

Санкт-Петербургский Государственный Университет  
Фундаментальная Информатика и Информационные Технологии

Выменец Максим Юрьевич

Применение вейвлет-преобразований к анализу и  
классификации изображений

Бакалаврская работа

Научный руководитель: Соловьёв Игорь Павлович,  
доцент кафедры информатики  
математико-механического факультета СПбГУ

Рецензент: Михайлова Елена Георгиевна,  
доцент кафедры информационно-аналитических систем  
математико-механического факультета СПбГУ

Санкт-Петербург  
2016

SAINT-PETERSBURG STATE UNIVERSITY

Fundamental Computer Science and Information Technology

Maxim Vymenets

Application of wavelet transforms to image analysis and  
classification

Bachelor's Thesis

Scientific supervisor: Igor Soloviev,  
Associate Professor, Computer Science Department,  
St. Petersburg State University

Reviewer: Elena Mikhailova,  
Associate Professor, Sub-Department of Analytical Information Systems,  
St. Petersburg State University

Saint-Petersburg  
2016

## Оглавление

1. Введение.....	4
2. Постановка задачи.....	5
2.1 Цель работы.....	5
2.2 Краткий обзор аналогов и обоснование актуальности.....	6
3. Дискретное вейвлет-преобразование.....	7
4. Описание приложения Wavelet Transformer.....	9
4.1 Общая информация.....	9
4.2 Функциональность.....	10
4.3 Интерфейс.....	11
4.4 Код преобразования.....	14
4.5 Код визуализации.....	16
4.6 Образец результатов работы приложения.....	17
5. Описание форматов файлов.....	19
5.1 Формат файлов .wavelet.....	19
5.2 Формат файлов .wldi.....	20
6. Перспективы применения.....	21
7. Заключение.....	22
8. Список литературы.....	23

# 1. Введение

Вейвлеты являются сравнительно новым изобретением в современной математике, однако интерес к ним за последние десятилетия многократно вырос [5]. Вейвлет-преобразования сигналов, заключающиеся в свертке сигнала со специальной функцией, собственно, вейвлетом, являются важным инструментом во многих областях информационных технологий [4]. Они позволяют разложить сигнал на низко- и высокочастотную составляющие, что позволяет отдельно рассматривать глобальные изменения сигнала и его локальные особенности.

Вейвлет-преобразование изображений широко применяется в сжатии [1, 3], анализе изображений и распознавании образов [4]. Для более успешного их применения необходимо исследовать эффективность различных вариаций преобразования применительно к большому количеству изображений из предметной области. Данная работа заключается в создании инструмента — программного продукта, помогающего в проведении подобных исследований. Написанная программа позволяет легко получать результаты преобразования набора изображений при помощи различных вейвлетов.

## **2. Постановка задачи**

### **2.1 Цель работы**

Целью данной работы являлось создание программного обеспечения, предназначенного для применения вейвлет-преобразований к изображениям. К программному обеспечению были предъявлены следующие требования:

1. оно должно предоставлять возможность применения преобразований к многим изображениям сразу;
2. оно должно предоставлять возможность применения различных преобразований к одному и тому же изображению и набору изображений;
3. оно должно предоставлять возможность использования потенциально неограниченного количества вейвлет-преобразований;
4. оно должно предоставлять возможность вывода на экран как загруженных изображений, так и результатов преобразований;
5. оно должно обладать удобным интерфейсом.

## **2.2 Краткий обзор аналогов и обоснование актуальности**

Разумеется, существует множество реализаций вейвлет-преобразований. Однако каждая из найденных реализаций обладала принципиальными недостатками. Ниже перечислены наиболее часто встречающиеся:

- реализация не предоставляет возможности проведения двумерных преобразований;
- реализация ограничена заранее заданным разработчиком набором вейвлет-преобразований;
- реализация обладает лишь консольным интерфейсом;
- реализация является не самостоятельным приложением, а библиотекой к конкретному языку программирования, либо частью большого пакета программ, которые необходимы для его использования;
- реализация обладает настолько избыточной функциональностью, что использование её для поставленных целей представляется крайне неэффективным.

Стоит заметить, что все обнаруженные реализации обладали как минимум двумя из перечисленных недостатков. Таким образом, конечный результат работы является актуальным и предоставляющим новые возможности продуктом.

### 3. Дискретное вейвлет-преобразование

При проведении дискретного вейвлет-преобразования вейвлет используется для создания двух фильтров — низко- и высокочастотного. Фильтры представляют собой векторы коэффициентов, которые сдвигаются до различных мест исходного сигнала и скалярно умножаются на вектор его значений для получения соответственно низко- и высокочастотной его компоненты. Так как используются два различных (и притом ортогональных) фильтра, для отсутствия потерь информации достаточно вычислить скалярные произведения со сдвигом лишь в каждую вторую точку исходного сигнала.

Векторы коэффициентов фильтров связаны между собой, высокочастотный фильтр получается из низкочастотного при записи его коэффициентов в обратном порядке и изменения знака каждого чётного коэффициента. Таким образом, хранить оба фильтра необязательно, достаточно хранить, например, низкочастотный [7]. Именно его коэффициенты в дальнейшем именуется коэффициентами вейвлет-преобразования.

В итоге, если обозначить коэффициенты низкочастотного фильтра как вектор  $c$  длины  $2n$ , вейвлет преобразование сигнала будет заключаться в домножении соответствующего ему вектора-столбца на матрицу следующего вида:

$$\begin{pmatrix} c_1 & c_2 & c_3 & \dots & c_{2n-1} & c_{2n} & 0 & 0 & 0 & \dots \\ -c_{2n} & c_{n-1} & -c_{n-2} & \dots & -c_2 & c_1 & 0 & 0 & 0 & \dots \\ 0 & 0 & c_1 & c_2 & c_3 & \dots & c_{2n-1} & c_{2n} & 0 & \dots \\ 0 & 0 & -c_{2n} & c_{n-1} & -c_{n-2} & \dots & -c_2 & c_1 & 0 & \dots \\ \dots & \dots \end{pmatrix}$$

Элементы получившегося вектора с нечетным индексом будут составлять низко-, а с четным — высокочастотную компоненту сигнала. В простейшем случае вейвлета Хаара  $n=1$ ,  $c_1=c_2=1/\sqrt{2}$ , а матрица выглядит так:

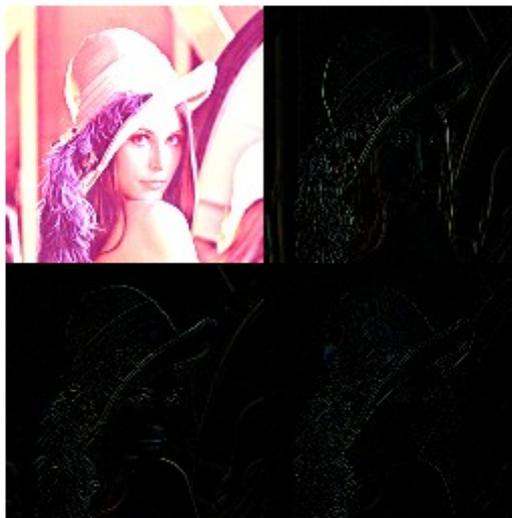
$$\begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 & 0 & \dots \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 & 0 & \dots \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 & \dots \\ 0 & 0 & -1/\sqrt{2} & 1/\sqrt{2} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

При работе с изображением вейвлет-преобразование применяется к каждой строке-горизонтالي или столбцу-вертикали по отдельности и фактически состоит в замене пары соседних пикселей (которые можно считать просто числами; если изображение цветное, то преобразование просто проводится отдельно для каждого цвета) на новую пару, получаемую как взвешенные суммы пикселей в некоторой окрестности той пары (включая её саму). Обычно для удобства восприятия и использования пиксели, составляющие

низкочастотную компоненту, отправляют в левую (если преобразование проводится по строкам) или верхнюю (если по столбцам) часть изображения, а составляющие высокочастотную компоненту — в правую или нижнюю часть соответственно [2, 3, 5]. Преобразование последовательно проводится по обеим координатам, в результате левая верхняя четверть результата преобразования (уже не являющегося в полной мере изображением, так как значения яркости цветов могут выходить за пределы таковых для исходного формата) является низкочастотной компонентой, правая верхняя — высокочастотной по горизонтали, левая нижняя — высокочастотной по вертикали, а правая нижняя — высокочастотной по диагонали. Это продемонстрировано на рисунке ниже.

Низкочастотная

Высокочастотная по горизонтали



Высокочастотная по вертикали

Высокочастотная по диагонали

## 4. Описание приложения Wavelet Transformer

### 4.1 Общая информация

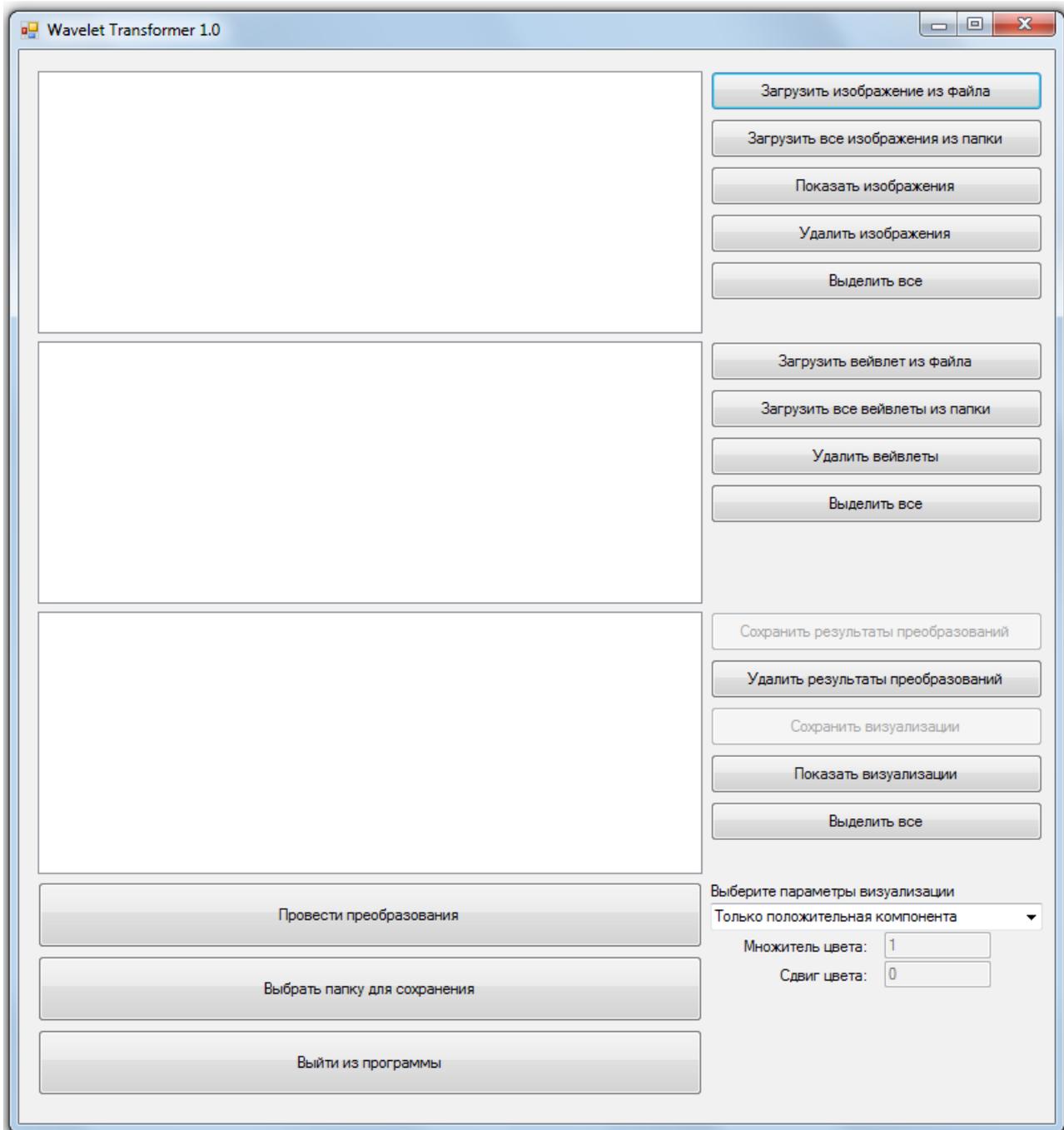
Приложение Wavelet Transformer было реализовано на языке программирования C# с применением программной платформы .NET Framework версии .NET Framework 4 Client Profile. Код приложения содержит более 1200 строк кода, 8 классов и 55 методов. Приложение позволяет быстро проводить вейвлет-преобразования большого числа файлов, менять параметры преобразования и визуализации его результатов, а также использовать коэффициенты преобразования, заданные пользователем в внешнем файле. Код приложения выложен в открытый доступ по адресу <https://drive.google.com/open?id=0B5f6vrnXKL3JQUxJZHFPR3YzZkk>

## 4.2 Функциональность

Приложение Wavelet Transformer позволяет:

- загружать большое количество изображений форматов BMP, JPG, PNG, GIF. (TODO: проверить поддержку других форматов);
- загружать большое количество наборов коэффициентов вейвлет-преобразований из файлов специального формата .wavelet;
- выбирать любое количество наборов коэффициентов вейвлетов из загруженных и изображений из загруженных и проводить каждое выбранное преобразование с каждым файлом по нажатию одной кнопки;
- выводить на экран исходные изображения, а также визуализации результатов преобразования;
- настраивать параметры визуализации результатов преобразования;
- сохранять результаты преобразований в специальном формате .wldi и визуализации результатов преобразований в формате BMP.

## 4.3 Интерфейс

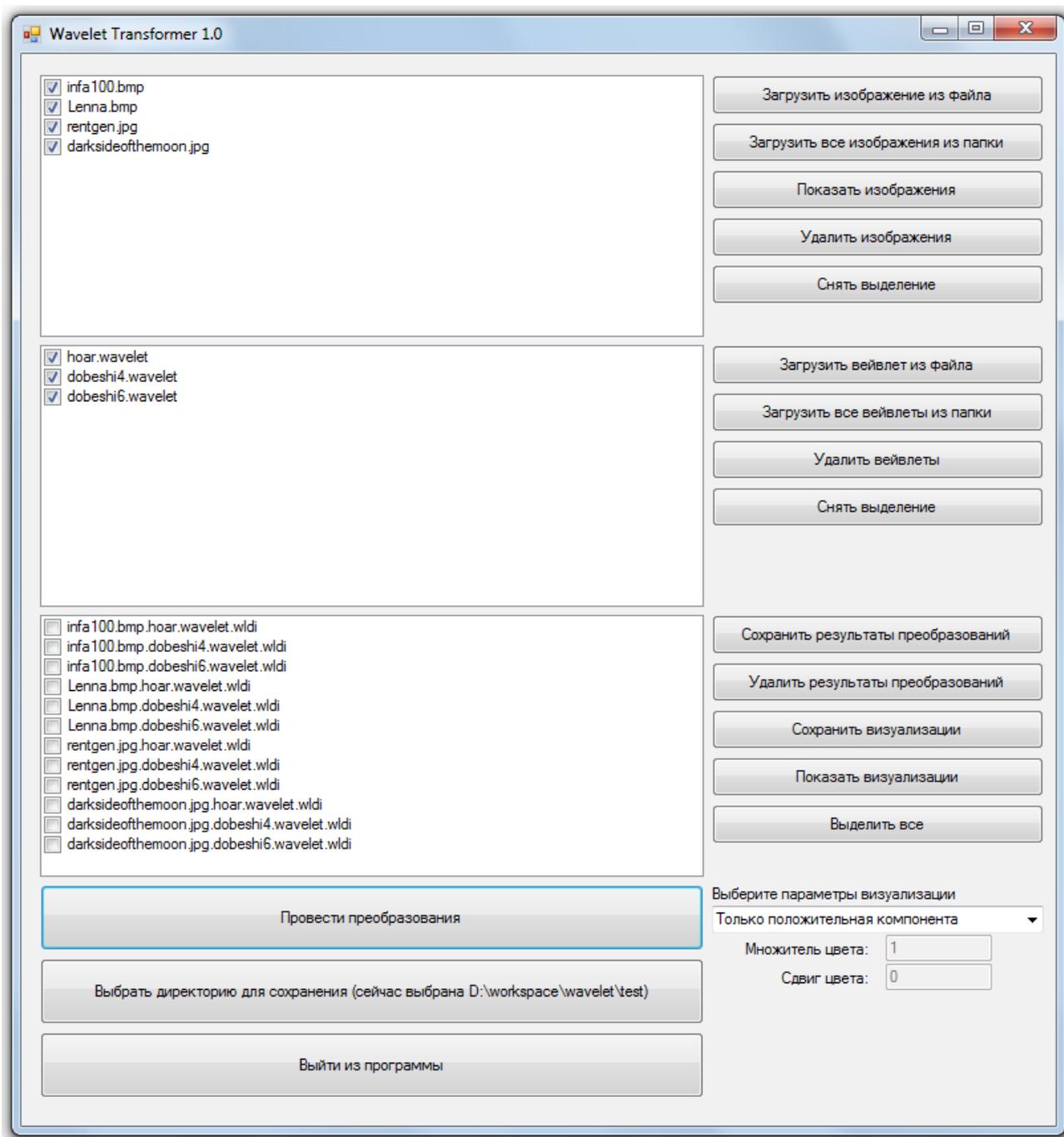


Основной частью интерфейса приложения являются три поля, содержащие списки загруженных изображений, загруженных наборов коэффициентов преобразования и полученных результатов преобразований соответственно.

Выделяя элементы списков (процесс выделения можно ускорить при помощи кнопки «Выделить все», расположенной справа от соответствующего поля), пользователь определяет, к каким элементам будут применены действия, инициированные кнопками.

Приложение предоставляет возможность загружать изображения и коэффициенты вейвлетов, как по отдельности, так и массово при помощи

кнопок «Загрузить все изображения из папки» и «Загрузить все вейвлеты из папки».



При нажатии кнопки «Провести преобразования» каждое загруженное и выделенное в списке изображение будет преобразовано при помощи каждого загруженного и выделенного набора коэффициентов вейвлета. Полученные результаты можно будет потом сохранить при помощи кнопок «Выбрать папку для сохранения» и «Сохранить результаты преобразований».

При помощи выпадающего меню «Выберите параметры визуализации» можно выбрать значения множителя и сдвига, применяемые к значениям яркости цветов пикселей результата преобразования перед выводом на экран или сохранения в файл BMP. Каждое значение последовательно домножается и

сдвигается на заданные числа, после чего значения, меньшие 0 и большие 255 приравниваются к 0 и 255 соответственно, что позволяет визуализировать нужный фрагмент информации. В меню представлено несколько кажущихся наиболее востребованными вариантов а также опция «Пользовательский вариант», позволяющая выбрать параметры произвольно, вписав их в соответствующие поля.

#### 4.4 Код преобразования

Программный код метода статического класса UseWavelet, реализующего вейвлет-преобразование изображения initialImage при помощи вектора коэффициентов низкочастотного фильтра вейвлет-преобразования c\_low:

```
public static DoubleImage waveletTransform(DoubleImage initialImage,
double[] c_low)
{
    int waveletOrder = c_low.Length;
    double[] c_high = new double[waveletOrder];
    int W = initialImage.Width;
    int H = initialImage.Height;
    DoubleImage middleImage = new DoubleImage(W, H);
    DoubleImage resultImage = new DoubleImage(W, H);
    for (int i = 0; i < waveletOrder; i++)
    {
        c_high[i] = c_low[waveletOrder - 1 - i];
        if (i % 2 == 0) c_high[i] *= -1;
    }

    for (int i = 0; i < W / 2; i++) //По строкам
        for (int j = 0; j < H; j++)
        {
            double R = 0, G = 0, B = 0;
            for (int d = 0; d < waveletOrder; d++)
            {
                R += c_low[d] * initialImage.GetPixel((i * 2 + d) % W, j).R;
                G += c_low[d] * initialImage.GetPixel((i * 2 + d) % W, j).G;
                B += c_low[d] * initialImage.GetPixel((i * 2 + d) % W, j).B;
            }
            middleImage.SetPixel(i, j, R, G, B); //Низкочастотная
            КОМПОНЕНТА
            R = 0;
            G = 0;
            B = 0;
            for (int d = 0; d < waveletOrder; d++)
```

```

    {
        R += c_high[d] * initialImage.GetPixel((i * 2 + d) % W, j).R;
        G += c_high[d] * initialImage.GetPixel((i * 2 + d) % W, j).G;
        B += c_high[d] * initialImage.GetPixel((i * 2 + d) % W, j).B;
    }
    middleImage.SetPixel(W / 2 + i, j, R, G, B); //Высококачественная
КОМПОНЕНТА
}
for (int i = 0; i < W; i++) //По столбцам
    for (int j = 0; j < H / 2; j++)
    {
        double R = 0, G = 0, B = 0;
        for (int d = 0; d < waveletOrder; d++)
        {
            R += c_low[d] * middleImage.GetPixel(i, (j * 2 + d) % H).R;
            G += c_low[d] * middleImage.GetPixel(i, (j * 2 + d) % H).G;
            B += c_low[d] * middleImage.GetPixel(i, (j * 2 + d) % H).B;
        }
        resultImage.SetPixel(i, j, R, G, B); //Низкокачественная
КОМПОНЕНТА
        R = 0;
        G = 0;
        B = 0;
        for (int d = 0; d < waveletOrder; d++)
        {
            R += c_high[d] * middleImage.GetPixel(i, (j * 2 + d) % H).R;
            G += c_high[d] * middleImage.GetPixel(i, (j * 2 + d) % H).G;
            B += c_high[d] * middleImage.GetPixel(i, (j * 2 + d) % H).B;
        }
        resultImage.SetPixel(i, H / 2 + j, R, G, B); //Высококачественная
КОМПОНЕНТА
    }
return resultImage;
}

```

#### 4.5 Код визуализации

Программный код метода класса DoubleImage, реализующего визуализацию результатов преобразования как обычного изображения:

```
public Bitmap ToBitmap(double colorMult, double colorShift) //Получить  
обычное изображение
```

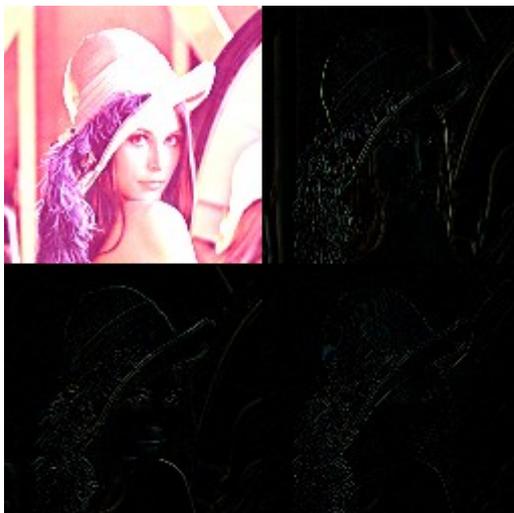
```
{  
    Bitmap BM = new Bitmap(W,H);  
    for (int i = 0; i < BM.Width; i++)  
        for (int j = 0; j < BM.Height; j++)  
        {  
            DoublePixel P = new DoublePixel(pixels[i, j]);  
            P.R = P.R * colorMult+ colorShift;  
            P.G = P.G * colorMult+ colorShift;  
            P.B = P.B * colorMult+ colorShift;  
            BM.SetPixel(i, j, P.ToNormalColor());  
        }  
    return BM;  
}
```

#### **4.6 Образец результатов работы приложения**

Ниже представлено исходное изображение и полученные при помощи приложения Wavelet Transformer визуализации результата вейвлет-преобразования Хаара, использующие различные параметры визуализации. Сами значения параметров colorMult (множитель цвета) и colorShift (сдвиг цвета) подписаны под каждым изображением. Эти значения параметров соответствуют вариантам, предоставляемым системой по умолчанию, однако пользователь может выбрать и другие.



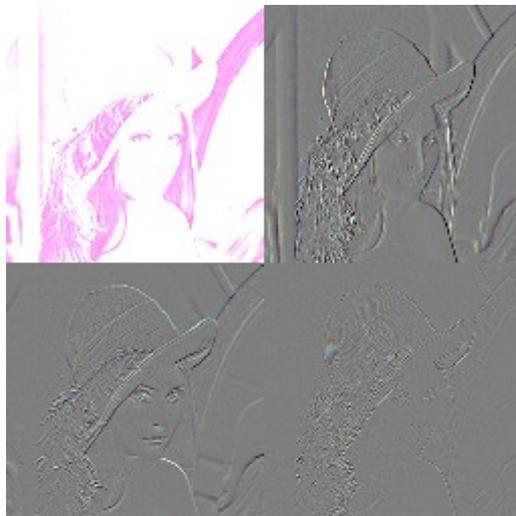
Исходное изображение



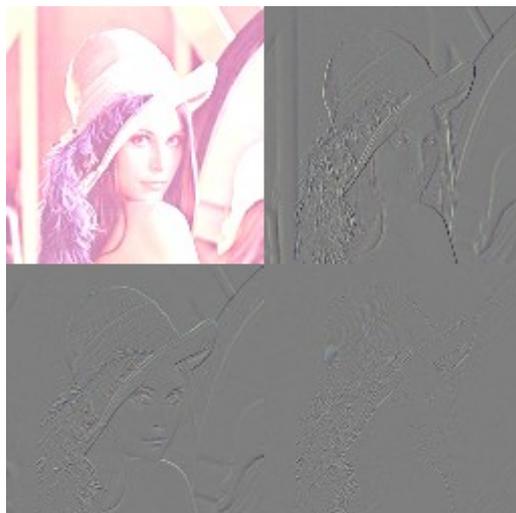
colorMult = 1, colorShift = 0



colorMult = -1, colorShift = 0



colorMult = 1, colorShift = 127.5



colorMult = 0.5, colorShift = 127.5

## 5. Описание форматов файлов

### 5.1 Формат файлов *.wavelet*

Формат файлов *.wavelet* используется программой для получения коэффициентов, используемых при вейвлет-преобразовании.

В первой строке файла записывается чётное целое число, обозначающее количество коэффициентов преобразования. Во второй и последующих строках записываются сами коэффициенты, по одному на строку, их количество должно соответствовать числу из первой строки. Коэффициенты являются вещественными числами; разделитель должен соответствовать используемому в операционной системе, в которой установлена программа.

Пример заполнения файла представлен ниже. Такой файл описывает вейвлет-преобразование Хаара.

```
2
0.70710678118
0.70710678118
```

## 5.2 Формат файлов *.wldi*

Формат файлов *.wldi* используется программой для сохранения результата вейвлет-преобразования. Необходимость использования отдельного формата вызвана тем, что если результат преобразования рассматривается просто как изображение, то оказывается, что значения яркости цветов в нём во-первых, практически всегда не являются целыми числами, а во-вторых могут выходить за границы отрезка от 0 до 255.

В первой строке файла через пробел записываются два целых числа, обозначающие размеры изображения по горизонтали и по вертикали соответственно. Во второй и последующих строках записываются значения яркости красного, синего и зеленого цвета в очередном пикселе. Данные по пикселям записываются последовательно по строкам. Количество пикселей должно равняться произведению чисел из первой строки.

Пример первых строк файла, полученного в результате преобразования, представлен ниже.

```
512 512
451.999999991629 273.999999994926 249.99999999537
451.999999991629 273.999999994926 249.99999999537
452.999999991611 276.99999999487 247.999999995407
450.999999991648 266.999999995055 239.999999995555
447.999999991703 258.999999995204 209.999999996111
446.999999991722 261.999999995148 215.999999996
```

## 6. Перспективы применения

Созданное приложение Wavelet Transformer может применяться для изучения вейвлетов, в частности оно позволяет быстро получать результаты различных вейвлет-преобразований одних и тех же изображений или различные визуализации одних и тех же результатов. Это полезно при изучении сравнительной эффективности применения вейвлет-преобразований и/или способов визуализации их результатов в той или иной области, такой, например, как анализ и классификация изображений.

Если говорить более конкретно про упомянутую область, в ней вейвлет-преобразования широко применяются при очистке зашумленных сигналов [8,9], устранении дефектов изображений и, совместно с нейронными сетями, в, собственно алгоритмах классификации [6].

## 7. Заключение

В ходе данной работы было создано приложение, позволяющее применять вейвлет-преобразования к изображениям. Данная реализация выгодно отличается от аналогов удобным интерфейсом, возможностью использования сразу большого количества изображений и преобразований и наличием настроек визуализации. В перспективе приложение можно развивать, добавляя новые возможности, например, возможность повторного преобразования низкочастотной компоненты изображения.

Приложение можно использовать для изучения вейвлет-преобразований и сравнения целесообразности применения различных преобразований в той или иной ситуации.

## 8. Список литературы

- [1] Kovacevic, J. *Fourier and Wavelet Signal Processing* / J. Kovacevic, V. K. Goyal, M. Vetterli. — Cambridge University Press, 2014. — 738 с.
- [2] Starck, J.-L. *Image and Data Analysis: the Multiscale Approach* / J.-L. Starck, F. Murtagh, A. Bijaoui. — Cambridge University Press, 1998. — 307 с.
- [3] Steinbuch, M. *Wavelet Theory and Applications A literature study* / M. Steinbuch, M.J.G. van de Molengraft — Eindhoven University of Technology, 2005. — 39 с.
- [4] Нагорнов, О.В. *Вейвлет-анализ в примерах: учебное пособие* / О.В. Нагорнов [и др.] — М.: НИЯУ МИФИ, 2010. — 120 с.
- [5] Уэлстид, С. *Фракталы и вейвлеты для сжатия изображений в действии: учебное пособие* / С. Уэлстид — М.: Издательство Триумф, 2003 — 320 с.
- [6] Lofti, M. *Combining wavelet transforms and neural networks for image classification* / M. Lofti [и др.] — 41st Southeastern Symp. on System Theory. — Tennessee, USA, 2009. — с. 44—48.
- [7] Mallat, S. G. *A Wavelet Tour of Signal Processing* / S. G. Mallat — Academic Press, 1999. — с. 637.
- [8] Московский, С.Б. *Очистка сигнала от шумов с использованием вейвлет-преобразования* / С.Б. Московский, Сергеев А.Н., Лалина Н.А. — *Universum: Технические науки : электрон. научн. журн.*, №2(15), 2015. — URL: <http://7universum.com/ru/tech/archive/item/1958>
- [9] Обидин, М.В. *Очистка сигнала от шумов с использованием вейвлет-преобразования и фильтра Калмана* / М.В. Обидин, А.П. Серебровский — *Информационные процессы*, Том 13, № 3, 2013. — с. 198—205.