|  |
| --- |
| УНИВЕРСИТЕТ ИТМО |
| Лабораторная работа №7 по дисциплине «Цифровая обработка сигналов» |
| Сокрытие данных в изображении-стегоконтейнере |
| Группа Р3402 |
| **Выполнила: Орлова Кристина Александровна** |
| **Преподаватель: Тропченко Андрей Александрович** |

|  |
| --- |
| *28.01.20* |

**Цель**

* Получение навыков сокрытия данных в цифровых полутоновых изображениях методами стеганографии.
* Получение навыков самостоятельного программирования алгоритмов обработки изображений в системе Matlab.

**Задание №1**

1. Используя соответствующие функции пакета Matlab, напишите программу, реализующую алгоритм сокрытия данных в младшем разрядном срезе изображения.
2. Проведите эксперименты по сокрытию данных и их извлечению для различных изображений и типов скрываемых данных (бинарных изображений, данных типа .txt, .dat).
3. Определите количественные характеристики, определяющие соответствие между пустым и заполненным стегоконтейнером.

Исходный код

function ret = uint32touint8 (val)

ret = repmat(val(:)', [4 1]);

ret = bitand(ret, repmat([0xff; 0xff00; 0xff0000; 0xff000000], 1, numel(val)));

ret = uint8(bitshift(ret, repmat(-[0; 8; 16; 24], 1, numel(val))));

end

function ret = uint8touint32 (val)

ret = uint32(reshape(val, 4, []));

ret = bitshift(ret, repmat([0; 8; 16; 24], 1, numel(val)/4));

ret = sum(ret);

end

function ret = uint8tological (val)

ret = repmat(val(:)', [8 1]);

ret = bitand(ret, repmat([0x1 0x2 0x4 0x8 0x10 0x20 0x40 0x80]', ...

[1 numel(val)]));

ret = (ret != 0)(:);

end

function ret = logicaltouint8 (val)

ret = reshape(bitand(val(:)', 1), 8, []);

ret = bitshift(ret, repmat([0 1 2 3 4 5 6 7]', [1 numel(val)/8]));

ret = sum(ret);

end

function hidden = hide\_last (img, data)

if (!isa(img, 'uint8'))

error('image must be of class uint8')

end

container = bitand(img, 0xFE);

if (isa(data, 'logical'))

if (numel(data) + 4 > numel(img))

error('data can not fit into image')

end

data = reshape(data, [], 1);

l = uint8tological(uint32touint8(uint32(numel(data))));

data = reshape(resize([l; data], [numel(container) 1]), size(container));

hidden = container + data;

else

if (isa(data, 'char'))

data = uint8(data);

end

if (!isa(data, 'uint8'))

error('data must be of class logical, char or uint8')

end

if (numel(data)\*8 > numel(img))

error('data can not fit into image')

end

l = uint8tological(uint32touint8(uint32(numel(data))));

secret = uint8tological(data);

secret = reshape(resize([l; secret], [numel(container) 1]), size(container));

hidden = container + secret;

end

end

function data = unhide\_last (img, clazz, shape = [])

if (!isa(img, 'uint8'))

error('image must be of class uint8')

end

data = logical(bitand(img(:), 1));

l = uint8touint32(logicaltouint8(data(1:32)));

data(1:32) = [];

switch (clazz)

case 'logical'

0;

case 'uint8'

data = logicaltouint8(data);

case 'char'

data = char(logicaltouint8(data));

otherwise

error('unsupported class requested')

end

data(l + 1:end) = [];

if (!isempty(shape))

data = reshape(data, shape);

end

end

img = imread('Lenna.gif');

secret1 = imresize(imread('cat.gif'), [128 128]) > 20;

secret2 = 'hidden text';

secret3 = uint8([1 2 3 9 8 7 6 4 5]);

container = bitand(img, 0xFE);

hidden1 = hide\_last(img, secret1);

hidden2 = hide\_last(img, secret2);

hidden3 = hide\_last(img, secret3);

figure 1

subplot(2,3,1), imshow(img), title('original')

subplot(2,3,2), imshow(container), title(['empty' char(10) num2str(corr2(img, container))])

subplot(2,3,3), imshow(hidden1), title(['filled' char(10) num2str(corr2(img, hidden1))])

subplot(2,3,4), hist(img(:), 256)

subplot(2,3,5), hist(container(:), 256)

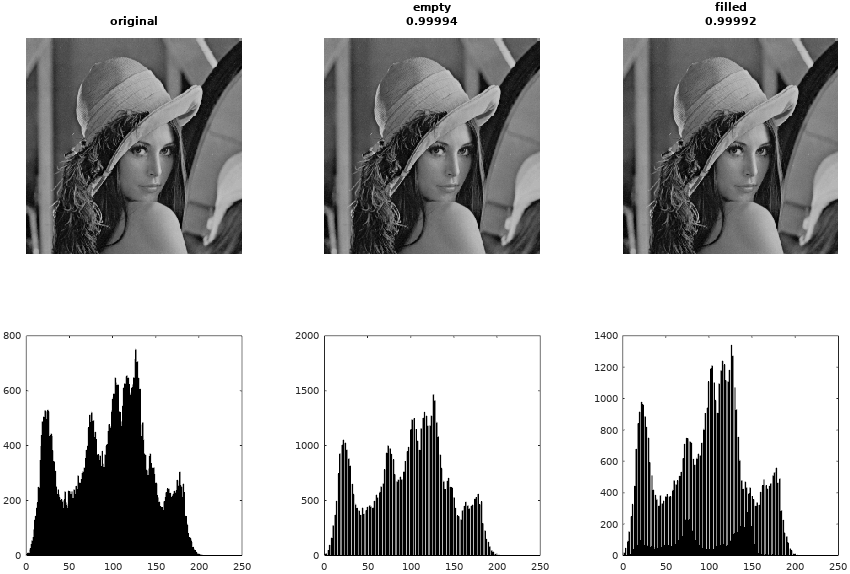
subplot(2,3,6), hist(hidden1(:), 256)

Результаты

На Рисунке 1 приведено бинарное изображение, скрываемое в стегоконтейнере. На Рисунке 2 приведены последовательно: исходное изображение, пустой контейнер, заполненный контейнер и соответствующие гистограммы. Можно заметить, что пустой контейнер имеет прерывистую гистограмму из-за наличия только четных байтов. После заполнения часть байтов становится нечётными, чем объясняется наличие сплошной области на гистограмме. Корреляция контейнера с исходным изображением высока.



*Рисунок 1 – Исходное изображение*



*Рисунок 2 – Сокрытие бинарного изображения в младшем разрядном срезе*

**Задание №2**

1. Самостоятельно реализуйте алгоритм сокрытия данных (текста) в псевдо-белых и псевдо-черных пикселах изображения и с использованием средств пакета Matlab напишите программу, реализующую данный метод.
2. Выполните п.п. 2 и 3 задания №1.
3. Сравните емкость стегоконтейнеров для обоих методов.
4. Самостоятельно провести подобные манипуляции с несколькими различными изображениями. Привести вид исходного стегоконтейнера и гистограмму яркости пикселов подлежащего сокрытию бинарного изображения, стегоконтейнера после помещения в него скрываемой информации (с гистограммой) и вид восстановленного бинарного изображения (для каждого из изображений). Проанализировать недостатки и достоинства методов и вариант модификации для помещения скрываемой текстовой (символьной) информации.

Исходный код

function ret = uint32touint8 (val) <…>

function ret = uint8touint32 (val) <…>

function ret = uint8tological (val) <…>

function ret = logicaltouint8 (val) <…>

function hidden = hide\_bw (img, data)

if (!isa(img, 'uint8'))

error('image must be of class uint8')

end

pixels\_w = find(img >= 0xf0);

pixels\_b = find(img <= 0x0f);

pixels = [pixels\_w; pixels\_b];

pixels\_n = numel(pixels);

img(pixels\_w) = 0xf0;

img(pixels\_b) = 0x00;

l = uint32(numel(data));

if (isa(data, 'logical'))

if (numel(data) + 32 > pixels\_n\*4)

error('data can not fit into image')

end

data = [uint8tological(uint32touint8(l)); data(:)];

if (rem(l, 4) != 0)

data = resize(data, (fix(l/4)+1)\*4, 1);

end

data = reshape(data, 4, []);

data = bitshift(uint8(data), repmat([0; 1; 2; 3], 1, numel(data)/4));

data = sum(data);

img(pixels) += resize(data(:), pixels\_n, 1);

hidden = img;

else

if (isa(data, 'char'))

data = uint8(data);

end

if (!isa(data, 'uint8'))

error('data must be of class logical, char or uint8')

end

if (numel(data)\*8 + 32 > pixels\_n\*4)

error('data can not fit into image')

end

data = [uint32touint8(l); data(:)]';

data = repmat(data, 2, 1);

data(1, :) = bitand(data(1, :), 0xf);

data(2, :) = bitshift(bitand(data(2, :), 0xf0), -4);

img(pixels) += resize(data(:), pixels\_n, 1);

hidden = img;

end

end

function data = unhide\_bw (img, clazz, shape = [])

if (!isa(img, 'uint8'))

error('image must be of class uint8')

end

pixels\_w = find(img >= 0xf0);

pixels\_b = find(img <= 0x0f);

pixels = [pixels\_w; pixels\_b];

data = bitand(img(pixels), 0xf);

l = data(1:8);

l = reshape(l, 2, []);

l(2, :) = bitshift(l(2, :), 4);

l = uint8touint32(sum(l)');

data(1:8) = [];

if (strcmp(clazz, 'logical'))

data(l/4+1:end) = [];

data = repmat(data(:)', 4, 1);

data(1, :) = bitand(data(1,:), 1);

data(2, :) = bitshift(bitand(data(2, :), 2), -1);

data(3, :) = bitshift(bitand(data(3, :), 4), -2);

data(4, :) = bitshift(bitand(data(4, :), 8), -3);

data(l+1:end) = [];

data = logical(data(:)');

elseif (strcmp(clazz, 'uint8') || strcmp(clazz, 'char'))

data(l\*2+1:end) = [];

data = reshape(data, 2, []);

data(2, :) = bitshift(data(2, :), 4);

data = sum(data);

if (strcmp(clazz, 'char'))

data = char(data);

end

else

error('unsupported class requested')

end

if (!isempty(shape))

data = reshape(data, shape);

end

уnd

img = imread('cat.gif');

secret1 = imresize(imread('Lenna.gif'), [350 350]) > 80;

secret2 = 'hidden text';

secret3 = uint8([1 2 3 9 8 7 6 4 5]);

pixels\_w = find(img >= 0xf0);

pixels\_b = find(img <= 0x0f);

pixels = [pixels\_w; pixels\_b];

disp('Number of pixels available for hiding:')

disp(numel(pixels))

container = img;

container(pixels\_w) = 0xf0;

container(pixels\_b) = 0;

hidden1 = hide\_bw(img, secret1);

hidden2 = hide\_bw(img, secret2);

hidden3 = hide\_bw(img, secret3);

unhidden1 = unhide\_bw(hidden1, 'logical', [350 350]);

figure 1

subplot(2,5,1), imshow(img), title('original')

subplot(2,5,2), imshow(container), title(['empty' char(10) num2str(corr2(img, container))])

subplot(2,5,3), imshow(secret1), title('secret')

subplot(2,5,4), imshow(hidden1), title(['filled' char(10) num2str(corr2(img, hidden1))])

subplot(2,5,5), imshow(unhidden1), title('unhidden')

subplot(2,5,6), hist(img(:), 256)

subplot(2,5,7), hist(container(:), 256)

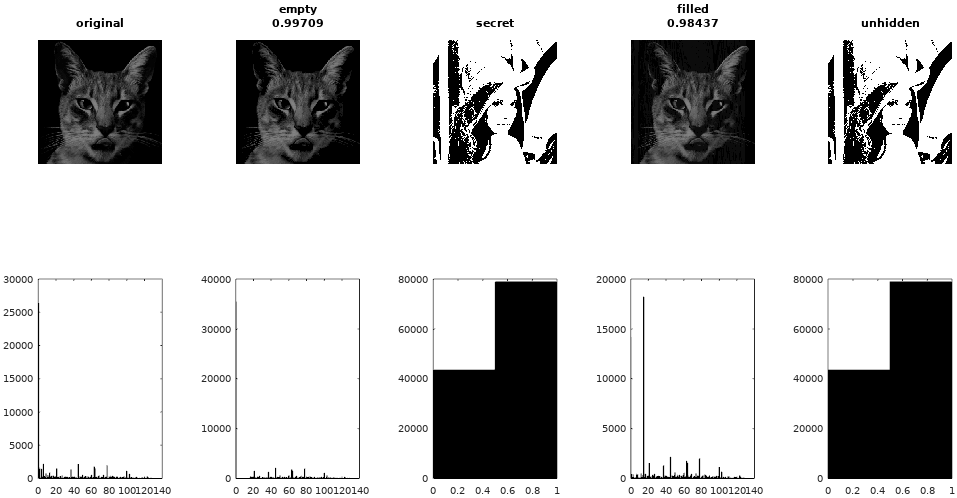
subplot(2,5,8), hist(secret1(:), 2)

subplot(2,5,9), hist(hidden1(:), 256)

subplot(2,5,10), hist(unhidden1(:), 2)

Результаты

Этот метод сильно зависит от изображения, используемого в качестве стегоконтейнера. В данном случае из-за большого количества черных пикселей в контейнере удалось поместить в него бинарное изображение большего размера. Однако корреляция с исходным изображением оказывается ниже, чем в предыдущем методе.



*Рисунок 3 – Сокрытие бинарного изображения в псевдо-белых и псевдо-черных пикселах*

**Выводы**

Метод сокрытия информации в младшем разрядном срезе можно применить для любого изображения, причём вместимость зависит только от размера.

Сокрытие в псевдо-белых и псевдо-черных пикселах сильно зависит от самого изображения, и может давать как намного меньшую вместимость, так и намного большую.