

Профессор
Игорь Н. Бекман

КОМПЬЮТЕРНЫЕ НАУКИ

Курс лекций

Лекция 5. ИНФОРМАЦИЯ В КОМПЬЮТЕРЕ

Содержание

1. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ	1
2. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ	2
2.1 Целые числа	3
2.2 Вещественные числа	4
3. ПРЕДСТАВЛЕНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ	7
4. ПРЕДСТАВЛЕНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ	11
4.1 Векторное представление	13
4.2 Растровое представление	14
4.3 Цветовые модели	16
4.3.1 RGB	16
4.3.2 CMYK	18
4.3.3 HSB	19
4.4 Форматы графических файлов	20
4.5 Фрактальная графика	21
5. ПРЕДСТАВЛЕНИЕ ЗВУКОВОЙ ИНФОРМАЦИИ	21

Компьютер, помогающий человеку хранить и обрабатывать информацию, приспособлен в первую очередь для обработки текстовой, числовой, графической информации.

В данной лекции мы рассмотрим способы представления информации в компьютере, в первую очередь – представление целых и вещественных чисел, текстовой и графической информации (включая векторное, растровое и фрактальное представления). В заключительной части лекции коротко остановимся на представлении в компьютере звуковой информации.

1. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ

Информация графическая и звуковая может быть представлена в аналоговой или дискретной форме. При аналоговом представлении информации физическая величина может принимать бесконечное множество значений. При дискретном представлении информации физическая величина может принимать конечное множество значений, при этом она изменяется скачкообразно. Примером аналогового и дискретного представления информации можно привести наклонную плоскость и лестницу. Положение тела на наклонной плоскости и на лестнице задается значениями координат X и Y . При движении тела по наклонной плоскости его координаты могут принимать бесконечное множество непрерывно изменяющихся значений из определенного диапазона, а при движении по лестнице – только конечный набор значений, изменяющихся скачкообразно.

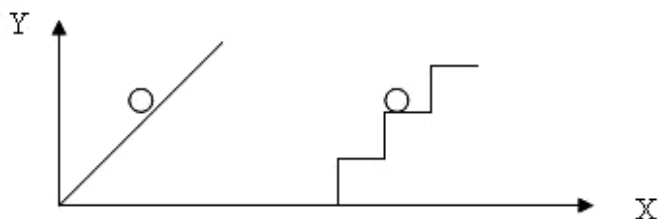


Рис. 1. Аналоговое и дискретное представление информации.

Примером аналогового хранения звуковой информации является виниловая пластинка (звуковая дорожка изменяет свою форму непрерывно), а дискретного – аудио компакт-диск (звуковая дорожка которого содержит участки с различной отражающей способностью).

Преобразование графической и звуковой информации из аналоговой формы в дискретную производится путем дискретизации, т.е. разбиения непрерывного графического изображения или

непрерывного (аналогового) звукового сигнала на отдельные элементы. В процессе дискретизации производится кодирование, т.е. присвоение каждому элементу конкретного значения в форме кода.

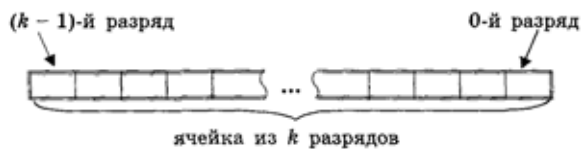
Дискретизация – это преобразование непрерывных изображений и звука в набор дискретных значений, каждому из которых присваивается значение кода.

В аналоговой форме звук представляет собой волну с непрерывно меняющейся амплитудой и частотой. При преобразовании звука в цифровую дискретную форму производится временная дискретизация, при которой в определенные моменты времени амплитуда звуковой волны измеряется и квантуется. С помощью специальных программных средств (редакторов звукозаписей) открываются широкие возможности по созданию, редактированию и прослушиванию звуковых файлов. Создаются программы распознавания речи и, в результате, появляется возможность управления компьютером с помощью голоса.

Числовая информация была первым видом информации, который начали обрабатывать ЭВМ, и долгое время она оставалась единственным видом. Поэтому не удивительно, что в современном компьютере существует большое разнообразие типов и представлений чисел. Прежде всего, это **целые и вещественные числа**, которые по своей сути и по представлению в машине различаются очень существенно. Целые числа, в свою очередь, делятся на числа со знаком и без знака, имеющие уже не столь существенные различия. Наконец, вещественные числа имеют два способа представления – с фиксированной и с плавающей запятой, правда, первый способ сейчас представляет в основном исторический интерес.

Существуют разные способы кодирования и декодирования информации в компьютере. Это зависит от вида информации: текст, число, графическое изображение или звук. Для числа также важно, как оно будет использовано: в тексте, или в вычислениях, или в процессе ввода-вывода. Вся информация кодируется в двоичной системе счисления: с помощью цифр 0 и 1. Эти два символа называют двоичными цифрами или битами. Такой способ кодирования технически просто организовать: 1 - есть электрический сигнал, 0 - нет сигнала. Недостаток двоичного кодирования - длинные коды. Но в технике легче иметь дело с большим числом простых однотипных элементов, чем с небольшим числом сложных.

Каждый регистр арифметического устройства компьютера, каждая ячейка памяти представляет собой физическую систему, состоящую из некоторого числа однородных элементов, обладающих двумя устойчивыми состояниями, одно из которых соответствует нулю, а другое – единице. Каждый такой элемент служит для записи одного из разрядов двоичного числа. Именно поэтому каждый элемент ячейки называют **разрядом**.



Вычислительная техника возникла как средство автоматизации вычислений. Сегодня компьютеры обрабатывают различные виды информации: числовую, текстовую, звуковую, графическую. Однако современный компьютер может хранить и обрабатывать только дискретную информацию. Следовательно,

любой вид информации, подлежащий компьютерной обработке, тем или иным способом должен быть закодирован с помощью конечной последовательности целых чисел, которая затем переводится в двоичный вид для хранения в компьютере.

Задача перевода информации естественного происхождения в компьютерную называется задачей **дискретизации** или **квантования**. Эту задачу необходимо решать для всех видов информации. Способы дискретизации для разных видов информации различны, но подходы к решению этой задачи построены на одинаковых принципах.

Кодирование обычно проводят в несколько этапов: 1) Определение объёма информации, подлежащей кодированию; 2) Классификация и систематизация информации; 3) Выбор системы кодирования и разработка кодовых обозначений; 4) Непосредственно кодирование

2. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

В двоичной системе счисления для представления используются две цифры 0 и 1. Сравните: в десятичной системе счисления $435,6710 = 4 \cdot 10^2 + 3 \cdot 10^1 + 5 \cdot 10^0 + 6 \cdot 10^{-1} + 7 \cdot 10^{-2}$ в двоичной системе счисления $10110,1012 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$ Действия с числами в двоичной системе счисления изучает наука двоичная арифметика. Все основные законы арифметических действий для таких чисел выполняются. Для сравнения рассмотрим два варианта кодирования для числа 45. При использовании числа в тексте каждая цифра кодируется 8 битами в соответствии с *ASCII* (т.е. потребуется 2 байта): 4 - 01000011, 5 - 01010011. При использовании в вычислениях код этого числа получается по специальным

правилам перевода из десятичной системы счисления в двоичную в виде 8-разрядного двоичного числа: 001011012, что потребует 1 байт.

2.1 Целые числа

Любое целое число можно рассматривать как вещественное, но с нулевой дробной частью, т.е. можно было бы ограничиться представлением в компьютере вещественных чисел и реализацией арифметических действий над ними. Однако для эффективного использования памяти, повышения скорости выполнения вычислений и введения операции деления нацело с остатком целые числа представляются специально для них предназначенными способами.

Введение специальных способов представления целых чисел оправдано тем, что достаточно часто в задачах, решаемых с помощью компьютера, многие действия сводятся к операциям над целыми числами. Например, в задачах экономического характера данными служат количества акций, сотрудников, деталей, транспортных средств и т.д., по своему смыслу являющиеся целыми числами. Целые числа используются и для обозначения даты и времени, и для нумерации различных объектов: элементов массивов, записей в базах данных, машинных адресов и т.п.

Для компьютерного представления целых чисел обычно используется несколько различных способов представления, отличающихся друг от друга количеством разрядов и наличием или отсутствием знакового разряда. Беззнаковое представление можно использовать только для неотрицательных целых чисел, отрицательные числа представляются только в знаковом виде.

При беззнаковом представлении все разряды ячейки отводятся под само число. При представлении со знаком самый старший (левый) разряд отводится под знак числа, остальные разряды – под собственно число. Если число положительное, то в знаковый разряд помещается 0, если отрицательное - 1. Очевидно, в ячейках одного и того же размера можно представить больший диапазон целых неотрицательных чисел в беззнаковом представлении, чем чисел со знаком. Например, в одном байте (8 разрядов) можно записать положительные числа от 0 до 255, а со знаком – только до 127. Поэтому, если известно заранее, что некоторая числовая величина всегда является неотрицательной, то выгоднее рассматривать её как беззнаковую.

Целые числа в компьютере хранятся в формате с фиксированной запятой.

Для получения компьютерного представления беззнакового целого числа в k -разрядной ячейке памяти достаточно перевести его в двоичную систему счисления и дополнить полученный результат слева нулями до k разрядов. Понятно, что существует ограничение на числа, которые мы можем записать в k -разрядную ячейку.

Максимально представимому числу соответствуют единицы во всех разрядах ячейки (двоичное число, состоящее из k -единиц). Для k -разрядного представления оно будет равно 2^k-1 . Минимальное число представляется нулями во всех разрядах ячейки, оно всегда равно нулю. **Табл. 1.** Максимальные числа для беззнакового представления при различных значениях k :

Количество разрядов	Максимальное число
8	255 ($2^8 - 1$)
16	65535 ($2^{16} - 1$)
32	4294967295 ($2^{32} - 1$)
64	18446744073709551615 ($2^{64} - 1$)

При знаковом представлении целых чисел возникают такие понятия, как прямой, обратный и дополнительный коды.

Представление числа в привычной для человека форме знак-величина, при которой старший разряд ячейки отводится под знак, остальные $k-1$ разрядов – под цифры числа, называется прямым кодом.

Например, прямые коды двоичных чисел 11001_2 и -11001_2 для восьмиразрядной ячейки равны 00011001 и 10011001 соответственно. Положительные целые числа представляются в компьютере с помощью прямого кода. Прямой код отрицательного целого числа отличается от прямого кода соответствующего положительного числа содержимым знакового разряда. Но вместо прямого для представления отрицательных целых чисел в компьютере используется дополнительный код.

Отметим, что максимальное положительное число, которое можно записать в знаковом представлении в k разрядах, равно $2^{k-1}-1$, что практически в два раза меньше максимального числа в беззнаковом представлении в тех же k разрядах.

Пример 1. Число $53=110101_2$ в восьмиразрядном представлении имеет вид:

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

Это же число 53 в 16 разрядах будет записано следующим образом:

0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

В обоих случаях неважно, знаковое или беззнаковое представление при этом используется.

Пример 2. Для числа $200=11001000_2$ представление в 8 разрядах со знаком невозможно, т.к. максимальное допустимое число в таком представлении равно 127, а в беззнаковом восьмиразрядном представлении оно имеет вид:

1	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---

Для представления в компьютере целых отрицательных чисел используют дополнительный код, который позволяет заменить арифметическую операцию вычитания операцией сложения, что существенно увеличивает скорость вычислений. Прежде чем вводить определение дополнительного кода, сделаем одно замечание.

В k -разрядной целочисленной компьютерной арифметике $2^k \equiv 0$.

Объяснить это можно тем, что двоичная запись числа 2^k состоит из одной единицы и k нулей, а в ячейку из k разрядов может уместиться только k цифр, в данном случае только k нулей. В таком случае говорят, что значащая единица вышла за пределы разрядной сетки.

k – разрядный дополнительный код отрицательного числа m – это запись в k разрядах положительного числа $2^k - |m|$, где $|m|$ – модуль отрицательного числа m , $|m| \leq 2^{k-1}$.

Обратный код является дополнением исходного числа до числа $2^k - 1$, состоящего из k двоичных единиц. Поэтому прибавление единицы к инвертированному коду позволяет получить его искомым дополнительным кодом.

Пример 3. Получим дополнительный код числа -52 для восьми- и шестнадцатиразрядной ячейки.

Для восьмиразрядной ячейки:

0011 0100 – прямой код числа $|-52|=52$;

1100 1011 – обратный код числа -52;

1100 1100 – дополнительный код числа -52.

Для шестнадцатиразрядной ячейки:

0000 0000 0011 0100 – прямой код числа $|-52|$;

1111 1111 1100 1011 – обратный код числа -52;

1111 1111 1100 1100 – дополнительный код числа -52.

Приведём значения границ диапазонов для знаковых представлений в ячейках с различной разрядностью:

Разрядность	Минимальное число	Максимальное число
8	-128	127
16	-32768	32767
32	-2147483648	2147483647
64	-9223372036854775808	9223372036854775807

2.2 Вещественные числа

Если при представлении целых чисел в компьютере ограничением может служить лишь величина записываемого числа, то при записи вещественного числа речь идёт о тонности его представления, т.е. о количестве значащих цифр, которые удаётся сохранить в ограниченном числе разрядов.

Любое число a в экспоненциальной форме представляется в виде $a = \pm m \cdot P^q$, - где основание системы счисления, m называется **мантиссой** числа, q – **порядком** числа.

Пример 4. Длину отрезка, равного 47,8 см, в экспоненциальной форме записи можно представить так:

- 1) $478 \cdot 10^{-1}$ см = 478 мм;
- 2) $4,78 \cdot 10^1$ см = 4,78 дм;
- 3) $47,8 \cdot 10^0$ см = 47,8 см;
- 4) $0,478 \cdot 10^2$ см = 0,478 м.

Из этого примера видно, что длину одного и того же отрезка можно записать с использованием различных экспоненциальных форм. Эта неоднозначность записи может приводить в определённых случаях к неудобству. Из курса алгебры известно, что если P фиксировано и $1/P \leq m < 1$, то представление числа в экспоненциальной форме единственно. Такая форма экспоненциального представления называется **нормализованной формой** и используется в компьютере для однозначности представления вещественных чисел.

Нормализованная запись отличного от нуля вещественного числа – это запись вида $a = \pm m \cdot P^q$, где q – целое число (положительное, отрицательное, или ноль), а m – правильная P -ичная дробь, у которой первая цифра после запятой не равна нулю, т.е. $1/P \leq m < 1$

Число ноль не может быть записано в нормализованной форме так, как она была определена. Поэтому относительно нормализованной записи нуля приходится прибегать к особым соглашениям. Условимся, что **запись нуля является нормализованной, если и мантисса, и порядок равны нулю.**

В нормализованной форме все числа записываются одинаково в том смысле, что запятая у них ставится в одном и том же месте – перед первой (самой левой) значащей цифрой мантиссы. Заметим, что в двоичной системе счисления первая цифра мантиссы нормализованного числа всегда равна 1 (за исключением числа ноль). Величина же числа (т.е. её порядок) указывается отдельно, с помощью соответствующей степени основания системы счисления, в которой это число было записано изначально. Количество цифр в мантиссе может оказаться меньше, чем число значащих цифр в исходном числе. Часто в нормализованной записи мантисса P -ичного числа записывается в P -ичной системе счисления, а порядок и само число P – в десятичной.

Пример 5. Приведём примеры нормализации чисел:

- 1) $0 = 0,0 \cdot 10^0$ (возможная нормализации нуля);
- 2) $3,1415926 = 0,31415926 \cdot 10^1$ (количество значащих цифр не изменилось);
- 3) $1000 = 0,1 \cdot 10^4$ (количество значащих цифр уменьшилось с четырёх до одной);
- 4) $0,123456789 = 0,123456789 \cdot 10^0$ (запятую передвигать не нужно);
- 5) $0,00001078 = 0,1078 \cdot 8^{-4}$ (количество значащих цифр уменьшилось с семи до трёх);
- 6) $1000,00012 = 0,100000012 \cdot 2^4$ (количество значащих цифр уменьшить невозможно).

При записи нормализованного числа в компьютере для записи мантиссы и порядка отводится заранее фиксированное количество разрядов.

В компьютерном представлении вещественных чисел максимально допустимое количество цифр в мантиссе определяет точность, с которой может быть представлено число.

Модуль разности между значением числа x и компьютерным его представлением x^* называется абсолютной погрешностью представления x .

Несмотря на то, что в абсолютном исчислении погрешность может быть значительно больше 1, относительно величины самого числа её порядок остаётся неизменным.

Относительная погрешность представления x – величина $\left| \frac{x - x^*}{x} \right|$.

Абсолютная погрешность говорит о том, на сколько полученный результат (например, результат представления числа в компьютере) отличается от истинного результата (самого числа). При решении реальных задач знание этой величины позволяет оценивать, насколько достоверный результат был получен. Если его точность нас не удовлетворяет, то следует выбрать другой (более точный) способ представления чисел.

Относительная же погрешность показывает, сколько верных старших значащих цифр содержит результат. Значение относительной погрешности непосредственно связано с количеством разрядов, отводимых для представления мантиссы нормализованного числа. На практике обычно известна относительная погрешность представления чисел, т.к. разрядность мантиссы фиксирована. Следовательно, можно предположить, сколько верных цифр содержит результат. Если из каких-либо априорных соображений известно значение вычисляемой величины, о можно оценить абсолютную погрешность результата.

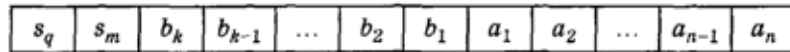
В компьютерной записи вещественных чисел с плавающей запятой количество цифр, отводимых под запись порядка, определяет, насколько большие и насколько маленькие положительные числа могут быть представлены.

Широкий диапазон представления чисел с плавающей запятой необходим при решении научных и инженерных задач. Такое представление чисел не только позволяет сохранять в разрядной сетке большое количество значащих цифр и тем самым повышать точность вычислений, но также упрощает действия над порядками и мантиссами.

Как и для целых чисел, при представлении вещественных чисел в компьютере используется чаще всего двоичная система счисления, следовательно, предварительно десятичное число должно быть переведено в двоичную систему, уж затем представлено в нормализованной форме с $P=2$. При представлении нормализованных чисел часть разрядов ячейки отводится для записи порядка числа, остальные разряды – для записи мантиссы. По одному разряду в каждой группе отводится для изображения

знака порядка и знака мантиссы. О таком представлении говорят, что число записано в формате с **плавающей запятой**.

Например, можно представить себе такое распределение разрядов ячейки памяти:



Первые два разряда служат для представления знаков порядка (s_q) и мантиссы (s_m) соответственно. Следующие k разрядов используется для представления абсолютной величины числа (q), остальные n разрядов – для представления абсолютной величины мантиссы. В каждом разряде ячейки может храниться одно из двух значений: 0 или 1.

Тогда изображённому на схеме состоянию ячейки соответствует число

$$(-1)^{s_m} \cdot 2^q \cdot \left(\frac{a_1}{2} + \frac{a_2}{2^2} + \dots + \frac{a_n}{2^n} \right),$$

где $q = (-1)^{s_q} (b_k \cdot 2^{k-1} + b_{k-1} \cdot 2^{k-2} + \dots + b_2 \cdot 2 + b_1)$.

При такой системе записи наибольшее по абсолютной величине число, которое может быть представлено в машине, равно $2^{2^k-1} \cdot (1 - 2^{-n})$, а наименьшее по абсолютной величине число, отличное от нуля, равно $2^{2^k-1} \cdot 2^{-n} = 2^{-2^k-n+1}$.

Вещественных чисел, точно представимых в компьютере, конечное число. Остальные числа либо приближаются представимыми, либо оказываются непредставимыми. Последнее относится к слишком большим и слишком маленьким вещественным числам.

Использование в компьютере представления чисел в формате с плавающей запятой усложняет выполнение арифметических операций.

При сложении и вычитании чисел сначала производится подготовительная операция, называемая выравниванием порядков. Она состоит в том, что мантисса числа с меньшим порядком сдвигается в своей ячейке вправо на количество разрядов, равное разности порядков данных чисел. После этой операции одноимённые разряды мантисс оказываются расположенными в одноимённых разрядах обеих ячеек, и теперь уже сложение или вычитание мантисс выполняется достаточно просто, так же как над числами с фиксированной запятой.

После операций над порядками и мантиссами мы получаем порядок и мантиссу результата, но последняя может не удовлетворять ограничениям, накладываемым на мантиссы нормализованных чисел. Так как от результата арифметических операций в компьютере требуется, чтобы он также был нормализованным числом, необходимо дополнительное преобразование – нормализация. В зависимости от величины получившейся мантиссы результата она сдвигается вправо или влево так, чтобы её первая значащая цифра попала в первый разряд после запятой. Одновременно порядок результата увеличивается или уменьшается на число, равное величине сдвига.

Над мантиссами в арифметическом устройстве могут выполняться все четыре арифметические действия, а также операции сдвига, тогда как над порядками производятся только действия сложения и вычитания. Отрицательные порядки можно записывать в дополнительном коде для того, чтобы операцию вычитания свести к операции сложения.

В ряде случаев, даже если некоторые два числа были представлены в формате с плавающей запятой абсолютно точно, результат выполнения над ними арифметической операций может быть заведомо неверным.

У вещественной арифметики есть несколько потенциально опасных особенностей. Вес они имеют общее происхождение, а именно, тот факт, что мантисса и порядок в представлении с плавающей запятой занимают фиксированное число разрядов.

При этом уже на стадии записи чисел в компьютер возникают ошибки округления, которые при выполнении арифметических действий нарастают; наличие погрешностей округления приводит к следующему правилу программирования: неразумно сравнивать в программе два вещественных числа на точное равенство; в результате вычитания возникают недостоверные значащие цифры, которые могут привести к серьёзной потере точности или получению неправильного результата; прибавление или вычитание малого числа может никак не сказаться на результате; получение очень больших чисел может вызвать переполнение, или исчезновение числа (превращение в нуль), это может привести к аварийному завершению программы.

3. ПРЕДСТАВЛЕНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ

Всякий текст состоит из символов – букв, цифр, знаков препинания и т.д., которые человек различает по начертанию. Однако для компьютерного представления текстовой информации такой метод неудобен, а для компьютерной обработки текстов и вовсе неприемлем.

Поскольку текст изначально дискретен – он состоит из отдельных символов, - для компьютерного представления текстовой информации используется другой способ: все символы **кодируются** числами, и текст представляется в виде набора чисел – **кодов символов**, его составляющих. При выводе текста на экран монитора или принтер необходимо восстановит изображения всех символов, составляющих данный текст. Для этого используются кодовые таблицы символов, в которых каждому коду символа ставится в сообщении изображение символа.

Все кодовые таблицы, используемые в любых компьютерах и любых операционных системах, подчиняются международным стандартам кодирования символов. На заре компьютерной эры, когда США были абсолютным лидером в этой области, стандарты разрабатывались Американским национальным институтом стандартизации (*ANSI*); впоследствии для разработки и принятия компьютерных стандартов была создана Международная организация стандартизации (*ISO*).

В программировании наиболее часто используются однобайтовые кодировки: в них код каждого символа занимает ровно 1 байт, или 8 бит. При этом общее количество различаемых символов составляет $2^8=256$, а коды символов имеют значения от 0 до 255.

Информационный объём блока информации называется количеством бит, байт или производных единиц (килобайт, мегабайт), необходимых для записи этого блока путём заранее оговоренного способа двоичного кодирования.

Пример 6. Оценить в байтах объём текстовой информации в Словаре из 740 страниц, если на одной странице размещается в среднем 60 строк по 80 символов (включая пробелы). Будем считать, что при записи используется кодировка «один символ - один байт». Количество символов во всём словаре равно $80 \cdot 60 \cdot 740 = 3552000$. следовательно, объём в байтах равен $3552000 \text{ байт} = 3468,75 \text{ Кбайт} \approx 3,39 \text{ Мбайт}$.

Основой для компьютерных стандартов кодирования послужил *ASCII (American Standard Code for Information Interchange)* – американский стандартный код для обмена информацией, разработанный в 1960-х годах и применяемый в США для любых видов передачи информации, в том числе и некомпьютерных (телеграф, факсимильная связь и т.д.). В нём используется 7-битовое кодирование: общее количество символов составляет $2^7=128$, из них первые 32 символа – управляющие, а остальные – «изображаемые», т.е. имеющие графическое изображение. Управляющие символы должны восприниматься устройством вывода текста как команды, например:

Код	Действие	Английское название
7	Подача стандартного звукового сигнала	Beep
8	Удаление предыдущего символа	Back Space (BS)
13	Перевод строки	Line Feed (LF)
26	Признак «Конец текстового файла»	End Of File (EOF)
27	Отмена предыдущего ввода	Escape (Esc)

К изображаемым символам в *ASCII* относятся буквы английского алфавита (прописные и строчные), цифры, знаки препинания и арифметических операций, скобки и некоторые специальные символы. Хотя в *ASCII* символы кодируются 7 битами, в памяти компьютера под каждый символ отводится ровно 1 байт, при этом код символа помещается в младшие биты, а старший бит не используется. Главный недостаток стандарта *ASCII* заключается в том, что он рассчитан на передачу только английского текста. Со временем возникла необходимость кодирования и неанглийских букв. Во многих странах для этого стали разрабатывать расширения *ASCII*- кодировки, в которых применялись однобайтовые коды символов; при этом первые 128 символов кодовой таблицы совпадали с кодировкой *ASCII*, а остальные (со 128-го по 255-й) использовались для кодирования букв национального алфавита, символов национальной валюты и т.п. Из-за несогласованности этих разработок для многих языков было создано по несколько вариантов кодовых таблиц (например, для русского языка их около десятка!).

КОИ8-Р является стандартом *de facto* для всех служб Интернета, кроме *WWW*. В частности, все службы электронной почты и новостей Рунета работают в этой кодировке. Что касается Веба, то здесь ситуация сложнее. Дело в том, что более 90% клиентских компьютеров Сети работает под управлением *Windows* разных версий. *Windows* использует собственную кодировку русских букв, которую принято называть по номеру кодовой страницы *Windows*-1251 или *CP1251*. Поскольку текстовые редакторы и средства

разработки *HTML*-страниц в *Windows* работают в этой кодировке, абсолютное большинство Веб-документов Рунета хранится в кодировке *Windows-1251*.

Табл. 2. Кодировка текстовой информации в стандарте *ASCII*.

[0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0		☺	☹	♥	♦	♣	♠	●		○							
1	▶	◀		!			_		↑	↓	→	←	↔	^	▼		
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□	
8	Ç	ü	é	â	ã	à	å	ç	ê	ë	è	ï	î	í	Ä	Å	
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	€	£	¥	ℳ	ƒ	
A		í	ó	ú	ñ	ª	º	¿	¬	½	¼	¿	«	»			
B	☐	☐	☐														
C	┌	└	┐	┌	└	┐	┌	└	┐	┌	└	┐	┌	└	┐	┌	└
D	┌	┐	┌	┐	┌	┐	┌	┐	┌	┐	┌	┐	┌	┐	┌	┐	┌
E	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	η	
F	≡	±	≥	≤			÷	≈	°	·	·	√	²	²	■	□	

Кодировка символов, предложенная IBM (соответствует ASCII - кодировке)

Пример 7. Слово *COMPUTER*, закодированное в кодах *ASCII* выглядит так:

C O M P U T E R
067 079 077 080 085 084 069 082

Если учесть, что в двоичном представлении для кодирования каждого символа используется 8-ми разрядный код, то получится двоичный код длиной в 64 символа
1000001110011111001101101000010101101010010001011010010

Впоследствии использование кодовых таблиц было несколько упорядочено: каждой кодовой таблице было присвоено особое название и номер. Указав кодовую таблицу, автоматически выбирают и язык, которым можно пользоваться в дополнение к английскому; точнее, выбирается то, как будут интерпретироваться символы с кодами более 127.

Для русского языка наиболее распространёнными являются однобайтовые кодовые таблицы CP-866 (*Code Page*), *Windows-1251* и КОИ-8. В них первые 128 символов совпадают с *ASCII*-кодировкой, а русские буквы размещены во второй части таблицы, однако коды русских букв в этих кодировках различны! Сравните, например, кодировки КОИ-8 (Код Обмена Информацией 8-битовый, международное название *koi-8r*) и *Windows-1251*, вторые половины которых приведены в **Табл.2** и **3** соответственно.

Несовпадение кодовых таблиц приводит к ряду неприятных эффектов, например, т.к. один и тот же текст имеет различное компьютерное представление в разных кодировках, то текст, набранный в одной кодировке, будет нечитабельным в другой!

Пример 8. Вот так будет выглядеть десятичный код слова «Диск» в разных кодировках:

КОИ-8
Windows-1251
CP-866

228 201 211 203
196 232 241 234
132 168 225 170

Однобайтовые кодировки обладают одним серьёзным ограничением: качество различных кодов символов в этих кодировках недостаточно велико, чтобы можно было пользоваться одновременно несколькими языками. Для устранения этого ограничения в 1993 году был разработан новый стандарт кодирования символов, получивший название *Unicode*, который, по замыслу его разработчиков, позволил бы использовать в текстах любые символы любых языков мира.

Табл. 3. Кодировка КОИ-8 КОИ – Код Обмена Информацией] – 8-разрядный код, ориентированный на обмен сообщениями в Интернет

—		Г	Г	Г	Г	Г	Г	Г	Г	Г	■	■	■	■	■
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
▒	▒	▒	Г	■	●	√	≈	≤	≥	nbsp	↓	◦	²	•	÷
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
=		F	ё	П	Г	Г	П	Г	Г	Г	Г	Г	Г	Г	Г
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
Г	Г	Г	Ё	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	©
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
Ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
п	я	р	с	т	у	ж	в	ь	ы	з	ш	э	щ	ч	ъ
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
П	Я	Р	С	Т	У	Ж	В	Ь	Ы	З	Ш	Э	Щ	Ч	Ъ
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

В *Unicode* на кодирование символов отводится 31 бит. Первые 128 символов (коды 0-127) совпадают с таблицей ASCII; далее размещены основные алфавиты современных языков: они полностью умещаются в первой части таблицы, их коды не превосходят 65536 (216). А в целом стандарт *Unicode* описывает алфавиты всех известных, в том числе и «мёртвых» языков; для языков, имеющих несколько алфавитов (например, японский и индийский), закодированы все варианты; в кодировку *Unicode* внесены все математические и иные научные символные обозначения и даже некоторые придуманные языки (например, письменности эльфов по Р.Р.Толкиену). потенциальная информационная ёмкость 31-битового *Unicode* столь велика, что используется менее одной тысячной части возможных кодов символов!

В современных компьютерах и операционных системах используется укороченная, 16-битовая версия *Unicode*, в которую входят все современные алфавиты; эта часть *Unicode* называется базовой многоязыковой страницей (*Base Multilingual Plane, BMP*). *UNICODE* – 16 – разрядная система кодирования (65536 символов) охватывает символы всех языков (включая языки, использующие иероглифы, например, китайский и японский). Стандарт *Unicode* 3.2 поддерживает следующие языки народов России с дополнительными кириллическими буквами: алтайский, башкирский, бурятский, долганский, калмыцкий, коми, корякский, марийский, нанайский, ненецкий, осетинский, саамский (без указания долготы гласных), татарский, тувинский, удмуртский, хакасский, хантыйский, чувашский, эвенкийский, эвенский, якутский, кавказские языки с буквой «палочка».

В *UNIX*-подобных операционных системах, где работа с *Unicode*-текстами невозможна из-за особенностей архитектуры, используются особые формы этого стандарта, которые называются *UTF* (*Unicode Transformation Form*), в них символы кодируются переменным количеством битов. Например, в *UTF-8* коды символов занимают от 1 до 6 байтов.

Табл. 4. Кодировка Windows-1251.

Character Map																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□
8	Ъ	Ґ	,	ѓ	„	…	†	‡	€	‰	Љ	‹	Њ	Ќ	Ѓ	Ѕ
9	ђ	‘	’	“	”	•	–	—	□	™	љ	›	њ	ќ	ћ	џ
A		Ÿ	ÿ	Ј	#	Ѓ	;	§	Ё	©	Є	«	¬	-	®	İ
B	°	±	ı	ı	г	μ	¶	·	ё	№	є	»	ј	ѕ	ѕ	ї
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Табл. 5. Дополнительные буквы языков народов России, поддерживаемые стандартом Unicode 3.2

	0	040	041	042	043	044	045	046	047	048	049	04A	04B	04C	04D	04E	04F
0	Ё	А	Р	а	р	ё	Ѡ	Ѳ	Ѵ	Г	К	Ѹ	І	Ă	З	Ÿ	
1	Ё	Б	С	б	с	ё	ѡ	ѳ	ѵ	Г	К	ѹ	Ѓ	ă	з	ÿ	
2	Ђ	В	Т	в	т	ђ	Ђ	Ѧ	Ѣ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ÿ	
3	Ѓ	Г	У	г	у	ѓ	Ђ	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ÿ		
4	Є	Д	Ф	д	ф	є	Є	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
5	Є	Е	Х	е	х	є	Є	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
6	І	Ж	Ц	ж	ц	і	І	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
7	İ	З	Ч	з	ч	ı	І	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
8	Ј	И	Ш	и	ш	ј	Ј	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
9	Љ	Й	Щ	й	щ	љ	Љ	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
A	Њ	К	Ъ	к	ъ	њ	Њ	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
B	Ѓ	Л	Ы	л	ы	ѓ	Ѓ	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
C	Ќ	М	Ь	м	ь	ќ	Ќ	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
D	Ѕ	Н	Э	н	э	ѕ	Ѕ	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
E	Ї	О	Ю	о	ю	ї	Ї	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		
F	Ѡ	П	Я	п	я	ѡ	Ѡ	Ѧ	Ѣ	Ѣ	Ѣ	Ђ	Ă	Й	Ч		

С точки зрения компьютера текст состоит из отдельных символов. К числу символов принадлежат не только буквы (заглавные или строчные, латинские или русские), но и цифры, знаки препинания, спецсимволы типа "=", "(", "&" и т.п. и даже пробелы между словами: пустое место в тексте тоже должно иметь свое обозначение. При нажатии клавиши клавиатуры сигнал посылается в компьютер в виде двоичного числа, которое хранится в кодовой таблице. Кодовая таблица - это внутреннее представление символов в компьютере. Например, буква S имеет код 01010011; при нажатии ее на клавиатуре происходит декодирование двоичного кода и по нему строится изображение символа на экране монитора. Каждый символ хранится в виде двоичного кода, который является номером символа. Можно сказать, что компьютер

имеет собственный алфавит, где весь набор символов строго упорядочен. Количество символов в алфавите также тесно связано с двоичным представлением и у всех ЭВМ равняется 256. Иными словами, каждый символ всегда кодируется 8 битами, т.е. занимает ровно один байт.

В компьютере хранится не начертание буквы, а её номер. Именно по этому номеру воспроизводится вид символа на экране дисплея или на бумаге. Поскольку алфавиты в различных типах ЭВМ не полностью совпадают, при переносе с одной модели на другую может произойти превращение разумного текста в «абракадабру». Такой эффект иногда получается даже на одной машине в различных программных средах: например, русский текст, набранный в *MS DOS*, нельзя без специального преобразования прочитать в *Windows*. Остается утешать себя тем, что задача перекодировки текста из одной кодовой таблицы в другую довольно проста и при наличии программ машина сама великолепно с ней справляется.

Интересно, что каждый символ текста имеет свой числовой код, но не каждому коду соответствует отображаемый на экране символ. Речь идет об управляющих символах, величина которых меньше шестнадцатиричного числа 20 (т.е. 32 в десятичной системе счисления). При получении этих кодов внешние устройства не изображают какого-либо символа, а выполняют те или иные управляющие действия. Так, код 07 вызывает подачу стандартного звукового сигнала, а код 0C - очистку экрана. Особую роль играют коды 0A (перевод строки, обозначаемый часто *LF*) и 0D (возврат каретки - *CR*). Первый вызывает перемещение в следующую строку без изменения позиции, а второй - на начало текущей строки. Таким образом, для перехода на начало новой строки требуются оба кода и в любом тексте эта «неразлучная пара» кодов хранится после каждой строки.

4. ПРЕДСТАВЛЕНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Подавляющая часть информации об окружающем мире человек получает с помощью зрения – доля зрительной информации составляет 80% общего потока информации. Важность зрения обусловлена историческим развитием человека как биологического вида, поэтому зрительные органы человека, и особенно зрительные центры мозга, прекрасно приспособлены к обработке информации с большой скоростью и в больших объёмах. С появлением компьютеров, способных быстро обрабатывать информацию, началась разработка компьютерных методов хранения и обработки изображений.

Рассмотрим теперь некоторые методы кодирования изображений

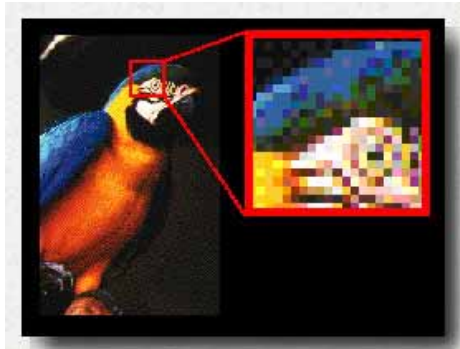
Под компьютерной (машинной) графикой понимается совокупность методов и приёмов преобразования при помощи компьютера данных в графическое представления. Изображения могут быть: штриховые или полутоновые, чёрно-белые или цветные. Различают три вида компьютерной графики: растровая графика; векторная графика; фрактальная графика.

Для хранения и обработки графической и звуковой информации в компьютере требуются значительные вычислительные ресурсы (память и процессорное время), и, кроме того, обрабатываемая информации естественного происхождения должна быть представлена в специальном компьютерном виде. Главная проблема разработки такого представления заключается в том, что компьютер может обрабатывать и хранить только ограниченный объём информации, в то время как любые естественные сигналы – носители информации, - непрерывны и неограниченны в пространстве и времени.

Для преобразования естественной информации в дискретную форму её подвергают **дискретизации и квантованию**.

Дискретизация – процедура устранения временной и/или пространственной непрерывности естественных сигналов, являющихся носителями информации.

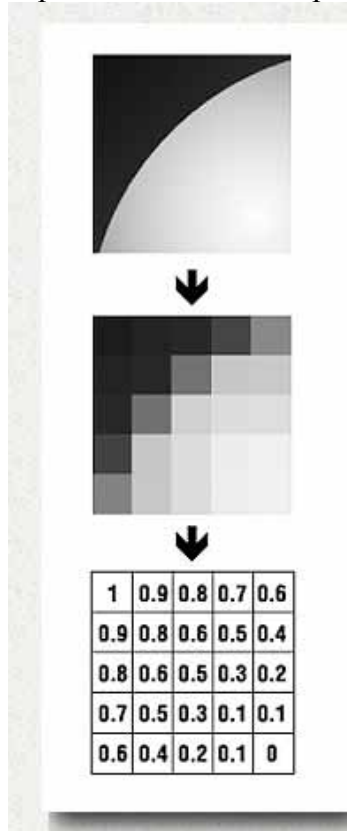
Рис. 2. Пространственная дискретизация изображения.



До того, как изображения можно будет обрабатывать на компьютере, их оцифровывают (преобразуют в электрические сигналы, представляющие нули и единицы, Например, белый цвет преобразуется в «1», а черный в «0»). Этот процесс называется выборка и дискретизация. Минимальный элемент оцифрованного изображения называется пиксел (picture element - элемент картины). Цифровая картина представляет собой набор пикселов. При **пространственной дискретизации** изображения его разбивают на небольшие области, в пределах которых характеристики изображения считают неизменными. С

информационной точки зрения графическое изображение является совокупностью световых сигналов на плоскости: отдельные световые сигналы различаются местоположением, цветовым оттенком и яркостью. Цвет и яркость – характеристики точек изображения, их можно измерять, т.е. выразить в числах. Как цвет, так и яркость могут измеряться непрерывно, поэтому их следовало бы выразить вещественными числами. Но в этом случае их невозможно абсолютно точно представить в компьютере. Поэтому все измеряемые непрерывные характеристики (например, яркость точек изображения) подвергают квантованию.

Рис. 3. Квантование чёрно-белого изображения.



Квантование – процедура преобразования непрерывного диапазона всех возможных входных значений измеряемой величины в дискретный набор выходных значений.

При квантовании диапазон возможных значений измеряемой величины разбивается на несколько диапазонов. При измерении определяется поддиапазон, в который попадает значение, и в компьютере сохраняется только номер поддиапазона.

Пример 9. При квантовании шкалы оттенков серого цвета надо каждому оттенку поставить в соответствие некоторое число. Чёрному цвету приписывают нулевой уровень яркости, белому – единственный, а промежуточным серым тонам – дробные числа в интервале от 0 до 1, выражающие яркость оттенка как долю от максимальной яркости. Эту величину выражают в процентах от максимальной яркости, т.е. чёрный имеет 0% яркости, белый – 100%. Такое описание цвета непригодно для непосредственного компьютерного представления, так как вещественные числа недискретны, для точной записи их значений необходимо хранить бесконечное число цифр! Следовательно, надо провести квантование, для чего диапазон значений яркости разбивают на поддиапазоны, или уровни. В примере диапазон яркости серого цвета разбиваем на 6 уровней разной ширины: 0-й уровень – 0-17% максимальной яркости, 1-й уровень – 0-17% максимальной яркости и т.п.

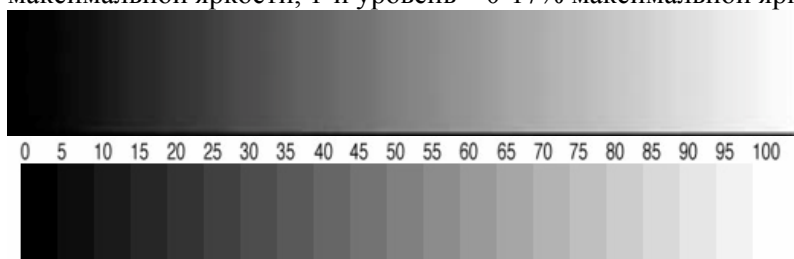


Рис. 4. Шкалы градаций серого цвета: а) непрерывная шкала; б) дискретная шкала

Тогда интервалы значений яркости и номера уровней связаны друг с другом простым соотношением: k -й уровень соответствует интервалу: $\left[k \cdot \frac{100}{6}; (k+1) \cdot \frac{100}{6} \right]$. По величине яркости легко вычислить номер соответствующего уровня: $k = [x \cdot 0,06]$, здесь квадратные скобки означают целую часть числа, x – величина яркости, k – номер уровня.

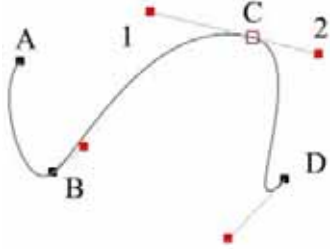
Пусть яркость серого оттенка составляет 70%. При квантовании $\left(\left[\frac{70}{100} \cdot 6 \right] = \left[\frac{420}{100} \right] = 4 \right)$ это значение попадает в 4-й поддиапазон (67-83%), поэтому в компьютере этот оттенок серого будет закодирован целым числом 4.

Дискретизация и квантование всегда приводят к потере некоторой доли информации. Так, компьютерное изображение живописного полотна всегда отличается от оригинала. Из-за больших размеров графических файлов они редко хранятся в компьютере в неупакованном виде. Для уменьшения их размеров используют сжатие информации.

При пространственной дискретизации изображений пользуются растровым и векторным представлением графической информации. Растровое представление можно охарактеризовать как поточечное представление, а векторное – как структурное представление изображения.

4.1 Векторное представление

Рис. 5. Создание рисунка из отдельных элементов.



Векторное представление описывает, как построить исходное изображение при помощи стандартных геометрических фигур из заранее определённого набора, например, из отрезков и дуг. Оно использует числовые формулы, дающие математическое описание объектов (геометрических фигур или линий, кроме того оно обычно многослойно). Когда векторные изображения пересылаются на монитор или принтер, то они преобразуются в пиксели в соответствии с размерами выводимого изображения. Геометрические фигуры из стандартного набора называются элементарными фигурами, или примитивами. Построение векторного представления называется векторизацией изображения. При выполнении векторизации изображение анализируют, разбивают на примитивы, а затем сохраняют их параметры: положение, размеры и цвет.

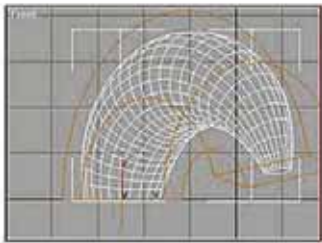


Рис. 6. Создание чертежа из отдельных линий.

Каждый элемент векторного изображения является объектом, который описывается с помощью специального языка (математические уравнения линий, дуг, окружностей и т.д.) и располагается в своём собственном слое. В основе векторной графики лежат математические представления о свойствах геометрических фигур. Основные графические примитивы: точка (задаётся двумя числами (x,y)), прямая линия (описывается уравнением $y=kx+b$), отрезок прямой (координаты начала и конца отрезка), кривая второго порядка ($x^2+a_1y^2+a_2xy+a_3x+a_4y+a_5=0$), кривая третьего порядка, кривые Безье.



Многие виды изображений по своей природе хорошо структурированы и поэтому удобны для векторизации: это графики, диаграммы, чертежи, схемы, плакаты, карты, символы, гербы и флаги, логотипы, всевозможные стилизованные изображения.

Рис. 7. Геометрические фигуры – чёрно-белые примитивы.

Все объекты имеют атрибуты (свойства). К этим свойствам относятся: форма линии, её толщина, цвет, характер линии (сплошная, пунктирная и т.п.). Объекты могут группироваться в слои с общими характеристиками. Количество цветов, в отличие от растровой графики, на размер файла практически не влияет. Файлы векторной графики способны содержать растровые изображения в качестве одного из типов объектов.

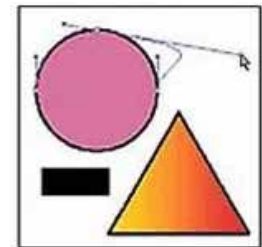


Рис. 8. Цветные примитивы.

Пример 10. Любой чертёж содержит отрезки, окружности, дуги. Положение каждого отрезка на чертеже можно задать координатами двух точек, определяющими его начало и конец; окружность – координатами центра и длиной радиуса; дугу – радиусом, а также координатами начала и конца дуги. Кроме того, для каждой линии можно указать её тип: тонкая, штрихпунктирная и т.д. Такая информация вводится в компьютер как обычная буквенно-цифровая и обрабатывается в дальнейшем специальными программами.

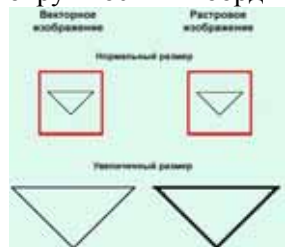
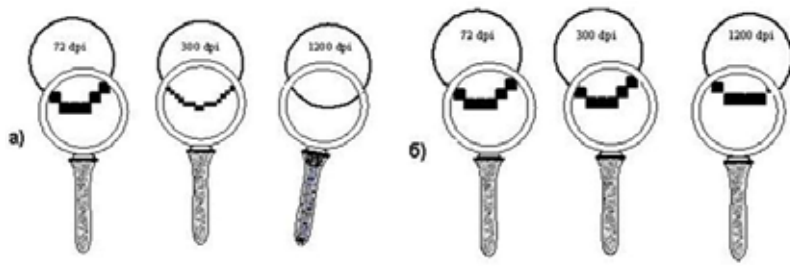


Рис. 9. Сравнение влияния увеличения на изображение при векторном и растровом подходе.



Печать окружности на принтерах с различным разрешением:
а векторная, б растровая.

Достоинства векторной графики: Объекты векторного изображения в отличие от растровой графики, могут изменять свои размеры без потери качества (при увеличении растрового изображения увеличивается зернистость).

Она использует все преимущества разрешающей способности любого устройства вывода, что позволяет изменять размеры векторного рисунка без потерь его качества. Векторные команды просто сообщают

устройству вывода, что необходимо нарисовать объект заданного размера, используя столько точек, сколько возможно. Векторная графика позволяет редактировать отдельные части рисунка, не оказывая влияния на остальные (в растровых изображениях пришлось бы редактировать каждый пиксел).

Векторные изображения, не содержащие растровых объектов. Занимают в памяти компьютера относительно небольшое место (в 10 – 1000 раз меньше, чем его растровый аналог). Векторные изображения позволяют описывать трёхмерные (объёмные) фигуры.

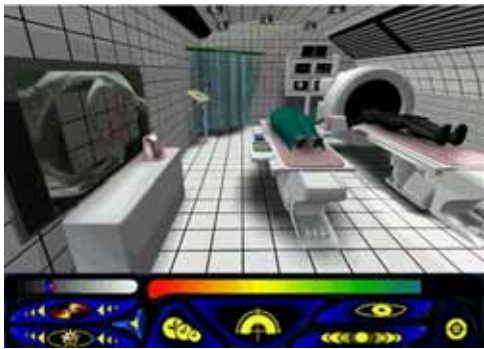


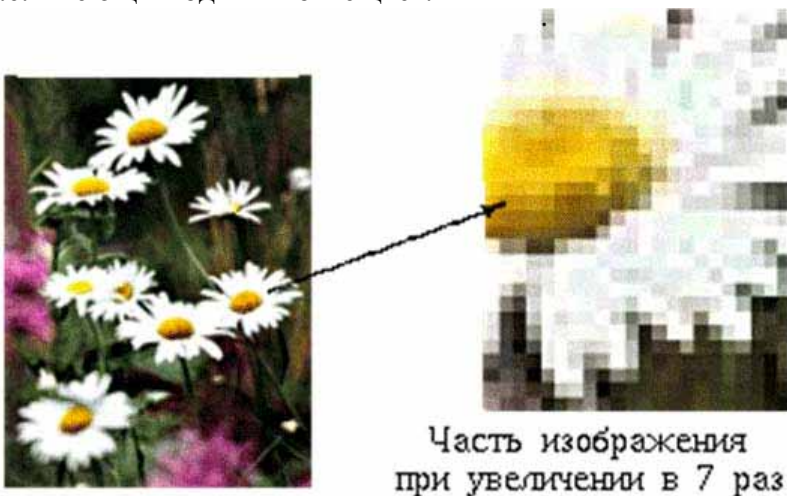
Рис. 10. Некая искусственность векторного изображения.

Недостатки векторной графики: Рисунки часто выглядят достаточно искусственно, так как основным компонентом векторного рисунка является прямая линия, а она в природе

встречается редко. Возможны проблемы печати, как правило, сложных рисунков на отдельных типах принтеров из-за того, что не все команды могут ими правильно интерпретироваться.

4.2 Растровое представление

В отличие от хорошо структурированных изображений существуют изображения, которые вообще не имеют чёткой структуры. К ним относятся фотографии, живописные полотна, рукописные тексты и т.д. Такие изображения крайне неудобны для векторизации. Для хранения подобных изображений используют **растровое представление**: всё изображение разбивается на множество очень маленьких элементов, причём, в отличие от векторного представления, размеры и положения элементов задаются заранее и совершенно не зависят от самого изображения. В пределах каждого такого элемента изображение считается однородным, т.е. имеющим один и то же цвет.



Часть изображения при увеличении в 7 раз

Рис. 11. Хранение и воссоздание растрового изображения

Порядок разбиения изображения на элементы называется **растром**, а сами элементы – пикселями. **Пиксели** – атомы растрового изображения, на меньшие части его не делят. Пиксели очень малы и их отождествляют с точками. Процедура разбиения изображения на пиксели называется растеризацией или оцифровкой изображения.

Хранение рисунка в векторной форме обычно на несколько порядков сокращает необходимый объём памяти по сравнению с растровой формой представления.

Растр – специальным образом организованная совокупность пикселей, представляющая изображение. Координаты, форма и размер пикселей задаются при определении растра. Изменяемым атрибутом является цвет.

Растровая графика является универсальным способом кодирования как полутоновых так и штриховых изображений. Это способ кодирования изображения, при котором оно представляется в виде матрицы элементов (*bitmap*). Элементы матрицы называются пиксель (*pixels*) – сокращение от *picture*

elements. Компьютер запоминает цвета всех пикселей подряд в определённом порядке. Поэтому растровые изображения требуют для хранения большой объём памяти.

В технике и компьютерной графике чаще всего используется прямоугольный растр, в котором пиксели составляют прямоугольную матрицу, её основными параметрами являются размеры растровой матрицы, т.е. количество столбцов и строк, составленных из пикселей. Главное преимущество прямоугольных растров заключается в том, что положение каждого пикселя на экране (или изображении) можно вычислить, зная только размеры растровой матрицы и линейные размеры пикселей либо плотность размещения пикселей, которую обычно измеряют в количестве точек на дюйм (*dpi, Dots Per Inch*). Для этого достаточно ввести правила перечисления пикселей. Например, в мониторах персональных компьютеров пиксели перечисляются слева направо и сверху вниз: сперва нумеруются все пиксели в верхней строке слева направо, затем нумерация продолжается на нижележащей строке и т.д.

Пример 11. Фотография сохранена в формате *JPEG* с размерами 768x576. Это значит, что матрица пикселей состоит из 768 столбцов и 576 строк. Операционные системы с графическим пользовательским интерфейсом (*Windows, MacOS*, графическая подсистема *X Windows* в *UNIX* и т.д.) представляют экран дисплея как растровое прямоугольное изображение некоторого размера (например, 800x600 или 1024x768 пикселей).

Квантование (кодирование) цвета базируется на математическом описании цвета, которое, в свою очередь, опирается на тот факт, что цвета можно измерять и сравнивать. Научная дисциплина, изучающая вопросы измерения цветовых характеристик, называется метрологией цвета, колориметрией. Человек обладает очень сложным цветовосприятием, метрологией цвета, или колориметрией. Человек обладает очень сложным цветовосприятием, достаточно заметить, что зрительные центры мозга у новорождённых детей в течение нескольких месяцев только тренируются видеть. Поэтому и математическое описание цвета тоже весьма нетривиально.

Законы Грассмана:

Закон трёхмерности: с помощью трёх линейно независимых цветов можно однозначно выразить любой цвет. Цвета считаются линейно независимыми, если никакой из них нельзя получить путём смешения остальных.

Закон непрерывности: при непрерывном изменении состава цветовой смеси регулирующий цвет также меняется непрерывно. К любому цвету можно подобрать бесконечно близкий цвет.

В компьютерной графике существуют два противоположных метода его описания: системы аддитивных и субтрактивных цветов.

Аддитивный цвет получается при соединении лучей света различных цветов. В этой системе используются три основных цвета: красный (*red*), зелёный (*green*) и синий (*blue*), (*RGB*). Смешивая их в разных пропорциях можно получить любой цвет. В системе субтрактивных цветов происходит обратный процесс: цвет получается, вычитая другие цвета из общего луча света. В системе субтрактивных цветов основными являются голубой (*cyan*), пурпурный (*magenta*) и жёлтый (*yellow*) цвета, (*CMY*) противоположны красному, зелёному и синему. Применяются так же *CMYK* (*Cyan-Magenta-Yellow-black*, голубой-пурпурный – жёлтый-чёрный); *HSB* (*Hue-Saturatin-Brightness*, цветовой оттенок – насыщенность-яркость).

Яркость – характеристика цвета, определение которой в основном совпадает с бытовым понятием яркости и физическим понятием освещённости или светимости. Ярко-красный и тёмно-красный цвета различаются именно яркостью. С физической точки зрения, яркость – количественная мера потока световой энергии, излучаемой или отражаемой предметом в сторону наблюдателя. Так, при ярком солнечном свете и в сумерках один и тот же цветной рисунок выглядит по-разному. При этом цветовые оттенки не меняются, различными оказываются лишь яркости цветов. Цветовой оттенок и насыщенность – две другие независимые характеристики цвета. Пусть у нас есть набор красок разного цвета. Смешением различных красок между собой мы будем получать новые цвета. Например, смесь равного количества жёлтой и синей красок даст зелёную краску. Цветовой оттенок, или цветовой тон рассматриваемого объекта связан со спектральным составом излучения. По цветовому тону объекта мы можем судить об окраске объекта – синей, зелёной, красной и т.д. Отдельные участки видимого спектра вызывают ощущения различных цветов.

Насыщенность характеризует степень разбавления цветового тона белым цветом. Например, если яркокрасную (насыщенную) краску разбавить белой, то её цветовой оттенок останется прежним, изменится только насыщенность. Ровно так же коричневый цвет, жёлтый и лимонный имеют один и тот же цветовой оттенок – жёлтый, их различие заключается в насыщенности цветового оттенка. Наибольшей насыщенностью обладает свет от монохромного источника. Отметим, что для белого и чёрного цветов насыщенность составляет 0%, т.е. эти цвета не обладают насыщенностью. Именно поэтому, подмешивая их к цветной краске, мы меняем её насыщенность, а не оттенок.

Важной характеристикой является количество цветов, закодированных в файле. Цвет каждого пиксела кодируется определённым числом бит, эта характеристика называется **глубиной цвета**. Если для кодировки отвести лишь один бит, то каждый пиксел может быть либо белым (значение 1), либо чёрным (значение 0). Такое изображение называют монохромным (monochrome). 8-бит – 256 различных цветов или серого цвета (полутоновое). 16-бит – 65536 различных цветов (*high color*). 24 бита – $2^{24}=16777216$ различных цветов и оттенков (*True Color*).

Достоинства растровой графики. Растровые изображения выглядят вполне реалистично. Это связано со свойствами человеческого глаза: он приспособлен для восприятия реального мира как огромного набора дискретных элементов, образующих предметы. Легко управлять выводом изображения на устройства, представляющие изображения в виде совокупности точек принтеры, фотонаборные автоматы.

Недостатки растровой графики. Большой объём памяти, требуемый для хранения изображения хорошего качества. Трудности редактирования изображений. Так как сами изображения занимают много памяти компьютера, то, очевидно, и для их редактирования потребуется также много памяти. Кроме того, применение фильтров специальных эффектов к таким изображениям может занять от нескольких минут до часа в зависимости от используемого оборудования.

4.3 Цветовые модели

4.3.1 RGB

RGB (*Red, Green, Blue* - **красный, зелёный, синий**) - аддитивная цветовая модель, описывающая способ синтеза цвета для цветовоспроизведения. Выбор основных цветов обусловлен особенностями физиологии восприятия цвета сетчаткой человеческого глаза. Цветовая модель RGB нашла широкое применение в технике. Аддитивной она называется потому, что цвета получаются путём добавления к чёрному. Иначе говоря, если цвет экрана, освещённого цветным прожектором, обозначается в *RGB* как (r_1, g_1, b_1) , а цвет того же экрана, освещённого другим прожектором, - (r_2, g_2, b_2) , то при освещении двумя прожекторами цвет экрана будет обозначаться как $(r_1+r_2, g_1+g_2, b_1+b_2)$. Изображение в данной цветовой модели состоит из трёх каналов. При смешении основных цветов (основными цветами считаются красный, зелёный и синий) - например, синего (*B*) и красного (*R*), мы получаем пурпурный (*M magenta*), при смешении зеленого (*G*) и красного (*R*) - жёлтый (*Y yellow*), при смешении зеленого (*G*) и синего (*B*) - циановый (*C cyan*). При смешении всех трёх цветовых компонентов мы получаем белый цвет (*W*).

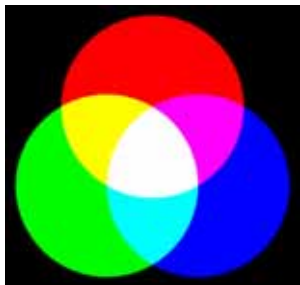
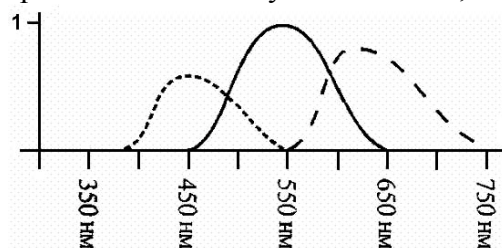


Рис. 12. Аддитивное смешение цветов.

Данная модель используется при отображении графических на экране монитора, телевизора, сотового телефона и т.д. В телевизорах и мониторах применяются три электронные пушки (светодиода, светофильтра) для красного, зелёного и синего каналов.

Смешением трёх основных цветов синтезируются все остальные цвета, их условные яркости (интенсивности) задаются вещественными числами от 0 до 1 (значение 1 соответствует максимальной яркости соответствующего цвета, которую может изобразить графическое устройство). Модель RGB



определяет пространство цветов и виде единичного куба с осями «яркость красной компоненты», «яркость зелёной компоненты» и «яркость синей компоненты».

Рис. 13. Относительная спектральная чувствительность человеческого глаза.

У цветовой модели RGB есть физиологическое обоснование. Человеческий глаз содержит четыре типа зрительных рецепторов: «палочки» (рецепторы интенсивности) и три типа «колбочек» (рецепторы цветовых оттенков). Колбочки каждого типа чувствительны к свету в своём узком диапазоне длин волн, для колбочек разных типов максимумы чувствительности приходятся на разные длины волн, диапазоны чувствительности частично перекрываются:

Тип колбочек	Диапазон длин волн	Максимум чувствительности
Красные	от 760 до 550 нм	~610 нм
Зеленые	от 650 до 450 нм	~550 нм
Синие	от 550 до 380 нм	~450 нм
Общий спектр видимого света	от 760 до 380 нм	555 нм (дневное зрение) 510 нм (ночное зрение)

Именно благодаря неравномерной спектральной чувствительности и перекрытию

диапазонов чувствительности человеческих глаз способен различать огромное количество цветов (10 миллионов). Если направить в глаз световой сигнал с правильно подобранным соотношением яркостей красного, зелёного и синего цветов, то зрительные центры мозга не смогут отличить подмену и сделают вывод, что наблюдается нужный цвет! Такой механизм синтеза цветовых оттенков используется во всех современных типах цветных мониторов, телевизоров, дисплеев сотовых телефонов.

Чтобы использовать математическую *RGB*-модель для реального компьютерного представления графической информации, необходимо произвести квантование цветового пространства, т.е. найти способ представлять вещественные значения яркостей цветовых компонент в дискретной форме. Наиболее простой способ добиться этого – перевести вещественные числа из интервала $[0;1]$ в интервал целых чисел от 0 до $N-1$ путём умножения на целое число N , с последующим округлением. Фактически, интервал $[0;1]$ разбивается на N равных подинтегралов вида:

$$\left[\frac{i}{N}; \frac{i+1}{N} \right), i \in \{0, \dots, N - 1\}.$$

Разбиению на подинтегралы подвергают каждую из цветовых осей. Количество подинтегралов на «красной», «зелёной» и «синей» осях (N_r, N_g, N_b) может быть различным, но чаще принимается, что $N_r=N_g=N_b=N$. После квантования каждый цвет представляется триадой целых неотрицательных чисел (k_r, k_g, k_b), $0 \leq k_i < N_i$. Числа N_i обычно выбирают равными степени двойки $N_i=2^{m_i}$, а величину M , равную сумме $m_r+m_g+m_b$, называют глубиной цвета или глубиной цветности.

Табл. 6. Наиболее распространённые видеорежимы (приведены количества отображаемых цветов).

Видеорежим	Глубина цвета	Количество отображаемых цветов
256 цветов	8	$2^8 = 256$
High Color	16	$2^{16} = 65\,536$
True Color	24	$2^{24} = 16\,777\,216$

Пример 12. В современных компьютерах в видеорежиме TrueColor на хранение информации об одной цветовой компоненте используется 1 байт, для сохранения цвета точки – 3 байта: $m_r=m_g=m_b$, $N_r=N_g=N_b=2^8=256$, $N_r \cdot N_g \cdot N_b=28+8+8=16777216$.

Глубины цветности 24 бита для мониторов вполне достаточно, чтобы создать видимость непрерывности шкалы цветовых оттенков. Особенности человеческого зрения таковы, что если на экране монитора изобразить две фигуры, цвета которых при глубине цвета 24 бита отличаются более чем на 1 в каждый цветовой компоненте, то человек не сможет заметить разницу.

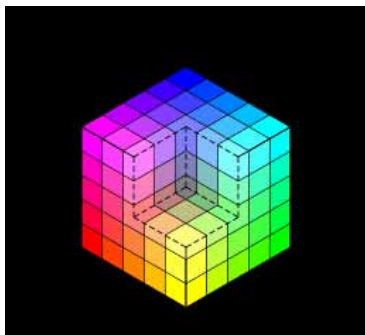


Рис. 14. *RGB*-цветовая модель представленная в виде куба

Для большинства приложений значения координат r, g и b можно считать принадлежащими отрезку $[0,1]$, что представляет пространство *RGB* в виде куба $1 \times 1 \times 1$. В компьютерах для представления каждой из координат традиционно используется один октет, значения которого обозначаются для удобства целыми числами от 0 до 255 включительно. Следует учитывать, что чаще всего используется гамма-компенсированное цветовое пространство *sRGB*, обычно с показателем 1.8 (Mac) или 2.2 (PC).

Пример 13. В видеорежиме *HighColor* цвет каждой точки кодируется 16 битами. На глубину красного и синего цвета отводится 5 бит, на глубину зелёного – 6 бит: $m_r=m_b=5, m_g=6$. Следовательно, шкала яркостей зелёного цвета содержит в два раза больше уровней, чем шкалы яркостей красного и синего цветов. Для экономии памяти биты цветовых компонент каждой точки записывают в два байта вместо трёх.

Цветовая модель *RGB* соответствует механизму синтеза цветов, используемому в мониторах. А поскольку в современных компьютерах именно мониторы являются главным, наиболее часто используемым устройством вывода информации, то практически все форматы графических файлов хранят изображения в *RGB*-представлении, и лишь очень немногие графические форматы используют другие цветовые модели.

Табл. 7. Двоичные значения уровней интенсивности некоторых цветов для стандарта *TrueColor*.

Название цвета	Интенсивность основных цветов		
	Красный	Зеленый	Синий
черный	00000000	00000000	00000000
красный	11111111	00000000	00000000
зеленый	00000000	11111111	00000000
синий	00000000	00000000	11111111
голубой	00000000	11111111	11111111
пурпурный	11111111	00000000	11111111
желтый	11111111	11111111	00000000
белый	11111111	11111111	11111111

Цветовая модель *RGB* имеет по многим тонам цвета более широкий цветовой охват (может представить более насыщенные цвета), чем типичный охват цветов *CMYK*, поэтому иногда изображения, замечательно выглядящие в *RGB*, значительно тускнеют и гаснут в *CMYK*.

4.3.2 CMYK

Цветовая модель *CMYK* также базируется на трёхкомпонентной теории цвета, но в отличие от модели *RGB*, основными цветами в ней являются голубой, пурпурный и жёлтый. Модель *CMYK* широко используется в цветной печати.

Четырёхцветная автотипия (*CMYK*: *Cyan*, *Magenta*, *Yellow*, *Key color*) — субтрактивная схема формирования цвета, используемая прежде всего в полиграфии для стандартной триадной печати. Схема *CMYK*, как правило, обладает сравнительно небольшим цветовым охватом. По-русски эти цвета часто называют так: голубой, пурпурный, жёлтый; но профессионалы подразумевают *cyan*, *magenta* и *yellow* (о значении *K* см. далее). Печать четырьмя красками, соответствующими *CMYK*, также называют печатью *триадными красками*.

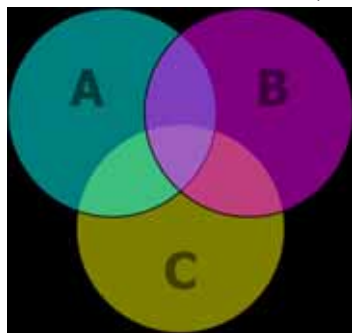


Рис. 15. Схема субтрактивного синтеза в *CMYK*.

Модель *CMYK* применяется в цветных принтерах общего назначения и в цветной офсетной печати низкого и среднего качества. Если рассмотреть под микроскопом цветные иллюстрации в какой-нибудь книге или цветной газете, то можно видеть, что цветные фрагменты напечатаны очень маленькими частично перекрывающимися цветными точками (офсетам). Офсетсы хорошо заметны на границах цветных участков и в местах с бледной окраской.

В *CMYK* используются четыре цвета, первые три в аббревиатуре названы по первой букве цвета, а в качестве четвёртого используется чёрный (*K* - сокращение от *black*). Дело в том, что при выводе полиграфических плёнок на них одной буквой указывался цвет, которому они принадлежат. Чёрный не стали обозначать *B*, чтобы не путать с *B* (*blue*) из модели *RGB*, а стали обозначать *K* (по последней букве).

Причиной появления модели *CMYK* является различие в принципах формирования цвета при его воспроизведении на мониторах и при печати. При восприятии цвета с экрана монитора мы видим излучаемый свет, а при рассматривании картинке, нарисованной на бумаге, - отражённый. Пиксели монитора излучают собственный свет; чтобы создать на экране основной цвет, надо включить субэлемент определённого типа (пиксель монитора состоит из трёх субэлементов: красного, зелёного и синего), а для получения составного цвета надо дополнительно включить (т.е. добавить) субэлементы другого типа, при этом суммарная яркость пикселя возрастает. Кстати, из-за такого принципа формирования составного цвета *RGB*-модель называют аддитивной цветовой моделью. В отличие от монитора, бумага отражает падающий свет, который обычно является «белым»: яркости всех его цветовых составляющих равны. Наносимые на бумагу краски являются поглощающими светофильтрами – они поглощают лучи определённого цвета, а остальные отражают. Видимый цвет краски определяется теми лучами, которые не были поглощены. Таким образом, краски могут только вычитать, или ослаблять цвета в отражаемом потоке света. По этой причине модель *CMYK* называют субтрактивной цветовой моделью. Основные цвета модели *CMYK* подобраны так, чтобы соответствующие краски поглощали свет в достаточно узкой области света: голубая краска сильно поглощает красный свет, пурпурная – зелёный, а жёлтая – синий.

Так как модель *CMYK* применяют в основном в полиграфии при цветной печати, а бумага и прочие печатные материалы являются поверхностями, отражающими свет, удобнее считать, какое количество света (и цвета) отразилось от той или иной поверхности, нежели сколько поглотилось. Таким образом, если вычесть из белого три

первичных цвета, *RGB*, мы получим тройку дополнительных цветов *CMY*. «Субтрактивный» означает «вычитаемый» — из белого вычитаются первичные цвета.

В идеальном случае трёх цветов (голубого, пурпурного и жёлтого) было бы вполне достаточно для формирования на бумаге любого цвета. Однако реально существующие краски не идеальны, они не поглощают цветовые компоненты полностью: если нанести на бумагу все три краски с наибольшей плотностью, то вместо чистого чёрного цвета получится тёмно-серый. Для коррекции цветовой гаммы используется четвёртая краска - чёрная.

Несмотря на то, что чёрный цвет можно получать смешением в равной пропорции пурпурного, голубого и жёлтого красителей, по ряду причин (чистота цвета, переувлажнение бумаги и др.) такой подход обычно неудовлетворителен. Основные причины использования дополнительного чёрного пигмента таковы: На практике смешение реальных пурпурного, голубого и жёлтого цветов даёт скорее грязно-коричневый или грязно-серый цвет; триадные краски не дают той глубины и насыщенности, которая достигается использованием настоящего чёрного. Так как чистота и насыщенность чёрного цвета чрезвычайно важна в печатном процессе, в печатный процесс был введён ещё один цвет. При выводе мелких чёрных деталей изображения или текста без использования чёрного пигмента возрастает риск неприводки (недостаточно точное совпадение точек нанесения) пурпурного, голубого и жёлтого цветов. Увеличение же точности печатающего аппарата требует неадекватных затрат. Смешение 100 % пурпурного, голубого и жёлтого пигментов в одной точке в случае струйной печати существенно смачивает бумагу, деформирует её и увеличивает время просушки. Чёрный пигмент (в качестве которого, как правило, используется сажа) существенно дешевле остальных трёх.

Пространство цветов модели *CMYK* также является единичным кубом. Яркости основных красок (или плотность закрашки) задаются вещественными числами от 0 до 1.

Квантование цвета в модели *CMYK* выполняется аналогично квантованию в модели *RGB*.

При печати в *CMYK* изображение растрируется, т. е. представляется в виде совокупности точек цветов *C*, *M*, *Y* и *K*. На расстоянии точки, расположенные близко друг к другу, сливаются, и создаётся ощущение, что цвета накладываются друг на друга. Глаз смешивает их и таким образом получает необходимый оттенок. Растрирование выделяют амплитудное (наиболее часто используемое, при котором, количество точек неизменно, но различается их размер), частотное (изменяется количество точек, при одинаковом размере) и стохастическое, при котором не наблюдается регулярной структуры расположения точек.

Каждое из чисел, определяющее цвет в *CMYK*, представляет собой процент краски данного цвета, составляющей цветовую комбинацию, а точнее, размер точки растра, выводимой на фотонаборном аппарате на плёнке данного цвета. Например, для получения цвета «хаки» следует смешать 30 % голубой краски, 45 % пурпурной краски, 80 % жёлтой краски и 5 % чёрной. Это можно обозначить следующим образом: (30,45,80,5). Иногда пользуются таким обозначением: C30M45Y80K5.

4.3.3 HSB

HSV (*Hue*, *Saturation*, *Value* — тон, насыщенность, значение) или *HSB* (*Hue*, *Saturation*, *Brightness* - оттенок, насыщенность, яркость) - цветовая модель, в которой координатами цвета являются: *Hue* - цветовой тон, (например, красный, зелёный или сине-голубой). Варьируется в пределах 0 -360°, однако иногда приводится к диапазону 0-100 или 0-1. *Saturation* - насыщенность. Варьируется в пределах 0-100 или 0-1. Чем больше этот параметр, тем «чище» цвет, поэтому этот параметр иногда называют чистотой цвета. А чем ближе этот параметр к нулю, тем ближе цвет к нейтральному серому. *Value* (значение цвета) или *Brightness* - яркость. Также задаётся в пределах 0-100 и 0-1. Модель является нелинейным преобразованием модели *RGB*. Цвет, представленный в *HSV*, зависит от устройства, на которое он будет выведен, так как *HSV* - преобразование модели *RGB*, которая тоже зависит от устройства. Для получения кода цвета, не зависящего от устройства, используется модель *Lab*.

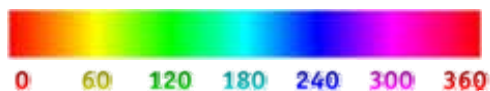


Рис. 16. Шкала оттенков - *Hue*.

Цветовая модель *HSB* описывает цветовое пространство через такие характеристики цвета, как цветовой оттенок, насыщенность и яркость. Чистый цветовой фон – один из цветов спектрального разложения света. Цветовой оттенок – смесь чистого цветового тона с серым цветом. Насыщенность цвета, или степень чистоты цвета – доля чистого тона в цветовой смеси (чем больше серого,

тем меньше насыщенность). Яркость характеризуется общей светлостью смешиваемых цветов (чем больше чёрного, тем меньше яркость).

В модели *HSB* цвет описывается тройкой чисел (цветовой оттенок, яркость, насыщенность). Рассмотрим ряд цветов: красный, тёмно-красный, красновато-чёрный, алый, розовый, бледно-розовый. В модели *HSB* эти цвета- производные от красного цвета и отличаются друг от друга только яркостью и насыщенностью красного оттенка. Такое описание цвета (в отличие от моделей *RGB* и *CMYK*) очень точно передаёт субъективное восприятие цвета человеком, а не технические особенности воспроизведения цветов. Подобные описания широко используются во всех областях искусства и производства, где приходится иметь дело с цветом.

Пространство цветов модели *HSB* имеет форму вложенных концентрических конусов с общей вершиной и общей осью симметрии. Цвета с одинаковой насыщенностью расположены на конической поверхности с определённым углом при основании. Цвета с одинаковой яркостью расположены по кругу – сечению объёмного конуса плоскостью, перпендикулярной его оси. При этом вершина конуса соответствует чёрному цвету. Цвета с одинаковым оттенком расположены в полуплоскости, проходящей через ось симметрии конуса. Оттенок изменяется по окружности конуса. Насыщенность цвета возрастает с удалением от оси конуса, а яркость — с приближением к его основанию.

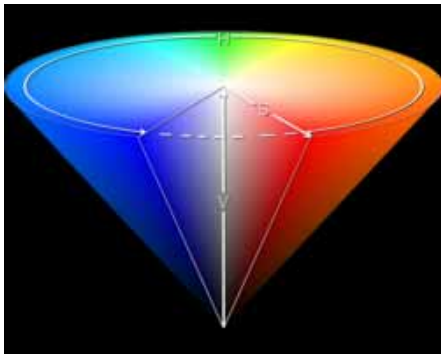


Рис. 17. Коническое представление модели

Чтобы использовать математическую модель *HSB* для компьютерного представления графической информации, надо, как и для моделей *RGB* и *CMYK*, провести квантование цветового пространства, т.е. непрерывно изменяемые значения компонент цвета представить в дискретной форме. В ОС *Windows* каждая из *HSB*-характеристик описывается одним байтом, т.е. шкала значений разделена на 256 уровней.

Все три рассмотренные цветовые модели (*RGB*, *CMYK*, *HSB*) описывают одно и то же реально существующее пространство. Их взаимный анализ позволяет отметить следующее:

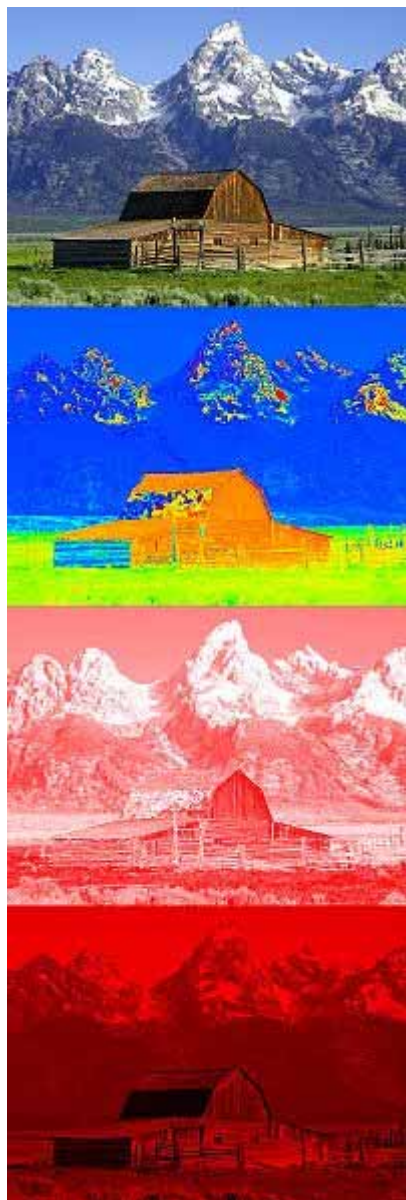
- в цветовом пространстве модели *HSB* очень хорошо видна связь между моделями *RGB* и *CMYK*: на цветовом круге основные цвета одной модели расположены точно напротив основных цветов другой модели;
- если на цветовом круге отметить точками основные цвета *RGB*-модели, то они образуют равносторонний треугольник, то же самое можно сказать и относительно модели *CMYK*;
- цвета модели *HSB*, которые не попадают в этот треугольник, в *RGB*-модели будут непредставимы. То же самое можно сказать и относительно модели *CMYK*.

Модель *HSB* позволяет представить (закодировать) практически все цвета, воспринимаемые человеком. Модели *RGB* и *CMYK* описывают возможности компьютерных устройств по воспроизведению цвета. И оказывается, что некоторые цвета в принципе не могут быть воспроизведены на компьютере.

Рис. 18. Изображение и его отдельные компоненты - *H*, *S*, *V*. На разных участках изображения можно проследить изменения компонент

4.4 Форматы графических файлов

Windows Bitmap (.BMP) формат файлов растровых рисунков, разработанный Microsoft. Главным достоинством является его простота и, как следствие, поддержка всеми без исключения программами, работающими с графикой. Хранит информацию о каждой точке без использования алгоритмов сжатия



Graphics Interchange Format (.GIF) формат файлов разработанный *ComuServe Inc.* Чаще всего применяется для размещения рисунков в интернете. К достоинствам формата можно отнести возможность создания рисунков с прозрачным фоном (*transparency*) и анимацией. Предусмотрен метод сжатия без потерь *LZW*.

Portable Network Graphics (.PNG) использует метод сжатия без потерь *LZW*, позволяющий достичь высокой степени сжатия (не хуже *GIF*). Примерно в два раза компактнее *BMP*. Имеет возможность черзстрочного вывода для быстрой черновой прорисовки изображения.

Joint Photographic Experts JPEG (.JPG) позволяет добиться наивысшей степени сжатия и минимальный размер выходного файла. Использует сжатие с потерей информации и предназначен для хранения, в основном, фотографических изображений с большим количеством оттенков и цветовых переходов. Не подходит для хранения однотонных и штриховых изображений. Сжатие и соответственно потеря качества происходит при каждом сохранении изображения, поэтому рекомендуется использовать как формат хранения. Используется в большинстве мультимедийных устройств.

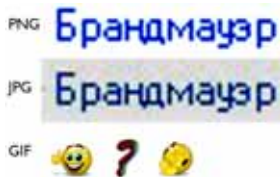


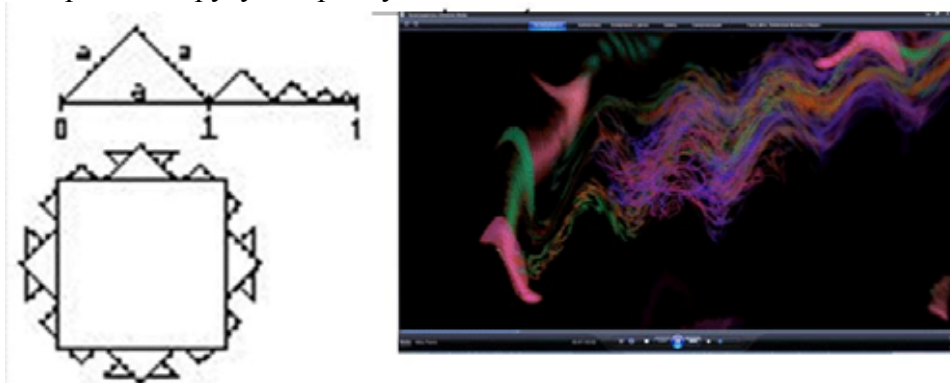
Рис. 19. Изображение в разных форматах.

TIFF (.TIF) позволяет сохранять изображения любой глубины цвета с использованием как модели *RGB*, так и *СМЯК*. Поддерживается несколько алгоритмов сжатия, как с потерей, так и без потери качества.

В отличие от текстового представления информации, когда минимальной единицей является символ, при отображении графики картинка строится из отдельных элементов - пикселей (*PICTure ELeMent* – «элемент картинки»). Часто пиксел совпадает с точкой дисплея, но это необязательно: например, в некоторых видеорежимах 1 пиксел может состоять из 2 или 4 точек экрана. Каждый пиксел характеризуется цветом. Как и вся остальная информация в ЭВМ, цвет кодируется числом. В зависимости от количества допустимых цветов, число двоичных разрядов на один пиксел будет различным. Так, для чёрно-белой картинки закодировать цвет точки можно одним битом: 0 - чёрный, 1 - белый. Для случая 16 цветов требуется уже по 4 разряда на каждую точку, а для 256 цветов - 8, т.е. 1 байт.

4.5 Фрактальная графика

Фрактальная графика, как и векторная – вычисляемая, но отличается от неё тем, что никакие объекты в памяти компьютера не хранятся. Изображение строится по уравнению (или по системе уравнений), поэтому ничего, кроме формулы, хранить не надо. Изменив коэффициенты в уравнении, можно получить совершенно другую картину.



5. ПРЕДСТАВЛЕНИЕ ЗВУКОВОЙ ИНФОРМАЦИИ

Звук – волновые колебания давления в упругой среде (воздухе, воде, металле и т.д.). Для обозначения звука часто используют термин «звуковая волна». Основные параметры любых волн, и звуковых в том числе, - частота и амплитуда колебаний. Частоту звука измеряют в герцах (Гц – количество колебаний в секунду). Человеческое ухо способно воспринимать звук в широком диапазоне частот, от 16 Гц до 20 кГц. В нетехнических областях (например, в музыке) вместо термина «частота» используют термин «тон».

Амплитуду звуковых колебаний называют звуковым давлением или силой звука, эта величина характеризует воспринимаемую громкость звука. Абсолютную величину звукового давления – паскалях (Па). Самые слабые, едва различимые звуки имеют амплитуду 20 мкПа ($2 \cdot 10^{-5}$ Па – порог слышимости). Самые сильные звуки, ещё не выводящие слуховые органы из строя, могут иметь амплитуду до 200 Па

(болевого порог). Из-за столь широкого диапазона значений (максимальное и минимальное значения отличаются на 6-7 порядков!) абсолютными величинами звукового давления пользоваться неудобно, на практике обычно используют логарифмическую шкалу децибелов.

Относительную силу звука, или уровень звука, определяют как логарифм отношения абсолютной величины звукового давления к величине порога слышимости, умноженный на некоторый постоянный коэффициент. Уровень звука измеряют в особых единицах – децибелах (обозначается дБ). Ниже приводится форма расчёта уровня звука:

$$L = 20 \cdot \lg(P_{зв}/P_{пс}),$$

где L – уровень звука (в дБ), $P_{пс}$ – порог слышимости ($2 \cdot 10^{-5}$ Па), $P_{зв}$ – давление измеряемого звука.

Логарифмическая шкала децибелов на практике весьма удобна, хотя поначалу пользоваться ей непривычно:

- весь диапазон слышимых звуков составляет 0-140 дБ: 0 дБ – порог слышимости, 140 дБ – болевой порог;
- человеческое ухо способно уловить различие в громкости, если звуки отличаются по силе не менее, чем на 10%, что соответствует разнице в уровнях примерно на 1 дБ;
- двухкратное различие в амплитуде звуков соответствует различию уровней 6 дБ;
- если уровни звуков отличаются на 20 дБ, а разница в 40 дБ соответствует 10-кратному различию в амплитудах.

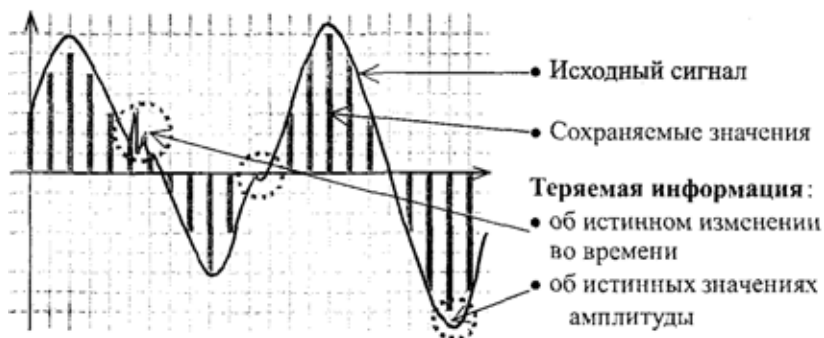
Табл. 8. Некоторые значения уровней звука.

Порог слышимости	0 дБ
Шорох листьев, шум слабого ветра	10–20 дБ
Шепот (на задней парте)	20–30 дБ
Разговор средней громкости (в кабинете директора)	50–60 дБ
Автомобильная магистраль с интенсивным движением	80–90 дБ
Авиадвигатели	120–130 дБ
Болевой порог	≈140 дБ

Звукозапись – процесс сохранения информации о параметрах звуковых волн. Способы хранения или записи звука разделяются на аналоговые и цифровые. При аналоговой записи на носителе размещается непрерывный «слепок» звуковой волны. Так, на грампластинке пропечатывается непрерывная канавка, изгибы которой повторяют амплитуду и частоту звука, а на магнитной ленте параметры звука сохраняются в виде намагниченности рабочей поверхности, степень намагниченности непрерывно изменяется, повторяя параметры звука.

В компьютерах применяется исключительно цифровая форма записи звука. При цифровой записи звук необходимо подвергнуть временной дискретизации и квантованию: параметры звукового сигнала измеряются не непрерывно, а через определённые промежутки времени (временная дискретизация; результаты измерений записываются в цифровом виде с ограниченной точностью (квантование).

В компьютер приходит не сам звук, а электрический сигнал, снимаемый с какого-либо устройства: например, микрофон преобразует звуковое давление в электрические колебания, которые в дальнейшем и обрабатываются. К компьютеру можно подключить и магнитофон, и радио, и эхолот – любое устройство, вырабатывающее электрические сигналы.



Цифровая запись вносит двойное искажение в сохраняемые параметры сигнала: во-первых, при дискретизации теряется информация об истинном изменении звука между измерениями, а во-вторых, при квантовании сохраняются не точные параметры, а только близкие к ним дискретные значения.

При **временной дискретизации** время разбивают на небольшие интервалы, в пределах которых характеристики природных сигналов считают неизменными.

Наглядным примером временной дискретизации может служить кино и телевидение. В них иллюзия подвижного изображения создаётся путём быстрой смены кадров. При этом сами кадры являются статическими изображениями. Компьютерное кодирование видеоинформации также основано на эффекте смены кадров, на которых изображены последовательные фазы движения.

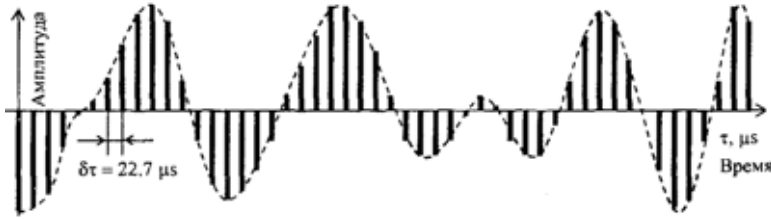
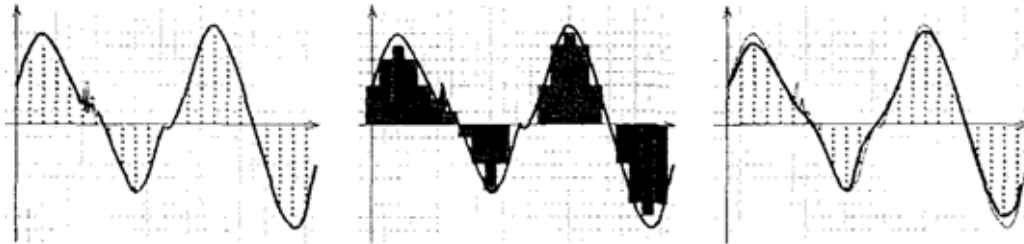


Рис. 20. Временная дискретизация звукового сигнала.

Импульсно-кодовая модуляция (*Pulse Code Modulation, PCM*) заключается в том, что звуковая информация хранится в виде значений амплитуды, взятых в определённые моменты времени (т.е. измерения проводятся «импульсами»). При записи звука в компьютер амплитуда измеряется через равные интервалы времени с некоторой достаточно большой частотой. При воспроизведении звука компьютер использует сохранённые значения для того, чтобы восстановить непрерывную форму выходного сигнала.



Исходный сигнал

Хранимая информация

Воспроизводимый сигнал

Процесс получения цифровой формы звука называют оцифровкой. Устройство, выполняющее оцифровку звука, называется аналого-цифровым преобразователем (АЦП). Устройство, выполняющее обратное преобразование, из цифровой формы в аналоговую, называется цифро-аналоговым преобразователем (ЦАП). В современных компьютерах основная обработка звука выполняется звуковыми картами. Помимо АЦП и ЦАП звуковые карты содержат сигнальный процессор – специализированный микрокомпьютер для обработки оцифрованного звука, выполняющий значительную часть рутинных расчётов при обработке звуков (смешение звуков, наложение спецэффектов, расчёт формы выходного сигнала и т.п.; центральный процессор не тратит время на выполнение этих работ).

Моменты измерения амплитуды называют отсчётами. Частоту, с которой производят измерения сигнала, называют частотой дискретизации.



Квантование звука заключается в следующем. Сначала мгновенные значения звукового давления измеряются с ограниченной точностью, затем, как и в случае с квантованием цветов, диапазон значений амплитуды разбивается на подуровни. По измеренному значению определяется подуровень, в который попадает значение, и в компьютере сохраняется только его номер. Количество бит, используемых для записи номеров подуровней, называется глубиной кодирования звука.

Если сравнить способы представления графической и звуковой информации, то импульсное

кодирование звука соответствует растровому представлению изображений:

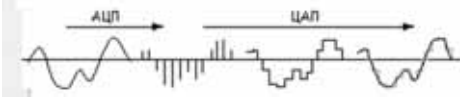
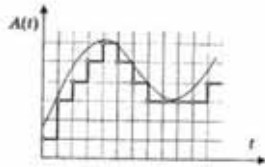
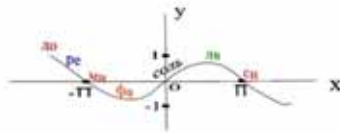
- структура звука (в графике – изображения) не анализируется;
- время (в графике – пространство) априори разбивается на небольшие области; в пределах этих областей параметры звука (изображения) считаются постоянными.

При рассмотрении представления графической информации упоминалось, что растровое представление изображения не требует хранения координат отдельных пикселей. Аналогично, при сохранении импульсного представления звука достаточно единожды сохранить параметры оцифровки (глубину кодирования, частоту дискретизации и длительность звукового фрагмента), а затем сохранять только номера подуровней единым потоком. Увеличивая частоту дискретизации и глубину кодирования, можно более точно сохранить и впоследствии восстановить форму звукового сигнала. При этом улучшается

субъективное качество оцифрованного звука, однако увеличивается объём сохраняемых данных. При цифровой записи звука в различных случаях используют разные значения частоты дискретизации и глубины кодирования. Например, в цифровых автоответчиках используют частоту дискретизации 8-11 кГц и 8 бит для записи амплитуды, а стандарт записи звука на компакт-дисках соответствует частоте дискретизации 44,1 кГц и 16 бит для амплитуды на каждый аудио-канал (стереозвук – 2 канала, моно – один канал).

Пример 14. Оценим объём стереоаудиофайла в формате PCM с глубиной кодирования 16 бит и частотой дискретизации 44,1 кГц, который хранит звуковой фрагмент длительностью звучания 1 сек. Объём такого звукового фрагмента равен: 16 бит x 44100 Гц x 2 (канала) = 1411200 бит = 176400 байт = 172,3 Кбайт.

Метод таблично-волнового (Wave-Table) синтеза
Метод FM (Frequency Modulation)



До какой степени можно уменьшать параметры оцифровки (а значит, и объём оцифрованного звукового фрагмента), чтобы при восстановлении звук оставался достаточно близок к исходному? В 1828 Гарри Найквист высказал утверждение, что частота дискретизации должна быть в два или более раза выше максимальной частоты измеряемого сигнала. В 1933 В.А.Котельников и независимо от него Клод Шеннон сформулировали и доказали теорему, более сильную, чем утверждение Найквиста, о том, при каких условиях и как по дискретным значениям можно восстановить форму непрерывного сигнала. Эта теорема в России называется теоремой

Котельникова, на Западе – теоремой Найквиста-Шеннона; есть у неё и нейтральное название – теорема об отсчётах.

Теорема Найквиста-Котельникова-Шеннона утверждает, что если имеется сигнал $U(t)$, спектр которого ограничен сверху частотой f , то после его дискретизации с частотой $F > 2f$ форму исходного сигнала можно точно восстановить по дискретным значениям (отсчётам), по формуле:

$$U(t) = \sum_{k=-\infty}^{+\infty} U(k\Delta t) \frac{\sin(2\pi F(t - k\Delta t))}{2\pi F(t - k\Delta t)},$$

где $\Delta t = 1/F$ – время между отсчётами; $k\Delta t$ - время k -го отсчёта; $U(k\Delta t)$ - значение k -го отсчёта.

Покажем, почему частота дискретизации должна быть как минимум вдвое выше частоты сигнала. Допустим, на вход АЦП был подан синусоидальный сигнал, и АЦП выдал его цифровое представление. Вопрос: можно ли однозначно определить форму входного сигнала?

Ответ на поставленный вопрос получается удручающий – решение не единственно, даже синусоидальных сигналов, имеющих такое цифровое представление, несколько. Точнее, подходящие синусоиды образуют два бесконечных семейства. На **Рис.21** по горизонтальной оси откладывается время, по вертикальной – амплитуда сигнала; толстые вертикальные линии обозначают измеренные значения; сплошной линией отмечен синусоидальный сигнал с низкой частотой; пунктирной линией отмечен синусоидальный сигнал с высокой частотой. Этот пример показывает, что по одним лишь отсчётам мы принципиально не можем определить характеристики входного сигнала (при дискретизации непрерывных сигналов часть информации утрачивается). Однако если потребовать, чтобы частота искомого сигнала не превосходила половины частоты дискретизации, то решение будет единственным.

Точная формулировка теоремы Найквиста-Котельникова применима только к сигналам с неизменными частотными характеристиками и бесконечной длительностью, так что для оцифровки реальных звуковых сигналов частоту дискретизации выбирают с небольшим запасом.

В 80-х годах прошлого века появились электронные музыкальные инструменты – синтезаторы, способные воспроизводить не только звуки многих существующих инструментов, но и абсолютно новые звуки. Было разработано соглашение о системе команд универсального синтезатора, получившее название стандарта *MIDI (Musical Instrument Digital Interface)*. Запись музыкального произведения в формате *MIDI* – последовательность закодированных сообщений синтезатору. Сообщение может быть командой, описанием параметров или управляющим сообщением.

MIDI-команды делают запись музыкальной информации более компактной, чем импульсное кодирование. Если сравнить способы представления графической и звуковой информации, то запись звука в виде *MIDI*-команд соответствует векторному представлению изображений.

Записанные звуковые файлы можно редактировать, т.е. вырезать, копировать и вставлять фрагменты из других файлов. Кроме того, можно увеличивать или уменьшать громкость, применять различные

звуковые эффекты (эхо, изменение скорости воспроизведения, воспроизведение в обратном направлении и др.), а также накладывать файлы друг на друга (микшировать).

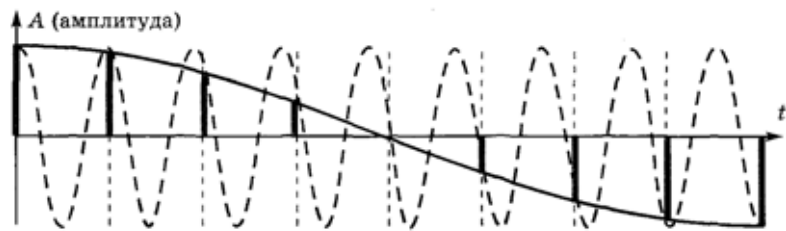


Рис.21. Потеря информации при дискретизации сигнала

При воспроизведении звука на компьютере цифровое представление сигнала преобразуют обратно в аналоговую форму. Расчёт параметров

выходного сигнала выполняет сигнальный процессор, а генерацию аналогового электрического сигнала выполняет ЦАП.

В современных компьютерах используются следующие форматы звуковых файлов:

WAVE (.wav) – наиболее широко распространённый звуковой формат, но не обеспечивает достаточно хорошего сжатия.

MPEG-3 (.mp3) Используя для оцифровки музыкальных записей. При кодировании применяется психоакустическая компрессия, при которой из мелодии удаляются звуки, плохо воспринимаемые человеческим ухом.

RealAudio (.ra, .ram) – формат, разработанный для воспроизведения звука в Интернет в реальном времени. Получающееся качество в лучшем случае соответствует посредственной аудиокассете.

MIDI (.mid) – цифровой интерфейс музыкальных инструментов (*Musical Instrument Digital Interface*). Интерфейс *MIDI* представляет собой протокол передачи музыкальных нот и мелодий. Т.е. в файле хранятся описания высоты и длительности звучания музыкальных инструментов. *MIDI* – файлы занимают меньший объём (единица звукового звучания в секунду), чем эквивалентные файлы оцифрованного звука.

Коротко остановимся на кодировании видеоинформации.

Аналоговое видео является самым ранним методом передачи видеосигнала. Композитное аналоговое видео комбинирует всё видео компоненты (яркость, цвет, синхронизацию и т.д. в один сигнал. Из-за объединения этих элементов в одном сигнале качество композитного видео далеко от совершенства. Цифровое видео – изображение или серия изображений, информация в которых хранится в цифровом виде. Оно использует цифровые сигналы и стандарты, отличные от международных стандартов для телевидения и вывода изображений на экран, используемых в аналоговом видео.

Известны следующие форматы сохранения видеоинформации:

Digital Video (.DV) – формат, разработанный для цифровых видеокамер и видеомagneтофонов. Сигнал компонентный, метод сжатия *MJPEG* с коэффициентом 5,1.

CD AVI (Audio Video Interleave) – чередование аудио и видео – позволяет одновременно хранить изображение и звук. При записи в этом формате используются несколько различных форматов сжатия (компрессии) видеоизображения: *Microsoft Video 1* (8 и 16-битный цвет), *Motion JPEG*, *Microsoft RLE* (8-битный цвет), *Indeo* и т.д.

MPEG (Motion JPEG) (.mpg, .mpeg, .dat) – формат для записи и воспроизведения видео, разработанный группой экспертов по движущимся изображениям (*MPEG – Moving Picture Expert Group*). Имеет собственный алгоритм компрессии, основанный на кодировании изменений ключевых кадров. Среди производных форматов известен *MPEG-2*, *MPEG-4*

Quick Time Movie (.qt, .mov) – наиболее распространённый формат для записи и воспроизведения потокового видео, разработанный фирмой *Adobe* в рамках технологии *Quick Time*. Поддерживает несколько различных форматов сжатия видео, в том числе *MPEG* и *Indeo*, а также свой собственный метод компрессии.