

1. Язык SQL. Операторы языка для описания и манипулирования данными.

SQL (*Structured Query Language* — «язык структурированных запросов») — универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных.

Стандарт SQL определяется ANSI (*Американским Национальным Институтом Стандартов*) и в данное время также принимается ISO (*Международной Организацией по Стандартизации*). Однако, большинство коммерческих программ баз данных расширяют SQL без уведомления ANSI, добавляя различные особенности в этот язык, которые, как они считают, будут весьма полезны. Иногда они несколько нарушают стандарт языка, хотя хорошие идеи имеют тенденцию развиваться и вскоре становятся стандартами "рынка" сами по себе в силу полезности своих качеств.

В язык SQL в качестве составных частей входят:

- *язык манипулирования данными* (*Data Manipulation Language, DML*) - используется, как это следует из его названия, для манипулирования данными в таблицах баз данных;
- *язык определения данных* (*Data Definition Language, DDL*) - используется для создания и изменения структуры базы данных и ее составных частей - таблиц, индексов, представлений (виртуальных таблиц), а также триггеров и сохраненных процедур;
- *язык управления данными* (*Data Control Language, DCL*) - используется для управления правами доступа к данным и выполнением процедур в многопользовательской среде. Более точно его можно назвать "язык управления доступом".

Операторы:

- операторы определения данных (*Data Definition Language, DDL*):
CREATE создает объект БД (саму базу, таблицу, представление, пользователя и т. д.)
ALTER изменяет объект
DROP удаляет объект
- операторы манипуляции данными (*Data Manipulation Language, DML*):
SELECT считывает данные, удовлетворяющие заданным условиям
INSERT добавляет новые данные
UPDATE изменяет существующие данные
DELETE удаляет данные
- операторы определения доступа к данным (*Data Control Language, DCL*):
GRANT предоставляет пользователю (группе) разрешения на определенные операции с объектом
REVOKE отзывает ранее выданные разрешения
DENY задает запрет, имеющий приоритет над разрешением
- операторы управления транзакциями (*Transaction Control Language, TCL*):
COMMIT применяет транзакцию.
ROLLBACK откатывает все изменения, сделанные в контексте текущей транзакции.
SAVEPOINT делит транзакцию на более мелкие участки.

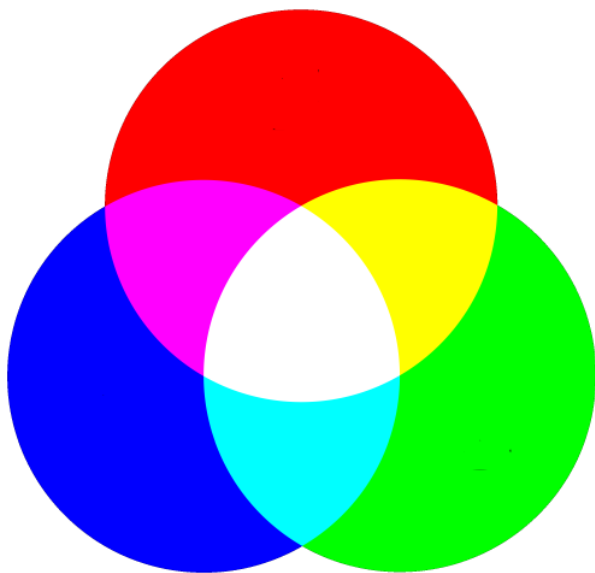
2. Цвет в машинной графике. Аппроксимация полутонами.

Субъективность в восприятии цвета при обработке изображений крайне нежелательна. Для обеспечения одинакового воспроизведения одного и того же цвета видеомониторами, принтерами и сканерами разных фирм-изготовителей необходимо наличие объективных измерительных систем, позволяющих установить однозначное определение цветовых координат. Для этих целей разработаны **цветовые модели**, предоставляющие средства для концептуального и количественного описания цвета.

По принципу действия цветовые модели можно условно разбить на три класса:

- аддитивные (RGB), основанные на сложении цветов;
- субтрактивные (CMY, CMYK), основу которых составляет операция вычитания цветов (субтрактивный синтез);
- перцепционные (HSB, HLS, Lab, YCC), базирующиеся на восприятии.

Аддитивный цвет получается на основе законов Грассмана путем соединения лучей света разных цветов. В основе этого явления лежит тот факт, что большинство цветов видимого спектра могут быть получены путем смешивания в различных пропорциях трех основных цветовых компонент: красный, зеленый и синий цвет. При попарном смешивании первичных цветов образуются вторичные цвета: голубой (Cyan), пурпурный (Magenta) и желтый (Yellow).

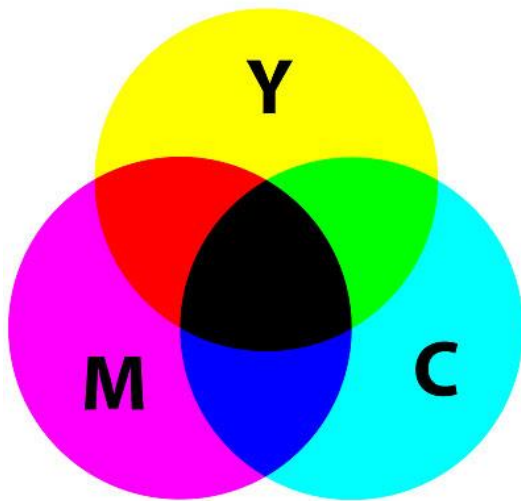


Аддитивные цвета нашли широкое применение в системах освещения, видеосистемах, устройствах записи на фотопленку, мониторах, сканерах и цифровых камерах.

Несмотря на то, что цветовая модель RGB достаточно проста и наглядна, при ее практическом применении возникают две серьезные *проблемы*:

- аппаратная зависимость;
- ограничение цветового охвата,

В отличие от экрана монитора, воспроизведение цветов которого основано на излучении света, печатная страница может только отражать цвет. Поэтому RGB-модель в данном случае неприемлема. Вместо нее для описания печатных цветов используется модель CMY, базирующаяся на **субтрактивных цветах**. Субтрактивные цвета в отличие от аддитивных цветов получают вычитанием вторичных цветов из общего луча света. В этой системе белый цвет появляется как результат отсутствия всех цветов, тогда как их присутствие дает черный цвет.



Основная идея аппроксимации полутонами состоит в том, что если правильно чередовать точки (для растра это будут пиксели), принимающие ограниченное количество значений атрибутов, начиная с определенного расстояния (которое зависит от пространственного разрешения устройства и конкретного человека), человек перестает различать точки по отдельности. Вместо этого возникнет картина усреднения значений атрибутов и, соответственно, иллюзия большого количества оттенков.

Аппроксимация полутонами — это метод, в котором используется минимальное число уровней интенсивности, обычно черный и белый, для улучшения визуального разрешения, то есть получения нескольких полутонов серого или уровней интенсивности.

Визуальное разрешение машинно-сгенерированных изображений можно улучшить с помощью метода, называемого конфигурированием. В данном методе обычно размеры клеток фиксированы. Для изображения с фиксированным разрешением несколько пикселей объединяются в конфигурации. Здесь ухудшение пространственного разрешения обменивается на улучшение визуального. На рис. 17.1а показана одна из возможных групп конфигураций для двухуровневого черно-белого дисплея. Для каждой клетки используется четыре пиксела. При такой организации получается пять возможных уровней или тонов серого (0-4).

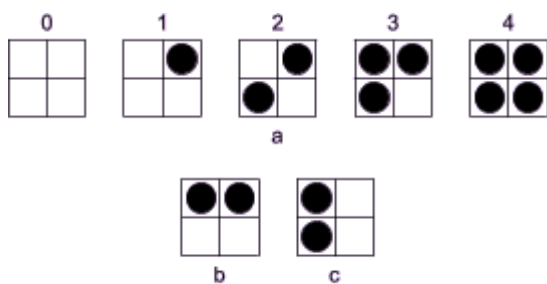


рис. 17.1

В общем случае для двухуровневого дисплея число возможных интенсивностей на единицу больше числа пикселей в клетке. При выборе конфигураций следует проявлять осторожность, так как иначе могут возникнуть нежелательные мелкомасштабные структуры. Например, не следует применять ни одну из конфигураций, изображенных на рис. 17.1b или рис. 17.1c, иначе это приведет к тому, что для большой области с постоянной интенсивностью на изображении появятся

нежелательные горизонтальные или вертикальные линии. Число доступных уровней интенсивности можно увеличить с помощью увеличения размера клетки. Конфигурации для клетки 3 * 3 пикселей приведены на рис. 17.2. Они дают десять уровней (с 0 по 9) интенсивности.

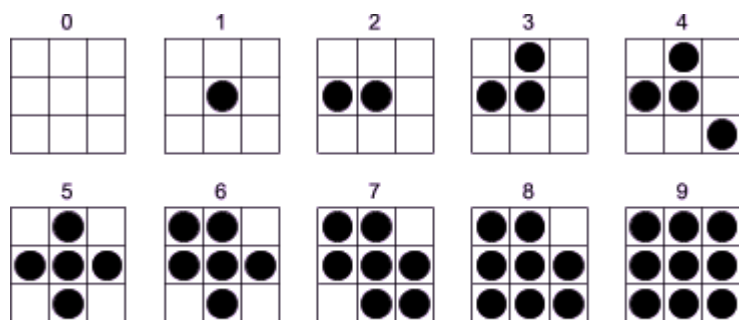


рис. 17.2

Если точки могут быть разного размера, то можно получить дополнительное количество уровней интенсивности. На рис. 17.4 представлены конфигурации для клетки 2 * 2 пиксела с двумя размерами точек. В результате получается 9 уровней интенсивности.

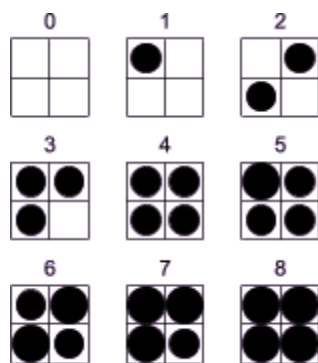


рис. 17.4

3. Представить алгоритм нахождения минимума функции двух переменных.

Для примера рассмотрим алгоритм метода градиентного спуска.

1. НАЧАЛО
2. Ввод h_1, h_2
3. Ввод $\Delta x_1, \Delta x_2$
4. Ввод точности ϵ
5. Ввод начального приближения (x_1, x_2)
6. $E = \epsilon$
7. Пока $E \geq \epsilon$
8. $df_{x1} = (f(x_1 + \Delta x_1, x_2) - f(x_1, x_2)) / \Delta x_1$
9. $df_{x2} = (f(x_1, x_2 + \Delta x_2) - f(x_1, x_2)) / \Delta x_2$
10. $x_1 = x_1 - h_1 * df_{x1}$
11. $x_2 = x_2 - h_2 * df_{x2}$
12. $y = f(x_1, x_2)$
13. $E = \text{abs}(df_{x1} + df_{x2})$

14. Конец цикла
15. Вывод у
16. КОНЕЦ