Усеченное блочное кодирование (BTC)

Название отражает тот факт, что изображение разбивается на небольшие прямоугольные куски одинакового размера, называемые блоками. Блоки обрабатываются независимо друг от друга.

Этот метод в отличие от большинства других подстраивает параметры квантования не под некоторую усредненную характеристику всего изображения, а под локальные особенности в пределах каждого блока.

Базовый алгоритм УБК устроен следующим образом. Изображение, представленное матрицей $||A_{ij}||$, разбивается на небольшие прямоугольные блоки размером $m \times n$ элементов. Каждый такой блок обрабатывается независимо от других, поэтому опишем алгоритм обработки одного блока.

Обработка блока начинается с вычисления порога и двух уровней квантования, затем проводится квантование блока на два уровня, после чего следует упаковка проквантованного блока.

Обработка блока начинается с вычисления порога и двух уровней квантования, затем проводится квантование блока на два уровня, после чего следует упаковка проквантованного блока.

Для определения уровней квантования сначала вычисляются два первых выборочных момента — среднее значение C и средний квадрат E:

$$C = \frac{1}{m \times n} \sum_i \sum_j A_{ij}, \quad E = \frac{1}{m \times n} \sum_i \sum_j A_{ij}^2,$$

где суммируются элементы изображения в пределах блока, и дисперсия

$$\sigma^2 = E - c^2.$$

Пороговая величина квантователя d полагается равной среднему C. Верхний a и нижний b уровни квантования вычисляются из условия сохранения (приближенно!) первых двух выборочных моментов, что приводит к следующим расчетным формулам:

$$a = C - \sigma\sqrt{q/(p-q)}, \quad b = C + \sigma\sqrt{(p-q)/q},$$

где $p = m \times n$ — число элементов блока, q — число элементов блока, превышающих порог d.

Квантование проводится по обычному правилу:

$$S_{ij} = \begin{cases} a, & \text{если} \quad A_{ij} < d, \\ b, & \text{если} \quad A_{ij} \geq d, \end{cases}$$

где S_{ij} — элементы изображения после квантования.

Квантование проводится по обычному правилу:

$$S_{ij} = \begin{cases} a, & \text{если} \quad A_{ij} < d, \\ b, & \text{если} \quad A_{ij} \geq d, \end{cases}$$

где S_{ij} — элементы изображения после квантования.

После квантования получается блок, содержащий только уровни a и b. Нетрудно показать, что среднее значение и средний квадрат исходного и кодированного блоков совпадают. Практически для удобства последующей упаковки вместо a записывается нуль, вместо b — единица. Уровни a и b запоминаются отдельно.

Упаковка состоит в том, что блок, содержащий теперь только нули и единицы, интерпретируется как двоичное число, имеющее $m \times n$ разрядов. Это число переписывается в переменную (элемент массива) соответствующей длины.

Восстановление закодированного изображения также проводится поблочно и состоит в распаковке и обратной подстановке: вместо нулей записывается нижний уровень квантования, вместо единиц — верхний.

Так как исходное изображение имеет до 256 уровней и представлено 8-разрядным двоичным кодом, то для хранения блока размером 4×4 элемента в «натуральном» виде требуется 16 байт. Для хранения того же блока в закодированном виде достаточно 4 байт: 16 бит для бинарного образа блока, что составит 2 байта, и по одному байту для хранения уровней квантования a и b. Таким образом, достигается 4-кратный выигрыш в объеме требуемой памяти.

Традиционно в качестве меры сходства берут среднеквадратическое отклонение, которое (с точностью до коэффициента) имеет вид

$$J = \sum_i \sum_j (A_{ij} - S_{ij})^2,$$

где A — блок исходного изображения, S — блок, проквантованный

на два уровня.

При кодировании решается задача на определение условного экстремума. Необходимо так подобрать порог и два уровня квантования, чтобы минимизировать среднеквадратическое отклонение.

Элементы исходного блока ранжируются в порядке возрастания. Переобозначим их в виде последовательности Z_k, $k = 1, 2 \ldots p$ ($p = m \times n$). Для некоторого порога d подсчитывается число q элементов, превышающих этот порог. Тогда уровни квантования определятся из соотношений

$$a = \frac{1}{p-q} \sum_{k=1}^{p-q} Z_k, \quad b = \frac{1}{q} \sum_{k=p-q+1}^{p} Z_k.$$