

Министерство образования Республики Беларусь

Учреждение образования

“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №1

по курсу “Обработка изображений в интеллектуальных системах”

Выполнил:
студент гр. 221701

Робилко Т. М.

Проверил:

Сальников Д. А.

Минск
2024

Задание:

- 1) Изучить алгоритм быстрого преобразования Фурье
- 2) Выполнить программную реализацию алгоритма БПФ.
- 3) На вход подать функцию $\sin(x)$ или $\cos(x)$ для N частоты и показать правильность работы преобразования.

Ход работы:

- 1) Описание алгоритма:
 - Преобразовать функцию в дискретное множество значений.
 - Рекурсивно разделить массив данных на четные и нечетные элементы по индексу.
 - Выполнить операции сложения и вычитания с использованием комплексного коэффициента W для получения новых подмассивов.
 - Рекурсивно произвести те же самые вычисления для каждого подмассива.
- 2) Средства разработки:
 - Язык программирования C#.
 - Сторонняя библиотека Complex для представления комплексных чисел..
- 3) Основные части кода:
 - Расчет дискретного множества значений функции на отрезке с заданной частотой дискретизации

```
public static ImmutableList<Complex> GetDiscreteValues(
    Func<double, double> func,
    double frequency,
    double min = 0,
    double max = Math.Tau)
{
    double step = (max - min) / frequency;

    List<Complex> data = [];

    for (double x = min; x < max; x += step)
        data.Add(func(x));

    return ImmutableList.CreateRange(data);
}
```

- Рекурсивное разбиение множества значений на подмножества и применение трансформаций с использованием коэф-та W

```

public static ImmutableList<Complex> FFT(ImmutableList<Complex> input)
{
    int samples = input.Count;
    if (samples == 1)
        return input;

    // Even array
    List<Complex> evenList = [];
    for (int i = 0; i < (samples / 2); i++)
    {
        evenList.Add(input[2 * i]);
    }
    evenList = [.. FFT([.. evenList])];

    // Odd array
    List<Complex> oddList = [];
    for (int i = 0; i < (samples / 2); i++)
    {
        oddList.Add(input[(2 * i) + 1]);
    }
    oddList = [.. FFT([.. oddList])];

    // Result
    var result = new Complex[samples];

    for (int i = 0; i < (samples / 2); i++)
    {
        double angle = -2.0 * i * Math.PI / samples;
        Complex w = new(Math.Cos(angle), Math.Sin(angle));
        Complex even = evenList[i];
        Complex odd = oddList[i];

        result[i] = even + (w * odd);
        result[i + (samples / 2)] = even - (w * odd);
    }
    return [.. result];
}

```

- Результат работы программы (входные и выходные множества значений для функции $\cos(x)$)

Source:	Processed:
<1; 0>	4.195962738671156E-15
<0.7071067811865476; 0>	2.2607160134655004E-15
<6.123233995736766E-17; 0>	8
<-0.7071067811865475; 0>	-1.0973953780200294E-16
<-1; 0>	-1.5314274795707802E-15
<-0.7071067811865477; 0>	-1.7096435511036433E-15
<-1.8369701987210297E-16; 0>	0
<0.7071067811865474; 0>	-4.4133292455985494E-16
<1; 0>	-1.1331077795295957E-15
<0.7071067811865477; 0>	-4.4133292455985494E-16
<3.061616997868383E-16; 0>	-8.881784197001252E-16
<-0.7071067811865467; 0>	-1.7096435511036429E-15
<-1; 0>	-1.5314274795707798E-15
<-0.7071067811865471; 0>	-1.0973953780200215E-16
<1.3477304596986769E-15; 0>	8
<0.707106781186549; 0>	2.2607160134655008E-15

Вывод:

В ходе выполнения лабораторной работы был изучен алгоритм быстрого преобразования Фурье, который позволяет сократить время перевода сигнала из одной системы координат в другую. Дискретное преобразование Фурье проигрывает быстрому преобразованию Фурье по временной оценке при показателях $O(N^2)$ и $O(N \log(N))$ для двух алгоритмов соответственно.