Санкт-Петербургский государственный университет
Прикладная математика и информатика

Отчет по научно-исследовательской работе
Встраивание информации (watermarking) в черно-белое изображение

Выполнил:
Самарин Игорь Александрович
Группа: 19.Б04-мм

Научный руководитель:
Кандидат физ.-мат. наук, доцент
Голяндина Нина Эдуардовна
Кафедра статистического моделирования

Санкт-Петербург
2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Проблема защиты и сокрытия информации известна из прошлого. С развитием систем информационных технологий и цифровизации общества она стала только актуальнее.

Системы обеспечения информационной безопасности различают на стеганографию и криптографию. Стеганография – это наука о тайной передаче информации путем сокрытия самого факта передачи. В отличие от криптографии, стеганография скрывает не только информацию, но и факт её наличия. Процесс стеганографии, как правило, включает в себя размещение сообщения в некотором носителе – контейнере, предназначенном для транспортировки. Контейнер обязан не терять свои естественные функции, а факт наличия скрытого сообщения должно быть как можно сложнее обнаружить. В качестве контейнеров зачастую используют медиа-форматы файлов, такие как MP3, JPEG, AVI и др. Они поддерживают хранение метаданных и, в силу своей распространенности, неочевидны как носители скрытой информации пользователям.

Целью данной курсовой работы является проектирование и написание программы встраивания информации в черно-белое изображение посредством использования сингулярного разложения матрицы (SVD).

ЛИНЕЙНАЯ АЛГЕБРА

На этапе разработки алгоритма были использованы следующие понятия из линейной алгебры.

СОБСТВЕННЫЕ ВЕКТОРЫ И ЗНАЧЕНИЯ

Собственным вектором матрицы $\boldsymbol{A}$ назовем ненулевой вектор v – такой, что умножение $\boldsymbol{A}$ на v изменяет лишь масштаб v:

$$\boldsymbol{Av} = \boldsymbol{\lambda v}$$

Скаляр λ будем называть собственным значением.

ОРТОГОНАЛЬНОСТЬ И ОРТОНОРМАЛЬНОСТЬ

Будем говорить, что вектора u, v ортогональны, если $(u, v) = 0$ и будем обозначать как $u \perp v$. Семейство $\{v_i\}_{i \in I}$ векторов называется ортогональным, если $v_i \perp v_j$ при $i \neq j$.

Базис $B = \{v_i\}_{i \in I}$ назовем ортонормированным, если:

$$(v_i, v_j) = \begin{cases} 1, \text{если } i = j \\ 0, \text{если } i \neq j \end{cases}$$

СИНГУЛЯРНОЕ РАЗЛОЖЕНИЕ

Сингулярное разложение (**Singular Value Decomposition**) – это способ разложения матрицы по сингулярным векторам и сингулярным числам. Сингулярное разложение есть у любой необязательно квадратной вещественной матрицы.

Исходная матрица $\boldsymbol{A}[\boldsymbol{m} \times \boldsymbol{n}]$ записывается в виде произведения трех матриц:

$$\boldsymbol{A} = \boldsymbol{UDV^T}$$

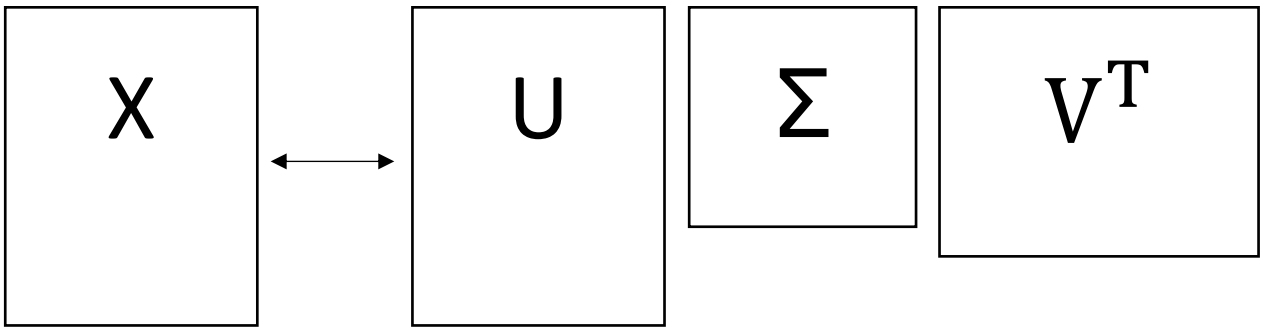
Элементы на диагонали $\boldsymbol{D}[\boldsymbol{m} \times \boldsymbol{n}]$ называются сингулярными значениями матрицы $\boldsymbol{A}$. Столбцы $\boldsymbol{U}[\boldsymbol{m} \times \boldsymbol{m}]$ называются левыми сингулярными векторами, а столбцы $\boldsymbol{V}[\boldsymbol{n} \times \boldsymbol{n}]$ – правыми сингулярными векторами.

Связь с собственными векторами и значениями: Левыми сингулярными векторами являются собственные вектора матрицы $\boldsymbol{AA^T}$, правыми сингулярными векторами – с.в. матрицы $\boldsymbol{A^TA}$, сингулярные числа – квадратные корни из ненулевых, положительных собственных значений матриц $\boldsymbol{AA^T}$ ($\boldsymbol{A^TA}$).

Матрицы U, V выбираются так, чтобы диагональные элементы матрицы D имели вид:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$$

Матрицы $\boldsymbol{U}[\boldsymbol{m} \times \boldsymbol{m}]$ и $\boldsymbol{V}[\boldsymbol{n} \times \boldsymbol{n}]$ являются ортогональными матрицами.



АЛГОРИТМ ВСТРАИВАНИЯ

Пусть дана конечная последовательность символов $\boldsymbol{T = t_1 t_2 \dots t_n}$, которую необходимо встроить в изображение. Каждый элемент последовательности может быть представлен в двоичном виде $\boldsymbol{t_i = t_1^* t_2^* \dots t_7^*}$, где $\boldsymbol{t_j^* \in \{0, 1\}}$. Заменим нулевые элементы последовательности значением **{-1}**.

Представим выбранное черно-белое изображение в виде матрицы $\boldsymbol{A[m \times n]}$, значения которой соответствуют оттенку серого.

$$\boldsymbol{A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, a_{ij} \in \{0 \dots 255\}}$$

Разделим матрицу $\boldsymbol{A}$ на равные по размеру блоки $\boldsymbol{B_i[k \times k]}$. Всего получим $\frac{m*n}{k^2}$ блоков. Каждый блок представим в виде произведения матриц $\boldsymbol{B_i = U_i D_i V_i^T}$ (**SVD**).

Будем встраивать сообщение в матрицу $\boldsymbol{U}$ по следующему правилу:

$$u_{ij}^* = |u_{ij}| * t_p^* \text{, где } i \in \{2 \dots (k - j + 1)\}$$
$$j \in \{2 \dots (k-1)\}$$
$$t_p^* \in \{-1, 1\}$$

Получим новую матрицу $\boldsymbol{U^*}$. Так как изначальная матрица $\boldsymbol{U}$ была ортогональной, необходимо сделать столбцы новой матрицы ортогональными друг другу. Для этого находим и заменяем последние, нетронутые элементы каждого столбца (начиная со второго) посредством решения соответствующих систем.

Кроме того, изменим сингулярные числа нашего разложения по следующему правилу:

$$\sigma_i = \sigma_2 - (i-1)\frac{\sigma_2 + \sigma_k}{k-2}, \qquad i \in \{2, \dots, k-1\}$$

Полученные положительные значения будут удовлетворять условиям порядка чисел в сингулярном разложении, а также будут равномерно распределены, что минимизирует ошибку при восстановлении матрицы $\boldsymbol{B_i}$ [3].

Путем перемножения матриц $\boldsymbol{U_i^* D_i V_i^T}$ получаем новый блок $\boldsymbol{B_i^*}$ с закодированным сообщением. Округляем значения блока до целых чисел и ограничиваем элементы интервалом допустимых значений $\{\mathbf{0} \dots \mathbf{255}\}$.

Из новых блоков получаем новое черно-белое изображение со скрытым сообщением.

16.gif / размер блока - 4 /

АЛГОРИТМ ДЕШИФРОВАНИЯ

Пусть дано черно-белое изображение и значение параметра размера блока, используемого при встраивании информации в изображение.

Представим изображение в виде матрицы значений оттенков серого цвета. Разделим матрицу на $\frac{m*n}{k^2}$ блоков, где m, n – размеры изображения, k – размер блока. Представим каждый блок в виде произведения трёх матриц $\boldsymbol{UDV^T}$ (**SVD**).

Будем записывать в результат биты по следующему правилу:

$$\begin{cases} 0\text{ , если } u_{ij} < 0 \\ 1\text{ , если } u_{ij} > 0 \end{cases} \quad \text{где} \quad \begin{matrix} i \in \{2 \dots (k-j+1)\} \\ j \in \{2 \dots (k-1)\} \end{matrix} \quad u_{ij} \in U$$

В результате получим строку вида: $\boldsymbol{t_1^* t_2^* \dots t_p^*}$, где $\boldsymbol{t_p^* \in \{0, 1\}}$.

Зная, что $\boldsymbol{t_i = t_1^* t_2^* \dots t_7^*}$, можно получить значение встроенного в изображение сообщения.

ПРОМЕЖУТОЧНЫЕ РЕЗУЛЬТАТЫ

Изображение	Размер блока	Норма разности	Кол-во бит	Ошибка
41.gif	4	2 869	49 152	0.0639
41.gif	8	6 731	86 016	0.0806
41.gif	16	12 514	107 520	0.1277
41.gif	32	20 096	119 040	0.2448
41.gif	64	29 494	124 992	0.3892
41.gif	128	42 328	128 016	0.4853
41.gif	256	49 930	129 540	0.4977

УЛУЧШЕНИЕ АЛГОРИТМА ВСТРАИВАНИЯ СООБЩЕНИЯ

ИЗБЫТОЧНОСТЬ СООБЩЕНИЯ

Существенным улучшением алгоритма является повторение кода сообщения.

Пусть дана конечная последовательность символов $\boldsymbol{T = t_1 t_2 \dots t_n}$. Каждый символ последовательности может быть представлен в виде $\boldsymbol{t_i = t_1^* t_2^* \dots t_7^*}$, где $\boldsymbol{t_j^* \in \{-1, 1\}}$. Повторим каждый $\boldsymbol{t_j^*}$ **m** раз, так что $\boldsymbol{T^* = (t_{1_1}^* t_{1_2}^* \dots t_{1_m}^*)(t_{2_1}^* t_{2_1}^* \dots t_{2_m}^*) \dots (t_{n_1}^* t_{n_2}^* \dots t_{n_m}^*)}$ – новая последовательность элементов, которую необходимо встроить в изображение.

При декодировании сообщения будем суммировать каждую группу содержащую **m** изначально равных битов. Знак суммы будем записывать в результат задачи декодирования.

Изображение	Избыточность	Размер блока	Кол-во бит	Ошибка
41.gif	1	8	49 152	0.0806
41.gif	3	8	28 672	0.0557
41.gif	5	8	17 203	0.0443
41.gif	7	8	12 288	0.0279
41.gif	9	8	9 557	0.0242
41.gif	15	8	5 734	0.0055
41.gif	21	8	4 096	0.0029

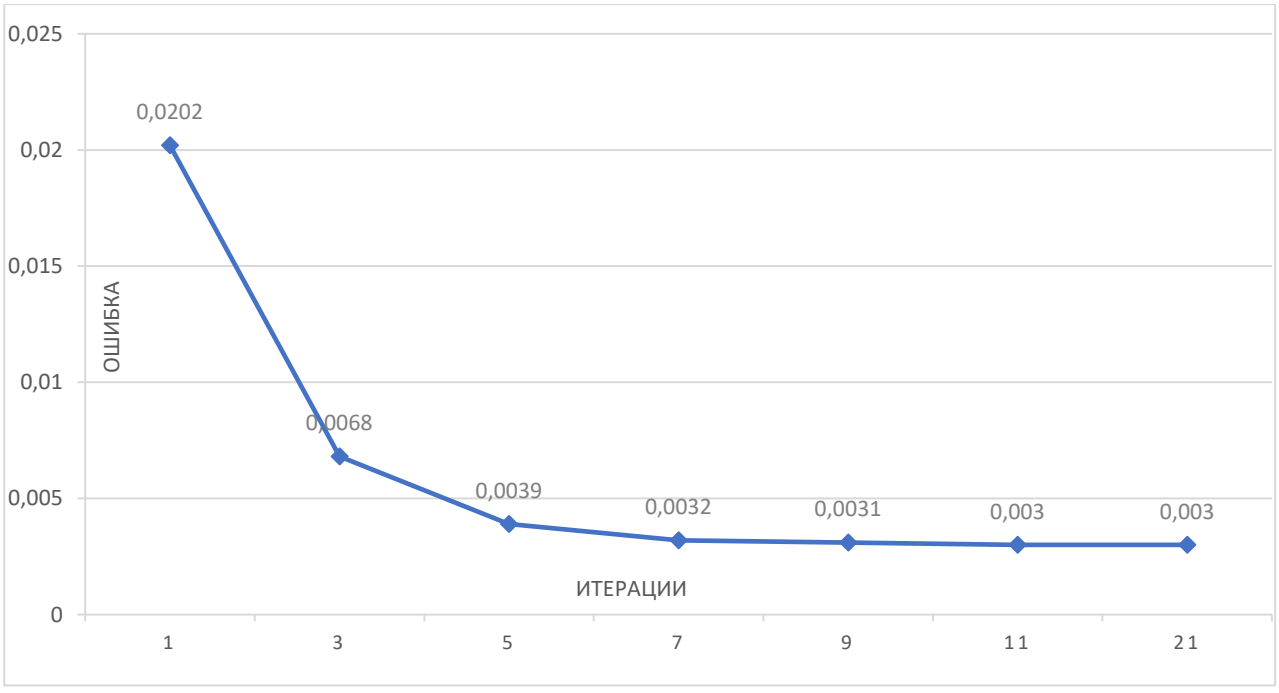
Как видно из таблицы, с увеличением количества повторов растет точность угадывания бита, но, в то же время, быстро уменьшается длина вмещаемого сообщения.

ДОБАВЛЕНИЕ ИТЕРАЦИЙ

Запустив алгоритм встраивания информации **m** раз, возможно сократить процент ошибки, не изменяя размеров исходного сообщения.

Таблица зависимости ошибки от количества итераций приведена ниже:

Изображение	Избыточность	Итерации	Размер блока	Кол-во бит	Ошибка
41.gif	3	1	4	16 384	0.0202
41.gif	3	3	4	16 384	0.0068
41.gif	3	5	4	16 384	0.0039
41.gif	3	7	4	16 384	0.0032
41.gif	3	9	4	16 384	0.0031
41.gif	3	11	4	16 384	0.0030
41.gif	3	21	4	16 384	0.0030



ПЕРВЫЕ N-НЕТРОНУТЫХ СТОЛБЦОВ

В стандартном варианте первый столбец матрицы **U** (разложения $\boldsymbol{UDV^T}$) всегда остается нетронутым. Увеличивая количество первых N-штук нетронутых столбцов, мы можем улучшить качество результирующего изображения. Тем самым, подобрав необходимое число столбцов, мы можем увеличивать размер блока, а соответственно и внедряемого сообщения, при этом незначительно жертвуя качеством и ошибкой.

Изображение	Итерации	Размер блока	N-столбец	Норма разности
41.gif	7	16	1	12 446
41.gif	7	16	2	11 137
41.gif	7	16	3	10 045
41.gif	7	16	4	9 216
41.gif	7	16	5	8 639
41.gif	7	16	6	8 231

41.gif / N = 1 /

41.gif / N = 3 /

41.gif / N = 6 /

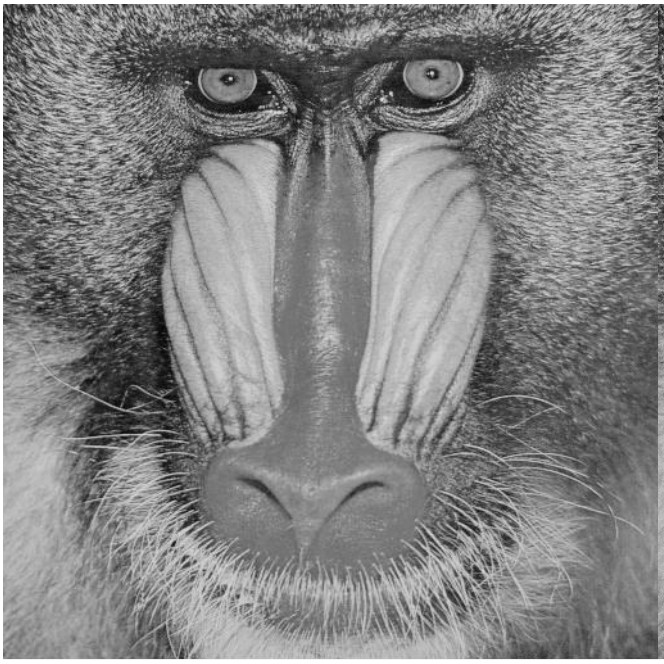
ПРОВЕРКА КАЧЕСТВА

Проверять качество написанного алгоритма будем путем сравнения битов исходного сообщения и битов результата работы алгоритма дешифрования.

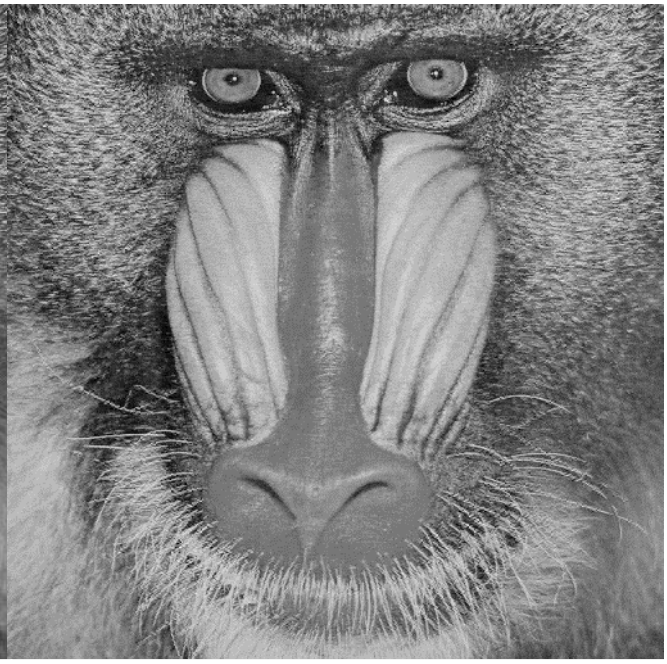
Кроме того, будем использовать генератор случайных чисел, равномерно распределенных на отрезке от 0 до 1. Полученные числа будем умножать на небольшую константу C и добавлять к каждому пикселю изображения для создания эффекта испорченности изображения.

Полученные результаты представлены ниже:

Изображение	Избыточность	Итерации	Размер блока	C	Ошибка
47.gif	3	7	4	0	0.0015
47.gif	3	7	4	0.1	0.0018
47.gif	3	7	4	0.5	0.0034
47.gif	3	7	4	1	0.0079
47.gif	3	7	4	1.5	0.0141
47.gif	3	7	4	3	0.0317
47.gif	3	7	4	5	0.0569
47.gif	3	7	4	10	0.1334



47.gif / Оригинал



47.gif / размер блока - 4, избыточность – 3, итерации – 7 /

43.gif / Оригинал

43.gif / размер блока - 4, избыточность – 3, итерации – 7 /

ЗАКЛЮЧЕНИЕ

Таким образом, мною была написана программа по встраиванию информации в черно-белое изображение и последующему её извлечению. Программа хорошо справляется с поставленной задачей в слабострессовых условиях и может быть применима в передаче сообщений по не портящим качество каналам.

ЛИТЕРАТУРА

1. Code [Электронный ресурс]. – URL: https://github.com/killerxzol/water-marking
2. Ian Goodfellow, Yoshua Bengio and Aaron Courville. Deep Learning. MIT Press, 2016
3. C. Bergman and J. Davidson, “Unitary Embedding for Data Hiding with the SVD”. Iowa State University, 2005
4. Dataset of Standart 512x512 Grayscale Test Images [Электронный ресурс]. – URL: https://ccia.ugr.es/cvg/CG/base.htm