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

GRAFICA PE CALCULATOR

ТЕМА 5. СЖАТИЕ ИЗОБРАЖЕНИЙ

l.u., dr. NASTAS Andrei

CONȚINUT

5.1. Численное представление изображений

Изображение представляет собой обычно прямоугольную поверхность, характеризующуюся на уровне любой ее точки определенным цветом. В идеале, цвет постоянно меняется в любом направлении. К сожалению, в числовых системах нельзя использовать непрерывно меняющиеся размеры, а только их дискретизированную форму.

Таким образом, изображение должно быть дискретизировано, прежде чем возникнет вопрос о численном представлении. Дискретизация заключается в разделении изображения на сетку, похожую на шахматную доску. Каждая часть изображения, ограниченная этой сеткой, будет считаться имеющей однородный цвет — среднее значение существующего цвета в этом разделе. Эти участки также называются пикселями или точками, количество которых определяет разрешение изображения. Например, для любого изображения с разрешением 640 x 480 пикселей это означает, что на его поверхности определена сетка, которая делит его по горизонтали на 640 секций и по вертикали на 480.

Следующим шагом является поиск цветового представления. Любой цвет может быть разбит на три основных цвета (красный – R, зеленый – G и синий – B), другими словами, любое изображение может быть получено путем аддитивного наложения трех световых излучений этих трёх цветов с различной интенсивностью. Итак, для того, чтобы численно представить цвет, достаточно представить световую интенсивность трех основных цветов.

5.1. Численное представление изображений

Если для каждого компонента выделено 8 бит, то можно закодировать 256 уровней интенсивности, таким образом, отсутствие цвета (нулевая интенсивность) кодируется значением 00000000 в двоичном или 00 в шестнадцатеричном формате, а максимальная интенсивность — наибольшим значением, которое может быть представлено на 8 битах, а именно 11111111 в двоичном или FF в шестнадцатеричном. Это представление, однако, больше связано с техническими способами захвата и воспроизведения изображения и меньше с физиологическим механизмом восприятия цвета. В ходе различных экспериментов было установлено, что с точки зрения способности воспринимать детали глаз более чувствителен к светящейся интенсивности цвета, чем к оттенку. По этой причине интерес представляет другой способ представления цвета, который учитывает это наблюдение, примером которого является представление YUV, используемое в цветном телевидении. В этом случае вместо трех первичных компонентов R,G,B используются три других производных от них размера, а именно:

$$\begin{cases} Y = 0,3R + 0,59G + 0,11B \\ U = R - Y = 0,7T - 0,59G - 0,11B \\ V = B - Y = -0,3R - 0,59 + 0,89B \end{cases} \qquad (5.1)$$

5.1. Численное представление изображений

В случае этого представления компонент Y соответствует воспринимаемой *световой интенсивности* для этого цвета (коэффициенты 0,30, 0,59 и 0,11 представляют *яркость* относительно белого, трех основных цветов красного, зеленого и синего соответственно). Этот компонент также известен как *яркость*. Компоненты U и V определяют оттенок цвета, по этой причине они называются *компонентами цветности*. Они рассчитываются как разница между красными и синими составляющими и компонентами яркости соответственно.

Преимущество представления YUV заключается в том, что оно отделяет компонент яркости, для которого глаз очень чувствителен к деталям, от компонентов оттенка, для которых чувствительность ниже. Это позволяет уменьшить информацию, связанную с изображением, используя более низкое разрешение для компонентов цветности. В случае цветного телевидения «сжатие» достигается путем ограничения полосы частот, выделенной сигналам цветности (например, в системе PAL сигналы U и V имеют полосу 1,3 МГц по сравнению с сигналом Y, который имеет полосу 6 МГц).

5.2. Представление изображения в несжатом формате

Изображение представлено в виде матрицы точек (обычно квадратной формы), каждая точка характеризуется цветом. Например, для изображения на рисунке 5.1 это можно увидеть, если увеличить часть изображения таким образом, чтобы точечная матрица стала видимой, как на рисунке 5.1, *б*.

Для преобразования такого изображения необходимо использовать числовой режим представления цвета. Для этого отталкивается от того, что любой цвет можно получить, смешивая в разных пропорциях три основных цвета (первичные цвета). В практике используются Красный (R), зеленый (G) и синий (B). Интенсивность света первичного цвета может быть численно представлена в виде 8-битного целого числа, значение 0 соответствует нулевой интенсивности, а максимальное (255) — максимальной интенсивности. Таким образом, цвет будет численно представлен 8-битным целочисленным триплетом (R,G,B). Например, цвет ЖЕЛТЫЙ будет иметь представление формы (255, 255, 0). В этих условиях изображение представлено в виде матрицы IM(Nx, Ny), где Nx представляет количество точек по горизонтали, а Ny — количество вертикальных точек, а элементы матрицы — триплеты целых чисел на 8 бит типа (R,G,B).

а)

б)

Рис. 5.1. Изображение, представленное в виде точечной матрицы

5.3. Методы и подходы к сжатию изображений

Ниже приведены некоторые методы, используемые в сжатии, подчеркивающие их применимость при сжатии изображений.

1. *Скалярный дизеринг* может быть использован для сжатия изображений, но его производительность посредственна. Например, 8-битное/пиксельное изображение может быть сжато путем масштабирования квантования путем исключения четырех наиболее незначительных битов каждого пикселя. Это приводит к коэффициенту сжатия 0,5, что, помимо того, что оно не является значительным, также приводит к тому, что количество цветов (или оттенков серого) одновременно уменьшается с 256 до 16. Такое сокращение не только снижает в целом качество восстановленного изображения, но даже может создавать полосы разных цветов, видимый и беспокоящий эффект.

5.3. Методы и подходы к сжатию изображений

Дизеринг — это термин, используемый для описания стратегического применения шума к изображению. Он традиционно использовался для улучшения внешнего вида изображений, когда вывод ограничен определенным цветовым диапазоном.

Например, 1-битное изображение является монохромным и может использовать только палитру из двух цветов: черного и белого. Дизеринг можно использовать для создания появления нескольких оттенков, варьируя расстояние между точками. Вы можете воспринимать несколько оттенков серого на изображении ниже, но присутствуют только черные и белые цвета:

5.3. Методы и подходы к сжатию изображений

Пример

Считается, например, ряд из 12 пикселей похожих цветов, начиная с 202 до 215. В двоичной системе этими значениями являются:

11010111 11010110 11010101 11010011 11010010 11010001

11001111 11001110 11001101 11001100 11001011 11001010.

Дизеринг выдаст следующие 12 4-битных значений:

1101 1101 1101 1101 1101 1101 1100 1100 1100 1100 1100 1100,

из которых 12 пикселей будут восстановлены, добавив 4 нуля к каждому количественному значению:

11010000 11010000 11010000 11010000 11010000 11010000

11000000 11000000 11000000 11000000 11000000 11000000.

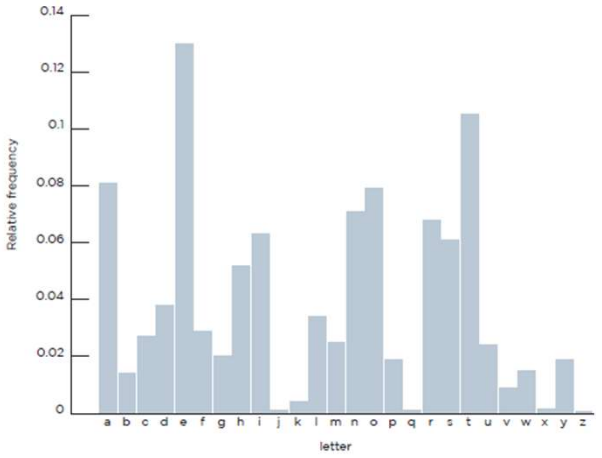
Первые шесть пикселей строки теперь имеют значение 110100002 = 208, а следующие шесть пикселей — 110000002 = 192. Если соседние строки имеют похожие пиксели, первые шесть столбцов образуют полосу, явно отличную от полосы, образованной следующими шестью столбцами. Это явление формирования полос, или контуров, очень заметно для глаз, так как они чувствительны к краям и разрывам на изображении.

5.3. Методы и подходы к сжатию изображений

2. *Векторный дизеринг* может быть использован более успешно для сжатия изображений.

Векторное квантование (CV) в общем смысле — это приближение сигнала непрерывной амплитуды дискретным амплитудным сигналом (состоящим из отдельных сигналов). В случае нашего приложения CV состоит из представления вектора *x* размерности *k* вектором *y* того же размера, принадлежащим конечному множеству, называемому *словарем*.

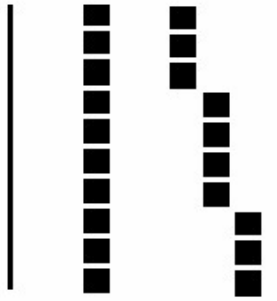
3. *Статистические методы* работают лучше, когда сжимаемые символы имеют разные вероятности. Входная последовательность, в которой сообщения имеют одинаковую вероятность, не будет эффективно сжиматься. В этом смысле в черно-белом или цветном изображении в непрерывных тонах разные цвета или оттенки серого часто оказываются имеющими примерно одинаковые вероятности. Именно поэтому статистические методы не являются хорошим выбором для сжатия таких изображений, и необходимы новые подходы. Изображения с цветовыми разрывами, в которых соседние пиксели имеют очень разные цвета, лучше сжимаются статистическими методами, но в этом случае предсказать пиксели непросто (рациональная операция предвосхищения события или явления).



5.3. Методы и подходы к сжатию изображений

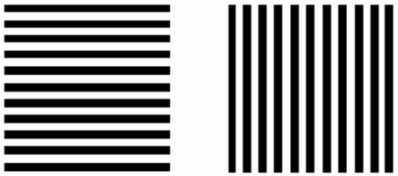
4. *Методы сжатия на основе словаря* также, как правило, не удачны в изображениях с непрерывным тоном. Такое изображение обычно содержит соседние пиксели в похожих цветах, но не содержит повторяющихся узоров. Даже изображение, содержащее повторяющиеся узоры, такие как вертикальные линии, может потерять их при оцифровке. Вертикальная линия в исходном изображении может стать слегка наклонной при оцифровке изображения. Идеальная вертикальная линия показана на рисунке 5.2, *а*. На рисунке 5.2, *б* предполагается, что линия идеально оцифрована в десяти пикселях, расположенных вертикально. Однако, если изображение помещено в слегка косой дигитайзер, процесс оцифровки может быть несовершенным, и полученные пиксели могут выглядеть так, как показано на рисунке 5.2, *в*.

Другая проблема со сжатием изображений на основе словаря заключается в том, что такие методы сканируют изображение строка за строкой и, таким образом, могут потерять вертикальные корреляции между пикселями. Примером являются два изображения на рисунке 5.3, *а*, *б*. Сохранение обоих изображений в GIF89, графическом формате на основе словаря, привело к файлам размером 1053 и 1527 байт соответственно.



а) б) в)

Рис. 5.2. Совершенная и несовершенная оцифровка



а) б)

Рис. 5.3. Сжатие параллельных строк на основе словаря

5.3. Методы и подходы к сжатию изображений

Традиционные методы неудовлетворительны для сжатия изображения, поэтому потребовались новые подходы, которые хоть и отличаются, но основаны на устранении избыточности из изображения, используя следующий принцип:

Принцип сжатия изображения. Если пиксель выбран случайным образом из изображения, весьма вероятно, что его соседи будут иметь тот же цвет или очень близкие цвета.

При сжатии изображения основываются на том факте, что соседние пиксели сильно коррелируют. Эта корреляция также называется *пространственной избыточностью*.

Пример: Ниже приведен простой пример, иллюстрирующий, как можно удалить избыточность из строки коррелированных пикселей. Следующая последовательность значений представляет интенсивность 24 смежных пикселей в ряду изображения непрерывного тона: 12, 17, 14, 19, 21, 26, 23, 29, 41, 38, 31, 44, 46, 57, 53, 50, 60, 58, 55, 54, 52, 51, 56, 60.

Только два из 24 пикселей идентичны. Их среднее значение составляет 40,3. Вычитание пар смежных пикселей приводит к последовательности: 12, 5, – 3, 5, 2, 4, – 3, 6, 11, – 3, – 7, 13, 4, 11, – 4, – 3, 10, – 2, – 3, – 3, 1, – 2, – 1, 5, 4.

Эти две последовательности показаны на рисунке 5.4.

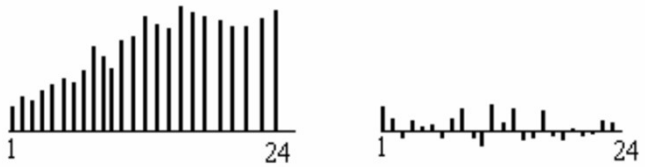


Рис. 5.4. Значения и различия 24 соседних пикселей

5.3. Методы и подходы к сжатию изображений

Последовательность разности значений имеет три свойства, показывающие ее потенциал сжатия:

1. Значения разностной последовательности меньше значений исходных пикселей. Их среднее значение составляет 2,58.

2. В последовательности разности есть значения которые повторяются. Существует только 15 различных значений в последовательности разности, поэтому в основном они могут быть закодированы четырьмя битами каждый.

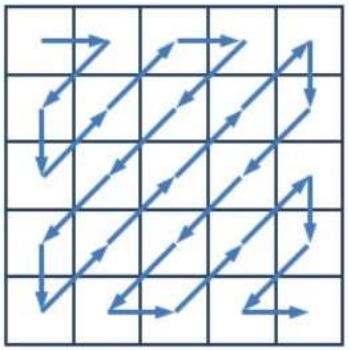
3. Значения последовательности разности декорелированы, т.е. смежные значения разности имеют разные значения. Это можно наблюдать, если произвести дальнейшее уменьшение, в результате чего получается последовательность 12, – 7, – 8, 8, – 3, 2, – 7, 9, 5, – 14, – 4, 20, – 11, 7, – 15, 1, 13, – 12, – 1, 4, – 3, 1, 6, 1. Его значения выше предыдущих различий.

В общем, методы сжатия для изображений предназначены для определенного типа изображений, и некоторые из этих конкретных методов, представленных ниже. Целевыми конкретными изображениями являются *двухуровневые изображения, изображения в градациях серого и цветные изображения*.

5.3. Методы и подходы к сжатию изображений

Подход 1. Этот подход используется для сжатия двухуровневых изображений. Каждый пиксел такого образа представляется одним битом. Применение принципа сжатия образов к компрессии двухуровневых изображений означает, что непосредственные соседи пиксела P стремятся совпадать с P. Поэтому имеет смысл использовать методы кодирования длин серий (RLE) для сжатия таких изображений. Метод сжатия сканирует образ строка за строкой и вычисляет длины последовательных черных и белых пикселов. Длины кодируются кодами переменной длины и записываются в сжатый файл. Примером такого сжатия является факсимильная компрессия.

Еще раз подчеркнем, что это всего лишь общий принцип сжатия, конкретные методы могут сильно отличаться друг от друга. Например, метод может сканировать образ по строкам, а может зигзагообразно или сканировать изображение область за областью с помощью заполняющих их кривых



5.3. Методы и подходы к сжатию изображений

Подход 2. Также метод для двухуровневых изображений. Опять используется принцип совпадения ближайших пикселов в расширенной форме: если текущий пиксел имеет цвет c (где c - белый или черный), то пикселы того же цвета, наблюдавшиеся в прошлом (а также те, которые появятся в будущем) будут иметь таких же ближайших соседей.

Этот подход рассматривает *n* последовательных соседей текущего пиксела и представляет их в виде *n*-битного числа. Это число называется контекстом пиксела. В принципе, может быть различных контекстов, но в силу избыточности образа мы ожидаем, что контексты распределены неравномерно. Некоторые контексты встречаются чаще, а некоторые - реже.

Кодер вычисляет сколько раз встречался каждый контекст для пиксела цвета c и присваивает контекстам соответствующие вероятности. Если текущий пиксел имеет цвет *c* и его контекст имеет вероятность *p*, то кодер может использовать адаптивное арифметическое кодирование для кодирования пиксела с этой вероятностью. Такой подход использован в методе JBIG (Joint Bi-level Processing Group).

Перейдем теперь к полутоновым изображениям. Пиксел такого изображения представлен *n* битами и может иметь одно из 2^n значений. Применяя принцип сжатия изображений, заключаем, что соседи пиксела *P* стремятся быть похожими на *P*, но не обязательно совпадают с ним. Поэтому метод RLE (run length encoding) здесь не годится.

5.3. Методы и подходы к сжатию изображений

Подход 3. Расслоить полутоновую картинку на *n* двухуровневых изображений и каждую сжать с помощью RLE и префиксных кодов. Здесь кажется, что можно применить основной принцип сжатия изображений, но по отношению к каждому отдельному слою полутонового изображения. Однако, это не так, и следующий пример проясняет ситуацию. Представим себе полутоновое изображение с $n = 4$ (т.е. 4 бита/пиксель или 16 оттенков серого). Ее можно расслоить на 4 двухуровневых изображения. Если два смежных пикселя исходного образа имели величины 0000 и 0001, то они похожи по цвету. Однако два пиксела со значениями 0111 и 1000 также близки на полутоновом изображении (это, соответственно, числа 7 и 8), но они различаются во всех 4 слоях.

Эта проблема возникает, так как в двоичном представлении соседние числа могут отличаться во многих разрядах. Двоичные коды для 0 и 1 отличаются в одном разряде, коды для 1 и 2 различаются в двух разрядах, а коды для 7 и 8 различаются уже во всех четырех битах. Решением этой проблемы может служить разработка специальных последовательностей двоичных кодов, в которых последовательные коды с номерами и различались бы ровно на один бит. Примером таких кодов служат рефлексные коды Грея.

5.3. Методы и подходы к сжатию изображений

Подход 4. Использует контекст пикселя *P* для прогнозирования его значения, который состоит из значений нескольких ближайших соседей. Выбираем несколько соседних пикселов, вычисляем среднее значение *A* их величин и делаем предсказание, что пиксел *P* будет равен *A*. Основной принцип сжатия изображений позволяет считать, что в большинстве случаев мы будем правы, во многих случаях будем почти правы и лишь в немногих случаях будем совершенно неправы. Можно сказать, что в предсказанной величине пиксела *P* содержится избыточная информация про *P*. Вычислим разность, $\Delta = P - A$, и присвоим некоторый (префиксный) код переменной длины величине Δ так, чтобы малым величинам (которые ожидаются часто) соответствовали короткие коды, а большим величинам (которые ожидаются редко) назначались длинные коды. Если значение *P* лежит в интервале от 0 до $m - 1$, то значения Δ попадают в интервал $[-(m-1), +(m-1)]$, для чего потребуется $2(m-1)+1$ или $2m-1$ кодов.

.

5.3. Методы и подходы к сжатию изображений

Экспериментирование с большими числами показывает, что значения Δ стремятся иметь распределение, близкое к распределению Лапласа, которое часто возникает в статистике. Тогда метод сжатия может использовать это распределение для присвоения вероятностей величинам и весьма эффективно применять арифметическое кодирование. Этот подход лежит в основе метода сжатия MLP (Meridian Lossless Packing).

Контекст пикселя может состоять из одного или двух его соседей. Однако лучшие результаты получаются, если использовать несколько пикселей, причем каждому пикселю присваивается некоторый вес для вычисления взвешенного среднего: более дальним пикселям присваивается меньший вес. Другое важное рассмотрение касается декодирования. Для декодирования изображения необходимо уметь вычислять контекст до самого пиксела. Это означает, что контекст должен состоять из уже декодированных пикселов. Если изображение сканируется строка за строкой (сверху вниз и слева направо), то контекст может состоять только из пикселей, которые расположены выше и левее данного пикселя.

5.3. Методы и подходы к сжатию изображений

Подход 5. Преобразование применяется к значениям пикселей и кодируется к преобразованным значениям. Понятие преобразования образа, а также наиболее важные преобразования, используемые при компрессии изображений. Напомним, что компрессия достигается путем удаления или сокращения избыточности. Избыточность изображения образуется за счет корреляции между пикселами, поэтому преобразования, которые делают декорреляцию, одновременно удаляют избыточность. Квантуя преобразованные значения, можно получить эффективное сжатие с частичной потерей информации. В процессе сжатья величины преобразовывать в независимые, так как для независимых величин легче построить простую модель для их кодирования.

Дальше рассмотрим кодирование цветных изображений. Пиксели таких изображений состоят из трех компонентов, красного, зеленого и синего. Большинство цветных картинок являются непрерывно-тоновыми или дискретно-тоновыми.

5.3. Методы и подходы к сжатию изображений

Подход 6. Принцип такого подхода заключается в разделении цветного изображения непрерывного тона на три полутоновых изображения и сжатии каждого из них по отдельности, используя подходы 2, 3 и 4.

Принцип сжатия непрерывно-тоновых изображений гласит, что цвет соседних пикселов мало изменяется, хотя и не обязательно постоянен. Однако близость цветов еще не означает близость величин пикселов. Рассмотрим, например, цветной 12-битовый пиксел, в котором каждая компонента имеет 4 бита. Итак, 12 бит 1000|0100|0000 обозначают пиксел, цвет которого получается смешением 8 единиц красного (около 50%, так как всего их 16), четырех единиц зеленого (около 25%) и совсем без синего. Представим себе теперь два пиксела со значениями 0011|0101|0011 и 0010|0101|0011. Их цвета весьма близки, поскольку они отличаются только в одной красной компоненте, и там их различие состоит только в одном бите. Однако, если рассмотреть их как 12-битовые числа, то два числа 0011|0101|0011=851 и 0010|0101|0011=595 различаются весьма значительно.

Важная особенность этого подхода состоит в использовании представления цвета в виде яркости и цветности вместо обычного RGB. Преимущество этого представления основано на том, что глаз чувствителен к маленьким изменениям яркости, но не цветности. Это позволяет допускать значительную потерю в компоненте цветности, но при этом совершать декодирование без значительного визуального ухудшения качества изображения.

5.3. Методы и подходы к сжатию изображений

Подход 7. Другие методы требуются для компрессии дискретно-тоновых изображений. Напомним, что такие изображения состоят из больших одноцветных областей, причем каждая область может появляться в нескольких местах образа. Хорошим примером служит содержимое экрана компьютера. Такой образ состоит из текста и пиктограмм (иконок). Каждая буква, каждая пиктограмма занимают некоторую область на экране, причем каждая из этих областей может находиться в разных местах экрана. Возможный способ сжатия такого изображение заключается в его сканировании и обнаружении таких повторяющихся областей. Если область B совпадает с ранее выделенной областью A, то можно сжать B, записав в файл указатель на область A. Примером такого метода служит алгоритм блоковой декомпозиции ABD (block decomposition algorithm).

5.3. Методы и подходы к сжатию изображений

Подход 8. Разделение изображения на части (перекрывающиеся или нет) и сжатие их одна за другой. Предположим, что следующая требующая сжатия часть имеет номер 15 (n). Постараемся сравнить ее с уже обработанными частями 1-14 ($1 \div n - 1$). Предположим, что ее можно выразить, например, с помощью комбинации частей 5 (растяжение) и 11 (поворот), тогда достаточно сохранить только несколько чисел, которые описывают требуемую комбинацию, а саму часть 15 можно отбросить. Если часть 15 не удается выразить в виде комбинации прежних частей, то ее придется сохранить в «сыром виде».

Другой подход является основой для различных фрактальных методов сжатия изображений. Здесь применяется основной принцип сжатия изображений, но вместо пикселов выступают части изображения. Этот принцип говорит о том, что «интересные» изображения (то есть те, которые будут сжиматься на практике) обладают некоторой степенью самоподобия. Часть изображения совпадает или близка всему изображению.

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 коррелируют) в представление, где они декорированы. 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 реконструирует исходные данные (точные или аппроксимированные) путем применения обратного преобразования. Преобразования, обсуждаемые ниже, являются ортогональными.

Термин декорарляция 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 вы превращаете изображение в представление, в котором пиксели украшены, избыточность удаляется, и изображение становится полностью сжатым.

.

ВОПРОСЫ