

### 1) Титульник

### 2) Задачи

### 3) Вейвлет-преобразования. Формулы

Вейвлет-преобразование (ВП) широко используется для анализа сигналов. Помимо этого, оно находит большое применение в области сжатия данных. ВП одномерного сигнала – это его представление в виде обобщенного ряда или интеграла Фурье по системе базисных функций.

$$\psi_{ab}(t) = \frac{1}{a} \psi\left(\frac{t-b}{a}\right), \quad (1)$$

сконструированных из материнского (исходного) вейвлета  $\psi(t)$ , обладающего определенными свойствами за счет операций сдвига во времени ( $b$ ) и изменения временного масштаба ( $a$ ).

Множитель  $1/a$  обеспечивает независимость нормы функций (1) от масштабирующего числа ( $a$ ). Для заданных значений параметров  $a$  и  $b$  функция  $\psi_{ab}(t)$  и есть вейвлет, порождаемый материнским вейвлетом  $\psi(t)$ .

### 4) Преобразование Хаара. Фото и пиксели

Рассмотрим фрагмент первой строки яркостей из известного изображения «Lenna». Разобьём все числа на пары и найдём полусуммы и полуразности значений в каждой из них. Полусумма — это среднее значение яркости пары пикселей. А полуразность несёт в себе информацию об отличиях между значениями в паре.

### 5) Преобразование Хаара. Графики точек

В самом деле, рассмотрим первые 2000 пар соседних пикселей и каждую пару представим на графике точкой. Все точки выстраиваются вдоль одной прямой линии. И так практически во всех реальных изображениях. Верхний левый и нижний правый углы изображения практически всегда пусты.

А теперь рассмотрим график, точками в котором будут полусуммы и полуразности. Видно, что полуразности находятся в гораздо более узком диапазоне значений. А это значит, что на них можно потратить меньше одного байта. Какое-никакое, а сжатие.

При аффинных преобразованиях может меняться площадь фигуры. Для этого достаточно умножить каждый элемент матрицы на корень из двух. На угол это не повлияет.

### 6) Преобразование Хаара. Отдельно полусуммы и разности

Полученные числа можно перегруппировать, разделив полусуммы и полуразности. Числа во второй половине последовательности как правило будут небольшими, потому что в реальных изображениях соседние пиксели редко отличаются друг от друга значительно.

- 7) **Название алгоритма**
- 8) **Синяя компонента**
- 9) **1 уровень разложения**
- 10) **2 уровень разложения**
- 11) **3 уровень разложения**
- 12) **Преобразование QR-кода**
- 13) **Вычисление порогов**
- 14) **Встраивание битов ЦВЗ в коэффициенты**
- 15) **Обратное преобразование Хаара**
- 16) **Восстановление исходной картинки (одинаковые)**
- 17) **Восстановление QR-кода (одинаковые)**
- 18) **Тесты. Синие квадраты**
- 19) **Тесты. Белые квадраты**
- 20) **Тесты. Черная кисть**
- 21) **Оптимизация 1. Встраивание в несколько**
- 22) **Оптимизация 2. Подсчет времени**
- 23) **Проверка на робастность**
- 24) **Оптимизация 3. Процент схожести декодированного QR**
- 25) **Спасибо за внимание**