# Введение

Стеганография. История. Применение

# Вейвлет-преобразования

## Общая теория

## Вейвлет-преобразования в методах внедрения ЦВЗ в изображения.

### Что это такое

### Разновидности

### Возможные уязвимости

# Алгоритм внедрения цифрового водяного знака в изображение с помощью вейвлет-преобразований

Краткая сводка: алгоритм является неслепым, бла-бла-бла…

В реализованном мною алгоритме И.Р. Кима используется трехуровневое разложение с использованием вейвлет-преобразования Хаара. ЦВЗ встраивается в коэффициенты всех уровней разложения последовательно, начиная с третьего и заканчивая первым. Для встраивания выбираются коэффициенты, превышающие значение определенного порога.

Для встраивания используется аддитивный алгоритм

|  |  |
| --- | --- |
| , | (1) |

где – измененный коэффициент, – выбранный для внедрения коэффициент, – бит ЦВЗ, коэффициент масштабирования, который регулируется для каждого уровня разложения. Для получения изображения, содержащего ЦВЗ, используется обратное 2D-ДВП.

Для извлечения ЦВЗ необходимо исходное изображение. Применяется инверсия формулы внедрения, учитывая адаптивно-уровневый коэффициент масштабирования ,

|  |  |
| --- | --- |
|  | (2) |

Количество встраиваемой информации относительно невелико. В среднем оно составляет примерно 1/120 от размера исходного изображения, однако при расчете возможного количества встраиваемой информации следует учитывать не только размер изображения, но и насыщенность цвета определенных составляющих.

## Описание алгоритма И.Р. Ким и его программной реализации

### Встраивание ЦВЗ

Обобщенная блок-схема встраиваемых данных водяного знака с помощью 2D-ДВП изображена на рис. 1.



Рис. 1 Обобщенная блок-схема внедрения ЦВЗ

**Шаг 1.** Импортированиеисходного изображение и извлечение его синей составляющей, изменения в которой наименее заметны человеческому глазу.

**Шаг 2.** Импортирование ЦВЗ и преобразование его в одномерный массив бит.

**Шаг 3.** Разложение синей составляющей исходного изображения с помощью вейвлет-преобразования Хаара на 3 уровня и формирование коэффициентов аппроксимации и детализации.

**Шаг 4.** Вычисление порогов, в зависимости от которых будут выбраны пригодные для встраивания коэффициенты.

Пороги вычисляются отдельно для каждого уровня. На каждом уровне разложения находится абсолютный максимум для каждого набора коэффициентов, аппроксимации и детализации – горизонтальных , вертикальных и диагональных . Затем из них берется максимум . Порог вычисляется по следующей формуле:

|  |  |
| --- | --- |
|  | (5) |

**Шаг 5.** Процесс встраивания ЦВЗ. Начиная с 3-го уровня разложения, перебираются коэффициенты аппроксимации и детализации. Каждый коэффициент сравнивается со значением порога, соответствующего данному уровню. Коэффициенты, превышающие значение порога, изменяются в соответствии с формулой 1. Аналогичные операции производятся на 2 и на 1 уровне. В процессе выполнения этого шага функция периодически проверяет, сколько символов ЦВЗ осталось встроить. Если ЦВЗ встроено полностью, процесс останавливается. Соответственно, ЦВЗ не всегда встраивается во все коэффициенты.

**Шаг 6.** Восстановление синей составляющей изображения, в которое был встроен ЦВЗ, из измененных коэффициентов с помощью обратного вейвлет-преобразования. Изображение визуально ничем не должно отличаться от исходного.

### Извлечение ЦВЗ

**Шаг 1.** Выполнение шага 1 и шага 3 алгоритма встраивания к изображению со встроенным ЦВЗ. В результате чего получаем коэффициенты аппроксимации и детализации 3 уровней разложения Хаара.

**Шаг 2.** Выполнение шага 1 и шага 3 алгоритма встраивания к исходному изображению. Это действие необходимо в связи с тем, что алгоритм И.Р. Ким является неслепым, а это значит, что для извлечения ЦВЗ необходимо исходное изображение.

**Шаг 3.** Вычисление порогов для коэффициентов. Вычисление производится с помощью исходного изображения аналогично шагу 4 предыдущего алгоритма.

**Шаг 4.** Извлечение ЦВЗ. Для извлечения ЦВЗ производится перебор коэффициентов синей составляющей исходного изображения, начиная с 3 уровня. Если коэффициент больше порога , то находим соответствующий коэффициент в изображении со встроенным ЦВЗ и извлекаем бит ЦВЗ в соответствии с формулой 2.

**Шаг 5.** Перевод одномерного извлеченного массива бит ЦВЗ к двумерному виду и сохранение его в формате изображения. Для выполнения этого шага необходимо указать размеры исходного ЦВЗ.

## Программная реализация алгоритма И.Р. Ким

Программа, реализующая алгоритм И.Р., описанный в пункте 2.1, написана на языке C# на платформе .NET Framework. Реализовано 7 классов, код которых приведен в приложениях А-Д. Далее следует описание основной функциональности каждого класса.

Для реализации хранения и обработки изображений было написано два калсса. Вспомогательный класс DoublePixel (Приложение А) дает возможность оперировать значениями пикселей. Он содержит функции для быстрого доступа к конкретному пикселю, замены его значения и приведения пикселя к формату RGB.

Класс DoubleImage (Приложение Б) был реализован с целью замены стандартного класса для обработки изображений Bitmap дает более быстрый доступ к отдельным пикселям в отличие от стандартного класса Bitmap. Класс DoubleImage позволяет извлекать отдельные цветовые компоненты, а также обновлять их. Этот класс содержит функцию для преобразования изображения в стандартный формат Bitmap для последующего сохранения.

**Шаг 2.** Импортирование ЦВЗ и преобразование его в одномерный массив бит.

Для данного шага потребовалось реализовать класс Watermark (Приложение В). Он позволяет преобразовывать изображение в двумерный массив бит и обратно.

**Шаг 3.** Разложение синей составляющей исходного изображения с помощью вейвлет-преобразования Хаара на 3 уровня и формирование коэффициентов аппроксимации и детализации.

Для данного этапа были реализованы два класса, HaarTransfrom (Приложение Г) и Wavelet (Приложение Д).

Класс HaarTransfrom содержит функцию Transform, принимающую на вход матрицу вещественных чисел, а также текущую ширину и высоту для преобразования с целью использования данной функции рекурсивно на одной матрице. Функция Transform сначала осуществляет преобразования по строкам матрицы, затем аналогично по столбцам. Для этого создается пустая матрица такого же размера для сохранения промежуточных результатов. В ходе преобразований по строкам для каждой строки *k* берется пара соседних чисел пробегает от 0 до *width/*2, где *width* – это ширина матрицы, с шагом 2, и преобразуется следующим образом:

|  |  |
| --- | --- |
|  | (3) |
|  | (4) |

Затем полученные значения аппроксимации *approx* и детализации *detal* сохраняются в промежуточную матрицу в строку *k* по индексам *i* и *width/2*, соответственно.

Преобразования по столбцам производятся на промежуточной матрице, полученной в результате преобразований по строкам. Преобразования производятся аналогично описанных выше преобразованиям по строкам. Результат сохраняется в конечную матрицу, которую возвращает функция Transform.

Класс HaarTransfrom также содержит функцию Untransform, которая производит обратные преобразования для восстановления исходной матрицы.

Класс Wavelet содержит функцию Transform, которая принимает на вход матрицу вещественных чисел и уровень разложения. После проверки на возможность разложения до указанного уровня запускается функция Transform класса HaarTransfrom. Размеры изменяемой части матрицы зависят от уровня разложения. Для первого уровня разложения требуется вся матрица, для последующих уровней размеры изменяемой части определяются следующим образом:

*variableWidth = width / i,*

*variableHeight = height / i,*

где *width и height –* исходные размеры изображения, а *i –* текущий уровень разложения.

Класс Wavelet содержит функцию Untransform, которая производит обратное вейвлет-преобразование.

Также класс Wavelet содержит функцию GetCoefficient, которая возвращает матрицу коэффициентов (аппроксимации и детализации – горизонтальных, вертикальных и диагональных) в зависимости от требуемого уровня разложения.

**Шаг 4.** Вычисление порогов, в зависимости от которых будут выбраны пригодные для встраивания коэффициенты.

## Анализ алгоритма

## Оптимизация алгоритма

# Заключение

# Список литературы