

КОМПЬЮТЕРНАЯ ГРАФИКА

ВВЕДЕНИЕ

© В. И. Зенкин, 2013



назад

закреть

Компьютерная графика — раздел информатики (computer science), который изучает средства и способы создания и обработки визуальной информации (в виде изображений, чертежей, анимации и т. д.) при помощи компьютерной техники.

В настоящее время компьютерная графика широко применяется во многих областях науки и техники, а также в искусстве, компьютерных играх, делопроизводстве, средствах массовой информации, коммерческой и управленческой деятельности и т. п.

Условно можно выделить три основных типа задач, относящихся к компьютерной графике:

1. Построение изображения по описанию.
2. Редактирование изображения.
3. Распознавание образов.

В данном курсе будем рассматривать задачи первых двух видов.

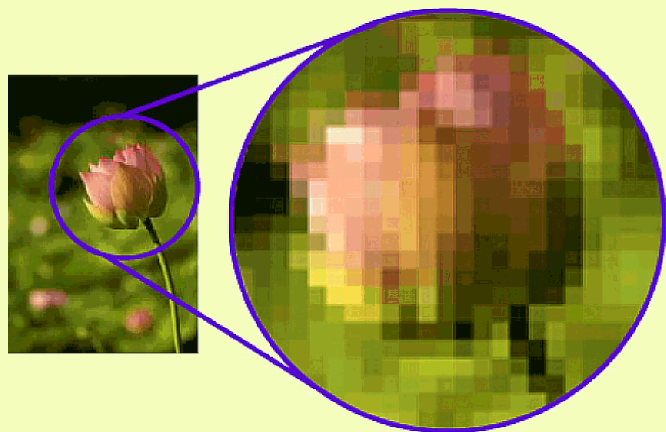
Основные виды компьютерной графики:

- Растровая графика.
- Векторная графика.

Они отличаются друг от друга способами формирования изображения при отображении на экране монитора или при печати на бумаге и форматом хранения графической информации в файлах.

Растровая графика (raster graphics) характеризуется представлением изображения в виде массива цветных «точек» — *пикселей*¹, наименьших элементов² двумерного цифрового растрового изображения.

Самый простой тип растрового изображения — *бинарный*, когда пиксели могут быть только двух цветов, например, черного и белого. Для кодирования такого цвета, очевидно, достаточно одного бита информации (0 — черный, 1 — белый). В настоящее время стандартным является 24-бита для ко-



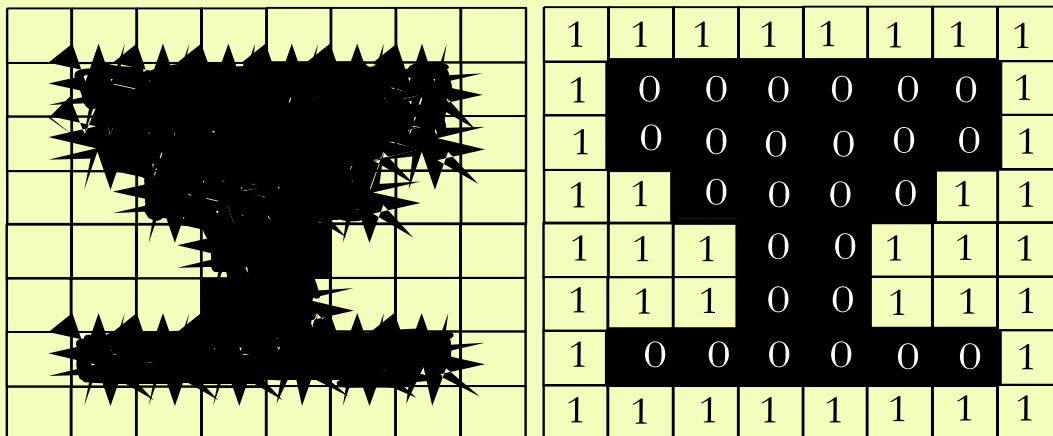
дирования цвета пиксела. Например, для записи небольшого изображения 500×600 пикселей требуется $500 \cdot 600 \cdot 3 = 900\,000$ байт. Т.о, растровая графика, как правило, требует для хранения значительного объема информации, что является проблемой, особенно, при передаче файлов в Интернете. Для решения этой проблемы применя-

ют *сжатие* данных. Однако, наиболее значительная степень сжатия достигается за счет потери качества изображения ("lossy" compression).

¹ англ. pixel — сокращение от picture element, элемент изображений.

² В компьютерной графике термин *пиксел* используется также в значении «наименьший элемент изображения монитора» и «точка, печатаемая принтером».

Оцифровка изображений. Поскольку память компьютера всегда ограничена, а информация, которую он хранит, имеет цифровой вид, ЭВМ не может содержать *аналоговые* (непрерывные)³ изображения, которые обычно встречаются в реальности. Поэтому применяют оцифровку (digitization) — перевод аналогового изображения в цифровой вид, пригодный для записи на электронные носители и последующей компьютерной обработки. Рассмотрим аналоговое изображение на плоскости (например,

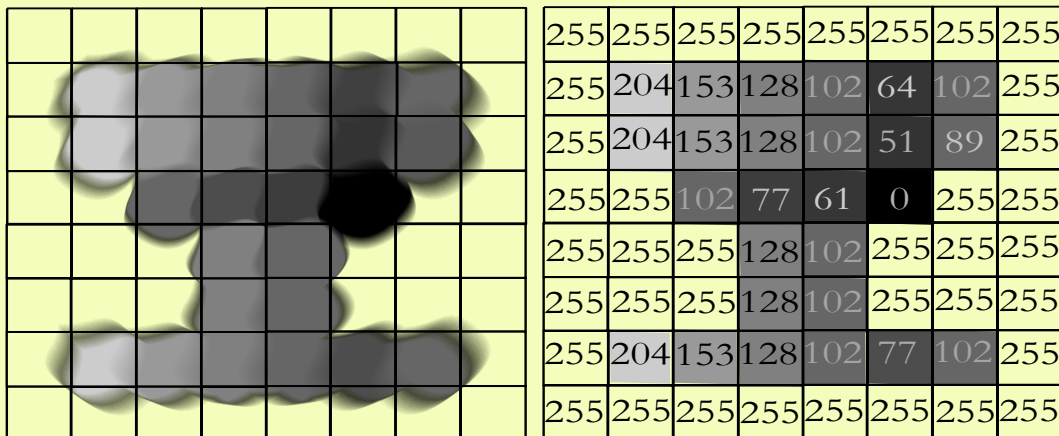


фотографию). Вначале предположим, что изображение имеет только два

³ Очевидно, что непрерывный сигнал в общем случае невозможно записать в цифровой форме в полном объёме ни на какое реальное физическое устройство, т. к. любой отрезок вещественной прямой содержит бесконечно много точек.

цвета, чёрный и белый. Исходное непрерывное изображение разбивается квадратной сеткой на области. Чем чаще сетка, тем больше оцифрованное изображение будет похоже на исходное и тем больший объем будет занимать файл, содержащий цифровые данные. Поскольку изображение двухцветное, для записи цвета достаточно одного бита: 0 для кодирования чёрного цвета, 1 — для белого.

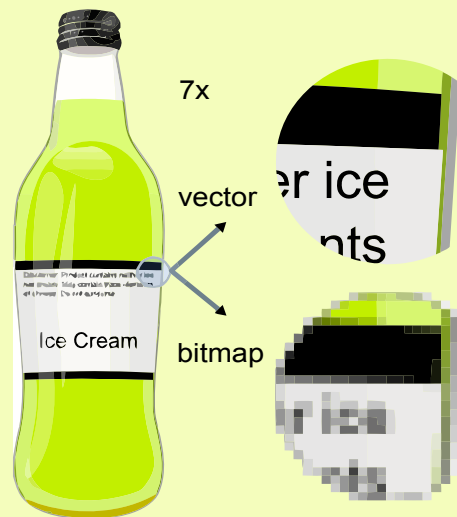
Для полутонового (grayscale) изображения (256 градаций серого, от чёрного до белого включительно) понадобится уже 8 бит: (в двоичном виде) от 00000000 (чёрный) до 11111111 (белый). Разрядность оцифровки в компьютерной графике называют *битовой глубиной цвета*.



назад

заккрыть

Векторная графика представляет изображение в виде комбинации из набора «элементарных» геометрических объектов (примитивов), как-то: точки, отрезки прямой, эллипсы, сплайны и многоугольники и т. д. Объекты в векторной графике описываются при помощи комбинации



компьютерных команд и математических формул. Главное преимущество векторной графики — меньший объем описаний и масштабируемость. Чтобы задать, скажем, окружность в растровом формате нужно записать цвета всех точек окружности и цвет фона, а для векторных изображений достаточно хранить координаты центра окружности, радиус, цвет линии и фона. При увеличении растровой картинки неизбежно пострадает ее качество, в отличие от векторной. Объединяя и комбинируя векторные графические примитивы можно

изобразить очень сложные объекты. Все базовые объекты имеют определенные свойства. Например, для линии это: форма, толщина, цвет, характер (сплошная, пунктирная и т. п.). Внутренняя область замкнутого контура может быть заполнена цветом или текстурой. На линиях располагаются *узлы*, которые имеют свойства, определяющие вид линий и способ сопряжения с другими линиями.

Преимущества векторной графики по сравнению с растровой:

- Масштабируемость без потери качества.
- Как правило, меньший объем файла.
- Удобство редактирования.
- Возможность независимого изменения *частей* рисунка. Объекты на векторном изображении могут перекрываться без всякого воздействия друг на друга.
- Простота и ясность описания графических объектов. Например, векторный формат EPS (Encapsulated PostScript) является одновременно специальным языком программирования для описания страниц. Пример — окружность радиуса 28 с центром в точке (30, 30) на EPS:

```
newpath  
30 30 28 0 360 arc  
closepath
```

Формат SVG (Scalable Vector Graphics — масштабируемая векторная графика), созданный Консорциумом Всемирной паутины (W3C) и рекомендуемый в настоящее время в качестве стандарта для векторной графики в сети Интернет, также является языком разметки. Зная языки, Файлы этих форматов можно читать и редактировать при помощи обычных текстовых редакторов.

Недостатки векторной графики:

- Отсутствие фотореализма. Векторные изображения выглядят искусственно.
- Векторное описание изображения не позволяет *автоматизировать* ввод графической информации, как это делает сканер для растровой графики.
- Ограниченность живописных средств изображения вследствие того, что минимальной областью, закрашиваемой однородным цветом, является не один пиксел, как в растровой графике, а один объект.
- Векторный формат невыгоден при передаче изображений с большим количеством оттенков или мелких деталей.



☞ Таким образом, по возможности всюду следует использовать векторную графику; исключение составляют случаи, когда необходимы фотореалистические изображения с плавными переходами оттенков цвета.

Модели цвета. Как известно, все видимые человеком предметы можно разделить на два класса: объекты, которые излучают свет (например, электрическая лампа, монитор) и объекты, свет отражающие (бумага, клавиатура).

В связи с таким разделением видимых объектов на излучающие и отражающие свет существуют два основных метода моделирования цвета: *аддитивный* для первого случая и *субтрактивный* — для второго.



Известно также, что цветовое восприятие человека обусловлено наличием трех видов световосприимчивых рецепторов на сетчатке глаза, максимумы спектральной чувствительности которых соответствуют синему, зеленому и красному цветам. Поэтому эта тройка цветов является базовой для одной из аддитивных моделей цвета — RGB (Red, Green, Blue). Любой цвет в RGB состоит из 3-х базовых цветов, взятых с определёнными коэффициен-

тами (как любой вектор является линейной комбинацией векторов базиса линейного пространства). Т.о. любой цвет может быть представлен тройкой целых чисел (R, G, B) , где $R, G, B = 0, \dots, 255$ (24-х битная глубина цвета). В частности, в модели RGB $(0, 0, 0)$ соответствует чёрному цвету, а $(255, 255, 255)$ — белому.

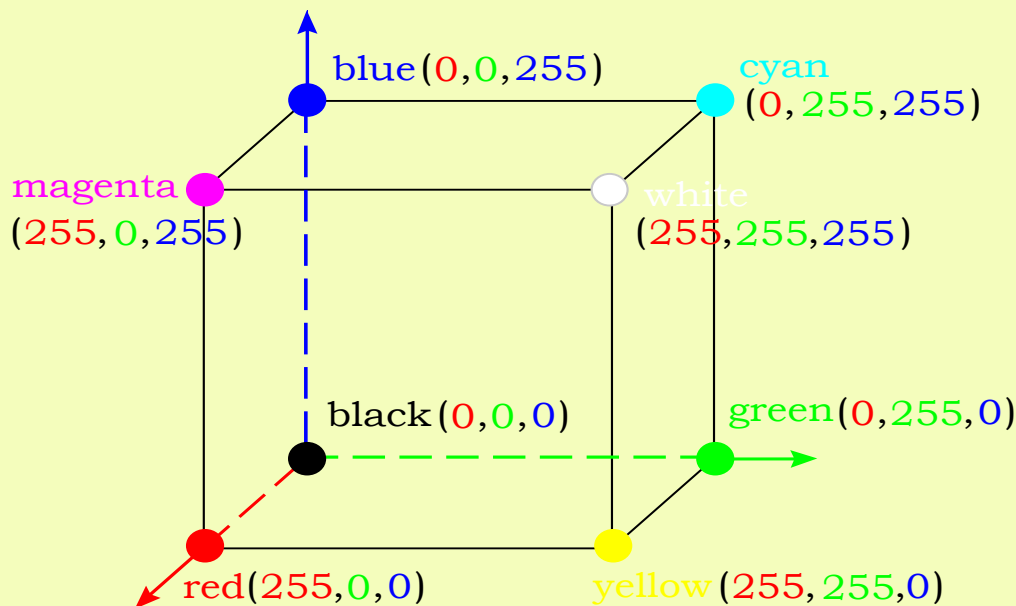


назад

закрыть

? Каковы компоненты серого цвета в цветовой модели RGB?

Наглядно систему RGB удобно представить в виде трёхмерного цветового куба. При этом любой из 2^{24} цветов соответствует целочисленной точке, лежащей внутри и на поверхности данного куба.



Субтрактивные модели CMY (Cyan, Magenta, Yellow) и CMYK (Cyan, Magenta, Yellow, black) используются для задания цвета на объектах, поглощающих свет, в частности для печати на бумаге. В этих моделях, в отличие от RGB, для того, чтобы получить требуемый цвет, базовые цвета *вычитаются* из белого цвета. Например, когда бумага окрашивается

цветом суан, то красный свет, падающий на бумагу, полностью поглощается, т. е. краситель суан как бы вычитает красный свет из падающего белого, который является суммой красного, синего и зелёного света. Переход от модели RGB к CMY задаётся соотношениями

$$C = 1 - r;$$

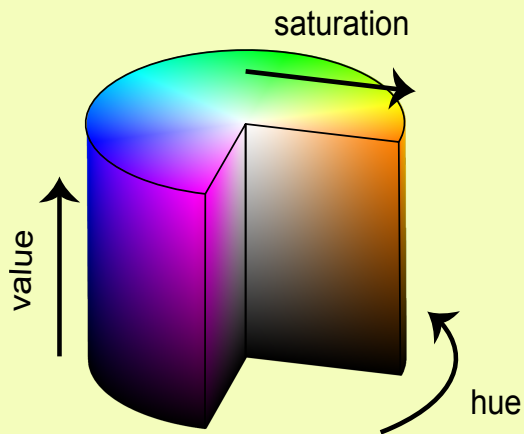
$$M = 1 - g;$$

$$Y = 1 - b,$$

где $r = R/255$, $g = G/255$, $b = B/255$. Теоретически чёрный цвет должен получаться в модели CMY как смесь базовых цветов суан, magenta и yellow, но практически⁴ приходится для достижения этой цели добавлять настоящий чёрный цвет, так получается модель CMYK.

⁴ В следствие химических реакций при смешивании основных цветов, вместо черного обычно получается грязно-бурый цвет.

Рассмотренные цветовые модели RGB и CMYK подходят, соответственно, для отображения на мониторе и полиграфии, и не очень удобны для человеческого восприятия цвета, для которого, как считается, важны тон, насыщенность и светлота цвета. Поэтому есть и другие модели.



Модель HSV⁵ (Hue — оттенок, Saturation — насыщенность, Value — величина). В цилиндрической системе координат H определяется полярным углом, S — радиус-вектором, а величина V — координатой Z. То есть, оттенок изменяется при движении вдоль окружности цилиндра, насыщенность — вдоль радиуса, а яркость — вдоль высоты. Применяется также модель представления с отображением на конус. В пользователь-

ских интерфейсах графических редакторов оттенок представляется в виде радужного кольца, а насыщенность и значение цвета выбираются при помощи вписанного в это кольцо треугольника. Его вертикальная ось, как правило, регулирует насыщенность, а горизонтальная позволяет изменять значение цвета.

⁵Употребляется также название HSB (англ. Hue, Saturation, Brightness — тон, насыщенность, яркость.)

Задачи

Пусть задано исходное изображение размерности $m \times n$ пикселей. Требуется подвергнуть его поэлементному (попиксельному) преобразованию f . Пусть $X_{i,j}$ — цвет пиксела с растровыми координатами (i, j) ($i = 1, \dots, m$, $j = 1, \dots, n$) исходного изображения, $Y_{i,j}$ — цвет соответствующего пиксела, получаемого после преобразования изображения, т. е. преобразование в общем случае задаётся отображением

$$Y_{i,j} = f(X_{i,j}).$$

Например, пусть задано полноцветное (24 бита) изображение, требуется преобразовать его в полутоновое (цвета от чёрного до белого с 256-ю градациями серого цвета). Тогда, как ясно из рассмотрения цветового куба, это преобразование можно задать соотношением

$$(R, G, B) \rightarrow \left(\frac{R + G + B}{3}, \frac{R + G + B}{3}, \frac{R + G + B}{3} \right),$$

где R, G, B — компоненты красного, зелёного и синего цветов цвета $X_{i,j}$.

Далее, в условиях задач, исходное изображение — полутоновое. Требуется подвергнуть его поэлементному преобразованию, указанному в заданиях.

1. Линейное контрастирование. Преобразование задаётся формулой

$$Y_{i,j} = (X_{i,j} - X_{\min}) \frac{Y_{\max} - Y_{\min}}{X_{\max} - X_{\min}} + Y_{\min},$$

где X_{\min} , X_{\max} — минимальное и максимальное значение интенсивностей входного изображения. Y_{\min} , Y_{\max} — желаемое минимальное и максимальное значение интенсивностей серого цвета полученного изображения. Написать программу, производящую линейное контрастирование. Входные параметры: файл изображения в формате BMP, Y_{\min} , Y_{\max} . Предусмотреть возможность сохранения полученной картинki.



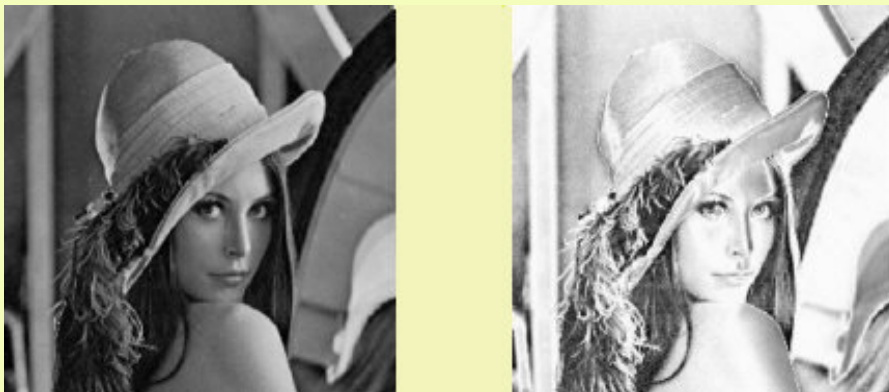
назад

заккрыть

2. Соляризация изображения (лат. solaris — солнечный, sol — солнце) — специальная фотографическая техника трансформации изображения. Фотографический эффект, основанный на получении частично обращённого изображения, используется в художественных фотографиях, рекламе, фильмах и т. д. Задаётся преобразованием

$$Y_{i,j} = kX_{i,j}(X_{\max} - X_{i,j}),$$

где k — константа, X_{\max} — максимальная интенсивность цвета входного изображения. Написать программу, производящую соляризацию. Входные параметры: файл изображения в формате BMP, k . Предусмотреть возможность сохранения полученной картинki.



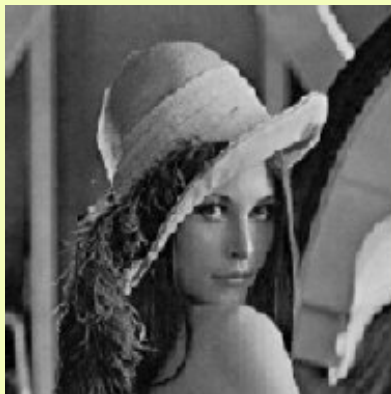
назад

закрыть

3. Бинаризация изображения. Конвертация полутонового изображения в бинарное (чёрно-белое) задаётся преобразованием

$$Y_{i,j} = \begin{cases} 0 \text{ (чёрный)}, & \text{если } X_{i,j} < T; \\ 1 \text{ (белый)}, & \text{иначе.} \end{cases}$$

где T — пороговое значение интенсивности цвета. Входные параметры: файл изображения в формате BMP, T . Предусмотреть возможность сохранения полученной картинке.



Список литературы

1. Корриган, Дж. Компьютерная графика: Секреты и решения. М.: Энтроп, 1995.
2. Роджерс Д., Адамс Дж. Математические основы машинной графики. М.: Мир, 2001.
3. Ф. В. Казанцев. Основы компьютерной графики. Казань, 2001.
4. Николай Тюкачев и др. Программирование графики в Delphi. СПб.: БХВ-Петербург, 2008.

[назад](#)[закреть](#)