

## СЖАТИЕ ПОЛНОЦВЕТНЫХ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ УОЛША

Карагодин М.А., Осокин А.Н.

В настоящее время актуальна следующая проблема: в связи с малой вычислительной мощностью приемника или передатчика цифровой графической информации ужесточаются требования к алгоритму сжатия этой информации. В качестве наиболее распространенных примеров можно привести задачу сжатия изображений в программах-терминалах и просмотр видео и или больших по размеру изображений на маломощных рабочих станциях.

На текущий момент для решения подобных проблем считается оптимальным использование алгоритма JPEG [1,2,3], однако данный алгоритм имеет низкую скорость работы и сложен в аппаратной реализации. В данной работе рассмотрен альтернативный алгоритм, основанный на дискретном двумерном преобразовании Уолша [4,5].

За основу был взят алгоритм JPEG. Проведена серия экспериментов по выбору оптимального алгоритма путем автоматического перебора различных вариантов шагов алгоритма. Исследовались цветовые системы YCbCr, YUV, YIQ, HSV; методы субдискретизации 1:1:1, 1:2:2, 1:4:4; матричные преобразования DCT, двумерное преобразование Уолша с упорядочением по Адамару, по Уолшу и по Пэли; матрицы квантования взятые из стандарта JPEG и модифицированные; методы дожатия RLE+Хаффман, RLE64+Хаффман. Эксперименты проводились на специально разработанном программном экспериментальном комплексе [5].

Исходя из результатов эксперимента, был разработан и реализован новый алгоритм сжатия изображений. Алгоритм включает 5 следующих шагов.

1. Преобразование из цветовой системы RGB в цветовую систему YCbCr.
2. Субдискретизация цветовых компонент.
3. Матричное преобразование компонент, разбитых на квадраты 8x8 пикселей.
4. Квантование полученных коэффициентов.
5. Дожатие коэффициентов методами сжатия без потерь.

Разработанный нами алгоритм отличается тем, что в качестве матричного преобразования для цветовой компонент используется преобразование Уолша [4,5].

Выбор преобразования Уолша так же определяет и вид матрицы квантования. В разработанном алгоритме использовались матрицы квантования, данные в описании стандарта JPEG. Однако, при преобразовании WHT потребовалось применить матрицу JPEG, поэлементно деленную на определенный коэффициент. Это обусловлено тем, что данное преобразование имеет меньшую точность в сравнении с преобразованием DCT.

Для дожатия коэффициентов был разработан вариант алгоритма RLE, получивший название RLE64 [5], т.к. используется для кодирования 64-байтных блоков коэффициентов преобразований. Сжатие сводится к тому, что на каждой итерации алгоритма определяется длина максимальной последовательности ненулевых байт (не больше 15) и длина следующей за ней последовательности нулевых байт (не больше 15). В выходной поток выдается контрольный байт, в младших 4 битах содержащий длину первой последовательности, а в старших – длину второй. После этого в выходной поток выдается обнаруженная последовательность ненулевых байт. Если до конца блока остались одни нули, алгоритм выдает в выходной поток нуль и для данного блока завершает выполнение.

Выход алгоритма RLE64 кодируется адаптивным алгоритмом Хаффмана.

Тестирование разработанного алгоритма проводилось на компьютере Pentium IV – 1,7GHz / 256 Mb DDR RAM / 60Gb HDD, chipset Intel i845D под управлением ОС Microsoft Windows 2000 (Service Pack 3). Использовался международный стандартный тестовый набор Waterloo Bragzone (Calgary Corpus) [6]. Результаты приведены в таблице 1. Сводная диаграмма сравнения разработанного алгоритма и реализации стандартного алгоритма JPEG приведена на рис. 1.

Таблица 1

Результаты тестирования оптимального алгоритма по коэффициенту сжатия

Имя файла	Размер (байт)	Коэффициент сжатия					
		WHT высокое качество	WHT среднее качество	WHT низкое качество	JPEG высокое качество	JPEG среднее качество	JPEG низкое качество
clegg.bmp	2150774	7,41	18,16	41,14	2,62	14,38	41,67
frymire.bmp	3708434	6,84	15,51	34,66	2,91	13,54	35,77
Lena.bmp	786486	21,03	38,10	63,10	3,86	32,23	82,22
monarch.bmp	1179702	18,93	34,73	58,25	4,42	30,95	74,75
peppers.bmp	786486	20,70	36,05	56,98	4,24	29,76	74,20
sail.bmp	1179702	11,60	24,28	52,84	3,29	18,76	57,18
serrano.bmp	1499126	9,82	23,29	43,15	3,25	18,59	51,59

tulips.bmp	1179702	14,24	28,49	50,29	3,80	22,72	63,43
<b>Среднее</b>		<b>13,82</b>	<b>27,33</b>	<b>50,05</b>	<b>3,55</b>	<b>22,62</b>	<b>60,10</b>

Как можно заметить, при высоком качестве восстановленного изображения разработанный алгоритм выигрывает в коэффициенте сжатия, фактически, в три раза, однако на среднем качестве превосходство алгоритма уже не столь значительно, а на низком – пропадает вовсе. Однако, тестирование при низком качестве восстановленного изображения проведено только для полноты исследования, практическая ценность кодирования изображений при низком качестве чрезвычайно мала.

Графики зависимостей среднеквадратичного отклонения от коэффициента сжатия для разработанного алгоритма и стандартного алгоритма JPEG приведены на рис. 2.

Среднеквадратичное отклонение восстановленного изображения от исходного вычислялось по следующей формуле [2]:

$$d(x, y) = \sqrt{\frac{\sum_{i=1, j=1}^{n, m} (x_{ij} - y_{ij})^2}{nm}},$$

где  $x_{ij}$  – значение пиксела исходного изображения с координатами  $i, j$ ;

$y_{ij}$  – значение пиксела восстановленного изображения с координатами  $i, j$ ;

$n, m$  – высота и ширина изображения.

Как показывают результаты тестирования (см. рис. 3) время сжатия и распаковки изображений практически постоянно на одних и тех же данных и почти не зависит от коэффициента сжатия. При этом, скорость сжатия и распаковки разработанного алгоритма в среднем в 1,5 раза выше, чем скорость сжатия и распаковки стандартного алгоритма JPEG. Конкретное время зависит от исходных данных изображения.

Также можно говорить о симметричности алгоритма по скорости сжатия и распаковки данных.

Итак, в результате данной работы нами был разработан алгоритм сжатия полноцветных изображений, обладающий следующими свойствами:

- Скоростью сжатия и распаковки в 1.5 большим, чем стандартный JPEG.
- Коэффициентом сжатия, превышающим в 3 раза стандартный JPEG при высоком качестве изображения.

Полученный алгоритм может быть реализован на недорогих ПЛИС ввиду простоты аппаратной реализации.

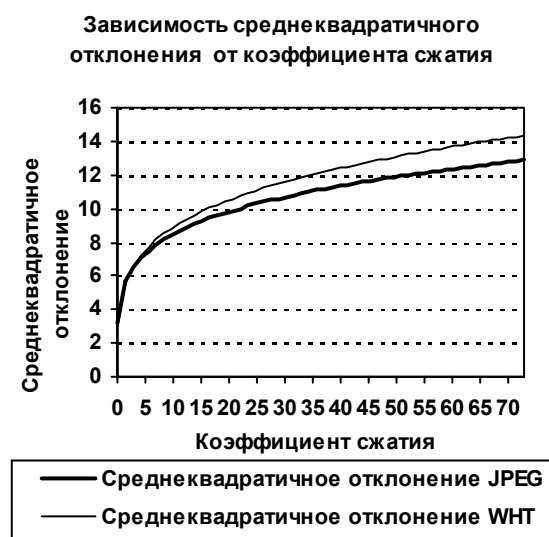
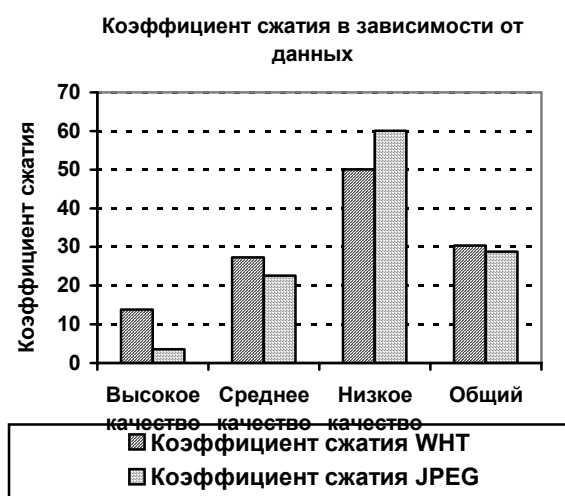


Рис. 1. Диаграмма сравнения разработанного алгоритма и стандартного алгоритма JPEG по коэффициенту сжатия

Рис. 2. График зависимостей среднеквадратичного отклонения от коэффициента сжатия для разработанного алгоритма и стандартного алгоритма JPEG



Рис. 3. График зависимостей времени сжатия от коэффициента сжатия для разработанного алгоритма и стандартного алгоритма JPEG

### СПИСОК ИСТОЧНИКОВ

1. Официальный сайт группы JPEG, [www.jpeg.org](http://www.jpeg.org).
2. Д. Ваголин, А. Ратушняк и др. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: Диалог МИФИ, 2002.
3. Мюррей Д.Д., Райнер У. Ван. Энциклопедия форматов графических файлов. Пер. с англ. - Киев: ВНУ, 1997, - 535с.
4. Ахмед Н., Рао К.Р. Ортогональные преобразования при обработке цифровых сигналов. - М.: Связь 1980.
5. Карагодин М.А., Осокин А.Н. Алгоритм сжатия изображений на основе функций Уолша. //Материалы Международной конференции "Информационные системы и технологии ИСТ'2003", в 3-х томах, том 2, Новосибирск, 2003. стр. 126-130.
6. Сайт Calgary Corpus (тестовые наборы изображений). <http://links.uwaterloo.ca/bragzone.base.html>.

