких операцій у понад 2 рази, тому предиктору *ProgresPredict1* слід надавати перевагу перед *ProgresPredict2* при однакових значеннях ентропії після їх застосування. Для порівняння: в процесі виконання нелінійного статичного предиктора послідовного обходу *PaethPredict* в середньому виконується 5.66, а *MedPredict* – 5 операцій порівняння. Тобто застосування трендово-шумового предиктора *ProgresPredict1* навіть замість нелінійних предикторів послідовного обходу прискорює кодування/декодування.

Додатково підвищити ефективність шумово-трендових предикторів можливо за рахунок аналогічного врахування в асиметричних предикторах на першому проході кожного шару значень суміжних опрацьованих елементів зліва і зверху цього ж проходу (на рис. 2 а вони позначені відповідно через *ad* і *ab*). Адже кольори пікселів найчастіше подібні до кольорів близьких пікселів саме по горизонталі чи вертикалі і значно рідше – до кольорів близьких пікселів по діагоналі. Цим (а не лише більшими відстанями до прогнозованого елемента) пояснюється значно нижча ефективність застосування симетричних предикторів на першому проході відносно другого для кожного шару довільного зображення.

Наведені результати дослідження вказують на те, що предиктори слід використовувати лише з того проходу і того шару, коли їх застосування починає зменшувати ентропію, а серед предикторів необхідно обирати той, який забезпечує найменшу ентропію після свого застосування. Більше того, рівень впливу і характер шумів та трендів для різних фрагментів зображення може відрізнитися, тому для них можуть виявитися ефективними різні ієрархічні предиктори. Підкреслимо, що відмовлятися від застосування предикторів слід для цілого проходу, а не для окремих його фрагментів, оскільки значення яскравостей компонентів зображень без та після використання предикторів мають різний характер нерівномірностей розподілу. Тому в процесі прогресуючого ієрархічного

стиснення для кожного проходу чергового шару доцільно спочатку встановити, чи застосування предикторів загалом зменшує ентропію яскравостей компонентів його пікселів, після чого, у випадку прийняття рішення про застосування предикторів, обрати для кожного однорідного фрагмента пікселів проходу саме той ієрархічний предиктор, який максимально зменшує його ентропію.

**За результатами** дослідження можна зробити такі **висновки**:

1. В нових версіях графічних форматів та нових форматах компресії зображень без втрат доцільно реалізувати прогресуюче ієрархічне стиснення, оскільки це дозволяє суттєво прискорити декодування, коли розміри області виводу менші від розмірів зображення.

2. Зменшення розмірів стиснутих прогресуючим ієрархічним способом зображень досягається в основному на останніх шарах, оскільки піксели, які при цьому використовуються предикторами, мають з прогнозованим пікселем в середньому найвищий рівень кореляції відносно інших шарів.

3. На першому проході кожного шару в процесі ієрархічного стиснення підвищити ефективність застосування симетричних предикторів, які враховують яскравості опрацьованих найближчих пікселів по діагоналі, можливо за рахунок врахування яскравостей опрацьованих на цьому ж проході найближчих суміжних пікселів по горизонталі та вертикалі.

4. Під час контекстно-незалежного стиснення без втрат для однорідних фрагментів зображень доцільно обирати той предиктор з декількох альтернативних, який дає змогу максимально зменшити ентропію.

Надалі, з метою додаткового зменшення розмірів файлів стиснутих зображень без втрат і прискорення декодування, планується адаптувати запропоновані контекстно-залежні методи компресії до ієрархічного обходу пікселів зображень.

1. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М.: ДИАЛОГ-МИФИ, 2003. – 384 с. 2. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон. – М.: Техносфера, 2006. – 368 с. – (Серия: Мир программирования: цифровая обработка сигналов). 3. Бредихин Д. Ю. Сжатие графики без потерь качества [Электронный ресурс] / Д. Ю. Бредихин. – 2004. – Режим доступа: [http://www.compression.ru/download/articles/i\\_less/bredikhin\\_2004\\_lossless\\_image\\_compression\\_doc.rar](http://www.compression.ru/download/articles/i_less/bredikhin_2004_lossless_image_compression_doc.rar). – Назва з екрану. 4. Миано Дж. Форматы и алгоритмы сжатия изображений в действии: Учеб. пособ. / Дж. Миано. – М.: Триумф, 2003. – 336 с., ил. – (Серия: Практика программирования).

Рецензент: д.т.н., професор Власюк А. П.