

СОДЕРЖАНИЕ

1 Задание на практическую работу	3
2 Краткая теоретическая часть	4
2.1 Метод QIM.....	5
2.2 Оценка эффективности встраивания.....	6
3 Программная реализация	8
3.1 Результаты работы программы.....	11
3.2 Результаты вычислительных экспериментов.....	12
3.3 Сравнение гистограмм изображений до и после встраивания	29
4 Выводы	32
5 Литература	33

1 Задание на практическую работу

Целью данной работы является приобретение навыков программной реализации стеганографического встраивания информации в цифровые изображения.

В рамках практической работы необходимо выполнить следующее:

- 1) Написать программную реализацию одного из следующих стеганографических методов для цифровых изображений по выбору студента: QIM, PVD, NMI;
- 2) Провести вычислительные эксперименты с полученной программной реализацией и сделать выводы об эффективности рассмотренного метода встраивания;
- 3) Подготовить отчет о выполнении работы.

2 Краткая теоретическая часть

Стеганография – это наука о скрытой передаче и хранении информации таким образом, чтобы сам факт наличия этой информации был тайной для третьих лиц. Цифровая стеганография работает с цифровой информацией. В качестве контейнеров для секретных сообщений могут выступать самые разные цифровые объекты: мультимедиа-данные, исполняемые файлы, интернет-трафик, данные с датчиков «интернета вещей» и т.д.

Одним из наиболее распространенных типов контейнеров для стеганографического встраивания являются цифровые изображения. В первую очередь, это связано с их высокой избыточностью: наиболее распространенная цветовая модель RGB содержит 2^{24} различных цветов, поэтому небольшое изменение оттенка отдельных пикселей остается незаметным для глаза человека. Кроме того, цифровые изображения обладают, так называемой, пространственной избыточностью, которая состоит в том, что соседние пиксели изображения в большинстве случаев похожи друг на друга. Это свойство также широко используется в методах обработки цифровых изображений.

Существует большое количество различных методов встраивания дополнительной информации в цифровые изображения, отличающихся разными особенностями. Все существующие методы делятся на два больших класса: методы пространственного встраивания и методы частотного встраивания. Методы пространственного встраивания напрямую изменяют значения пикселей изображения, в то время как методы частотного встраивания вносят изменения в значения частотных коэффициентов, полученных после применения некоторого частотного преобразования к матрице пикселей.

2.1 Метод QIM

Метод модуляции индекса квантования (Quantization Index Modulation, QIM) [3] состоит в изменении значений пикселей изображения в зависимости от значений встраиваемых битов сообщения. Данная операция называется модуляцией. Встраивание бита m_i осуществляется в пиксель контейнера P_i по формуле (2.1.1):

$$P'_i = q \cdot \left\lfloor \frac{P_i}{q} \right\rfloor + \frac{q}{2} \cdot m_i, \quad (2.1.1)$$

где q – шаг квантования (четное число), $\lfloor \dots \rfloor$ – взятие целой части от деления.

Для извлечения моделируется ситуация встраивания в пиксель P'_i нулевого и единичного битов, после чего результирующий бит определяется на основании того, к какому из двух значений ближе имеющееся значение P'_i . Формула извлечения одного бита представлена в формуле (2.1.2):

$$m_i = \arg \min_{e \in \{0,1\}} |P'_i - [P''_i]_e|, [P''_i]_e = q \cdot \left\lfloor \frac{P'_i}{q} \right\rfloor + \frac{q}{2} \cdot e \quad (2.1.2)$$

2.2 Оценка эффективности встраивания

Для оценки качества стеганографического встраивания информации в изображения используются различные характеристики. Наиболее распространенными из них являются незаметность и емкость встраивания. Незаметность отражает различие между изображениями до и после встраивания и обычно оценивается метриками PSNR, MSE, RMSE, SSIM, формулы для нахождения данных метрик (2.2.1), (2.2.2), (2.2.3), (2.2.4) соответственно, представлены ниже:

$$\text{PSNR (dB)} = 10 \times \log_{10} \left(\frac{255^2}{\text{MSE}} \right) \quad (2.2.1)$$

$$\text{MSE} = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (P_{ij} - P'_{ij})^2 \quad (2.2.2)$$

$$\text{RMSE} = \sqrt{\text{MSE}}, \quad (2.2.3)$$

где P_{ij} – значение пикселя изображения-контейнера, P'_{ij} – значение пикселя изображения с вложением;

$$\text{SSIM} = \frac{(2\mu_P\mu_S + K_1) \times (2\sigma_{PS} + K_2)}{(\mu_P^2 + \mu_S^2 + K_1) \times (\sigma_P^2 + \sigma_S^2 + K_2)}, \quad (2.2.4)$$

где μ_P – среднее значение пикселей изображения-контейнера, μ_S – среднее значение пикселей изображения с вложением, σ_P^2 – дисперсия пикселей изображения-контейнера, σ_S^2 – дисперсия пикселей изображения с вложением, σ_{PS} – ковариация между пикселями обоих изображений, K_1 и K_2 – константы.

Емкость показывает, сколько битов встроенной информации приходится на каждый пиксель изображения, и выражается следующей формулой (2.2.5):

$$\text{EC (bpp)} = \frac{B}{M \times N}, \quad (2.2.5)$$

где $M \times N$ – размеры изображения, B – это общее число битов, встроенных в изображение.

Для оценки устойчивости встраивания к деструктивным воздействиям, таким как изменение яркости, JPEG-сжатие и т.д. обычно применяются метрики BER и NCC, их формулы представлены под номерами (2.2.6) и (2.2.7):

$$\text{BER} = \frac{B_e}{B}, \quad (2.2.6)$$

где B_e – количество ошибок, возникших при извлечении (измененных битов);

$$\text{NCC} = \frac{\sum_{x=1}^M \sum_{y=1}^N (W(x, y) \times W_{ext}(x, y))}{\sqrt{\sum_{x=1}^M \sum_{y=1}^N (W^2(x, y))} \sqrt{\sum_{x=1}^M \sum_{y=1}^N (W_{ext}^2(x, y))}}, \quad (2.2.7)$$

где $W(x, y)$ – исходное сообщение, $W_{ext}(x, y)$ – извлеченное сообщение.

Метрика NCC обычно применяется, когда встроенное сообщение является изображением.

Важным показателем качества стеганографического встраивания является устойчивость перед стегоанализом. Существует множество схем стегоанализа для пространственной области изображений. Простейшие из них анализируют сходство гистограмм изображений до и после встраивания.

После чего, происходит извлечение битов сообщения из стегоизображения, проходит по формуле (2.1.2). Код функции представлен на рисунке 3.5.

```

189 def str_to_bin(msg):
190     step_size = len(msg)
191     w = int((len(msg) + 1) / 2)
192
193     step_size = step_size * convert('M1') # В зависимости от выбранного метода в форме M1
194     step_size = step_size * convert('M2') # В форме w * step_size * convert(w, 1)
195
196     est_message_binary = [] # Инициализация массива для хранения сообщения
197
198     for v in range(len(msg) * step_size):
199         pos = 0
200         for i in range(step_size * step_size):
201             pos = (i * w)
202             # В зависимости от выбранного метода M1 и M2
203             r_modified = pos - step_size * convert(step_size, 1)
204             r2 = pos
205
206             # В зависимости от выбранного метода P 2 и w * convert(w, 1)
207             P_0 = q * (r_modified // 4)
208             P_1 = q * (r_modified // 4) + (q // 2)
209
210             a_1 = min(abs(r_modified_P_0), abs(r_modified_P_1))
211             (r_modified_P_0, r_modified_P_1) = (r_modified_P_1, r_modified_P_0)
212             est_message_binary.append(a_1)
213         else:
214             est_message_binary.append(1)
215
216     est_message_str = ""
217
218     est_message_str = binary_to_string(est_message_binary)
219
220     return est_message_str

```

Рисунок 3.5 – Функция извлечения EXTRACTION

Весь программный код доступен по ссылке: https://github.com/il3241/stegano_QIM.git

3.1 Результаты работы программы

Теперь перейдем к результатам работы программы. Для начала встроим сообщение, а после извлечем его. Пусть сообщением для встраивания будет “Never gonna give you up Never gonna let you down”, а изображение-контейнер представлено на рисунке 3.1.1, размер изображения 256 на 256 пикселей.



Рисунок 3.1.1 – Цветное изображение контейнер

Запустим программу, введем все необходимые настройки и выберем шагом квантования 12, данные действия представлены на рисунке 3.1.2.

```
Введите название входного файла (файл должен быть сохранен в папке /in_img):
cat.png
Что необходимо сделать с входным файлом:
1 - встраивание;
2 - извлечение;
1
Введите сообщение которое необходимо встроить в файл, оно должно быть сохранено в папке /message:
Never.txt
Введите шаг квантования:
12
```

Рисунок 3.1.2 – Запуск программы на встраивание изображения

Результатом этого действия является новое изображение с сокрытым в нем сообщением, показано на рисунке 3.1.3.



Рисунок 3.1.3 – Цветное изображение контейнер

Теперь извлечем сообщение из только что созданного стегоизображения. Для этого выполним запуск программы с настройками, представленными на рисунке 3.1.4.

```
Введите название входного файла (файл должен быть сохранен в папке /in_img):
cat_stego_never.png
Что необходимо сделать с входным файлом:
1 - встраивание;
2 - извлечение;
2
Введите шаг квантования:
12
Never gonna give you up Never gonna let you down
```

Рисунок 3.1.4 – Запуск программы на извлечение сообщения

Как видно из рисунка 3.1.4 извлечение прошло успешно.

3.2 Результаты вычислительных экспериментов

Встроим в изображение максимальный объем информации. Стоит заметить, что максимальное количество бит информации, которое можно вставить вычисляется по формуле (3.2.1).

$$I = h * w, \quad (3.2.1)$$

Где I – кол-во доступных пикселей для встраивания,

h – высота изображения,

w – ширина изображения.

В ходе экспериментов будет использовано изображение размером 256 на 256 пикселей, вычислим I , вычисления представлены в формуле (3.2.2).

$$I = 256 * 256 = 65536 \text{ пикселей.} \quad (3.2.2)$$

Изображение, в которое будет происходить вставка представлено на рисунке 3.2.1.



Рисунок 3.2.2 – Цветное изображение контейнер

Встроим в него максимальный возможный объем информации и оценим незаметность этой операции, при разных шагах квантования, представлено на рисунке 3.2.3 – 3.2.6.

```
Введите название входного файла (файл должен быть сохранен в папке /in_img):
cat.png
Что необходимо сделать с входным файлом:
1 - встраивание;
2 - извлечение.
1
Введите сообщение которое необходимо встроить в файл, оно должно быть сохранено в папке /message:
longues.txt
Введите шаг квантования:
8
-----
PSNR
12.111087205078125
-----
PSNR
12.4679911151912
-----
RMSE
1.41069300215262047
-----
SSIM
0.95084151002383925
-----
EC
1.0
```

Рисунок 3.2.3 – Запуск программы на встраивание максимального объема сообщения с оценкой незаметности при шаге квантования 8



Рисунок 3.2.4 – Полученное изображение, при встраивании максимального объема информации с шагом квантования 8

```

Введите название входного файла (файл должен быть сохранен в папке ./in_img):
cat.png
Что необходимо сделать с входным файлом:
1 - встраивание;
2 - извлечение.
1
Введите сообщение, которое необходимо встроить в файл, оно должно быть сохранено в папке ./message:
message.txt
Введите шаг квантования:
200
=====
MSG:
[1625, 7285, 3888, 1789]
RGB:
[7, 476228284999235]
MSG:
[187, 8273888, 3632, 304]
SDR:
[8, 0419585579955762]
EC:
1.0

```

Рисунок 3.2.5 – Запуск программы на встраивание максимального объема сообщения с оценкой незаметности при шаге квантования 200

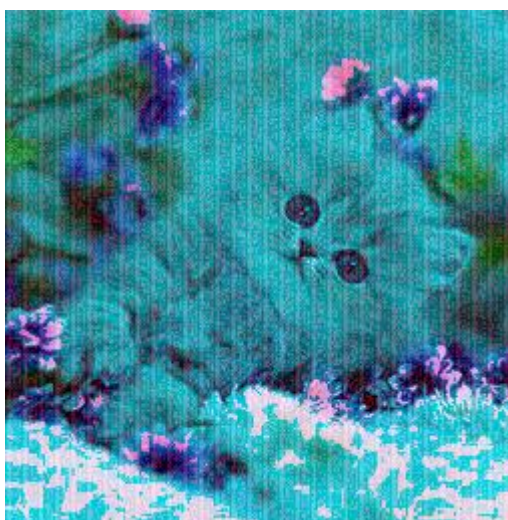


Рисунок 3.2.6 – Полученное изображение, при встраивании максимального объема информации с шагом квантования 200

Первый параметр, на который стоит обратить внимание это ЕС. Данная величина показывает отношение количества встроенных бит к размеру изображения, соответственно, чем ближе ЕС к 1, тем больше пикселей из доступных было использовано для встраивания. Как можно заметить при проведении эксперимента данная величина в обоих случаях была

равна 1 – это значит, что было встроено сообщение максимальной длины. Теперь перейдем к остальным метрикам. MSE – метрика показывающая разницу между значениями пикселей изображения до и после встраивания, соответственно, чем меньше это значение, тем незаметнее встраивание. Как видно из рисунков 3.2.3 и 3.2.5, при шаге квантования 8 MSE равно 11.810, а при шаге в 200 MSE равно 11626.728, что в разы больше, а следовательно, при шаге квантования 8 встраивание незаметнее. Сразу отметим такую метрику, как RMSE, которая вычисляется, как корень из MSE. PSNR, метрика, так же характеризующая незаметность, при этом чем больше данное значение тем вставка незаметнее, в рассматриваемом случае, при шаге квантования 8 PSNR равно 37.407, а при шаге 200 PSNR равно 7.476. Так же одной из метрик демонстрирующей незаметность встраивания является SSIM. Значение данной метрики лежит в диапазоне от 0 до 1, где 1 говорит о полном сходстве изображения до и после встраивания. Относительно нашего случая при шаге в 8 SSIM равно 0.958, что говорит о практически полном сходстве изображений, а при шаге в 200 SSIM равно 0.041. Отдельно стоит заметить, что при большом шаге квантования встраивание можно обнаружить просто невооруженным взглядом.

Теперь встроим в изображение разные объемы информации при шаге квантования 10. Для этого используем все то же цветное изображение, показанное на рисунке 3.2.2. Выполним программу с настройками, продемонстрированными ниже на рисунках 3.2.7, 3.2.8 и 3.2.9

```

Введите название входного файла (файл должен быть сохранен в папке ./in_img):
cat.png
Что необходимо сделать с входным файлом:
1 - встраивание
2 - извлечение
1
Введите сообщение которое необходимо встроить в файл, оно должно быть сохранено в папке ./message:
shortaes.txt
Введите шаг квантования:
10
=====
MSE
0.01104736328125
-----
PSNR
37.69821725294463
-----
RMSE
0.10510043796776336
-----
SSIM
0.9999864572986229
-----
EC
0.00048838125
-----

```

Рисунок 3.2.7 – Запуск программы на встраивание сообщения размером 0.04% от доступного объема с оценкой незаметности, при шаге квантования 10

```

Введите название входного файла (файл должен быть сохранен в папке ./in_img):
cat.png
Что необходимо сделать с входным файлом:
1 - встраивание
2 - извлечение
1
Введите сообщение которое необходимо встроить в файл, оно должно быть сохранено в папке ./message:
mes.txt
Введите шаг квантования:
10
=====
MSE
0.878895881445313
-----
PSNR
18.19411870863245
-----
RMSE
0.9375295221887035
-----
SSIM
0.950713848689183
-----
EC
0.587568359275
-----

```

Рисунок 3.2.8 – Запуск программы на встраивание сообщения размером 50 % от доступного объема с оценкой незаметности, при шаге квантования 10

```

Введите название исходного файла (файл должен быть сохранен в папке /in_img):
cat.png
Что необходимо сделать с исходным файлом:
1 - встраивание
2 - извлечение
3
Введите сообщение, которое необходимо встроить в файл, оно должно быть сохранено в папке /message:
70.txt
Введите шаг квантования:
10
-----
PGE
14.596306665839832
-----
PSNR
10.408481157035870
-----
RMSE
3.820498222090948
-----
SSIM
0.9506009108544142
-----
EC
0.7528076171875
-----

```

Рисунок 3.2.9 – Запуск программы на встраивание сообщения размером 75% от доступного объема с оценкой незаметности, при шаге квантования 10

При этом результаты встраивания представлены на рисунках 3.2.10, 3.2.11 и 3.2.12 для соответственно 0.04%, 50% и 75% занимаемых сообщением объема.



Рисунок 3.2.10 – Полученное изображение, при встраивании информации размером 0.04% от доступного объема с шагом квантования 10



Рисунок 3.2.11 – Полученное изображение, при встраивании информации размером 50% от доступного объема с шагом квантования 10



Рисунок 3.2.12 – Полученное изображение, при встраивании информации размером 75% от доступного объема с шагом квантования 10

В данном случае, невооруженным глазом встраивание не обнаружить. Обратимся к собранным числовым показателям, собранным в таблице 3.2.1.

Таблица 3.2.1 – Метрики собранные при встраивании разного объема информации, при шаге квантования 10

	0.04%	50%	75%
MSE	0.011	9.878	14.596
PSNR	67.698	38.184	36.488
RMSE	0.105	3.142	3.820
SSIM	0.999	0.966	0.958
EC	0.0004	0.507	0.752

Как можно заметить из таблицы, чем больше размер сообщения, тем заметнее встраивание, однако стоит заметить, что в любом случае SSIM остается близок к 1, а значит все три встраивания можно считать незаметными. Следовательно, при малых шагах квантования незаметность мало зависит от объема встраиваемой информации.

Проведем тоже самое, но при большом шаге квантования равном 100. Настройки, используемые при запуске программ приведены на рисунках 3.2.13, 3.2.14 и 3.2.15, а соответствующие им изображения, получаемые при выполнении программы показаны на рисунках 3.2.16, 3.2.17 и 3.2.18 соответственно.

```

Введите название входного файла (файл должен быть сохранен в папке /in_img)
cat.png
Что необходимо сделать с входным файлом:
1 - встраивание;
2 - извлечение.
1
Введите сообщение которое необходимо встроить в файл, оно должно быть сохранено в папке /message
shortes.txt
Введите шаг квантования:
100
=====
MSE
1.15118486263125
PSNR
47.5183553489797
RMSE
1.072834685325808
SSIM
0.995768763123626
EC
0.00048828125
=====

```

Рисунок 3.2.13 – Запуск программы на встраивание сообщения размером 0.04% от доступного объема с оценкой незаметности, при шаге квантования 100


```

Введите название входного файла (файл должен быть сохранен в папке /in_img):
cat.png
Что необходимо сделать с входным файлом:
1 - встраивание;
2 - извлечение;
1
Введите сообщение, которое необходимо встроить в файл, оно должно быть сохранено в папке /message:
mes.txt
Введите шаг квантования:
100
=====
PSE
0002.1099416825
-----
PSNR
18.1213072106772
-----
RPSE
01.657383679167944
-----
SSIM
0.5900308004052000
-----
EC
0.587568359375
-----

```

Рисунок 3.2.14 – Запуск программы на встраивание сообщения размером 50% от доступного объема с оценкой незаметности, при шаге квантования 100

```

Введите название входного файла (файл должен быть сохранен в папке /in_img):
cat.png
Что необходимо сделать с входным файлом:
1 - встраивание;
2 - извлечение;
1
Введите сообщение, которое необходимо встроить в файл, оно должно быть сохранено в папке /message:
m.txt
Введите шаг квантования:
100
=====
PSE
0448.043371526312
-----
PSNR
16.52110877271766
-----
RPSE
06.061047955060294
-----
SSIM
0.4241696513070443
-----
EC
0.7528076171875
-----

```

Рисунок 3.2.15 – Запуск программы на встраивание сообщения размером 75% от доступного объема с оценкой незаметности, при шаге квантования 100



Рисунок 3.2.16 – Полученное изображение, при встраивании информации размером 0.04% от доступного объема с шагом квантования 100



Рисунок 3.2.17 – Полученное изображение, при встраивании информации размером 50% от доступного объема с шагом квантования 100



Рисунок 3.2.18 – Полученное изображение, при встраивании информации размером 75% от доступного объема с шагом квантования 10

Снова обратимся к собранным числовым показателям, собранным в таблице 3.2.2.

Таблица 3.2.2 – Метрики собранные при встраивание разного объема информации, при шаге квантования 100

	0.04%	50%	75%
MSE	1.151	1002.189	1448.643
PSNR	47.519	18.121	16.521
RMSE	1.0729	31.657	38.061
SSIM	0.999	0.590	0.424
EC	0.0004	0.507	0.752

В данном случае незаметным можно считать только встраивание информации, занимающей всего 0.04% от доступного объема, стоит заметить, что встраивание таких малых объемов,

будет заметно, только при очень большом шаге квантования. Для остальных двух объемов встраивание стало заметно. Теперь его можно обнаружить, не только по метрикам, но и даже невооруженным глазом. Проанализируем полученные результаты для 50% и 75%: MSE и соответственно RMSE возросли для обоих объемов, а PSNR уменьшился, это все говорит о усилившейся заметности встраивания. Метрика SSIM теперь близка не к 1, а к 0.5, что опять же указывает на сильную заметность такого встраивания. Из этих опытов можно сделать вывод, что незаметность встраивания в первую очередь зависит от шага квантования, а не от объема информации.

Теперь проведем эксперимент с извлечением сообщения из стегоизображения в условиях отсутствия постобработки и при наличии различных операций обработки изображений. Для этого воспользуемся тем же изображением, представленным на рисунке 3.1.1. При этом при постобработке получаемое стегоизображение будет подвергаться следующим фильтрам:

- Размытие
- Изменение резкости
- Сглаживание
- Выделение контура
- Выделение деталей
- Выпуклость
- ЧБ фильтр

После наложения на стегоизображение фильтра происходит попытка выделения сообщения. Для оценки этого выделения применяется мерка BER, для каждого из фильтров данная мерка имеет название в формате BER_filtername. При этом встраиваемое сообщение будет занимать 100% объема контейнера, а шаг квантования при этом будет равен 10, 100, 1000.

Сначала проведем вычисления при шаге квантования равном 10, настройки для запуска программы представлены на рисунке 3.2.19, а изображения для соответствующих фильтров представлены на рисунках 3.2.20 – 3.2.26.



Рисунок 3.2.19 – Запуск программы на встраивание сообщения размером 100% от доступного объема с подсчетом метрики BER и шагом квантования 10



Рисунок 3.2.20 – Полученное изображение, при использовании фильтра размытия с шагом квантования 10



Рисунок 3.2.21 – Полученное изображение, при использовании фильтра изменения резкости с шагом квантования 10



Рисунок 3.2.22 – Полученное изображение, при использовании фильтра сглаживания с шагом квантования 10



Рисунок 3.2.23 – Полученное изображение, при использовании фильтра выделения контура с шагом квантования 10



Рисунок 3.2.24 – Полученное изображение, при использовании фильтра выделения деталей с шагом квантования 10

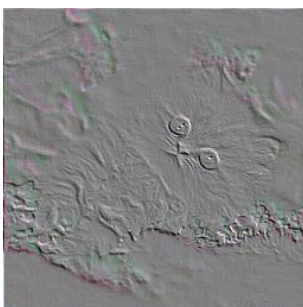


Рисунок 3.2.25 – Полученное изображение, при использовании фильтра выпуклости с шагом квантования 10



Рисунок 3.2.26 – Полученное изображение, при использовании чёрно-белого фильтра с шагом квантования 10

Как видно из рисунка 3.2.19 все метрики `BER_filtername` находятся в районе 0.5, что означает, что извлечь исходное сообщение невозможно, слишком много ошибок.

Проведем тот же опыт с шагом квантования равным 100. Настройки программы приведены на рисунке 3.2.27, а картинки полученных изображений на рисунках 3.2.28 – 3.2.33.

```

Введите название входного файла (файл должен быть сохранен в папке C:\_img):
cat.png
Введите название файла с выходным файлом:
1 - астрамем;
2 - астрамем;
3 - астрамем;
1
Введите сообщение, которое необходимо встроить в файл, оно должно быть сохранено в папке /message:
message.txt
Введите шаг квантования:
100
=====
HEX
2254.65686309375
=====
PDM
14.68686309375
=====
HEX
47.486309375
=====
HEX
0.27086309375
=====
EC
1.0
=====
Введите устойчивость астрамем к деструктивным воздействиям:
да - 1
нет - 2
1
=====
BER_size
0.5487757265625
=====
BER_changen
0.47686309375
=====
BER_message
0.486309375
=====
BER_message1
0.25086309375
=====
BER_detail1
0.27686309375
=====
BER_message2
0.5117686309375
=====
BER_message3
0.45186309375
=====

```

Рисунок 3.2.27 – Запуск программы на встраивание сообщения размером 100% от доступного объема с подсчетом метрики BER и шагом квантования 100



Рисунок 3.2.28 – Полученное изображение, при использовании фильтра размытия с шагом квантования 100



Рисунок 3.2.29 – Полученное изображение, при использовании фильтра изменения резкости с шагом квантования 100



Рисунок 3.2.30 – Полученное изображение, при использовании фильтра сглаживания с шагом квантования 100

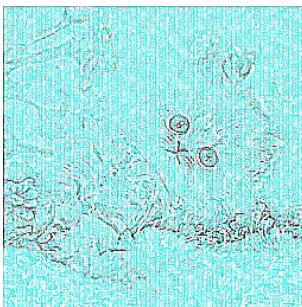


Рисунок 3.2.31 – Полученное изображение, при использовании фильтра выделения контура с шагом квантования 100



Рисунок 3.2.32 – Полученное изображение, при использовании фильтра выделения деталей с шагом квантования 100

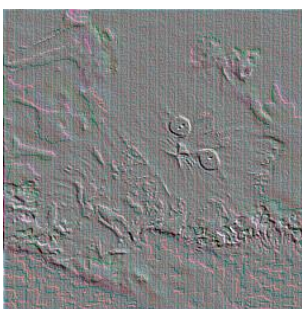


Рисунок 3.2.33 – Полученное изображение, при использовании фильтра выпуклости с шагом квантования 100

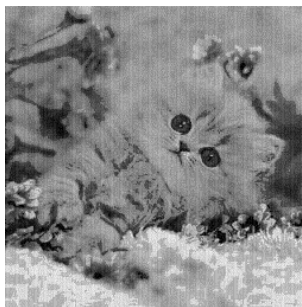


Рисунок 3.2.34 – Полученное изображение, при использовании чёрно-белого фильтра с шагом квантования 100

В данном случае метрика BER для фильтра сглаживания и выделения контура существенно снизилась. При извлечении с фильтром сглаживания даже появляются осмысленные обрывки исходного сообщения, показано на рисунке 3.2.35.



Рисунок 3.2.35 – Обрывки сообщения при извлечении из стегоизображения с фильтром сглаживания

Для фильтра выделения контура метрика BER все еще принимает слишком высокое значение, чтобы можно было извлечь осмысленное сообщение.

Проведем тот же опыт с шагом квантования равным 1000. Настройки программы приведены на рисунке 3.2.36, а картинки полученных изображений на рисунках 3.2.37 – 3.2.43.

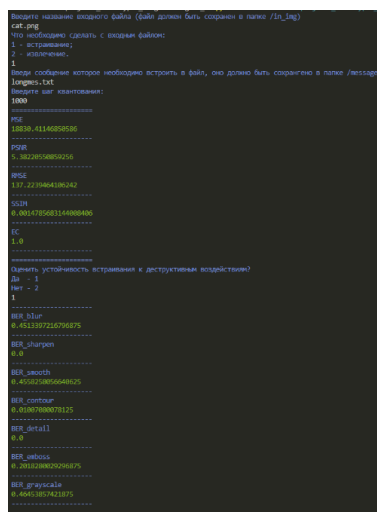


Рисунок 3.2.36 – Запуск программы на встраивание сообщения размером 100% от доступного объема с подчетом метрики BER и шагом квантования 1000

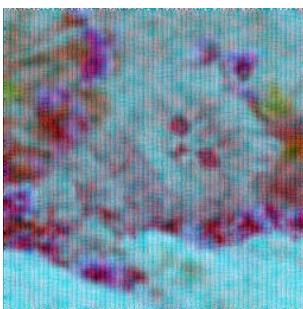


Рисунок 3.2.37 – Полученное изображение, при использовании фильтра размытия с шагом квантования 1000

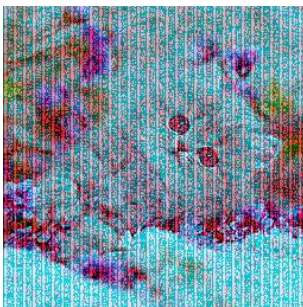


Рисунок 3.2.38 – Полученное изображение, при использовании фильтра изменения резкости с шагом квантования 1000

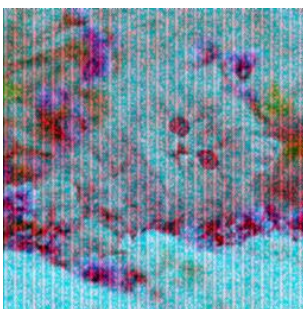


Рисунок 3.2.39 – Полученное изображение, при использовании фильтра сглаживания с шагом квантования 1000

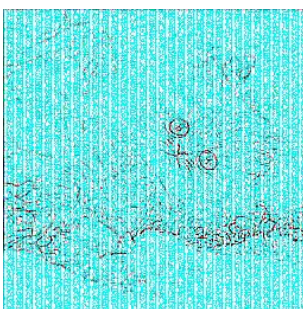


Рисунок 3.2.40 – Полученное изображение, при использовании фильтра выделения контура с шагом квантования 1000

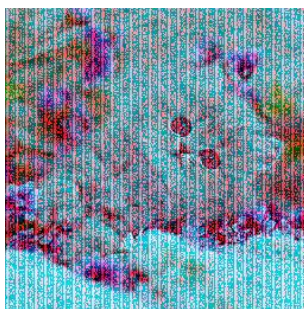


Рисунок 3.2.41 – Полученное изображение, при использовании фильтра выделения деталей с шагом квантования 1000

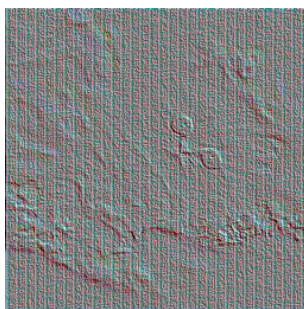


Рисунок 3.2.42 – Полученное изображение, при использовании фильтра выпуклости с шагом квантования 1000

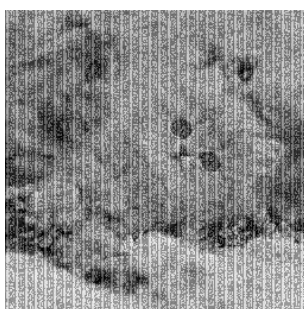


Рисунок 3.2.43 – Полученное изображение, при использовании чёрно-белого фильтра с шагом квантования 1000

Как можно заметить из рисунка 3.2.36 можем заметить, что при большом шаге квантования метрика BER для таких фильтров, как изменение резкости и выделение деталей стало равно нулю, а для фильтра выделения контура практически равна нулю. Из этого можно сделать вывод, что извлечение сообщения из стегоизображений с данным фильтрами приведет к получению исходной информации, приведено на рисунках 3.2.44 - 3.2.46.

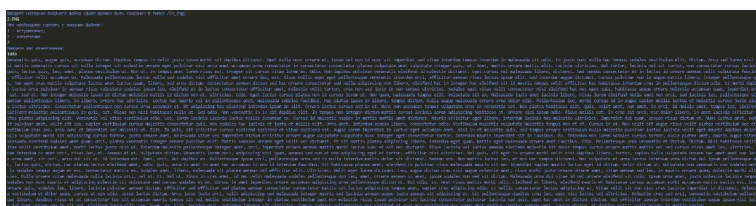
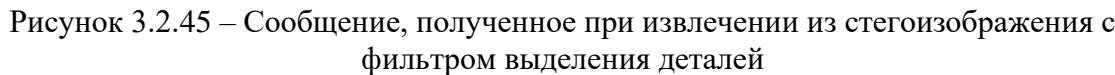
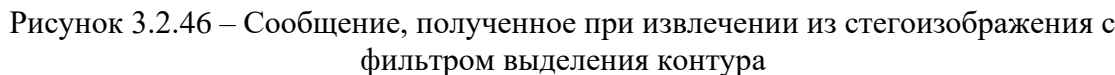


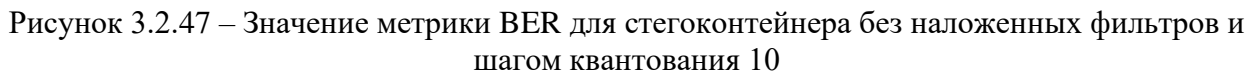
Рисунок 3.2.44 – Сообщение, полученное при извлечении из стегоизображения с фильтром изменения резкости



Стоит заметить, что у данного фильтра показатель BER был не 0, а 0.01, следовательно в сообщении могут быть ошибки.



Отдельно стоит отметить, что для стегоизображения, не подвергнутого фильтрам и другим преобразованиям значение метрики BER, в данном случае она имеет название BER_0, будет равно нулю, продемонстрировано на рисунках 3.2.47 – 3.2.49.



[illegible]

Рисунок 3.2.48 – Значение метрики BER для стежоконтейнера без наложенных фильтров и шагом квантования 100.

[illegible]

Рисунок 3.2.49 – Значение метрики BER для стежоконтейнера без наложенных фильтров и шагом квантования 1000

3.3 Сравнение гистограмм изображений до и после встраивания

Гистограммы построены для оригинального изображения и стегоизображений полученных встраиванием информации, занимающей 100% доступного объема, при шаге квантования 10, 100, 1000, с помощью библиотеки matplotlib [2]. Гистограмма для оригинального изображения представлена на рисунке 3.3.1 Настройки программы при запуске продемонстрированы на рисунке 3.3.2, 3.3.4, 3.3.6, а соответствующие гистограммы показаны на рисунках 3.3.3, 3.3.5, 3.3.7.

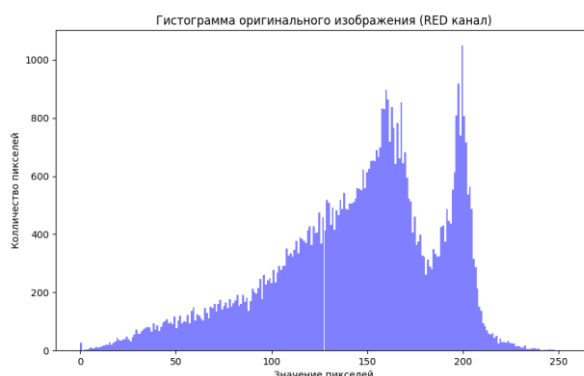


Рисунок 3.3.1 – Гистограмма красного канала оригинального изображения



Рисунок 3.3.2 – Настройки запуска программ для демонстрации гистограмм и шагом квантования 10

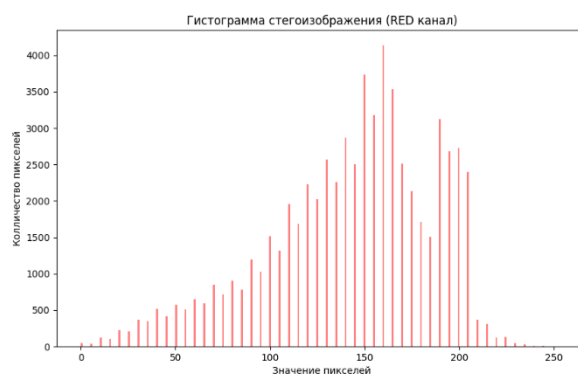


Рисунок 3.3.3 – Гистограмма красного канала стегоизображения и шага квантования 10

```

#!/usr/bin/perl
# Программа для демонстрации гистограмм и шага квантования
# 1. Ввод данных
# 2. Обработка
# 3. Вывод результатов
# 4. Сохранение в файл
# 5. Завершение

# Параметры
my $input = "input.png";
my $output = "output.png";
my $step = 100;

# Проверка наличия файлов
if (!-f $input) {
    print "Файл $input не найден\n";
    exit 1;
}

# Загрузка изображения
my $img = image_load($input);

# Проверка формата
if ($img->format != 'RGB') {
    print "Изображение не в формате RGB\n";
    exit 1;
}

# Квантование
my $img_quant = image_quant($img, $step);

# Сохранение
image_save($img_quant, $output);

# Вывод
print "Гистограмма построена\n";
print "Шаг квантования: $step\n";
print "Файл сохранен: $output\n";

```

Рисунок 3.3.4 – Настройки запуска программ для демонстрации гистограмм и шагом квантования 100

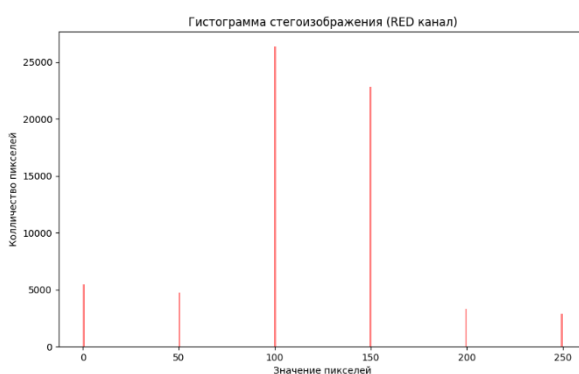


Рисунок 3.3.5 – Гистограмма красного канала стегоизображения и шага квантования 100

```

#!/usr/bin/perl
# Программа для демонстрации гистограмм и шага квантования
# 1. Ввод данных
# 2. Обработка
# 3. Вывод результатов
# 4. Сохранение в файл
# 5. Завершение

# Параметры
my $input = "input.png";
my $output = "output.png";
my $step = 1000;

# Проверка наличия файлов
if (!-f $input) {
    print "Файл $input не найден\n";
    exit 1;
}

# Загрузка изображения
my $img = image_load($input);

# Проверка формата
if ($img->format != 'RGB') {
    print "Изображение не в формате RGB\n";
    exit 1;
}

# Квантование
my $img_quant = image_quant($img, $step);

# Сохранение
image_save($img_quant, $output);

# Вывод
print "Гистограмма построена\n";
print "Шаг квантования: $step\n";
print "Файл сохранен: $output\n";

```

Рисунок 3.3.6 – Настройки запуска программ для демонстрации гистограмм и шагом квантования 1000

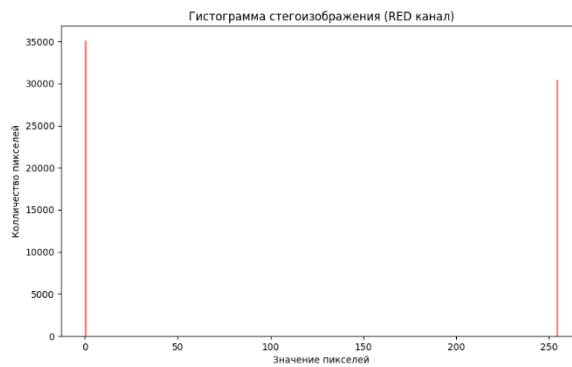


Рисунок 3.3.7 – Гистограмма красного канала стегоизображения и шага квантования 1000

Как можно заметить, чем больше шаг квантования, тем меньше количество возможных значений пикселя. Это вытекает из формулы (2.3.1), используемой при встраивании. К примеру, при шаге квантования в 1000 пиксель может принимать только значения 256, либо 0 в зависимости от бита сообщения.

4 Выводы

В данной практической работе были приобретены навыки программной реализации стенографического встраивания информации в цифровые изображения. Был программно реализован алгоритм QIM, а также проведены вычислительные эксперименты с различными параметрами, такими как размер встраиваемого изображения, шаг квантования, наложение фильтров и оценка робастности. Так же были построены гистограммы для оригинального изображения и стегоконтейнеров с разными шагами квантования. При увеличении сообщения в меньшей мере и при увеличении шага квантования в большой мере уменьшается незаметность встраивания, но при этом сообщение становится более стойким к фильтрам и другим видам постобработки, а также при увеличении шага квантования уменьшается количество возможных значений для пикселя, что вытекает из формулы используемой при встраивании, отдельно стоит заметить, что при шаге квантования больше 256, значение пикселя после встраивания перестает зависеть от его значения до встраивания и принимает значение равное либо нулю, либо половине шага квантования, но не больше 256.

5 Литература

- [1] Документация библиотеки PIL – URL:
<https://pillow.readthedocs.io/en/stable/index.html> Fridrich J. Steganography in Digital
- [2] Документация библиотеки Matplotlib – URL:
<https://matplotlib.org/stable/index.html>
- [3] Mitekin V. A new QIM-based watermarking algorithm robust against multi-image histogram attack / V. Mitekin, V. Fedoseev // Procedia Engineering. – 2017. – Vol. 201. – P. 453-462.