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 ввиде обобщенного ряда или интеграла Фурье по системе базисных функций.

ψab(t)=1aψ(t−ba), (1)

сконструированных из материнского (исходного) вейвлета ψ(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 ψab(t) и есть вейвлет, порождаемый материнским вейвлетом ψ(t).

1. **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.

1. **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.

1. **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.

1. **Название алгоритма**
2. **Синяя компонента**
3. **1 уровень разложения**
4. **2 уровень разложения**
5. **3 уровень разложения**
6. **Преобразование QR-кода**
7. **Вычисление порогов**
8. **Встраивание битов ЦВЗ в коэффициенты**
9. **Обратное преобразование Хаара**
10. **Восстановление исходной картинки (одинаковые)**
11. **Восстановление QR-кода (одинаковые)**
12. **Тесты. Синие квадраты**
13. **Тесты. Белые квадраты**
14. **Тесты. Черная киса**
15. **Оптимизация 1. Встраивание в несколько**
16. **Оптимизация 2. Подсчет времени**
17. **Проверка на робастность**
18. **Оптимизация 3. Процент схожести покоцанного QR**
19. **Спасибо за внимание**