УДК 004.932.2

АНАЛИЗ ВТОРИЧНОЙ ИНФОРМАЦИИ В JPEG

П. С. Санкин,[1]
аспирант
М. Ю. Литвинов,
соискатель
Санкт-Петербургский государственный университет аэрокосмического приборостроения

Рассматриваются особенности передачи информации в последовательностях изображений, сжатых по алгоритму JPEG. Исследуется проявление вторичной информации в JPEG при разборе кодером случайных данных. Для повышения эффективности обработки при маскировании изображений вводится математическое описание модели, обеспечивающей эффективную стратегию сокрытия всей информации в JPEG-изображениях.

Ключевые слова *— свойства изображений, вторичная информация, JPEG.*

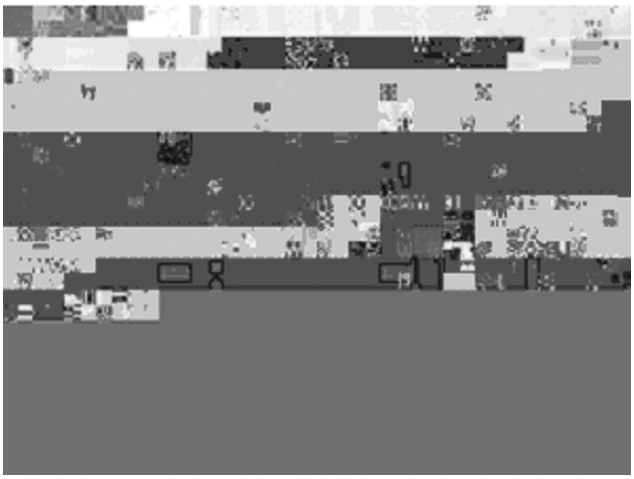
Введение

Цифровые видеопотоки, передаваемые в системах видеоконтроля, охранного телевидения, дистанционного мониторинга распределенных охраняемых объектов и др., для сохранения их конфиденциальности, как правило, маскируют путем приведения тела кадра к шумоподобному виду. Однако анализ маскированных данных показывает, что имеется вторичная информация, косвенно характеризующая передаваемые изображения [1]. Видеоданные обладают большой степенью временной и пространственной избыточности, удаление которой приводит к появлению переменной битовой скорости передачи, являющейся слабым местом потока при анализе данных в канале от цифровой камеры к видеосерверу. Для форматов, использующих сжатие, такой информацией является размер кадра.

Анализ изменения параметров, зависящих от характеристик изображения, позволяет сделать выводы о наличии или отсутствии движения в наблюдаемой области, о ее освещенности и контрастности, что крайне нежелательно.

В защищенном изображении, сжатом до маскирования по стандарту JPEG, при отображении его декодером проявляются однотонные полосы, которые периодически довольно резко меняют свой цвет. Кроме того, часто при заданном разрешении изображения отображается не вся область, а только некоторая часть, после которой идет фон (рис. 1).

Для объяснения данных эффектов в маскированных (т. е. фактически для случайных) данных необходимо подробно описать формат сжатия JPEG и этапы разбора данных кодером. Так как JPEG является блоковым кодеком, мы имеем блоковые искажения: каждому блоку соответствует один блок MCU (Minimum Coded Unit — минимальный кодируемый блок). Для рассмотрения интерпретации шума применительно к разбору JPEG-декодером блоков изображения нужна математическая модель, описывающая средний размер MCU для случайных значений коэффициентов.



■ ***Рис. 1****. Маскированный кадр JPEG*

[1] Научный руководитель — доктор технических наук, профессор, заведующий кафедрой безопасности информационных систем Санкт-Петербургского государственного университета аэрокосмического приборостроения *Е. А. Крук*.

Структура кадра

Согласно спецификации [2], изображение, закодированное алгоритмом JPEG, имеет строго определенную структуру (рис. 2). Файл состоит из сегментов маркеров и закодированных энтропийным кодером сегментов данных (ECS — Entropy-Coded Data).

Самой крупной единицей кодирования во всех режимах является образ (image), т. е. само изображение. Изображение содержит только один кадр (frame) в последовательном или прогрессивном режимах, и этот кадр идентичен самому изображению. В иерархическом режиме изображение разделяется на несколько кадров.

Следующий уровень разбиения данных — скан (scan), который содержит часть информации изображения. Разделение на сканы в разных режимах осуществляется по-разному. Закодированные кодером данные помещаются в сегменты ECS, которые состоят из блоков MCU.

Сегменты маркеров содержат признак маркера и тело маркера, состоящего, в свою очередь, из поля с размером маркера и набора параметров, характерных для каждого маркера. Признак маркера — это двухбайтовое значение, всегда начинающееся с байта FF. Файл JPEG может содержать следующие маркеры.

1. SOI и EOI. Каждый кадр JPEG должен начинаться с маркера SOI (Start of Image — начало изображения) и завершаться маркером EOI (End of Image — конец изображения). Эти два маркера не имеют тела.

2. DHT (Define Huffman Tables), в теле которого задаются таблицы Хаффмана для сжатия без потерь.

3. DQT (Define Quantization Table), в теле которого определяются таблицы квантования.

4. SOF (Start of Frame) — маркер, определяющий заголовок кадра, в теле которого описываются разрешение, число компонент, формат выборки (прореживания) для каждой компоненты, индекс таблицы квантования для компоненты и т. п.

5. SOS (Start of Scan) — маркер, описывающий скан. В теле этого маркера содержится описание числа компонент в скане, индексы таблиц для энтропийного кодера и т. п.

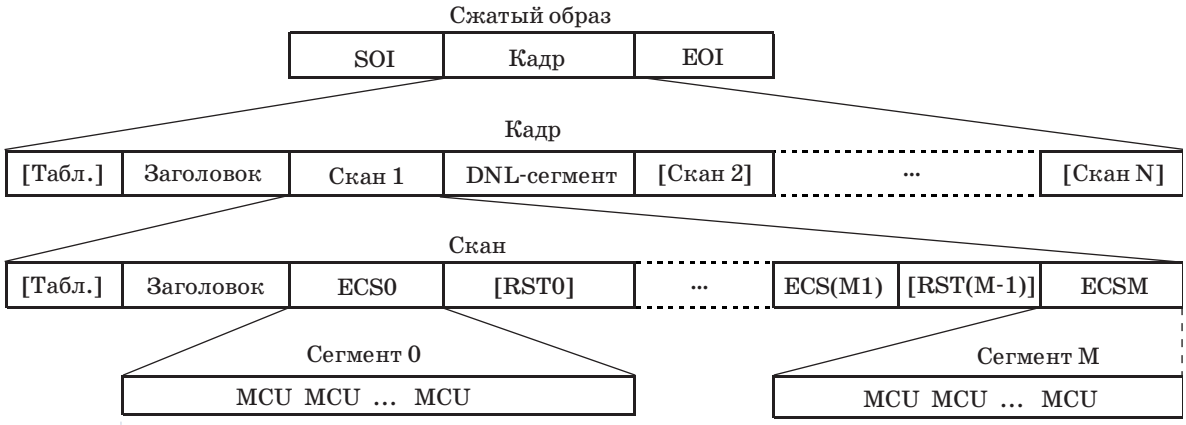
6. Маркеры DNL, RST, DRI (маркеры числа строк, рестарта, интервала рестарта) разделяют сегменты ECS и позволяют синхронизироваться декодеру после ошибок передачи закодированных данных.

7. APPn — пользовательские маркеры, позволяющие добавлять в файл дополнительные данные.

Следует отметить, что маркеры в файле JPEG могут идти в различной последовательности и повторяться (например, для задания двух таблиц квантования используются два разных маркера DQT). Для вычисления размера заголовка, т. е. длин всех маркеров, необходимо просматривать файл и суммировать длины всех сегментов маркеров. Формат не допускает появления в этом блоке других маркеров, поэтому во избежание сбоя при заполнении блока производится замена всех похожих на маркеры данных — позади встретившегося байта FF ставится байт 00.

Изначально все изображение разбивается на блоки 8 × 8 пикселей. Такой блок называется единицей данных DU (Data Unit). Обработка ведется макроблоками, размер которых определяется в зависимости от типа изображения (цветное, оттенки серого и т. п.). Согласно формату, такой макроблок называется минимальным кодируемым блоком.

Для изображения в оттенках серого (grayscale mode) используется одна компонента — яркостная. В этом случае MCU состоит из одного блока 8×8. Обход файла осуществляется построчно слева направо сверху вниз, и режим называется «без чередования» (non-interleaved order).



■ *Рис. 2. Общая структура файла JPEG*

Для цветного изображения обычно, а в основном режиме (baseline jpeg) — всегда, используются три компоненты (одна яркостная (Y) и две цветоразностные (Cr, Cb)). Так как глаз маловосприимчив к цветовой составляющей по сравнению с яркостной, для цветоразностных компонент обычно используется прореживание, что позволяет дополнительно сжать изображение.

Каждый минимальный кодируемый блок сжимается без потерь кодером Хаффмана (либо арифметическим кодером, который используется редко). Блок DU состоит из одного DC-коэффициента и до 63 AC-коэффициентов. Каждый коэффициент состоит из двух полей: длины поля данных и поля данных. Значение длины закодировано по соответствующей таблице Хаффмана.

Кодирование DC- и AC-коэффициентов осуществляется по-разному. Для кода Хаффмана используются таблицы кодовых слов, описание которых хранится в файле JPEG. Всего может быть использовано до четырех таблиц для кодирования коэффициентов: 1) DC яркостной компоненты; 2) AC яркостной компоненты; 3) DC цветоразностных компонент; 4) AC цветоразностных компонент.

Математическая модель блока данных

При разборе шума декодером важную роль играют области, описывающие размер поля данных каждого коэффициента. Построим вероятностную модель по стандартным таблицам Хаффмана, описывающим DC- и AC-коэффициенты. Будем рассматривать данные как некий неупорядоченный шум, распределенный по нормальному закону. Проанализируем внутреннюю структуру данных, описывающих изображение в оттенках серого.

Коэффициент DC соответствует среднему (яркостному или цветоразностному) значению блока 8×8, поэтому для соседних блоков он принимает близкие значения. Таким образом, DC сначала подвергается разностному кодированию, потом — кодированию Хаффмана. Для описания серого изображения достаточно использовать только яркостные компоненты.

Коэффициент состоит из двух полей: SIZE и AMPLITUDE, где AMPLITUDE — значение разности с соседним блоком (DIFF), SIZE — размер в битах, который требуется для хранения AMPLITUDE.

При разборе шума мы имеем дело со случайными двоичными данными. Вероятность появления значения в некоем диапазоне будет равна 2^{SIZE}. В зависимости от кода, определяющего поле SIZE, выбирается следующая порция данных, характеризующая значение поля AMPLITUDE.

По стандартной таблице кодов Хаффмана можно легко найти вероятности длин и диапазонов значений DC-коэффициентов. Для более наглядного представления построим таблицу вероятностей для яркостных коэффициентов (табл. 1).

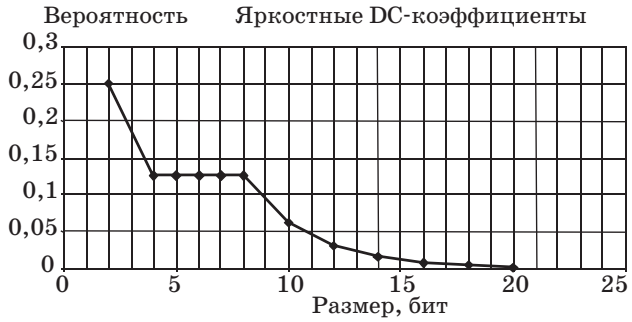
Под ошибкой понимаются все коды, отсутствующие в таблице Хаффмана. В зависимости от программной реализации декодера ошибка может не вызывать краха декодирования, например в случае, когда последний коэффициент всегда разбирается по ветке else алгоритма. Из табл. 1 видно, что вероятности появления большего значения и появления длинных полей уменьшаются.

Исходя из данных табл. 1 построим диаграмму вероятностей размера яркостного DC-коэффициента (рис. 3).

Найдем среднее значение длины яркостных DC-коэффициентов, для них применима формула математического ожидания дискретной случайной величины:

■ ***Таблица 1****. Таблица вероятностей значений яркостных DC-коэффициентов*

Категория	Длина кода	Кодовое слово	Вероятность	Значение	Общая длина
0	2	00	0,25	0	2
1	3	010	0,125	–1, 1	4
2	3	011	0,125	–3, –2, 2, 3	5
3	3	100	0,125	–7, ..., –4, 4, ..., 7	6
4	3	101	0,125	–15, ..., –8, 8, ..., 15	7
5	3	110	0,125	–31, ..., –16, 16, ..., 31	8
6	4	1110	0,0625	–63, ..., –32, 32, ..., 63	10
7	5	11110	0,03125	–127, ..., –64, 64, ..., 127	12
8	6	111110	0,015625	–255, ..., –128, 128, ..., 255	14
9	7	1111110	0,0078125	–511, ..., –256, 256, ..., 511	16
10	8	11111110	0,00390625	–1023, ..., –512, 512, ..., 1023	18
11	9	111111110	0,001953125	–2047, ..., –1024, 1024, ..., 2047	20
		Ошибка	0,001953125		



■ ***Рис. 3.*** *Распределение вероятностей длины полей*

$$M\xi = \sum_{i=1}^{n} p_i x_i. \qquad (1)$$

Подставив из таблицы вероятностей в качестве p вероятности, а в качестве x длины коэффициентов, получим $M_{\text{DCY}} = 5{,}7$ бит.

Таким образом, при интерпретации шума JPEG-декодер будет давать небольшие отклонения средней яркости блоков.

Найдем рассеивание длин DC-коэффициентов. Формула для дисперсии имеет вид

$$D\xi = M(\xi - M\xi)^2. \qquad (2)$$

Для дискретной случайной величины получим формулу

$$D\xi = \sum_{i=1}^{n} (x_i - M\xi)^2 p_i. \qquad (3)$$

Суммируя все значения вероятности разностей размеров отдельных коэффициентов и их среднего размера, получаем дисперсию $D_{\text{DCY}} = 10{,}2$.

Аналогичные расчеты, используя соответствующие таблицы Хаффмана, можно провести и для AC-коэффициентов. Всего в стандартной таблице Хаффмана для AC-коэффициентов используется 161 кодовое слово. Из-за их большого числа оценка принимаемых значений очень сильно затруднена, поэтому проведем оценку их размера. Построим сводную таблицу вероятностей размеров яркостных AC-коэффициентов, которые собраны на основе стандартной таблицы. В табл. 2 видно проявление неравномерности используемого кода — преобладающими являются коэффициенты с небольшими короткими длинами.

Вероятность появления кодов с полем более 10 бит очень мала, заметно преобладание коротких кодов. По сводной таблице, используя формулы (1) и (3), можно найти математическое ожидание и дисперсию для одного такого коэффициента: $M_{\text{DCY}} = 5{,}19$, $D_{\text{DCY}} = 5{,}9$.

Стандартом предусмотрена последовательность длиной до 63 таких коэффициентов. Завершающим будет блок EOB (End-of-Block) длиной 4 бита

■ ***Таблица 2.*** *Вероятности появления яркостных AC-коэффициентов*

Общий размер поля, бит	Сумма вероятностей полей
3	0,25
4	0,3125
5	0,0625
6	0,15625
7	0,0625
8	0,078125
9	0,00390625
10	0,048828125
11	0,004394531
12	0,001464844
13	0,011230469
14	0,000488281
15	0,004150391
16	0,000732422
17	$6{,}10352 \cdot 10^{-5}$
18	0,001083374
19	0,000183105
20	0,000198364
21	0,000213623
22	0,000228882
23	0,000228882
24	0,000228882
25	0,000244141
26	0,000244141
Ошибка	$1{,}52588 \cdot 10^{-5}$

и имеющий вероятность появления 0,0625. Всего в последовательности может быть один коэффициент, с увеличением числа AC-коэффициентов вероятность появления EOB будет изменяться следующим образом:

$$p_0 = 0{,}0625;$$
$$p_1 = p_0(1 - p_0);$$
$$p_2 = p_0(1 - (p_1 + p_0));$$
$$\ldots$$

Таким образом, получаем общую формулу вероятности отдельно взятого коэффициента

$$p_n = p_0\left(1 - \sum_{i=0}^{n-1} p_i\right). \qquad (4)$$

Приближенно будем считать случайные величины независимыми, найдем среднее значение для 63 коэффициентов. Так как данная случайная величина дискретна, для нее справедлива формула (1), математическое ожидание количества AC-коэффициентов будет $M_{\text{NY}} = 13{,}73$.

Математическое ожидание блока яркостных AC-коэффициентов будет равным сумме математических ожиданий его составляющих:

$$M_{\text{б ACY}} = M_{\text{NY}} M_{\text{ACY}} + EOB = \\ = 13{,}73 \cdot 5{,}19 + 4 = 75{,}2.$$

Проведем расчет полного размера отображаемого блока данных. Средний размер MCU будет равен сумме математических ожиданий составляющих его коэффициентов. Для изображения в оттенках серого это будет один блок данных, представленный яркостными коэффициентами:

$$M_{\text{MCU серого}} = M_{\text{DCY}} + M_{\text{б ACY}} = \\ = 5{,}7 + 75{,}2 = 80{,}9 \text{ бит}.$$

В таком режиме MCU будет кодировать блок из 64 пикселей, исходя из этого можно определить «степень сжатия», характеризующую средний объем случайных данных, необходимый для описания одного пикселя. Она будет равной 1,26 бит/пиксель.

Размер файла правильно соотносить с числом и структурой составляющих его блоков. Оценим разброс значений размеров **MCU. Найдем дисперсию** для блоков изображения как сумму дисперсий всех составляющих MCU:

$$D_{\text{MCU}} = D_{\text{DC}} + D_{\text{б AC}}.$$

Точное количество AC-коэффициентов в блоке неизвестно, поэтому за значение дисперсии блока возьмем сумму среднего числа коэффициентов:

$$D_{\text{б AC}} = \sum_{i=1}^{N} D_{\text{AC}i}.$$

Коэффициенты между собой независимы, а дисперсии равны между собой, поэтому эту сумму можно заменить произведением и подставить в общую формулу:

$$D_{\text{MCU}} = D_{\text{DC}} + N_{\text{Y}} \cdot D_{\text{AC}}. \qquad (5)$$

В данном случае число N_{Y} равно математическому ожиданию числа коэффициентов M_{NY}. По формуле (5) рассчитаем дисперсию для серого изображения, она будет представлена только яркостными компонентами:

$$D_{\text{MCU серого}} = D_{\text{DCY}} + N_{\text{Y}} \cdot D_{\text{ACY}} = \\ = 10{,}2 + 13{,}73 \cdot 5{,}92 = 91{,}48.$$

Определим разброс значений размера MCU, найдем среднеквадратичное отклонение для серого и цветного блоков:

$$\sigma = \sqrt{D};\ \sigma_{\text{серого}} = \sqrt{91{,}48} = 9{,}56.$$

Согласно закону нормального распределения, 99,73 % всех значений будут попадать в диапазон трех среднеквадратичных отклонений в любую сторону от среднего. Таким образом, разброс объема данных, необходимых для описания одного блока, в случае серого MCU в округленном виде будет составлять (81 ± 29) бит.

Итак, мы вычислили объем данных, необходимый для описания одного блока MCU. По нему можно найти эффективный размер кадра для изображения любого размера.

Экспериментальные данные

Для оценки объема случайных данных, требуемых для заполнения картинки без фоновых полос, были проведены следующие манипуляции с набором изображений. Исследовались файлы изображений в формате JPEG с размером картинки, кратной размеру MCU. Заголовки файлов содержали стандартные таблицы Хаффмана, блок данных после заголовка заменялся случайными данными. Заполнение производилось после блока Start-of-Scan (SOS), обозначенного FFDA. Во избежание сбоя декодера на случайно появившемся маркере, в тексте блоки FF заменялись на FF00. Такая замена вносит погрешность в виде увеличения размера примерно на 0,04 %, однако на число MCU это влияния не оказывает. В конце «шума» ставился маркер EOS. Таким образом, блок, заполненный случайными данными, находится между значениями блока FFDA и маркера FFD9.

Оценка изображения проводилась в несколько этапов:

1) вычисление размера блока данных;

2) подсчет числа блоков MCU (до начала фона, выдаваемого JPEG-декодером);

3) расчет размера MCU при неполном заполнении картинки.

Для оценки размеров MCU изображения, представленного только одной яркостной компонентой, тестовый файл cameraman.bmp (256 × 256 пикселей в оттенках серого) был сохранен в формате JPEG с различным качеством, после чего блок файла, отвечающий за описание картинки, был заменен эквивалентным объемом случайных данных. Результаты сопоставления отображаемых блоков MCU с размером блока данных, полученного заполнением шумом, сведены в табл. 3.

В ходе сравнения были получены средние значения для одного блока от 78 до 91 бита, при этом результаты для заполненных полностью изображений считались избыточными и не учитывались в дальнейшем.

Исходное изображение и картинки, полученные путем записи и отображения порций случайных данных, показаны на рис. 4. Параметр «качество» характеризует JPEG-файл, согласно

■ ***Таблица 3.*** *Экспериментальная оценка среднего размера MCU*

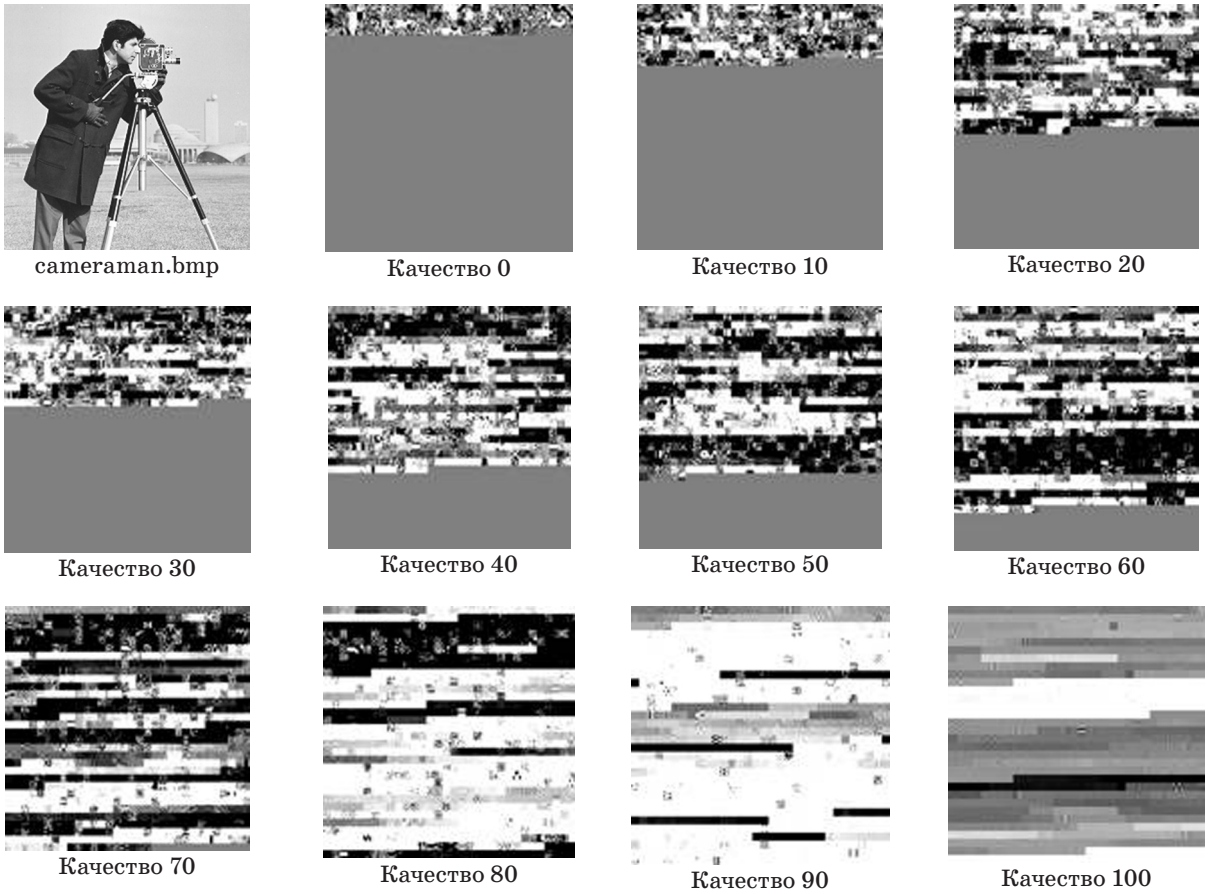
Качество	Блок данных, Б	Число MCU	Размер MCU, бит
0	1392	123	91
10	2705	244	89
20	4252	409	83
30	5624	527	85
40	6704	685	78
50	7718	710	87
60	8833	843	84
70	10662	1011	84
80	13455	1024*	105
90	19922	1024*	156
100	48961	1024*	383

* Полное заполнение изображения.

заголовку которого производится отображение шума.

Ошибки, приводящие к краху процесса декодирования, не проявились — на экране отображался шум. Полученный шум носил блоковый характер, прослеживалась зависимость наполненности картинки от размера файла. Размер MCU серого изображения каждого типа зачастую чуть превышает расчетные 81 бит, но находится в пределах расчетной погрешности.

С увеличением параметра «качество» характер отображаемых картинок меняется — переходы между блоками становятся более плавными. Это связано с тем, что заголовки файлов для хранения изображения заданного качества имеют разные таблицы квантования. Поскольку таблица квантования стандартом не регламентируется, но ее знание необходимо для последующего восстановления изображения, она передается в заголовке выходных данных. Значения в таблице связаны с требуемым качеством изображения, так как в таблице определен шаг значений коэффициентов, данные между которыми будут потеряны. При интерпретации эквивалентного объема шума декодер для изображений низкого качества делает резкие переходы яркости между блоками и внутри них. При высоком качестве переходы между значениями минимальны — это приводит к появлению больших однотонных полос.



■ ***Рис. 4.*** *Блоковый шум изображения в оттенках серого*

Заключение

Рассмотренный в работе подход применим и к оценке защищенных файлов сжатых изображений. Анализ поведения декодера при разборе шума показал, что **JPEG-декодер способен разбирать** шум, при этом искажение изображения в кадре будет блоковым. При разборе шума в теле **JPEG-файла кодер способен графически отобразить** все данные при условии, что в этих данных нет двухбайтовых блоков, похожих на маркер. Возможно неполное заполнение отображаемой области, при этом уровень заполнения визуализируемой картинки зависит от степени сжатия изображения. Стандарт не предусматривает фиксацию размера выходного файла, и такого рода информацию можно использовать для анализа последовательностей изображений.

Системы видеонаблюдения, использующие в своей основе **JPEG-сжатие, уязвимы для статистического** анализа битовой скорости. Особенности работы алгоритма сжатия по удалению избыточности того или иного рода приводят к появлению вторичных данных. Их наличие и знание основных особенностей алгоритма делает систему уязвимой.

Для сокрытия скорости передачи рекомендуется передавать данные равными порциями в любой момент времени, независимо от достижимой степени компрессии при любых качественных характеристиках изображений. Данные о размере блока можно использовать для маскирования вторичной информации в файле — это позволяет сместить вверх нижнюю границу размера кадра. Качественные характеристики изображения и параметры сжатия кодером влияют на размер конечного файла. Однако эти параметры не оказывают никакого влияния на объем данных, требуемых для заполнения выводимой картинки блоковым шумом. Анализ вторичной информации возможен как по битовой скорости потока, так и по размеру отображаемой части изображений. Информацию о размере блока можно использовать для маскирования незаполненных частей визуализируемой картинки.

Литература

1. **Санкин П. С., Литвинов М. Ю.** Особенности оценки содержимого сжатого видеопотока // Информационно-управляющие системы. 2009. № 3. С. 45–48.
2. **CCITT** Rec. T.81 (1992 E) | ISO/IEC 10918-1: 1993(E). Information technology — Digital compression and coding of continuous-tone still images: Requirements and guidelines. http://www.w3.org/Graphics/JPEG/itu-t81.pdf (дата обращения: 20.08.2008)