

Фрактальное сжатие изображений. Некоторый опыт сравнения алгоритмов

Александр Бубличенко

ДонНТУ, КЭМ-07М

al@bliqo.com

Руководитель: В.Н. Беловодский

Основы фрактального сжатия изображений

- IFS (Iterated Function System) — система итерируемых функций (аффинных преобразований), переводящих одно изображение в другое по заданным коэффициентам преобразования. Преобразованию подвергаются участки изображения (x координата, y координата, яркость).

$$w \begin{pmatrix} x \\ y \\ I \end{pmatrix} = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & u \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ I \end{pmatrix} + \begin{pmatrix} e \\ f \\ v \end{pmatrix}$$

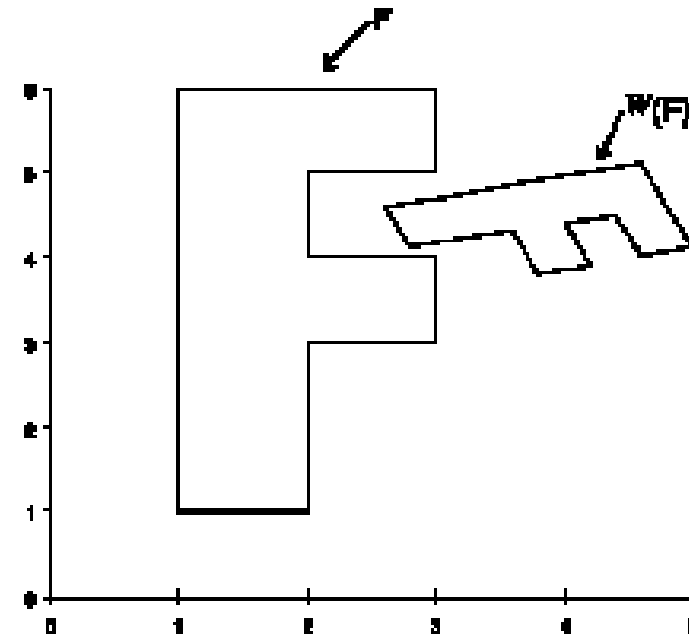
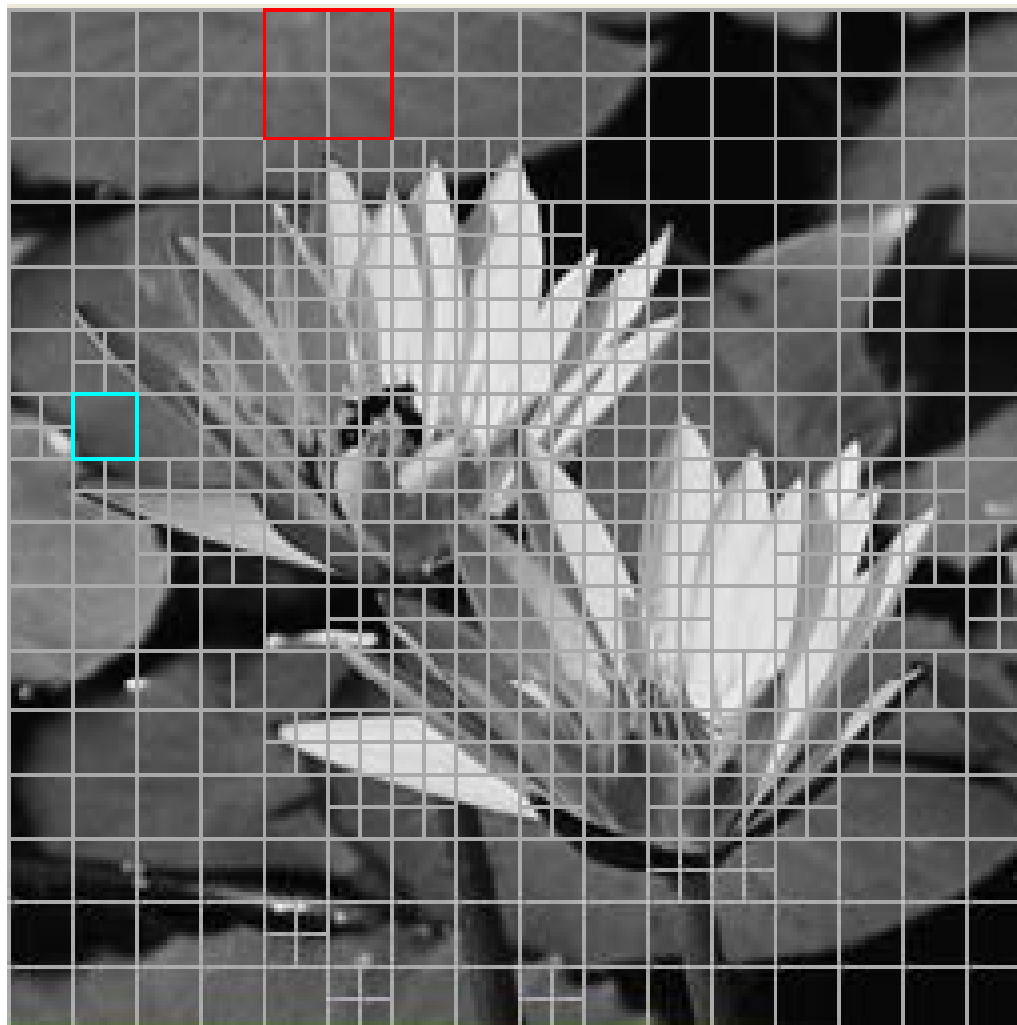


Abb.3

Основная идея фрактального сжатия изображений

- Заключается в поиске коэффициентов сжимающих преобразований, которые отображают доменные блоки изображения в ранговые блоки.
- Доменные блоки могут пересекаться, а ранговые блоки полностью покрывают изображение и не пересекаются.
- Доменные блоки по размерам должны быть больше ранговых блоков.
- Полученный в результате, этот набор коэффициентов преобразований — есть код изображения.
- Закодированное изображение можно декодировать путем итеративного применения каждого преобразования с известными коэффициентами к соответствующему участку произвольного начального изображения.

Отображение доменных блоков изображения в ранговые



- Доменный блок



- Ранговый блок

Свойства алгоритмов фрактального сжатия изображений

- 1) высокий (от нескольких единиц до тысяч) коэффициент сжатия¹, особенно для изображений, обильных самоподобием;
- 2) время кодирования намного больше времени декодирования;
- 3) качество декодированного изображения одинаково при различных масштабах его просмотра («искусственные детали»)

1 – коэффициент сжатия характеризует во сколько раз сжатое изображение меньше исходного по объему информации

Качество декодированного изображения при различных масштабах его просмотра



Базовый алгоритм фрактального сжатия изображений

Шаг 1. В изображении f выделяют множество доменных блоков, с возможностью их перекрытия.

Шаг 2. Изображение разбивают на не перекрывающиеся ранговые блоки $\{R_i\}$. Блоки R_i могут быть одинакового размера, но более эффективно использование адаптивного разбиения с переменным размером блоков (например, методом квадрато-дерева). Это дает возможность плотно заполнять ранговыми блоками маленького размера части изображения, содержащие мелкие детали.

Базовый алгоритм фрактального сжатия изображений

Шаг 3. Для каждого рангового блока перебирают все доменные блоки. При этом над доменами выполняют операции изменения ориентации (всего 8, включая исходную ориентацию без изменений). При каждом из вариантов ориентации домен сжимают до размеров рангового блока и вычисляют оптимальные значения коэффициентов a_{ij} и b_{ij} преобразования

$$w_{ij}(d_{xy}) = a_{ij} \cdot d_{xy} + b_{ij}$$

методом наименьших квадратов для получения наилучшего соответствия ранговому блоку.

Где $r_{xy} \in R_i$, $d_{xy} \in D'_{ij}$, D'_{ij} — i -ый доменный блок в j -ой ориентации, сжатый до размеров рангового блока.

Базовый алгоритм фрактального сжатия изображений

Шаг 3 (продолжение). Далее вычисляют нормированное значение параметра

$$k_{ij} = \sum_{R_i} |w_{ij}(d_{xy}) - r_{xy}| / (N_{R_i} \cdot 256)$$

который характеризует соответствие преобразованного (с вычисленными коэффициентами) i -го доменного блока D'_{ij} в его j -ой ориентации ранговому блоку R_i .

Где:

N_{R_i} — количество пикселей в ранговом блоке,
 $r_{xy} \in R_i$, $d_{xy} \in D'_{ij}$

Базовый алгоритм фрактального сжатия изображений

Шаг 3 (продолжение). Если для некоторого i -го домена и его j -ой ориентации значение k_{ij} не превышает заданной допустимой погрешности (например, $k_{ij} \leq 0.05$), то процедуру перебора доменов останавливают, рассматриваемый ранговый блок считают покрытым i -ым доменным блоком в его j -ой ориентации и переходят к обработке следующего ранга.

Если после перебора всех доменных блоков не нашлось таких, значения k_{ij} которых не превышают значение заданной допустимой погрешности, то делается проверка — находится ли рассматриваемый ранговый блок на максимально допустимом уровне разбиения рангов.

- Если нет, то этот ранговый блок разбивают на меньшие блоки и проводят данный шаг алгоритма для них.
- Если да, то из всех доменов выбирают тот домен и его вариант ориентации D_{ij} , значение k_{ij} которого является минимальным среди остальных, и считают рассматриваемый ранговый блок покрытым этим доменом.

Базовый алгоритм фрактального сжатия изображений

Недостаток:

- для каждого рангового блока R_i алгоритм перебирает все доменные блоки и все варианты их ориентации, проводя над каждым занимающие много машинного времени попиксельные операции изменения ориентации и нахождения коэффициентов преобразования

FE-алгоритм фрактального сжатия изображений

FE – Feature Extraction (выделение особенностей)

- выделение 5 характеристик, характеризующих доменные и ранговые блоки:
 - 1) Стандартное отклонение
 - 2) Асимметрия
 - 3) Межпиксельная контрастность
 - 4) Коэффициент бета (β) (насколько сильно отличаются значения пикселей от значения центрального пикселя)
 - 5) максимальный градиент (g)
- первоначально, проводится сравнение доменов и рангов именно по этим характеристикам, что значительно сокращает (в десятки раз) последующий объем вычислений, по сравнению с базовым алгоритмом.

FE-алгоритм фрактального сжатия изображений

1) Стандартное отклонение:

$$\sigma = \sqrt{\frac{\sum (p_{x,y} - \mu)^2}{N_I}}$$

2) Асимметрия:

$$a = \frac{\sum (p_{x,y} - \mu)^3}{N_I \cdot \sigma^3}$$

Где: l — сегмент изображения, N_I — количество пикселей в сегменте l , $p_{x,y}$ — значение пикселя в точке (x, y) , μ — среднее значение пикселя в сегменте l ;

FE-алгоритм фрактального сжатия изображений

3) Межпиксельная контрастность:

$$c = \frac{\sum_I \left| p_{x,y} - p_{x-d,y} \right| + \left| p_{x,y} - p_{x,y-d} \right|}{N_I}$$

Где: I — сегмент изображения, N_I — количество пикселей в сегменте I , $p_{x,y}$ — значение пикселя в точке (x, y) , μ — среднее значение пикселя в сегменте I , d — константа - расстояние (в пикселях) до соседнего пикселя;

FE-алгоритм фрактального сжатия изображений

- 4) Коэффициент бета (β), характеризующий насколько сильно отличаются значения пикселей от значения центрального пикселя:

$$\beta = \frac{\sum_{x=1}^{I_W} \sum_{y=1}^{I_H} \left(\omega - \sqrt{\left(x - \frac{I_W}{2}\right)^2 + \left(y - \frac{I_H}{2}\right)^2} \right) \cdot (p_{x,y} - \mu)}{\sum_{x=1}^{I_W} \sum_{y=1}^{I_H} \left(\omega - \sqrt{\left(x - \frac{I_W}{2}\right)^2 + \left(y - \frac{I_H}{2}\right)^2} \right)^2}$$

Где: I — сегмент изображения, μ — среднее значение пикселя в сегменте I , d — константа - расстояние (в пикселях) до соседнего пикселя, ω :

$$\omega = \frac{\sum_{x=1}^{I_W} \sum_{y=1}^{I_H} \sqrt{\left(x - \frac{I_W}{2}\right)^2 + \left(y - \frac{I_H}{2}\right)^2}}{N_I}$$

FE-алгоритм фрактального сжатия изображений

- 5) Максимальный градиент (g) — максимум из горизонтального (h) и вертикального (v) градиентов

$$g = \max(v, h)$$

$$h = \frac{\sum_{x=1}^{I_W} \sum_{y=1}^{I_H} \left(x - \frac{I_W}{2} \right) \cdot (p_{x,y} - \mu)}{\sum_{x=1}^{I_W} \sum_{y=1}^{I_H} \left(x - \frac{I_W}{2} \right)^2}$$

$$v = \frac{\sum_{x=1}^{I_W} \sum_{y=1}^{I_H} \left(y - \frac{I_H}{2} \right) \cdot (p_{x,y} - \mu)}{\sum_{x=1}^{I_W} \sum_{y=1}^{I_H} \left(y - \frac{I_H}{2} \right)^2}$$

FE-алгоритм фрактального сжатия изображений

Шаг 1. Аналогично шагу 1 базового алгоритма.

Шаг 2. Вычисление и хранение значений вектора характеристик для каждого доменного блока.

Шаг 3. При обработке рангового блока сначала вычисляют его вектор характеристик. Затем вычисляют расстояния между вектором характеристик данного ранга и вектором характеристик каждого домена по формуле

$$d = \sum_{j=1}^5 |f_j^R - f_j^D|$$

Где: f_j^R и f_j^D — это j -ые характеристики рангового и доменного блоков соответственно.

FE-алгоритм фрактального сжатия изображений

Для последующего сравнения выбирается только заданный q процент доменов ($q = 2\%$) с минимальными значениями расстояния d к данному рангу. Последующие действия аналогичны тем, которые выполняются в шаге 3 базового алгоритма, с той лишь разницей, что при последующем переборе доменов рассматриваются только выбранные q %.

FE-алгоритм фрактального сжатия изображений

Преимущество:

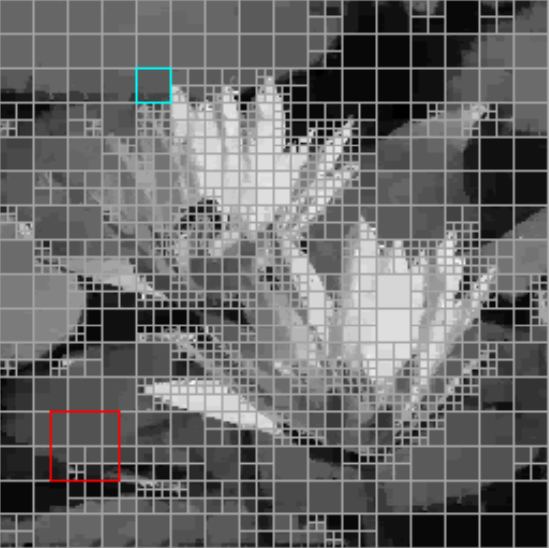
- Процедура выбора только доменов с самыми близкими расстояниями вектора характеристик к рангу является своеобразным фильтром, который существенно ограничивает количество доменов для перебора и занимающих много машинного времени операций вычисления коэффициентов преобразования.

Разработанное программное обеспечение

IFS Image Encoder/Decoder © Alexander Bublichenko, 2008

Файл Вид Инструменты Помощь

Открыть Кодировать Декодировать Итераций: 8 Масштаб: 100 %



Свойства Структура

Misc	
Время кодирования	00:01:02.5496262
Высота	256
Использование доменов, %	19.552827707145987
Использовано рангов	1228
Настройки кодирования	
Алгоритм кодирования	FeaturesExtraction
Допустимая погрешность	0.05
Искать лучший домен	False
Козф. перекрытия доменов	0.25
Макс. уровень доменов	4
Макс. уровень рангов	6
Нач. уровень доменов	3
Нач. уровень рангов	4
Процент отбираемых доменов	0.01
Обработано рангов	1552
Пикселей	65536
Средн. пикс. ошибка, %	0.0365181565284729
Ширина	256

Использование доменов, %
Показывает сколько % из всех доменов было поставлено в соответствии рангам в результате кодирования

E:\University\Master Project\Images_test4.FE.ifs | 256x256 | 00:01:02.5496262 | 00:00:03.2796975

Разработанное программное обеспечение

IFS Image Encoder/Decoder © Alexander Bublichenko, 2008

Файл Вид Инструменты Помощь

Открыть Кодировать Декодировать Итераций: 8 Масштаб: 100 %


Свойства Структура

Домены: 4562

Индекс	Положение и размеры	Ранги
0	0, 0, 32, 32	10
1	8, 0, 32, 32	0
2	16, 0, 32, 32	2

Ранги: 1228 (обработано 1552)

Исп.	Иер.инд.	Ур.	Дом	L	Abs. L	П	Инд
<input checked="" type="checkbox"/>	0000	4	0	0.011...	2.87...	0.. F	0
<input checked="" type="checkbox"/>	0001	4	30	0.013...	3.54...	1.. F	1
<input checked="" type="checkbox"/>	0002	4	0	0.010...	2.61...	0.. F	2
<input checked="" type="checkbox"/>	0003	4	30	0.009...	2.30...	1.. F	3
<input checked="" type="checkbox"/>	0010	4	4	0.014...	3.77...	3.. F	4
<input checked="" type="checkbox"/>	0011	4	641	0.019...	4.94...	4.. F	5
<input checked="" type="checkbox"/>	0012	4	33	0.014...	3.63...	3.. F	6
<input checked="" type="checkbox"/>	0013	4	2	0.011...	2.97...	4.. F	7
<input checked="" type="checkbox"/>	0020	4	805	0.046...	11.7...	0.. F	8
<input checked="" type="checkbox"/>	0021	4	698	0.039...	10.0...	1.. F	9
<input type="checkbox"/>	0022	4	586	0.066...	16.8...	0.. F	10



E:\University\Master Project\Images_test4.FE.ifs | 256x256 | 00:01:02.5496262 | 00:00:03.2796975

Сравнительный анализ алгоритмов. Эксперименты

В экспериментах используется изображение размером 256x256 пикселей

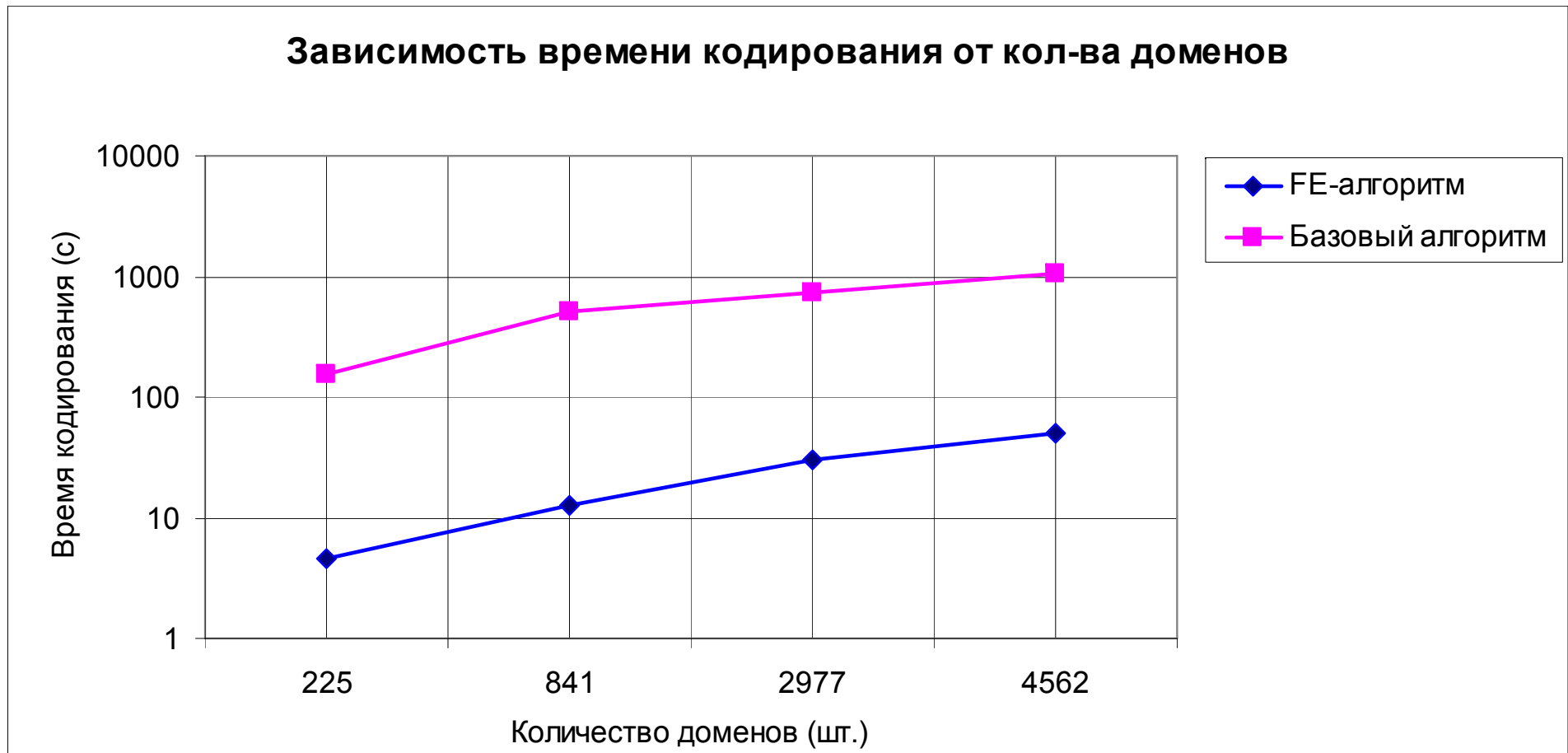


Сравнительный анализ алгоритмов. Эксперименты

Параметры	Эксперимент			
	1	2	3	4
FE-алгоритм				
Доменов	225	841	2977	4562
Рангов	1651	1504	1264	1228
Допуст.погр.	0.05	0.05	0.05	0.05
Ср. пикс. Ошибка, %	3.79	3.92	3.73	3.69
Кэфф.сжатия	3.35	3.68	4.38	4.51
Время кодирования, (с)	4.58	12.86	30.25	49.39
Время декодирования, (с)	1.91	1.62	1.41	1.25
Базовый алгоритм				
Доменов	225	841	2977	4562
Рангов	1150	1075	940	928
Допуст.погр.	0.05	0.05	0.05	0.05
Искать лучший	0	0	0	0
Ср. пикс. Ошибка, %	4.01	4.18	4.37	4.29
Кэфф.сжатия	4.81	5.14	5.88	5.50
Время кодирования, (с)	154.81	508.69	735.28	1068
Время декодирования, (с)	1.15	1.05	0.97	0.95

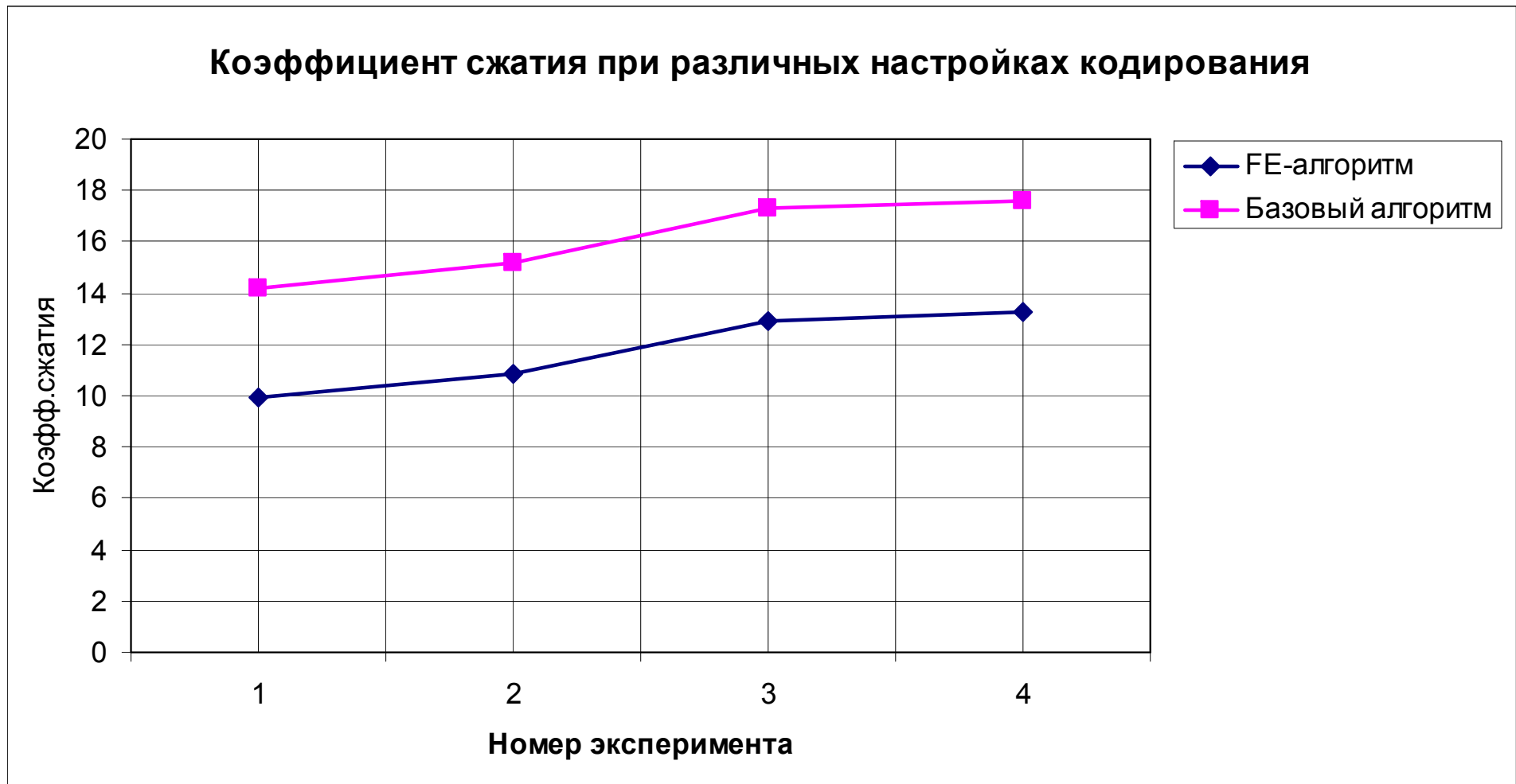
Сравнительный анализ алгоритмов. Результаты

- 1) FE-алгоритм, по сравнению с базовым, позволяет сжимать изображения за десятки раз меньшее время при одинаковых параметрах настройки алгоритмов (количество доменов, допустимая погрешность, нач. и макс. уровни рангов)



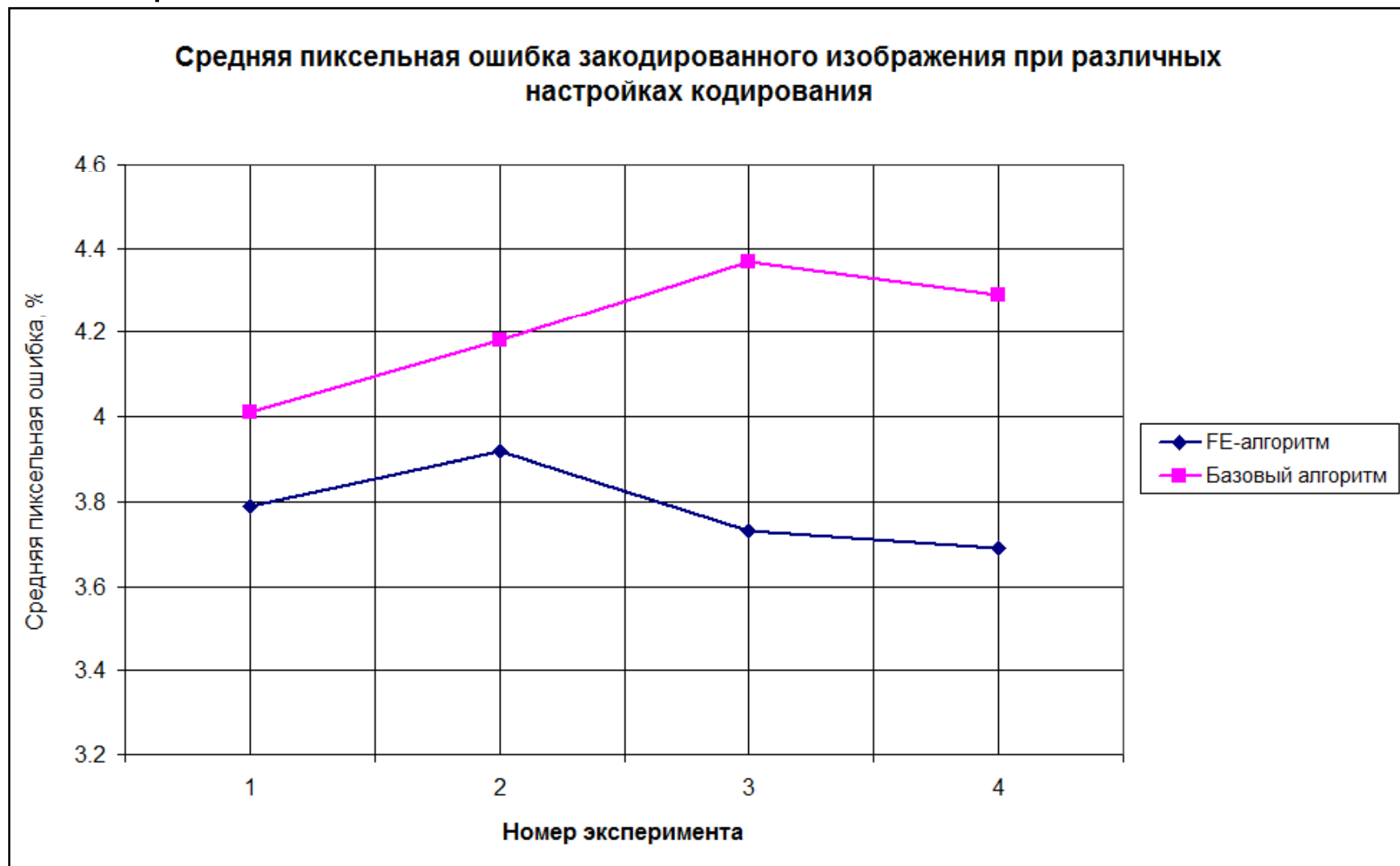
Сравнительный анализ алгоритмов. Результаты

- 2) базовый алгоритм обеспечивает в среднем на 30% более высокий коэффициент сжатия



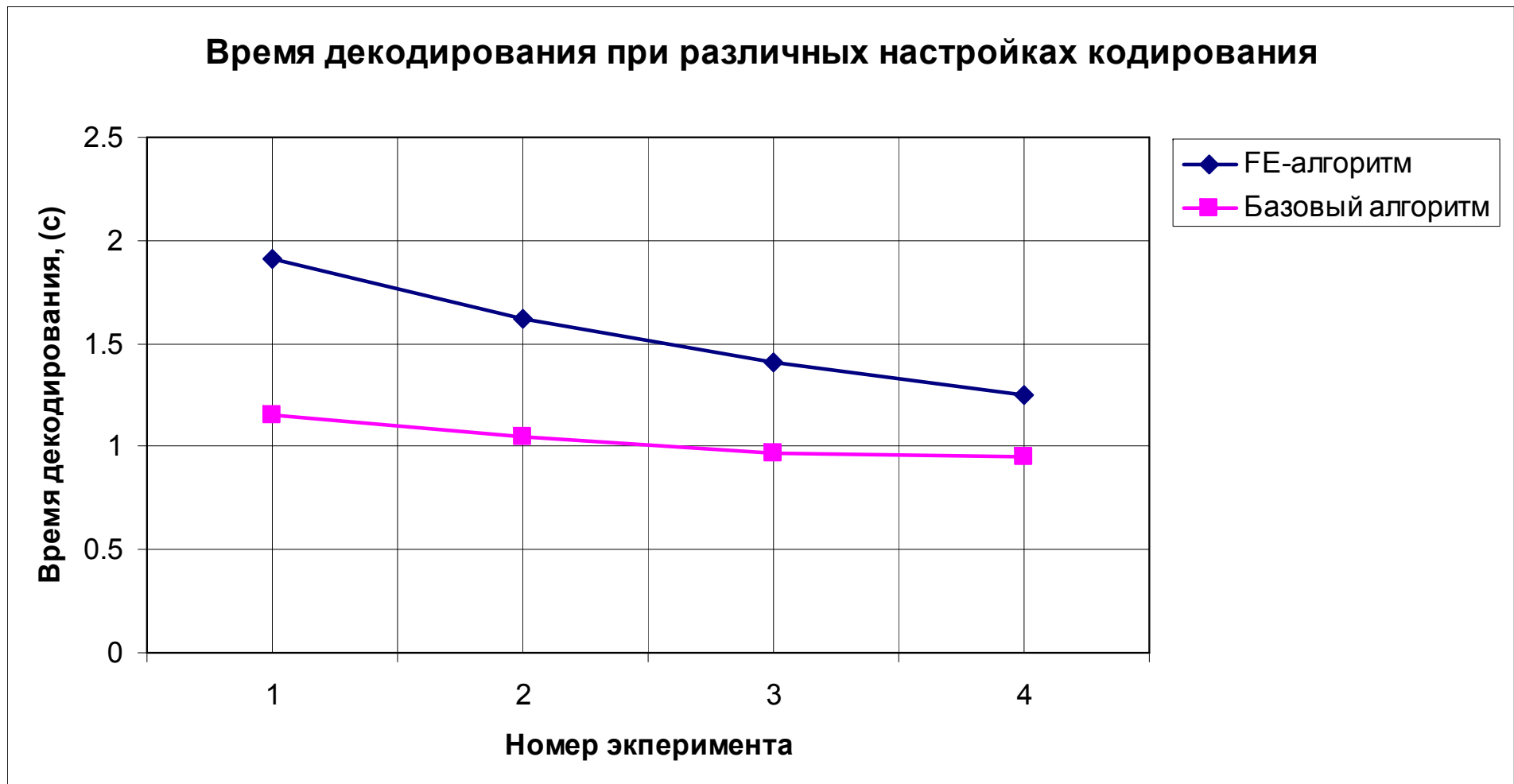
Сравнительный анализ алгоритмов. Результаты

- 3) Средняя пиксельная ошибка в результате работы обоих алгоритмов отличается незначительно



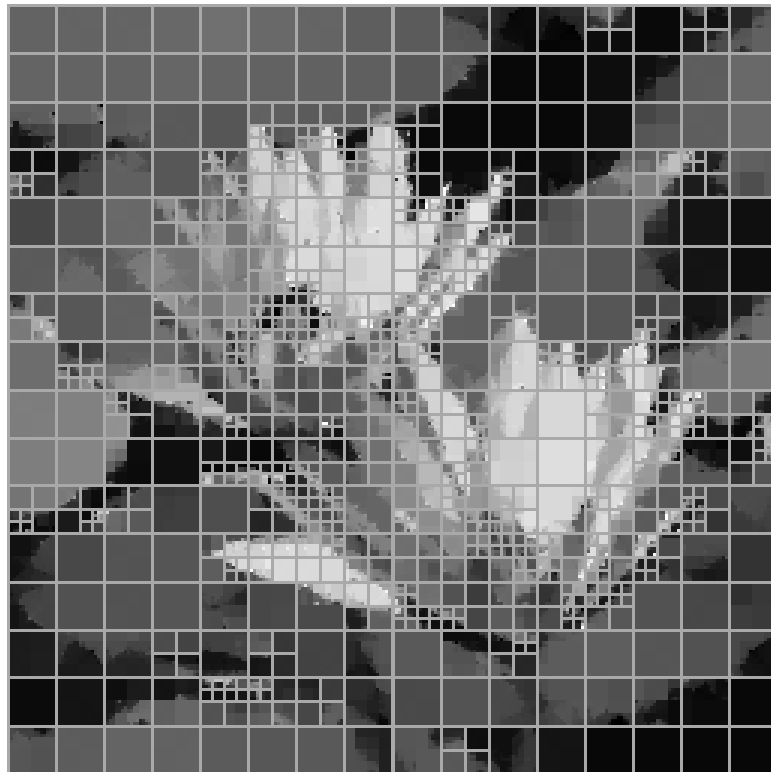
Сравнительный анализ алгоритмов. Результаты

- 4) Время декодирования закодированного изображения меньше при использовании базового алгоритма во всех экспериментах. Но при большом количестве доменов разница не существенна.

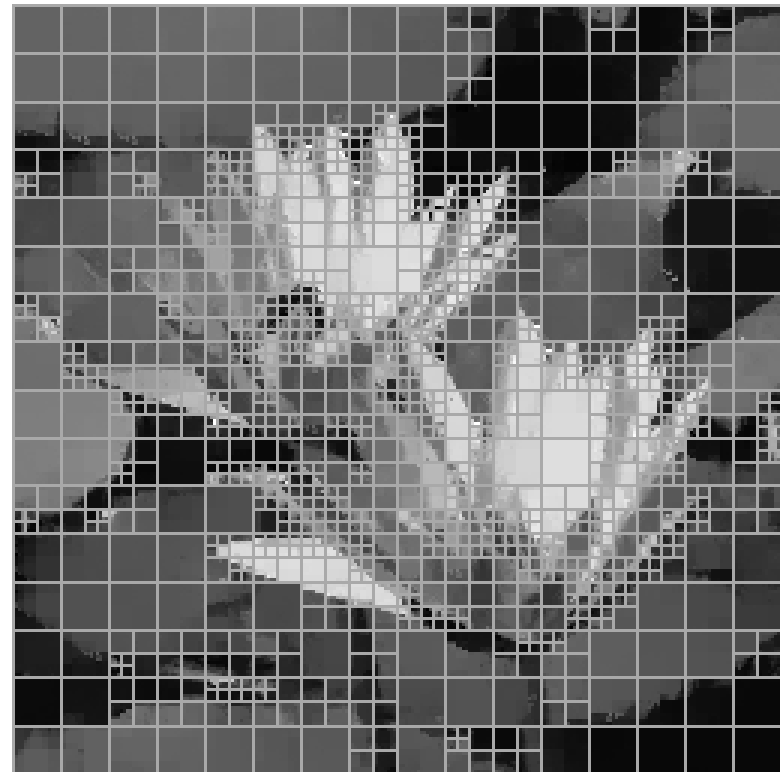


Сравнительный анализ алгоритмов. Результаты кодирования

При базовом алгоритме в результате кодирования получается меньшее количество рангов, что объясняет более высокий коэффициент сжатия

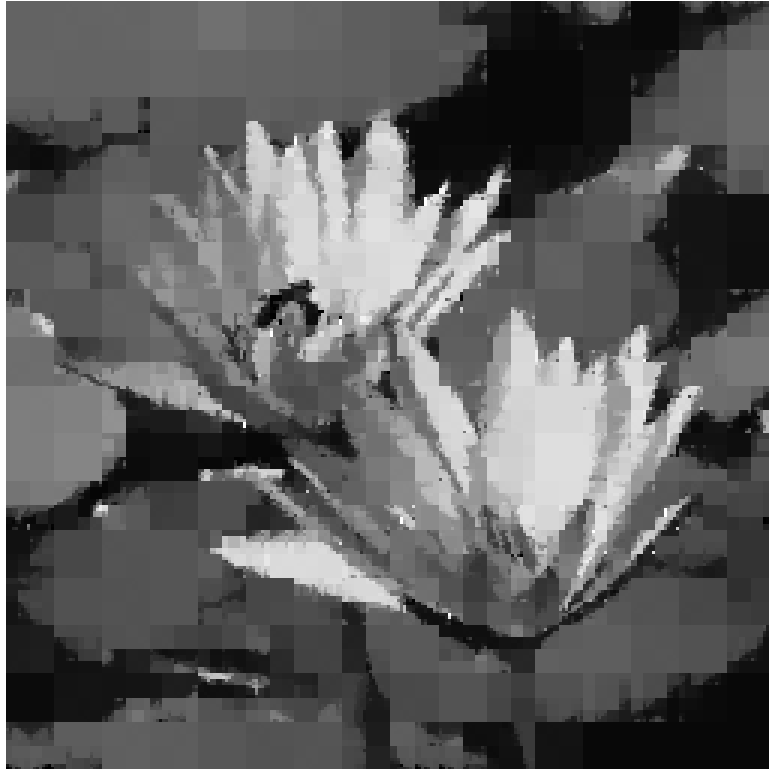


Изображение с сеткой рангов, закодированное базовым алгоритмом



Изображение с сеткой рангов, закодированное FE-алгоритмом

Декодированное изображение



Декодированное изображение,
предварительно закодированное
базовым алгоритмом



Декодированное изображение,
предварительно закодированное FE-
алгоритмом

Анимация первых 8 итераций декодирования изображения



Анимация первых 8 итераций декодирования изображения, предварительно закодированного базовым алгоритмом



Анимация первых 8 итераций декодирования изображения, предварительно закодированного FE-алгоритмом

Выводы

- FE-алгоритм, по сравнению с базовым, позволяет сжимать изображения в десятки раз меньшее время при одинаковых параметрах настроек алгоритмов
- Различаются средние пиксельные ошибки алгоритмов. В зависимости от настроек, различие находится в пределах 5-16%.
- В результате кодирования базовым алгоритмом во всех случаях достигается в среднем на 30-40% более высокий коэффициент сжатия.
- Визуальное качество декодированного изображения заметно лучше при использовании FE-алгоритма.
- Время декодирования закодированного изображения при различных настройках кодирования меньше при использовании базового алгоритма во всех экспериментах.

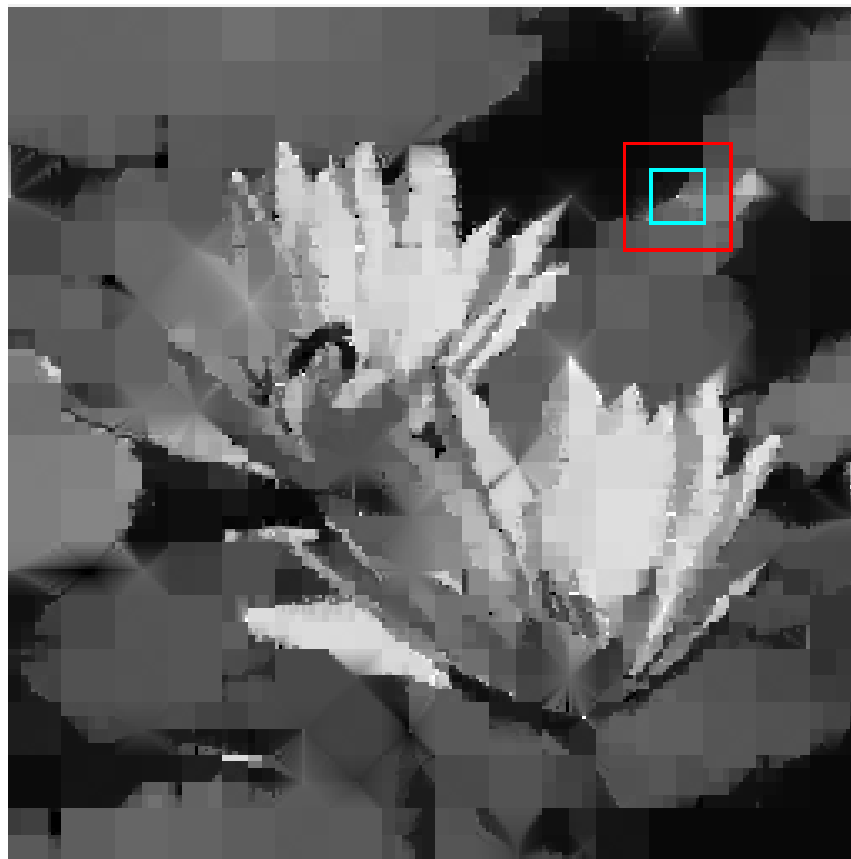
Дополнительные эксперименты. Вывод

- В результате проведения дополнительных экспериментов было установлено, что для 99.73 % ранговых блоков FE-алгоритм выбрал другие домены, т.е. — не наилучшие.
- Таким образом, по крайней мере, для данного изображения можно утверждать, что процедура отбора доменов, принятая в FE-алгоритме, не вполне адекватно отражает близость сравниваемых блоков.

Модификации и предложения

- Генерирование окаймляющих доменов.
- Многоуровневый анализ. Использование основного критерия к уменьшенным копиям пар домен-ранг.
- Применение нелинейных отображений (на стадии разработки).

Генерирование окаймляющих доменов



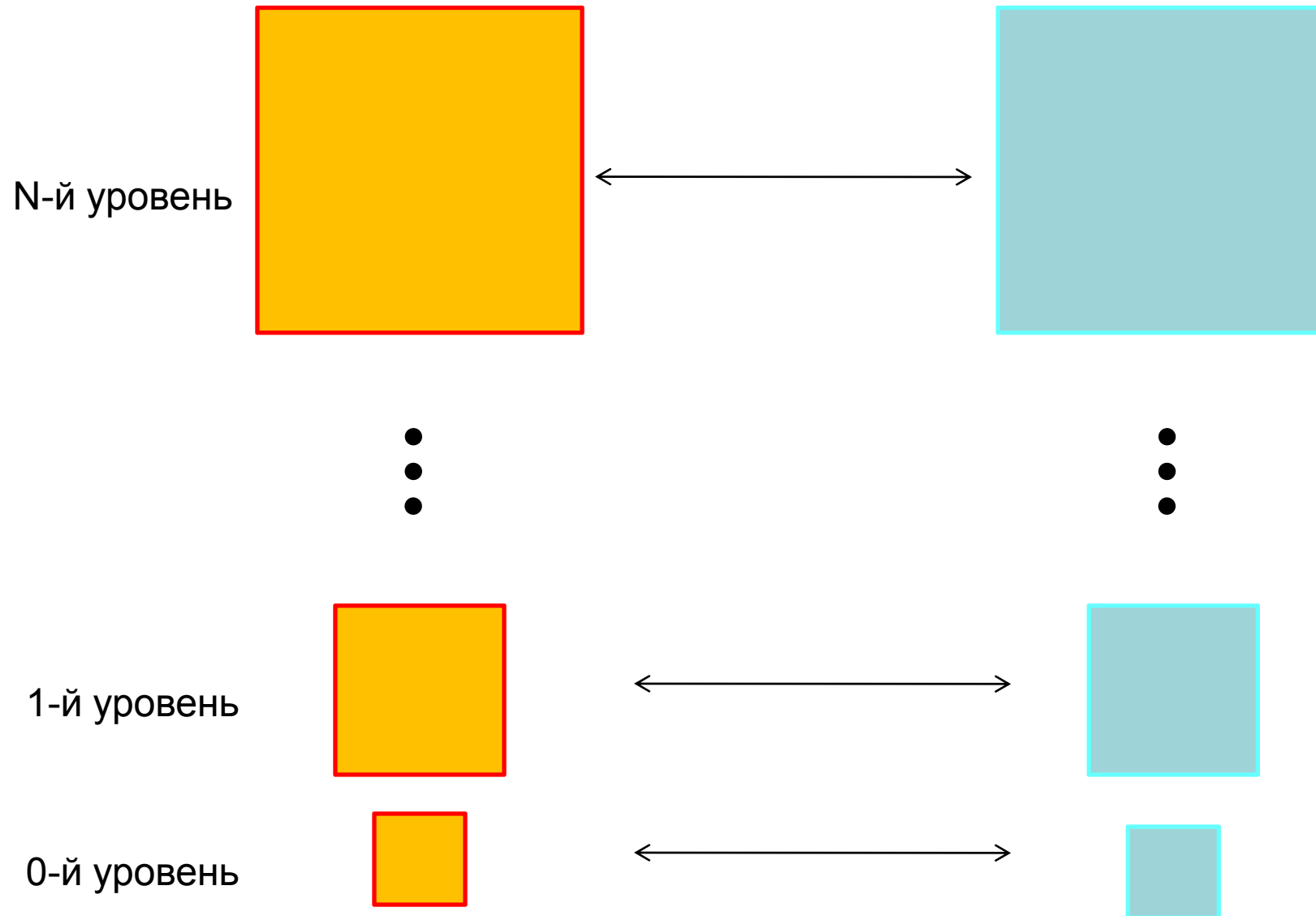
 - Доменный блок

 - Ранговый блок

Генерирование окаймляющих доменов

- Эффективность сжатия в немалой степени зависит от начального разбиения исходного изображения.
- Невключение удачного домена или его ошибочное отбрасывание сказывается как на степени сжатия, так и на качестве декодирования.
- Априори представляется правдоподобным, что различие соседних близких пикселей, в целом, менее разительно, чем в областях, выбираемых случайным образом.
- Поэтому, включение окаймляющих доменов в первоначально формируемое их множество представляется перспективным.

Многоуровневый анализ. Использование основного критерия к уменьшенным копиям пар домен-ранг.



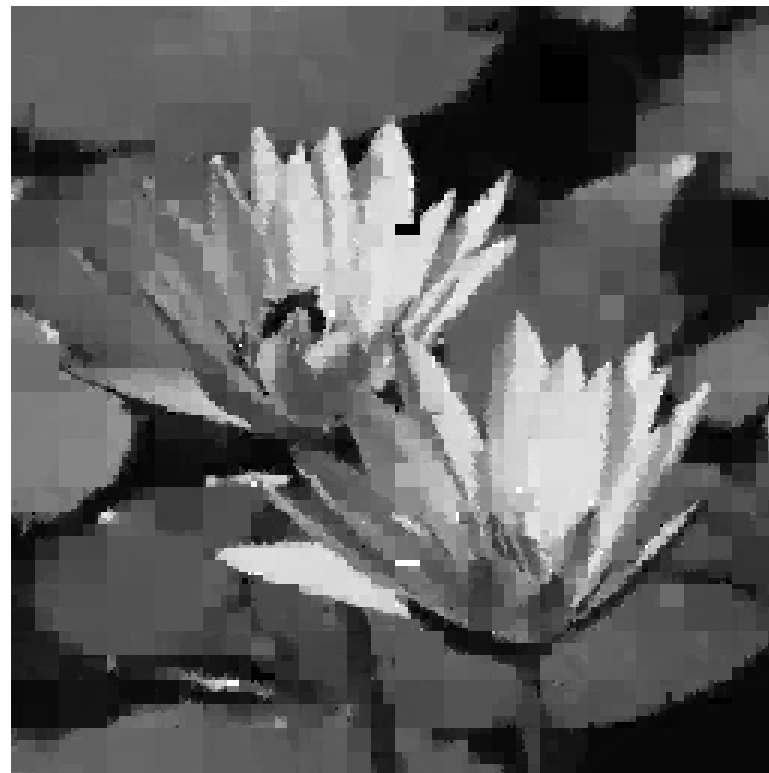
Эксперименты

Параметры	Эксперимент			
	1	2	3	4
Алгоритм с окаймляющими доменами				
Доменов	2261	2781	4785	6350
Рангов	946	886	787	772
Допуст.погр.	0.05	0.05	0.05	0.05
Ср. пикс. Ошибка, %	4.46	4.55	4.6	4.71
Кэфф.сжатия	5.84	6.24	7.02	7.15
Время кодирования, (с)	158	500.54	795.57	1187.97
Время декодирования, (с)	5.09	3.52	2.77	2.5
Алгоритм с многоуровневым анализом				
Доменов	225	841	2977	4562
Рангов	1708	1705	1708	1687
Допуст.погр.	0.05	0.05	0.05	0.05
Искать лучший	0	0	0	0
Ср. пикс. Ошибка, %	3.29	3.37	3.25	3.56
Кэфф.сжатия	3.24	3.25	3.24	3.28
Время кодирования, (с)	175.23	657.65	1645.47	2278.28
Время декодирования, (с)	3	3.12	3.2	3.3

Декодированное изображение



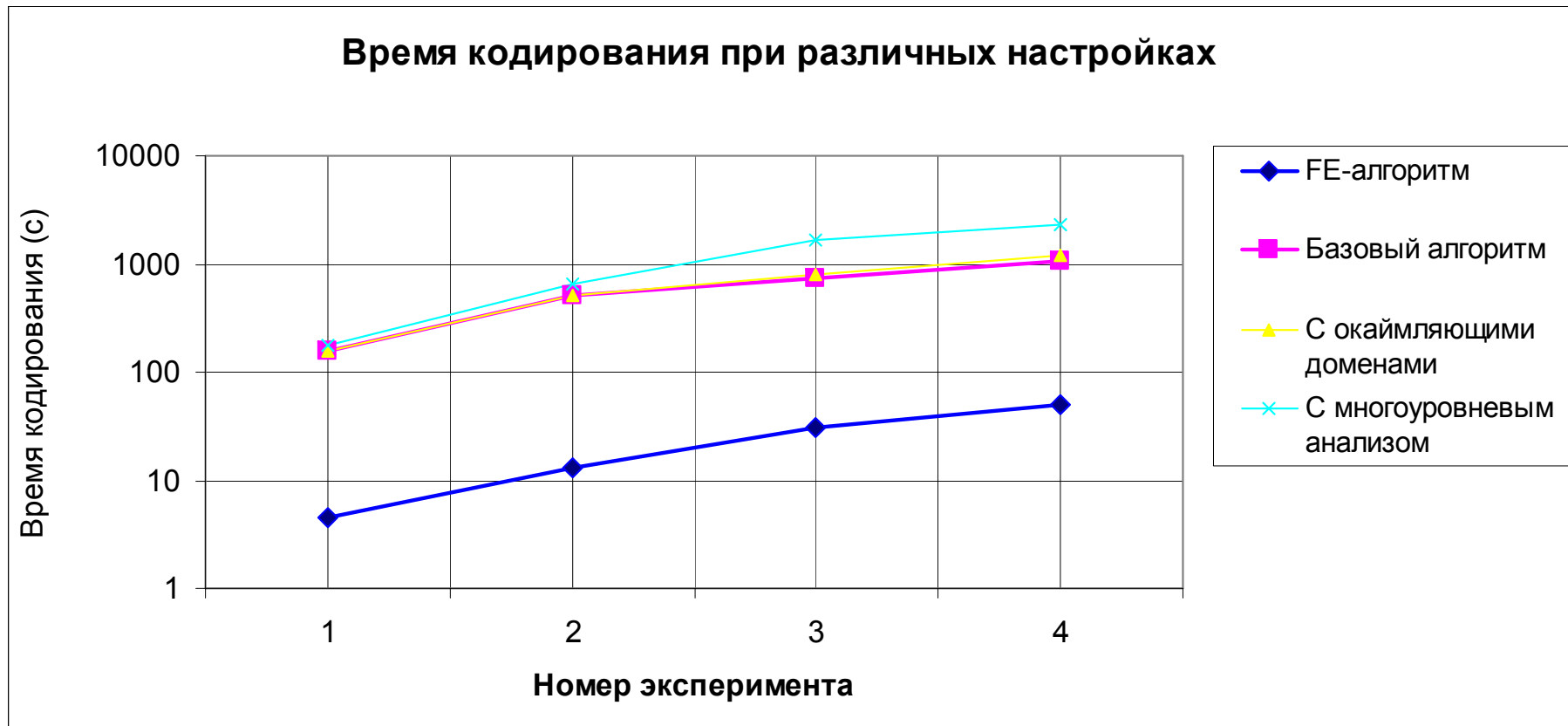
Декодированное изображение,
предварительно закодированное
алгоритмом с окаймляющими
доменами



Декодированное изображение,
предварительно закодированное
алгоритмом с многоуровневым
анализом

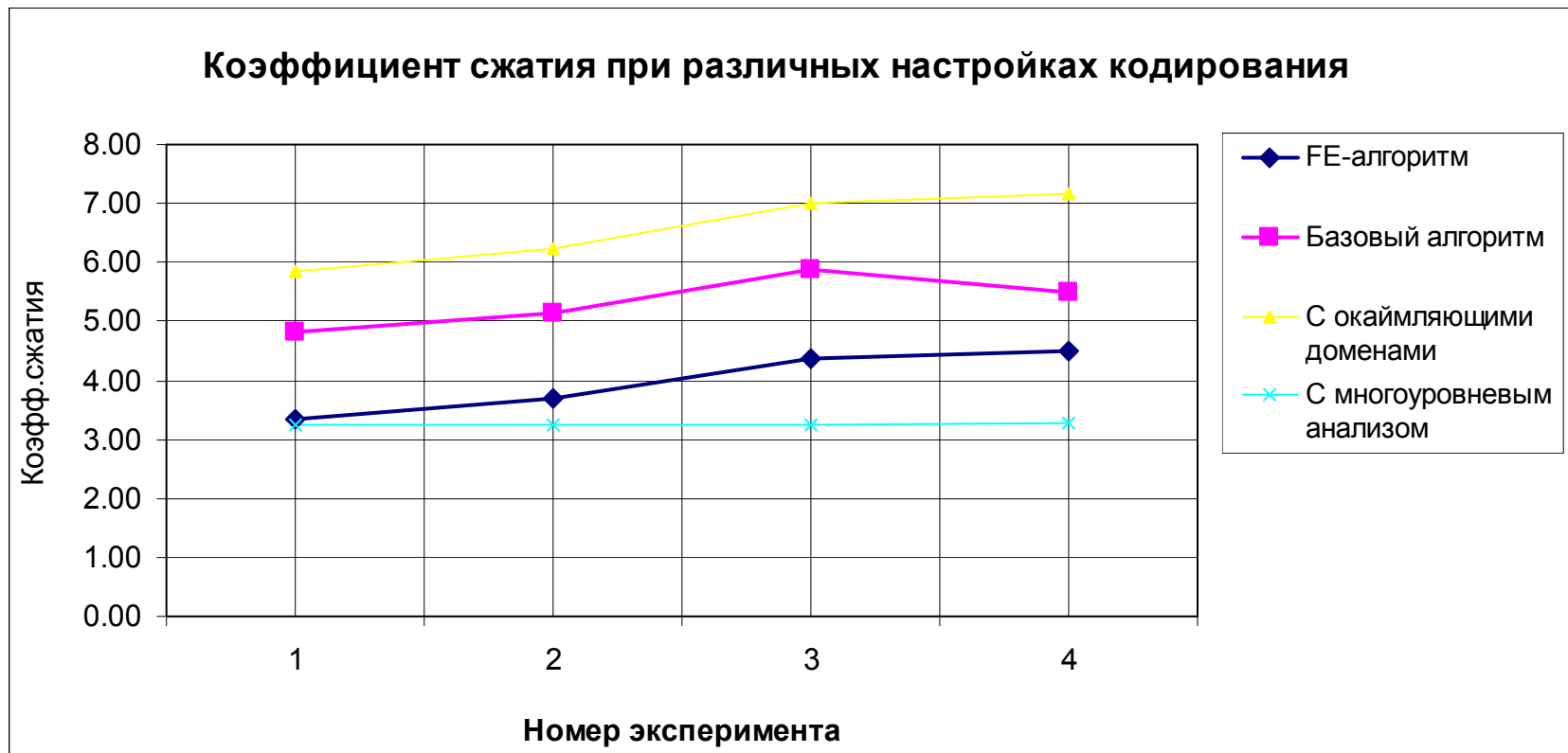
Результаты экспериментов

- Время кодирования при использовании алгоритма с окаймляющими доменами выросло незначительно, а при использовании алгоритма с многоуровневым анализом – значительно.

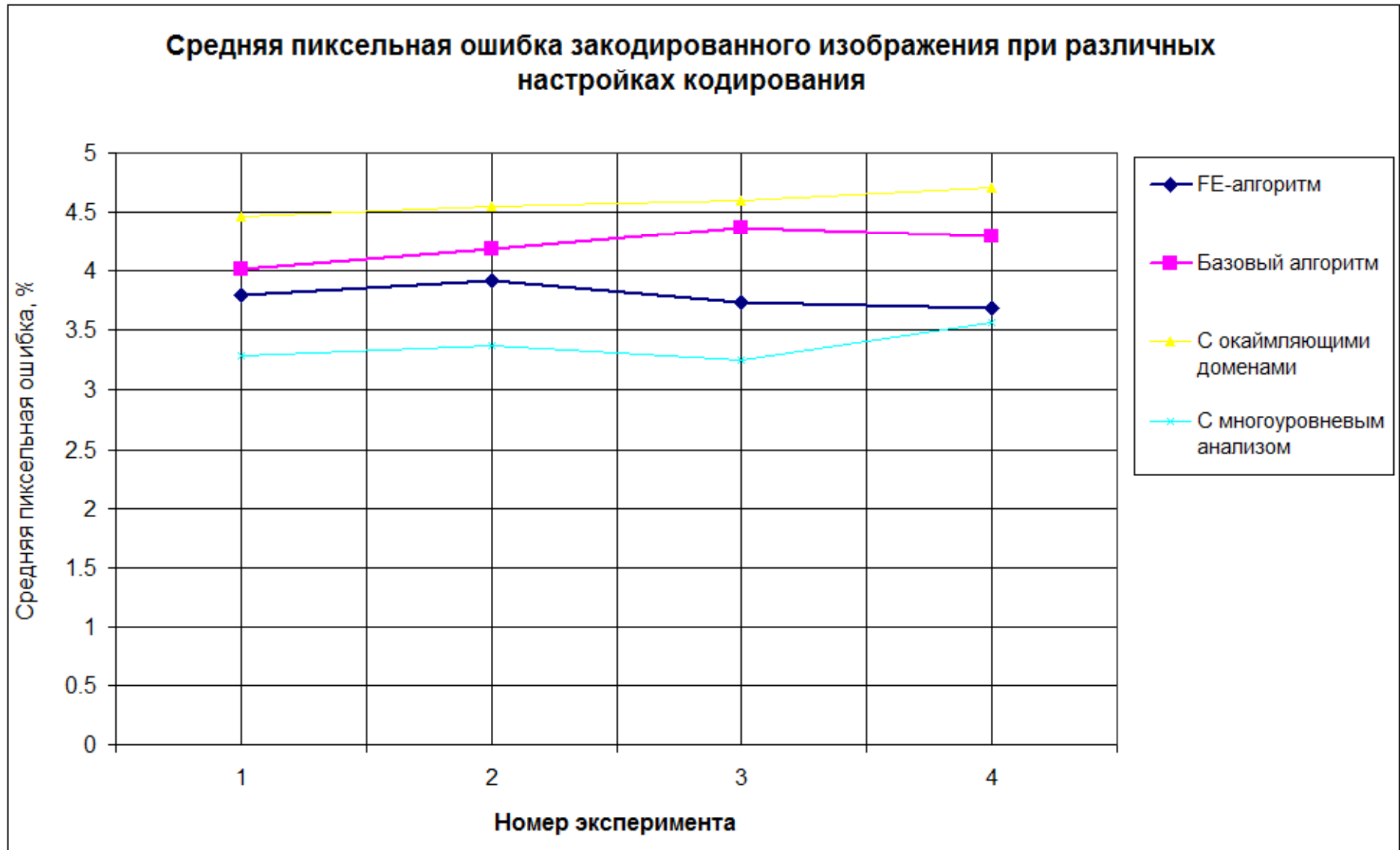


Результаты экспериментов

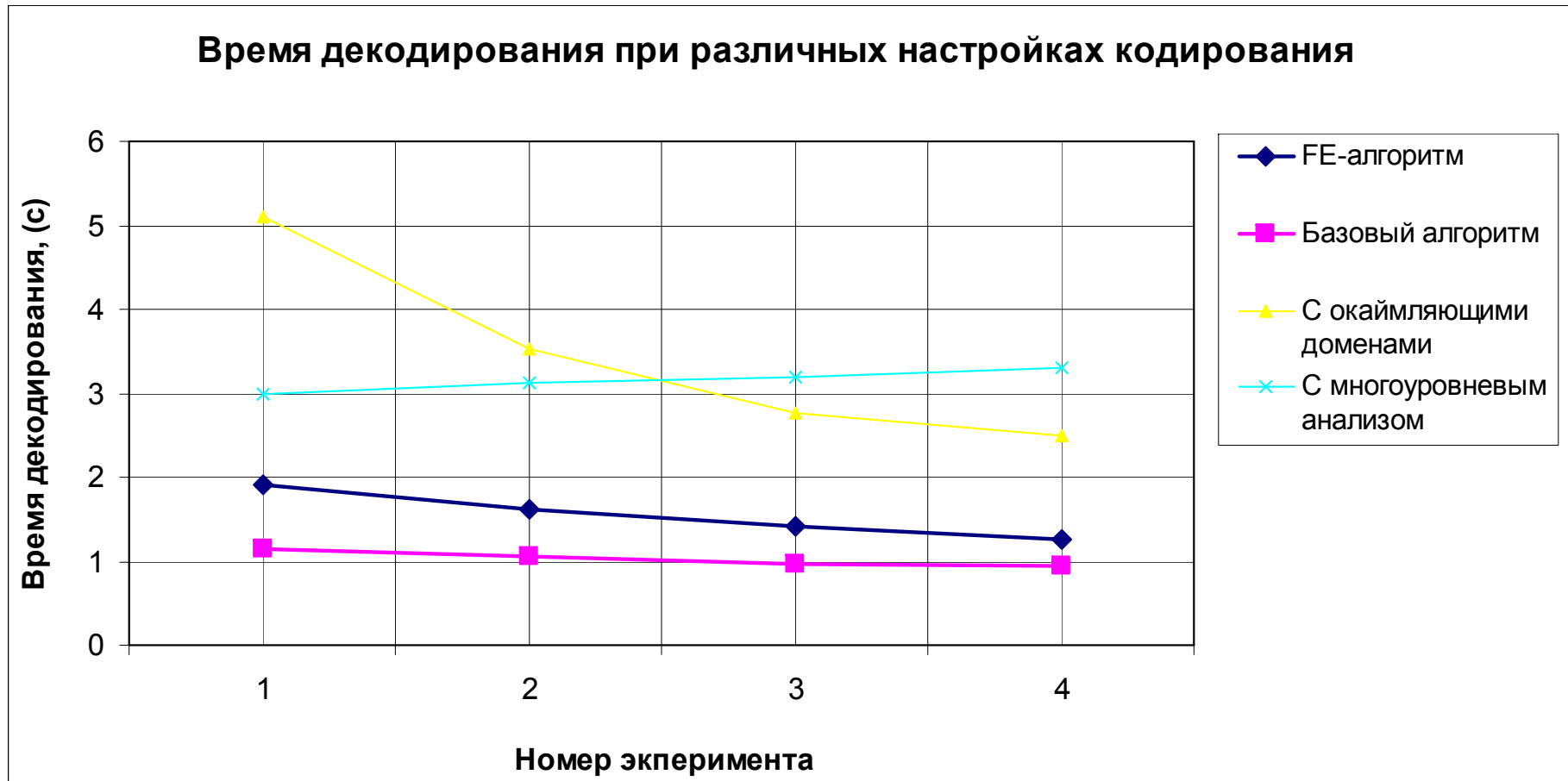
- Алгоритм с окаймляющими доменами дает примерно на 20% лучший коэффициент сжатия.
- Алгоритм с многоуровневым анализом улучшений не дал.



Результаты экспериментов



Результаты экспериментов



Выводы

- Алгоритм с окаймляющими доменами обеспечивает в среднем на 20% более сильное сжатие. При этом он незначительно уступает базовому алгоритму по времени кодирования и средней пиксельной ошибке. Его недостатком же является более продолжительное время декодирования.
- В дальнейшем планируется оптимизация этого алгоритма. В частности – модифицировать алгоритм так, чтобы перебор доменов начинался с окаймляющих, что это позволит сократить время кодирования.

Выводы

- Алгоритм с многоуровневым анализом не дал улучшений практически по всем параметрам эффективности.
- Причина лежит в его недостаточной оптимизации.
- В дальнейшем планируется оптимизация этого алгоритма.

Презентация завершена!
Спасибо за внимание.

Задавайте вопросы.