

UNIVERSITATEA TEHNICĂ A MOLDOVEI
Facultatea Calculatoare, Informatică și Microelectronică
Departamentul Informatică și Ingineria Sistemelor

GRAFICA PE CALCULATOR

ТЕМА 5. СЖАТИЕ ИЗОБРАЖЕНИЙ

I.u., dr. NASTAS Andrei

- 5.1. Численное представление изображений
- 5.2. Представление изображения в несжатом формате
- 5.3. Методы и подходы к сжатию изображений
- 5.4. Преобразования, используемые при сжатии изображений
 - 5.4.1. Ортогональные преобразования
 - 5.4.2. Двумерные преобразования
 - 5.4.3. Преобразование Кархунена-Лоева (KLT)
 - 5.4.4. Преобразование Уолш-Адамар (WHT)
 - 5.4.5. Преобразование Хаара
 - 5.4.6. Дискретное косинусное преобразование (DCT)
 - 5.4.7. Дискретное синусоидальное преобразование
- 5.5. Сжатие JPEG (Объединенная группа экспертов в области фотографии)
 - 5.5.1. Базовый режим
 - 5.5.2. Режим кодирования с расширенными потерями
 - 5.5.3. Последовательное сжатие JPEG без потерь
 - 5.5.4. Иерархическое сжатие JPEG

5.4. Преобразования, используемые при сжатии изображений

Математическая концепция преобразования важна во многих областях, включая сжатие изображений. Изображение может быть сжато путем преобразования его пикселей (которые коррелированы или имеют статистическую взаимосвязь) в представление, где эта связь отсутствует. Сжатие получается, если новые значения в среднем ниже исходных. Сжатие с потерей информации может быть получено путем квантования преобразованных значений. Декодер получает преобразованные значения из сжатой последовательности и воспроизводит исходные данные (точные или приближительные) путем применения обратного преобразования. Преобразования, обсуждаемые ниже, являются ортогональными.

Термин декорреляция относится к тому, что преобразованные значения независимы друг от друга. В результате они могут быть закодированы независимо, что упрощает построение статистической модели. Изображение может быть сжато, если его представление имеет избыточность. Избыточность в изображениях вытекает из корреляции пикселей. Если изображение превращаете в представление, в котором пиксели декоррелированы, избыточность удаляется, и изображение становится полностью сжатым. Считается, что изображение сканируется в порядке раstra и группируются соседние пары пикселей. Поскольку пиксели коррелируют, два пикселя пары обычно имеют одинаковые значения.

5.4. Преобразования, используемые при сжатии изображений

Дальше пары пикселей рассматриваются как точки в двумерном пространстве, и их представляют графически. Известно, что все точки вида (x, y) расположены на биссектрисе, $y = x$, поэтому ожидается, что рассматриваемые точки будут сосредоточены вокруг неё. На рисунке 5.4 показаны результаты пиксельного представления любого изображения, в котором пиксель имеет значения от 0 до 255. Большинство точек сгруппированы вокруг этой линии, и только несколько точек расположены поодаль от нее. Теперь изображение преобразуется путем поворота всех точек на 45° по часовой стрелке вокруг оси координат, таким образом чтобы биссектриса теперь совпадала с осью x , как показано на рис. 5.4, *b*.

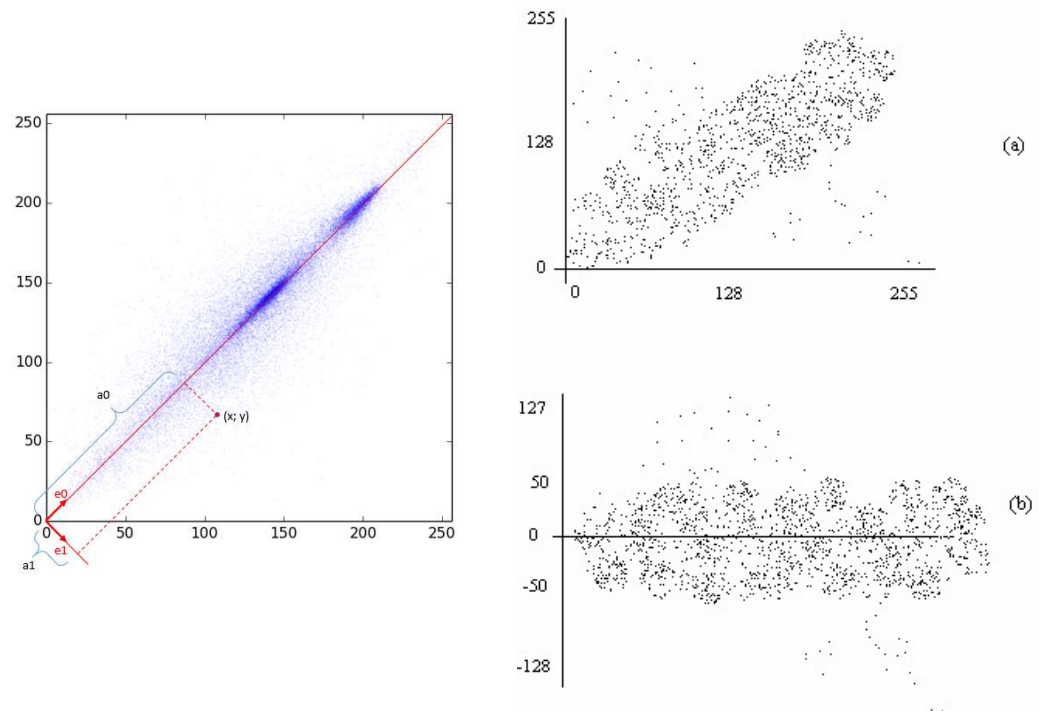


Рис. 5.4. Поворот группы точек

5.4. Преобразования, используемые при сжатии изображений

Это делается с помощью простого преобразования

$$(x^*, y^*) = (x, y) \begin{pmatrix} \cos 45^\circ & -\sin 45^\circ \\ \sin 45^\circ & \cos 45^\circ \end{pmatrix} = (x, y) \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = (x, y) \mathbf{R} \quad (5.2)$$

где матрица вращения \mathbf{R} является ортонормальной (то есть внутреннее произведение ряда с самим собой равно 1, внутреннее произведение разных строк равно 0, то же самое для столбцов). Обратное преобразование выглядит так

$$(x, y) = (x^*, y^*) \mathbf{R}^{-1} = (x^*, y^*) \mathbf{R}^T = (x^*, y^*) \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \quad (5.3)$$

Очевидно, что большинство точек в конечном итоге имеют координаты y нулевыми или почти нулевыми, в то время как координаты x не очень сильно меняются. Распределения координат x и y (т.е. нечетных и четных пикселей в изображении) перед вращением не сильно отличаются, тогда как после вращения распределение координат x остается почти таким же, но распределение координат y сосредоточено около нуля.

5.4. Преобразования, используемые при сжатии изображений

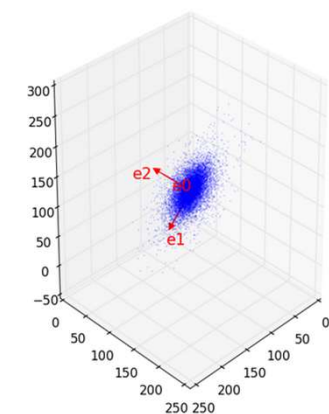
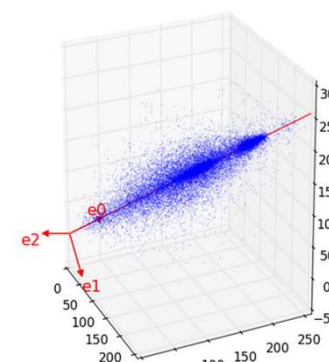
Поскольку координаты точек известны до и после вращения, легко измерить редукцию, которая появляется в корреляции точек, вычислив корреляционную функцию $\sum_i x_i y_i$ между точками. В этом случае сжатие изображения без потерь может быть осуществлено простым использованием пикселей, преобразованных в сжатую последовательность. Если поддерживается сжатие с потерей информации, то все пиксели можно квантизировать, получив еще меньшие числа. Также можно разделить все пиксели на нечетные (те, которые определяют координату x пары), за ними следуют четные пиксели. Эти две последовательности называются *векторами коэффициентов преобразования*. Вторая последовательность состоит из небольших чисел и может иметь, после преобразования, строки нулей, что может привести к еще лучшему сжатию.

Можно показать, что общая дисперсия пикселей не изменяется при вращении, так как матрица вращения является ортонормальной. Однако, поскольку дисперсия новых координат y невелика, большая часть дисперсии теперь сосредоточена к координате x . Дисперсию иногда называют *энергией распределения пикселей*, поэтому можно констатировать, что вращение концентрировало (или уплотнило) энергию в координате x и выполнило таким образом сжатие.

5.4. Преобразования, используемые при сжатии изображений

Это простое преобразование может быть легко распространено на любое количество размеров. Вместо выбора смежных пар пикселей можно выбрать триплеты. Каждый триплет становится точкой в трехмерном пространстве, и эти точки образуют область, сконцентрированную вокруг линии, которая образует углы 45° с тремя координатными осями. Когда эта линия поворачивается так, чтобы она совпала с осью x , координаты y и z преобразованных точек становятся малыми числами. Преобразование производится путем умножения каждой точки с ортонормальной матрицей вращения размером 3×3 . Преобразованные точки затем разделяются на три вектора коэффициентов, последние два из которых состоят из небольших чисел. Для максимального сжатия каждый вектор коэффициента должен быть количественно определен отдельно.

Эта идея может распространяться на более чем три измерения, с той лишь разницей, что нельзя просматривать пространства размером более трех. Некоторые методы сжатия, такие как JPEG, делят изображение на блоки по 8×8 пикселей каждый и поворачивают каждый блок дважды. Это двойное вращение производит набор из 64 преобразованных значений, первое из которых, называется «коэффициентом DC» или коэффициентом постоянного тока, является большим, а остальные 63, называемые «коэффициентами AC» или переменного тока, обычно малы. Таким образом, это преобразование концентрирует энергию в первом из 64 значений.



5.4.1. Ортогональные преобразования

Преобразования изображений, используемые на практике, должны быть быстрыми и предпочтительно простыми в реализации. Это приводит к мысли использования линейных преобразований. При таком преобразовании каждое преобразованное значение c_i представляет собой сумму пикселей d_j , где каждое значение умножается на множитель (или коэффициент преобразования) w_{ij} . Таким образом

$$c_i = \sum_j d_j w_{ij}, \text{ для } i, j = 1, 2, \dots, n. \quad (5.5)$$

Для $n = 4$ предыдущую формулу можно записать в форме матрицы

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix}$$

В целом

$$C = W \cdot D. \quad (5.5')$$

Каждая строка W называется *вектором основания*.

5.4.1. Ортогональные преобразования

Желательно определить значения w_{ij} так, чтобы первое преобразованное значение c_1 было большим, а остальные значения c_2, c_3, \dots быть маленьким. Из соотношения $c_i = \sum_j d_j w_{ij}$, видно что c_i будет большим, когда каждое значение w_{ij} усиливает вклад d_j в c_i . Это происходит, например, когда векторы w_{ij} и d_j имеют схожие значения и знаки. И наоборот, c_i будет маленьким, если все значения w_{ij} маленькие, а половина из них имеет противоположный знак d_j . Таким образом, когда получается большой c_i , это означает, что вектор w_{ij} аналогичен вектору данных d_j . С другой стороны малое значение c_i , означает, что w_{ij} и d_j различны. В заключение, векторы базы w_{ij} можно интерпретировать как средство извлечения характеристик из вектора данных.

5.4.1. Ортогональные преобразования

На практике значения должны быть независимыми от данных. В противном случае значения должны быть включены в сжатую последовательность для использования декодером. Это, в сочетании с тем фактом, что данные представляют собой пиксельные значения, которые не являются отрицательными, предлагает способ выбора векторов базы. Первый вектор, тот, который производит c_1 , должен состоять из положительных значений, возможно, даже идентичных. Это усилит неотрицательные значения пикселей. Каждый из остальных векторов должен иметь половину положительных элементов, другую половину – отрицательных. При умножении на неотрицательные значения данных, такие векторы, как правило, дают небольшое значение. Хорошим выбором базовых векторов было бы, если бы они сильно отличались друг от друга и, таким образом, могли извлекать больше характеристик. Это приводит к мысли, что векторы основания ортогональны. Если матрица преобразования W ортогональна, то само преобразование называется ортогональным. Другое наблюдение, которое помогает выбрать базовые векторы, заключается в том, что они должны иметь более и более высокие частоты, тем самым извлекая более высокие частотные характеристики из данных во время расчета преобразованных значений.

5.4.1. Ортогональные преобразования

Эти соображения удовлетворяются ортогональной матрицей:

$$W = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \quad (5.6)$$

Первый вектор основания (верхняя строка W) состоит только из значения 1, поэтому его частота равна нулю. Каждый из следующих векторов имеет два +1 и два из -1, поэтому они производят преобразования в небольшие значения, и их частоты (измеряемые как количество изменений сигнала вдоль базового вектора) становятся больше. Эта матрица похожа на преобразование Уолша-Адамара.

5.4.2. Двумерные преобразования

Двумерные преобразования будут рассмотрены в 7-ой теме.

5.4.3. Преобразование Кархунена-Лоева (KLT)

Преобразование Кархунена-Лоева (KLT) имеет наилучшую эффективность с точки зрения уплотнения энергии (или, другими словами, декорреляции пикселей), но оно имеет скорее теоретическое значение, чем практическое.

При таком преобразовании изображение, делится на k блоков по n пикселей каждый, где n обычно равно 64, но может иметь и другие значения, а k зависит от размера изображения. Рассмотрим векторные блоки $b^{(i)}$, где $i = 1, 2, \dots, k$. На их основе вычисляется среднее значение вектора $\bar{b} = (\sum_i b^{(i)})/k$. Определяется новый набор векторов $v^{(i)} = b^{(i)} - \bar{b}$, которые будут иметь среднее значение нулевым. Матрица преобразования KLT имеет размерность $n \times n$ и обозначается A . Результатом преобразования вектора $v^{(i)}$ является вектора $w^{(i)} = A \cdot v^{(i)}$. Среднее значение $w^{(i)}$ также равно нулю. Строится матрица V , столбцы которой являются векторами $v^{(i)}$, и другая матрица, столбцы которой являются векторами веса $w^{(i)}$.

$$V = (v^{(1)}, v^{(2)}, \dots, v^{(k)}), W = (w^{(1)}, w^{(2)}, \dots, w^{(k)}). \quad (5.7)$$

Матрицы V и W имеют n линий и k столбцов. Из определения $w^{(i)}$ следует, что $W = A \cdot V$.

Эти n векторы коэффициентов $c^{(j)}$ в преобразовании Кархунена-Лоева задаются отношением:

$$c^{(j)} = (w^{(1)}, w^{(2)}, \dots, w^{(k)}), j = 1, 2, \dots, n. \quad (5.8)$$

Таким образом, вектор $c^{(j)}$ состоит из элементов "j" всех векторов $w^{(i)}$ где $i = 1, 2, \dots, k$ ($c^{(j)}$ — координата j векторов $w^{(i)}$).

5.4.3. Преобразование Кархунена-Лоева (KLT)

Дальше рассмотрим элементы матрицы умножения $(W \cdot W^T)$ (это матрица размером $n \times n$). Любой элемент, в строке a и столбце b , этой матрицы является суммой произведений:

$$(W \cdot W^T)_{ab} = \sum_{i=1}^k w_a^{(i)} w_b^{(i)} = \sum_{i=1}^k c_i^{(a)} c_i^{(b)} = \mathbf{c}^{(a)} \mathbf{c}^{(b)}, \text{ pentru } a, b \in [1, n] \quad (5.9)$$

Поскольку среднее значение каждого вектора $w^{(i)}$ равно нулю, элемент $(W \cdot W^T)_{jj}$ по основной диагонали полученной матрицы, представляет собой вариацию или дисперсию элемента j (или координату j) вектора $w^{(i)}$.

Элементы вне диагонали являются ковариантами векторов $w^{(i)}$, т.е. элемент $(W \cdot W^T)_{ab}$ — ковариация координат a и b векторов $w^{(i)}$, которая равна внутреннему произведению $c^{(a)} c^{(b)}$. Основным недостатком преобразований, применяемых к изображению, является декорреляция координат векторов. Из теории вероятностей известно, что две координаты декоррелированы, если их ковариация равна нулю. Другим недостатком является уплотнение энергии, которое на самом деле тесно связано с первым. Имея это в виду, он направлен на поиск матрицы преобразования A , чтобы произведение $(W \cdot W^T)$ было диагональной матрицей.

Из определения матрицы W получается:

$$W \cdot W^T = (AV) \cdot (AV)^T = A(V \cdot V^T)A^T. \quad (5.10)$$

5.4.3. Преобразование Кархунена-Лоева (KLT)

Матрица $V \cdot V^T$ симметрична, а ее элементами являются ковариации координат векторов $v^{(i)}$:

$$(V \times V^T)_{ab} = \sum_{i=1}^k v_a^{(i)} v_b^{(i)}, \text{ pentru } a, b \in [1, n] \quad (5.11)$$

Поскольку матрица $(V \cdot V^T)$ симметрична, ее собственные векторы ортогональны. Эти векторы нормализуются, чтобы они были ортонормальными и выбираются в качестве линий матрицы A . Полученный результат:

$$W \times W^T = A(V \times V^T)A^T = \begin{pmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_n \end{pmatrix} \quad (5.12)$$

Такой выбор матрицы A приводит к диагональной $W \times W^T$ матрице, элементами которой на главной диагонали являются собственные значения матрицы $V \times V^T$. Матрица A — матрица преобразования Кархунена — Лоева, её линии являются векторами основания KLT, а энергии (дисперсии) преобразованных векторов — собственными значениями $\lambda_1, \lambda_2, \dots, \lambda_n$ матрицы $(V \times V^T)$. Базовые векторы преобразования KL не фиксированы, они зависят от данных, вычисляемых на основе пикселей исходного изображения. По этой причине они должны быть включены в последовательность сжатых данных, что снижает эффективность данного преобразования.

5.4.4. Преобразование Уолш-Адамар (WHT)

Это преобразование имеет низкую эффективность сжатия, и редко используется на практике. Однако она является быстрой, так как может быть рассчитана только сложениями, вычитаниями, а иногда и одним сдвигом вправо (чтобы эффективно разделить на степень 2).

Учитывая $N \times N$ блок пикселей P_{xy} (где N должен быть степенью 2, $N = 2^n$), соответствующее двумерное WHT и его обратный определяются уравнениями 5.16 и 5.17 соответственно:

$$\begin{aligned}
 H(u, v) &= \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} P_{xy} g(x, y, u, v) = \\
 &= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} P_{xy} (-1)^{\sum_{i=0}^{n-1} [b_i(x)p_i(x) + b_i(y)p_i(y)]}
 \end{aligned}
 \tag{5.16}$$

$$\begin{aligned}
 P_{xy} &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} H(u, v) h(x, y, u, v) = \\
 &= \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} H(u, v) (-1)^{\sum_{i=0}^{n-1} [b_i(x)p_i(x) + b_i(y)p_i(y)]}
 \end{aligned}
 \tag{5.17}$$

где $H(u, v)$ — результаты преобразования (т. е. коэффициенты WHT), величина $b_i(u)$ — бит i двоичного представления целого числа u , а $p_i(u)$ определяется согласно $b_j(u)$ уравнением 5.18.

$$\begin{aligned}
 p_0(u) &= b_{n-1}(u), \\
 p_1(u) &= b_{n-1}(u) + b_{n-2}(u), \\
 p_2(u) &= b_{n-2}(u) + b_{n-3}(u), \\
 &\dots\dots\dots \\
 p_{n-1}(u) &= b_1(u) + b_0(u).
 \end{aligned}
 \tag{5.18}$$

5.4.4. Преобразование Уолш-Адамар (WHT)

Например, считается $u = 6 = 110_2$. Биты ноль, единица и два "6" равны соответственно 0; 1 и 1, поэтому $b_0(6) = 0$, $b_1(6) = 1$, и $b_2(6) = 1$.

Размеры $g(x, y, u, v)$ и $h(x, y, u, v)$ называются ядрами (kernels) (или базовыми изображениями) WHT. Эти массивы идентичны. Их элементы составляют всего +1 и -1, и умножаются на множитель $1/N$. В результате преобразование WHT состоит из умножения каждого пикселя на изображении на +1 или -1, сложения и деления суммы на N . Поскольку $N = 2^n$, деление на 2^n можно сделать, переместив n позиций вправо.

Ядра WHT показаны в графическом виде для $N = 4$ на рисунке 5.5, где белый представляет +1, а черный -1 (коэффициент $1/N$ игнорируется).

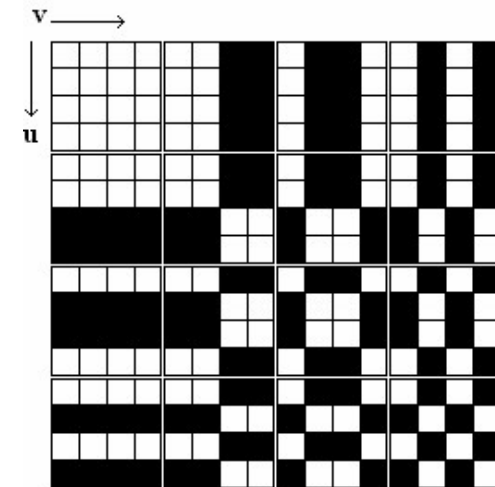


Рис. 5.5. Упорядоченное ядро WHT для $N = 4$

5.4.4. Преобразование Уолш-Адамар (WHT)

Строки и столбцы блоков на этом рисунке соответствуют значениям u и v , соответственно, от 0 до 3. Количество изменений знаков в течение строки или столбца массива называется *последовательностью строк или столбцов*.

Матрицы, соответствующие преобразованию DWHT, также могут быть получены рекурсивно как перестановки дискретных матриц Адамара, которые имеют особое значение в теории кодирования. Матрица Адамара размера N имеет свойство $HH^T=NI$, где I — единичная матрица.

Матрицы Адамара, размеры которых являются степенями двух, могут быть построены следующим

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} \quad (5.19)$$

с $H_1 = [1]$. Вот как они получаются

$$H_2 = \begin{bmatrix} H_1 & H_1 \\ H_1 & -H_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (5.20)$$

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (5.21)$$

$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (5.22)$$

5.4.4. Преобразование Уолш-Адамар (WHT)

Матрица H преобразования DWHT может быть получена из матрицы Адамара, умножив ее на коэффициент нормализации так, что $HH^T = I$, и переупорядочивая линии в порядке убывания частоты. Нормализация предполагает умножение матрицы на $\frac{1}{\sqrt{N}}$. Переупорядочивая H_8 , получается

$$H = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \quad (5.23)$$

Поскольку матрица без коэффициентов масштабирования состоит из ± 1 , операция преобразования состоит только из сложения и вычитания. По этой причине преобразование полезно в ситуациях, когда минимизация расчетного выигрыша очень важна.

5.4.5. Преобразование Хаара

Преобразование Хаара основано на функции Хаара $h_k(x)$, которая определяется для $x \in [0, 1]$ и для $k = 0, 1, \dots, N - 1$, где $N = 2^n$.

Любое целое число k может быть выражено как сумма, поэтому $k = 2^p + q - 1$, где $0 \leq p \leq n - 1$, $q = 0$ или 1 для $p = 0$, и $1 \leq q \leq 2^p$ для $p \neq 0$. Например, для $N = 4 = 2^2$ получить $0 = 2^0 + 0 - 1$, $1 = 2^0 + 1 - 1$, $2 = 2^1 + 1 - 1$, и $3 = 2^1 + 2 - 1$.

Основная функция преобразования Хаара определяется: $h_0(x) = h_{00}(x) = \frac{1}{\sqrt{N}}$, pentru $0 \leq x \leq 1$ (5.24)

$$\text{и } h_k(x) \stackrel{\text{def}}{=} h_{pq}(x) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2}, \frac{q-1}{2^p} \leq x < \frac{(q-1)/2}{2^p} \\ -2^{p/2}, \frac{(q-1)/2}{2^p} \leq x < \frac{q}{2^p} \\ 0, \text{ în rest} \end{cases} \quad (5.25)$$

Матрица преобразования Хаара, A_N порядка $N \times N$, может быть построена следующим образом: неким элементом матрицы (i, j) является базовая функция $h_i(j)$, где $i = 0, 1, \dots, N - 1$ и $j = 0/N, 1/N, \dots, (N-1)/N$ ($i = 1$ подразумевает $p = 0$ и $q = 1$). Например:

$$A_2 = \begin{pmatrix} h_0(0/2) & h_0(1/2) \\ h_1(0/2) & h_1(1/2) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (5.26)$$

5.4.6. Дискретное косинусное преобразование (DCT)

Дискретные косинусные и синусоидальные преобразования (DCT, DST) относятся к семейству тригонометрических преобразований с приложениями в сжатии/декомпрессии данных. Из них DCT гораздо чаще используется на практике из-за его свойства уплотнения энергии.

• Одномерный DCT

На практике используется двумерный DCT, но для удобства понимания он сначала считается одномерным DCT. Рассмотрим формы сигналов $w(f) = \cos(f\vartheta)$, для $0 \leq \vartheta \leq \pi$, с частотами $f = 0, 1, \dots, 7$ и

$$\theta = \frac{\pi}{16}, \frac{3\pi}{16}, \frac{5\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}, \frac{15\pi}{16} \quad (5.27)$$

Каждая форма сигнала $w(f)$ сэмплируется в восьми точках, чтобы сформировать вектор основания v_f . Восемь результирующих векторов v_f , $f = 0, 1, \dots, 7$ (всего 64 числа) показаны в таблице 5.1. Они являются основой для одномерного DCT. Обратите внимание на сходство между этой таблицей и W-матрицей в уравнении (5.5).

Таблица 5.1. Результирующие векторы v_f

θ	0,196	0,589	0,982	1,374	1,767	2,160	2,553	2,945
$\cos 0\theta$	1	1	1	1	1	1	1	1
$\cos 1\theta$	0,981	0,831	0,556	0,195	-0,195	-0,556	-0,831	-0,981
$\cos 2\theta$	0,924	0,383	-0,383	-0,924	-0,924	-0,383	0,383	0,924
$\cos 3\theta$	0,831	-0,195	-0,981	-0,556	0,556	0,981	0,195	-0,831
$\cos 4\theta$	0,707	-0,707	-0,707	0,707	0,707	-0,707	-0,707	0,707
$\cos 5\theta$	0,556	-0,981	0,195	0,831	-0,831	-0,195	0,981	-0,556
$\cos 6\theta$	0,383	-0,924	0,924	0,383	-0,383	0,924	-0,924	0,383
$\cos 7\theta$	0,195	-0,556	0,831	0,981	0,981	-0,831	0,556	-0,195

5.4.6. Дискретное косинусное преобразование (DCT)

Эти восемь векторов v_i являются ортонормальными (из-за частного выбора восьми точек выборки) и могут быть организованы в матрицу преобразования 8×8 . Поскольку эта матрица является ортонормальной, она является матрицей вращения, поэтому одномерный DCT можно интерпретировать как вращение в восьми измерениях.

Другая интерпретация одномерного DCT заключается в том, что можно рассматривать восемь ортонормальных векторов v_i как основу векторного пространства, а любой другой вектор p может быть выражен в этом пространстве как линейная комбинация этих v_i . Например, в качестве тестовых данных выбираются 8 коррелированных чисел, $p = (0,6; 0,5; 0,4; 0,5; 0,6; 0,5; 0,4; 0,55)$. Выражаем вектор p как линейную комбинацию восьми векторов основания, $p = \sum w_i v_i$. Решение этой системы из восьми уравнений дает восемь значений:

$$w_0 = 0,506, w_1 = 0,0143, w_2 = 0,0115, w_3 = 0,0439, \\ w_4 = 0,0795, w_5 = -0,0432, w_6 = 0,00478, w_7 = -0,0077.$$

5.4.6. Дискретное косинусное преобразование (DCT)

Значения w_0 мало чем отличается от элементов вектора p , но остальные семь значений намного меньше. Это указывает на то, как DCT (или любое другое ортогональное преобразование) производит сжатие. 8 значений будут просто представлять сжатые элементы вектора p . Количественно оценивая восемь значений, можно значительно увеличить сжатие, потеряв при этом лишь небольшой объем данных.

Рисунок 5.6 графически иллюстрирует эту линейную комбинацию. Каждый из восьми векторов v_i представлен в виде ряда из восьми маленьких серых прямоугольников, где значение $+1$ окрашено в белый цвет, а в число -1 — в черный. Каждый из восьми элементов вектора p выражается как взвешенная сумма восьмиуровневой шкалы серого.

Самый простой способ рассчитать одномерный DCT, на практике, это отношение

$$G_f = \frac{1}{2} C_f \sum_{t=0}^7 p_t \cos\left(\frac{(2t+1)f\pi}{16}\right) \quad (5.28)$$

где

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0, \\ 1, & f > 0, \end{cases} \quad \text{pentru } f = 0, 1, \dots, 7. \quad (5.29)$$

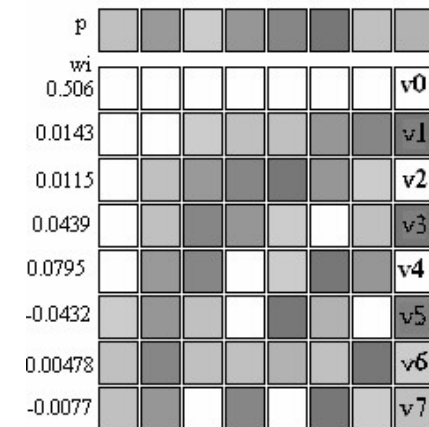


Рис. 5.6. Графическое представление одномерного DCT

5.4.6. Дискретное косинусное преобразование (DCT)

Начинается с набора из восьми значений данных p_t (пикселей, звуковых сэмплов или других данных) и получится набор из восьми коэффициентов DCT, G_f . Декодер получает коэффициенты DCT в наборы из восьми и применяет обратное преобразование DCT (IDCT) для восстановления исходных значений данных (также в группах по восемь человек). IDCT вычисляется с помощью формулы

$$p_t = \frac{1}{2} \sum_{j=0}^7 C_j G_j \cos\left(\frac{(2t+1)j\pi}{16}\right), \quad \text{pentru } t=0, 1, \dots, 7 \quad (5.30)$$

5.4.6. Дискретное косинусное преобразование (DCT)

- **Двумерный DCT**

Из опыта известно, что пиксели изображения коррелируют по двум измерениям, а не только по одному (пиксель коррелирует со своими соседями слева и справа, выше и ниже). Вот почему методы сжатия изображений используют двумерный DCT, заданный отношением

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \quad (5.31)$$

для $0 \leq i, j \leq n - 1$. Изображение разбивается на блоки $n \times n$ пикселей p_{xy} (обычно используется $n = 8$), а уравнение (5.31) используется для получения блока коэффициентов 8×8 DCT, G_{ij} , для каждого блока пикселей. Если сжатие происходит с потерей информации, коэффициенты количественно оцениваются. Декодер реконструирует блок значений данных (аппроксимированных или точных) путем вычисления IDCT.

$$p_{xy} = \frac{1}{\sqrt{2n}} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right) \quad (5.32)$$

где

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0, \\ 1, & f > 0, \end{cases} \quad \text{pentru } f = 0, 1, \dots, 7$$

5.4.6. Дискретное косинусное преобразование (DCT)

Двумерный DCT может быть интерпретирован двумя различными способами, как вращение (по сути, два отдельных вращения) и как основа n -мерного векторного пространства. В первой интерпретации он считается блоком $n \times n$ пикселей. Во-первых, каждая строка этого блока рассматривается как точка $(p_{x,0}; p_{x,1}; \dots; p_{x,n-1})$ в n -мерном пространстве, а точка поворачивается посредством преобразования, заданного суммой внутри

$$G1_{x,j} = C_j \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \quad (5.33)$$

уравнения (5.31). Это преобразование приводит к блоку $G1_{x,j}$ коэффициентов $n \times n$, где первый элемент каждой строки является доминирующим, а остальные элементы малы. Внешняя сумма уравнения (5.31) равна

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i \sum_{x=0}^{n-1} G1_{x,j} \cos\left(\frac{(2x+1)i\pi}{2n}\right) \quad (5.34)$$

5.4.6. Дискретное косинусное преобразование (DCT)

Здесь столбцы $G1_{x,j}$ считаются точками в n -мерном пространстве и поворачиваются. В результате получается высокий коэффициент в левом верхнем углу блока и $n^2 - 1$ маленькие коэффициенты в остальном. Эта интерпретация рассматривает двумерный DCT как два отдельных вращения, каждое из которых находится в n измерениях. Интересно отметить, что два вращения в n измерениях быстрее, чем одно в n^2 измерениях, так как во втором случае требуется матрица вращения размером $n^2 \times n^2$.

Вторая интерпретация (предполагающая $n = 8$) использует уравнение (5.31) для создания 64 блоков по 8×8 значений каждый. Затем 64 блока используются в качестве основы пространства 64-мерных векторов (они являются базовыми изображениями). Основные изображения, используемые в двумерном DCT, приведены на рисунке 5.7. Любой блок B в 8×8 пикселей может быть выражен как линейная комбинация основных изображений, а 64 веса этой линейной комбинации являются коэффициентами DCT блока B .

5.4.6. Дискретное косинусное преобразование (DCT)

Сжатие изображения с помощью DCT включает в себя выполнение следующих шагов:

- Разделение изображения на k блоков B_i , $i=1,2,\dots,k$, из $n \times n$ (обычно, 8×8) пикселей каждый.

- Применяет двумерный DCT к каждому B_i -блоку. Это преобразование выражает блок в виде линейной комбинации 64 основных изображений. В результате получается блок, называемый вектором $W^{(i)}$ из 64 значений $w_j^{(i)}$, где $j = 0, 1, \dots, 63$.

- Векторы k матрицы $W^{(i)}$ ($i = 1, 2, \dots, k$) разделены на 64 вектора коэффициентов $C^{(i)}$, где k элементы вектора $C^{(i)}$ равны $(w_j^{(1)}, w_j^{(2)}, \dots, w_j^{(k)})$. Первый вектор коэффициентов $C^{(0)}$ состоит из k коэффициентов DC.

- Каждый вектор коэффициента $C^{(i)}$ количественно определяется отдельно для получения квантованного вектора $Q^{(i)}$, который представляет сжатые данные.

Декодер считывает 64 вектора квантованных коэффициентов $Q^{(i)}$, использует их для построения k векторов значений $W^{(i)}$ и применяет IDCT к каждому значению вектора для реконструкции 64 пикселей блока B_i .

5.4.6. Дискретное синусное преобразование (DST)

Дискретное синусоидальное преобразование, DST, дополняет DCT. DCT обеспечивает производительность, близкую к оптимальному преобразованию K-L, с точки зрения сжатия, когда корреляция коэффициентов ρ высока, а DST обеспечивает производительность, близкую к оптимальному преобразованию K-L, когда ρ мала. Благодаря этому свойству она часто используется в качестве дополнительного преобразования DCT в кодировании изображения и звука.

Элементы преобразованного массива для DST размера $n \times n$ задаются формулой

$$[S]_{i,j} = \sqrt{\frac{2}{n}} \sin \frac{\pi(i+1)(j+1)}{2n}; \quad i, j = 0, 1, \dots, n-1 \quad (5.35)$$

Чтобы обосновать гораздо более частое использование DCT над DST, далее представлены различия между синусовыми и косинусными функциями и почему эти различия приводят к неэффективному дискретному синусоидальному преобразованию.

Синусоидальная функция является нечетной функцией, а косинусная функция — четной. Хотя единственным различием между двумя функциями является фаза (то есть косинусная функция является фазовой версией синуса), этой разницы достаточно, чтобы обратить вспять их четность. Для того чтобы понять разницу между DCT и DST, должен быть рассмотрен одномерный случай. Одномерный DCT, заданный уравнением (5.28), использует функцию $\cos((2t+1)f\pi/16)$ для $f = 0, 1, \dots, 7$. Для первого члена, где $f = 0$, эта функция становится $\cos(0)$, что равно 1. Этот член является коэффициентом DC, который дает среднее значение восьми значений данных, подвергающихся преобразованию.

5.4.6. Дискретное синусное преобразование (DST)

DST аналогично основан на функции $\cos((2t + 1) f\pi/16)$, что приводит к нулевому первому члену (поскольку $\sin(0) = 0$), который ничего не вносит в преобразование, поэтому DST не имеет коэффициента DC.

Этот недостаток можно увидеть при рассмотрении примера восьми одинаковых значений данных, подлежащих преобразованию. Такие значения, конечно, прекрасно коррелируют. При нанесении они становятся горизонтальной линией. Применяя DCT к этим значениям, получается только один коэффициент DC; все коэффициенты AC являются нулевыми. DCT сжимает всю энергию данных в один коэффициент DC, значение которого идентично значению данных. IDCT может перестроить в точности восемь значений (за исключением некоторых незначительных изменений, связанных с ограниченной точностью расчета). Применение DST к тем же восьми значениям, с другой стороны, приводит к семи коэффициентам DC, сумма которых представляет собой форму сигнала, которая проходит через восемь точек, соответствующих данным, но колеблется между этими точками. Такое поведение, имеет три недостатка, в основном:

1. Энергия исходных данных не уплотняется;
2. Семь коэффициентов не коронованы (при этом данные идеально коррелированы);
3. Путем количественной оценки семи коэффициентов можно добиться значительного снижения качества реконструкции, выполняемой обратным DST.

5.5. Сжатие JPEG (Joint Photographers Experts Group)

Область сжатия (кодирования) изображений связана с минимизацией количества битов, необходимых для восстановления изображения, с приложениями, особенно в передаче и хранении изображений. Применение в области передачи изображений находят в радиовещании, космической связи, радиолокационных и гидролокационных, телекоммуникационных сетях, факсимильных передачах, телеконференциях и т.д. Сжатие изображений имеет важное значение с точки зрения хранения (хранения) изображений в приложениях медицинской визуализации, в цифровых видеотехнологиях, для создания мультимедийных документов и т. Д. Новые технологии сжатия изображений обеспечивают возможное решение для интеграции приложений цифровых изображений и видео. Степень сжатия теперь достигла до 1:100, в зависимости от качества восстановленного изображения. Техники сжатия недостаточно, чтобы иметь возможность решать проблемы, возникающие в мультимедийных приложениях. Для того чтобы иметь возможность достичь переносимости цифровых видеоизображений и последовательностей приложений на нескольких системах, необходимо внедрить стандарты сжатия мультимедийных данных. Эти стандарты устанавливают порядок хранения и передачи сжатых данных с целью их возможного использования. Наиболее широко используемым стандартом сжатия статических изображений является стандарт JPEG, созданный Объединенной группой экспертов по фотографии

ВОПРОСЫ