

МИНИСТЕРСТВО ОБЩЕГО  
И ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

П.В.Вельтмандер

КОМПЬЮТЕРНАЯ ГРАФИКА  
(Учебное пособие в 3-х книгах)

Книга 1

ВВОДНЫЙ КУРС

Новосибирск  
1998

УДК 681.3.06  
ББК В 185.2я73–1

Вельтмандер П.В. Машинная графика. Вводный курс: Учеб. пособие/ Новосиб. ун — т. Новосибирск, 1997. 123 с., ил.

ISBN 5–230–13583–2

В вводном курсе дается общее введение в машинную графику — история, предмет, области приложений, затем рассмотрены зрительный аппарат человека, модели цветов в машинной графике, устройство и параметры технических средств документирования, архитектура векторных и растровых дисплеев, дисплеи на альтернативных принципах, устройства ввода, аппаратная реализация некоторых базовых алгоритмов.

Рецензент  
канд. физ.-мат. наук, С.И. Упольников

ISBN 5–230–13583–2

© Новосибирский государственный университет, 1997

# Оглавление

Введение . . . . .	9
0.1 История, предмет, приложения компьютерной графики . . . . .	10
0.1.1 Классификация устройств вывода . . . . .	10
0.1.2 История компьютерной графики . . . . .	10
0.1.3 Направления компьютерной графики . . . . .	13
0.1.4 Приложения компьютерной графики . . . . .	15
0.2 Зрительный аппарат человека . . . . .	21
0.2.1 Устройство глаза . . . . .	21
0.2.2 Чувствительность глаза . . . . .	22
0.3 Физические принципы формирования оттенков . . . . .	24
0.4 Цветовые модели . . . . .	26
0.4.1 Задание цветов . . . . .	28
0.4.2 Интерполяция цветов . . . . .	29
0.5 Печатающие устройства . . . . .	30
0.5.1 Разрешение устройств . . . . .	30
0.5.2 Матричный принтер . . . . .	32
0.5.3 Струйная печать . . . . .	33
0.5.4 Лазерный принтер . . . . .	36
0.5.5 Принтеры на твердых красителях . . . . .	38
0.6 Графопостроители . . . . .	40
0.6.1 Планшетные графопостроители . . . . .	40
0.6.2 Графопостроители с перемещающимся носителем . . . . .	40
0.6.3 Электростатические графопостроители . . . . .	41
0.7 Электронно-лучевые трубки . . . . .	44
0.7.1 Черно-белые кинескопы . . . . .	44
0.7.2 Цветные кинескопы . . . . .	44
0.7.3 Разрешение мониторов . . . . .	49
0.7.4 Люминофоры . . . . .	49
0.7.5 Гамма-коррекция . . . . .	50
0.8 Дисплеи с произвольным сканированием луча (каллиграфические или векторные дисплеи) . . . . .	54
0.8.1 Дисплейный генератор . . . . .	56
0.8.2 Преобразования . . . . .	61
0.8.3 Дисплейный контроллер . . . . .	62
0.8.4 Варианты архитектуры графических систем . . . . .	62
0.8.5 Области применения векторных дисплеев . . . . .	64

0.9	Растровые дисплеи . . . . .	65
0.9.1	Видеоконтроллер . . . . .	65
0.9.2	Создание изображения . . . . .	72
0.9.3	Видеопамять . . . . .	77
0.10	Другие типы дисплеев . . . . .	82
0.10.1	Дисплеи на запоминающей трубке . . . . .	82
0.10.2	Дисплеи с плазменной панелью . . . . .	83
0.10.3	Дисплеи с жидкокристаллическим индикатором . . . . .	86
0.10.4	Дисплеи с электролюминисцентным индикатором . . . . .	90
0.10.5	Дисплеи с эмиссией полей . . . . .	90
0.11	Дисплейные подсистемы IBM PC . . . . .	92
0.11.1	Видеоадаптеры . . . . .	92
0.11.2	Мониторы . . . . .	92
0.11.3	Замечание об адресации в IBM PC . . . . .	94
0.11.4	Архитектура видеоадаптеров . . . . .	94
0.11.5	Видеоадаптер CGA . . . . .	94
0.11.6	Видеоадаптеры EGA/VGA . . . . .	95
0.12	Устройства ввода . . . . .	99
0.12.1	Клавиатуры (Keyboards) . . . . .	99
0.12.2	Кнопки (Buttons) . . . . .	99
0.12.3	Световое перо (Lightpen) . . . . .	99
0.12.4	Планшеты (Tablets) . . . . .	100
0.12.5	“Мышь” (Mouse), трекбол (Trackball), джойстик (Joystick) . . . . .	105
0.12.6	Потенциометр . . . . .	106
0.12.7	Растровый сканер . . . . .	107
0.12.8	Кодировщик . . . . .	108
0.13	Нетрадиционные устройства . . . . .	109
0.13.1	Сенсорная панель (Touch Screen) . . . . .	109
0.13.2	Речевой диалог . . . . .	109
0.13.3	Средства диалога для систем виртуальной реальности . . . . .	110
0.14	Приложение 1. Преобразование моделей цветов . . . . .	116
0.14.1	V_HSVRGB — перевод из HSV в RGB . . . . .	116
0.14.2	V_RGBHSV — перевод из RGB в HSV . . . . .	117
0.14.3	Тест процедур V_HSVRGB и V_RGBHSV . . . . .	118
0.14.4	V_HLSRGB — перевод из HLS в RGB . . . . .	119
0.14.5	V_RGBHLS — перевод из RGB в HLS . . . . .	120
0.14.6	Тест процедур V_HLSRGB и V_RGBHLS . . . . .	122

# Список рисунков

0.1.1	Последовательность процесса познания . . . . .	14
0.1.2	Общий цикл разработки изделия . . . . .	16
0.1.3	Обжatie капсулы с газом. . . . .	18
0.2.1	Поперечный разрез глаза . . . . .	22
0.2.2	Спектральная чувствительность глаза . . . . .	23
0.3.1	Модели смешения цветов . . . . .	24
0.3.2	Цвет несамосветящегося объекта . . . . .	25
0.4.1	Цветовой куб модели RGB . . . . .	26
0.4.2	Цветовой куб модели CMY . . . . .	26
0.4.3	Цветовая модель HSV . . . . .	27
0.4.4	Цветовая модель HLS . . . . .	28
0.5.1	Классификация механических методов печати . . . . .	30
0.5.2	Классификация немеханических методов печати . . . . .	31
0.5.3	Классификация методов печати текстов . . . . .	31
0.5.4	Физическое разрешение растровой печати . . . . .	32
0.5.5	Схема устройства печатающей головки матричного принтера . . . . .	32
0.5.6	Схема устройства отдельного сопла в струйном принтере . . . . .	34
0.5.7	Схема образования капель в струйном принтере фирмы HP . . . . .	34
0.5.8	Управление формой и полетом капель в новых принтерах Stylus . . . . .	35
0.5.9	Струйная печать по технологиям Micro Dot и традиционной . . . . .	35
0.5.10	Схема устройства черно-белого лазерного принтера . . . . .	36
0.5.11	Изменение формы пятна в лазерном принтере . . . . .	37
0.5.12	Схема устройства цветного лазерного принтера . . . . .	37
0.5.13	Схема устройства принтера на твердых красителях . . . . .	39
0.5.14	Принтер Phaser 840 на твердых красителях . . . . .	39
0.6.1	Схема планшетного графопостроителя . . . . .	40
0.6.2	Схема графопостроителя с перемещающимся носителем . . . . .	41
0.6.3	Направления перемещения при использовании шаговых двигателей . . . . .	41
0.6.4	Схема электростатического графопостроителя . . . . .	42
0.7.1	Устройство черно-белого кинескопа . . . . .	44
0.7.2	Общая схема цветного кинескопа . . . . .	45
0.7.3	Устройство цветного кинескопа . . . . .	45
0.7.4	Электронно-оптическая система цветного кинескопа . . . . .	45
0.7.5	Схема формирования цветов в масочном кинескопе . . . . .	46
0.7.6	Структура экрана и маски планарного кинескопа . . . . .	46
0.7.7	Структура экрана и маски трехлучевого тринитрона . . . . .	47

0.7.8	Принцип работы трехлучевого хроматрона . . . . .	47
0.7.9	Принцип работы однолучевого хроматрона . . . . .	48
0.7.10	Принцип работы индексного кинескопа . . . . .	48
0.7.11	Определение количества строк разложения . . . . .	50
0.7.12	Разрешение цветного кинескопа . . . . .	50
0.7.13	Зависимость световой эмиссии люминофора от времени . . . . .	50
0.8.1	Общая архитектура простой графической системы . . . . .	55
0.8.2	Структура цифрового генератора векторов . . . . .	59
0.8.3	Генерация вектора по алгоритму Брезенхема . . . . .	59
0.8.4	Варианты использования генератора векторов . . . . .	60
0.8.5	Модель процесса вывода в графической системе . . . . .	63
0.9.1	Основные компоненты растрового дисплея . . . . .	66
0.9.2	Чересстрочная развертка . . . . .	66
0.9.3	Потери на обратный ход строчной и кадровой разверток . . . . .	66
0.9.4	Блок-схема простого видеоконтроллера . . . . .	68
0.9.5	Таблица цветности для дисплея с 6-ти битным пикселом . . . . .	69
0.9.6	Преобразование изображений в темпе разверток . . . . .	71
0.9.7	Закраска многоугольников “на лету” . . . . .	72
0.9.8	Кодирование изображения длинами серий в виде списка . . . . .	74
0.9.9	Простой текстовый дисплей . . . . .	75
0.9.10	Сравнение двух подходов к мультипроцессированию . . . . .	77
0.9.11	Простейший растровый дисплей . . . . .	79
0.9.12	Распределение цикла памяти в растровой системе . . . . .	79
0.9.13	Структура VRAM . . . . .	81
0.10.1	Схема запоминающей электронно-лучевой трубки . . . . .	82
0.10.2	Схема плазменной панели . . . . .	84
0.10.3	Вольтамперная характеристика разряда в газе . . . . .	84
0.10.4	Временная диаграмма работы плазменной ячейки . . . . .	85
0.10.5	Ячейка цветного дисплея с плазменной панелью . . . . .	85
0.10.6	Структурная формула жидкого кристалла . . . . .	86
0.10.7	Фазы жидкого кристалла . . . . .	87
0.10.8	Прохождение неполяризованного света через поляризаторы . . . . .	88
0.10.9	Вакрученный нематический жидкий кристалл . . . . .	88
0.10.10	Электронно-оптическое переключение жидким кристаллом . . . . .	88
0.10.11	Схема жидкокристаллического индикатора . . . . .	89
0.10.12	Схема электролюминисцентного индикатора . . . . .	90
0.10.13	Схема индикатора с эмиссией поля . . . . .	91
0.11.1	Структура видеопамати для CGA . . . . .	95
0.11.2	Блок-схема видеоадаптеров EGA/VGA . . . . .	96
0.11.3	Блок-схема видеоадаптера VGA . . . . .	96
0.11.4	Структура видеопамати для VGA-режима 320×200×256 . . . . .	97
0.12.1	Схема работы светового пера . . . . .	101
0.12.2	Устройство потенциометрического планшета . . . . .	102
0.12.3	Устройство акустического планшета . . . . .	102
0.12.4	Причина ошибки в акустическом планшете . . . . .	103

0.12.5	Устройство планшета с емкостным съемом информации . . . . .	104
0.12.6	Устройств магнитострикционного планшета . . . . .	105
0.12.7	Устройство механической мышки . . . . .	106
0.13.	Перчатка данных фирмы VPL . . . . .	111

# Список таблиц

0.7.1	Параметры некоторых люминофоров . . . . .	51
0.7.2	Зависимость моделей цветов от системной гаммы . . . . .	53
0.9.1	Характерные времена для некоторых мониторов . . . . .	67
0.9.2	Параметры некоторых микросхем памяти . . . . .	80
0.10.	Параметры дисплеев с плазменной панелью . . . . .	87
0.11.	Характеристики видеоадаптеров IBM PC . . . . .	93



# Введение

В данной, первой части курса лекций дается общее введение в компьютерную графику. Основное внимание при этом уделяется техническим средствам.

В первом разделе рассмотрены история, основные направления и некоторые приложения компьютерной графики.

В втором разделе рассмотрены зрительный аппарат человека и психофизиологические особенности восприятия цветов, которые необходимо учитывать при построении систем компьютерной графики.

В третьем разделе рассмотрены физические принципы формирования оттенков — аддитивный, используемый для самосветящихся объектов, и субтрактивный, используемый для несамосветящихся объектов.

В четвертом разделе рассмотрены основные цветовые модели, используемые в компьютерной графике: аппаратно-ориентированные модели — RGB, CMY, YIQ и ориентированные на человека — HSV, HLS.

В пятом разделе рассмотрены классификация методов печати, принципы работы основных принтеров — матричных, струйных, лазерных, на твердых красителях и протоколы их работы.

В шестом разделе приведена классификация графопостроителей и дано описание работы основных их типов — планшетных, с перемещающимся носителем (рулонных, с абразивной головкой, барабанных) и электростатических.

В седьмом разделе рассмотрены основные устройства отображения в компьютерной графике — электроннолучевые трубки, их основные типы, люминофор и соотношения, определяющие качество отображения.

В восьмом разделе рассмотрены дисплеи с произвольным сканированием луча (векторные или каллиграфические дисплеи), дисплеи с растровым сканированием луча (растровые дисплеи), их архитектура и работа основных компонент.

В девятом разделе рассмотрены дисплеи, использующие другие типы индикаторов, — запоминающую трубку, плазменную панель, жидкокристаллический индикатор и электролюминесцентную панель.

В десятом разделе рассмотрены архитектуры дисплейных подсистем IBM PC.

В одиннадцатом разделе рассмотрены устройства ввода, используемые в компьютерной графике.

В двенадцатом, заключительном разделе рассмотрены нетрадиционные устройства (сенсорная панель, речевой диалог, а также устройства, используемые в системах виртуальной реальности).

В приложении приведены алгоритмы перехода от одной цветовой модели к другой.

## 0.1 История, предмет, приложения компьютерной графики

В данном разделе:

- рассмотрена история развития компьютерной графики, начиная от ее появления в 1950 г;
- пояснены основные направления компьютерной графики:
  - изобразительная компьютерная графика,
  - обработка и анализ изображений,
  - анализ сцен (перцептивная компьютерная графика),
  - компьютерная графика для научных абстракций (когнитивная компьютерная графика — способствующая познанию);
- дано общее описание некоторых важных приложений компьютерной графики, подробно рассматривается приложение для компьютерного моделирования, как нового инструмента исследований, и направление “виртуальная реальность”, возникшее в последние годы.

### 0.1.1 Классификация устройств вывода

Устройства вывода можно классифицировать следующим образом:

1. По принципам записи (обновления) изображения:

- с произвольным сканированием луча, при котором изображение формируется при перемещении луча по экрану в соответствии с координатами строящихся элементов изображения (каллиграфические, штриховые устройства);
- с растровым сканированием луча, при котором изображение представляется в виде матрицы точек. Изображение на экране формируется при перемещении луча в соответствии с разверткой слева-направо по строке и сверху-вниз по строкам с подсветкой требуемых точек.

2. По принципам отображения:

- периодическая регенерация информации на экране из неотображающей памяти;
- использование отображающего устройства сохранения изображения.

3. По технологическим способам вывода (свечение люминофора, вычерчивание пишущим узлом, перенос красителя и т.п.).

### 0.1.2 История компьютерной графики

Развитие компьютерной графики, особенно на ее начальных этапах, в первую очередь связано с развитием технических средств и в особенности дисплеев. Рассмотрение начнем с истории развития технологий вывода (векторные, растровые и иные дисплеи), затем приведем краткую хронологию становления дисплейной техники у нас в стране и в заключение рассмотрим этапы развития методов и приложений.

#### История технологий вывода

**Произвольное сканирование луча.** Дисплейная графика появилась как попытка использовать электроннолучевые трубки (ЭЛТ) с произвольным сканированием луча для вывода изображения из ЭВМ. Как пишет Ньюмен [1] “по-видимому, первой машиной, где ЭЛТ использовалась в качестве устройства вывода была ЭВМ Whirlwind-I (Ураган-I), изготовленная в 1950 г.” в

Массачусетском технологическом институте. С этого эксперимента начался этап развития векторных дисплеев (дисплеев с произвольным сканированием луча, каллиграфических дисплеев). На профессиональном жаргоне вектором называется отрезок прямой. Отсюда и происходит название “векторный дисплей”.

При перемещении луча по экрану в точку, на которую попал луч, возбуждается свечение люминофора экрана. Это свечение достаточно быстро прекращается при перемещении луча в другую позицию (обычное время послесвечения — менее 0.1 с). Поэтому, для того чтобы изображение было постоянно видимым, приходится его перевыдавать (регенерировать изображение) 50 или 25 раз в секунду. Необходимость перевыдачи изображения требует сохранения его описания в специально выделенной памяти, называемой памятью регенерации. Само описание изображения называется дисплейным файлом. Понятно, что такой дисплей требует достаточно быстрого процессора для обработки дисплейного файла и управления перемещением луча по экрану.

Обычно серийные векторные дисплеи успевали 50 раз в секунду строить только около 3000–4000 отрезков. При большем числе отрезков изображение начинает мерцать, так как отрезки, построенные в начале очередного цикла, полностью погасают к тому моменту, когда будут строиться последние.

Другим недостатком векторных дисплеев является малое число градаций по яркости (обычно 2–4). Были разработаны, но не нашли широкого применения двух-трехцветные ЭЛТ, также обеспечивавшие несколько градаций яркости.

В векторных дисплеях легко стереть любой элемент изображения — достаточно при очередном цикле построения удалить стираемый элемент из дисплейного файла.

Текстовый диалог поддерживается с помощью алфавитно-цифровой клавиатуры. Косвенный графический диалог, как и во всех остальных дисплеях, осуществляется перемещением перекрестия (курсора) по экрану с помощью тех или иных средств управления перекрестием — координатных колес, управляющего рычага (джойстика), трекбола (шаровой рукоятки), планшета и т.д. Отличительной чертой векторных дисплеев является возможность непосредственного графического диалога, заключающаяся в простом указании с помощью светового пера объектов на экране (линий, символов и т.д.). Для этого достаточно с помощью фотодиода определить момент прорисовки (и следовательно начала свечения люминофора) любой части требуемого элемента.

Векторные дисплеи обычно подключаются к ЭВМ высокоскоростными каналами связи.

Первые серийные векторные дисплеи за рубежом появились в конце 60-х годов. В 1963 г. был разработан прототип дисплейной станции IBM 2250 (до осени 1964 г. работы были засекречены).

Подробному рассмотрению векторных дисплеев посвящен разд. 0.8.

**Растровое сканирование луча.** Прогресс в технологии микроэлектроники привел к тому, с середины 70-х годов подавляющее распространение получили дисплеи с растровым сканированием луча. Подробному рассмотрению этих дисплеев посвящен разд. 0.9.

**Запоминающие трубки.** В конце 60-х годов появилась запоминающая ЭЛТ, которая способна достаточно длительное время (до часа) прямо на экране хранить построенное изображение. Следовательно, не обязательна память регенерации и не нужен быстрый процессор для выполнения регенерации изображения. Стирание на таком дисплее возможно только для всей картинки в целом. Сложность изображения практически не ограничена. Разрешение, достигнутое на дисплеях на запоминающей трубке, такое же как и на векторных или выше — до 4096 точек.

Текстовый диалог поддерживается с помощью алфавитно-цифровой клавиатуры, косвенный графический диалог осуществляется перемещением перекрестия по экрану обычно с помощью координатных колес.

Такие дисплеи подключаются к ЭВМ низкоскоростными каналами связи (типа телефонных).

Появление таких дисплеев с одной стороны способствовало широкому распространению компьютерной графики, с другой стороны представляло собой определенный регресс, так как распространялась сравнительно низкокачественная и низкоскоростная, не слишком интерактивная графика. Более подробно о таких дисплеях см. в п. 0.10.1.

**Плазменная панель.** В 1966 г. была изобретена плазменная панель, которую упрощенно можно представить как матрицу из маленьких разноцветных неоновых лампочек, каждая из которых включается независимо и может светиться с регулируемой яркостью. Ясно, что системы отклонения не нужно, не обязательна также и память регенерации, так как по напряжению на лампочке можно всегда определить горит она ли нет, т.е. есть или нет изображение в данной точке. В определенном смысле эти дисплеи объединяют в себе многие полезные свойства векторных и растровых устройств. К недостаткам следует отнести большую стоимость, недостаточно высокое разрешение и большое напряжение питания. В целом эти дисплеи не нашли широкого распространения. Более подробно о них см. п. 0.10.2.

**Жидкокристаллические индикаторы.** Дисплеи на жидкокристаллических индикаторах работают аналогично индикаторам в электронных часах, но, конечно, изображение состоит не из нескольких сегментов, а из большого числа отдельно управляемых точек. Эти дисплеи имеют наименьшие габариты и энергопотребление, поэтому широко используются в портативных компьютерах несмотря на меньшее разрешение, меньшую контрастность и заметно большую цену, чем для растровых дисплеев на ЭЛТ. Более подробно о дисплеях на жидкокристаллических индикаторах см. п. 0.10.3.

**Электролюминисцентные индикаторы.** Наиболее высокие яркость, контрастность, рабочий температурный диапазон и прочность имеют дисплеи на электролюминисцентных индикаторах. Благодаря достижениям в технологии они стали доступны для применения не только в дорогих высококлассных системах, но и в общепромышленных системах. Работа таких дисплеев основана на свечении люминофора под воздействием относительно высокого переменного напряжения, прикладываемого к взаимноперпендикулярным наборам электродов, между которыми находится люминофор. Более подробно о таких дисплеях см. п. 0.10.4.

**Дисплеи с эмиссией поля.** Дисплеи на электронно-лучевых трубках, несмотря на их относительно дешевизну и широкое распространение, механически непрочны, требуют высокого напряжения питания, потребляют большую мощность, имеют большие габариты и ограниченный срок службы, связанный с потерей эмиссии катодами. Одним из методов устранения указанных недостатков, является создание плоских дисплеев с эмиссией поля с холодных катодов в виде сильно заостренных микроигл. Более подробно о таких дисплеях см. п. 0.10.5.

## **Хронология некоторых отечественных разработок**

В следующем ниже перечислении приведены далеко не все проведенные в стране разработки. К сожалению по многим из них у меня не было точной информации. Могу лишь упомянуть, что дисплеи разрабатывались в Новосибирске, Протвино, Ленинграде, Москве, Зеленограде, Воронеже, Львове, Виннице, Минске (ИТК АН БССР), Киеве и, конечно, во многих других. Одним

из первых векторных дисплеев был дисплей УПГИ, серийно выпускавшийся для комплектации АРМ машиностроительного направления. Для автоматизации проектирования печатных плат и простого цветного вывода использовался дисплей ЦРД (цветной растровый дисплей). Вероятно первой законченной конструкторской проработкой векторного дисплея был "экран динамического вывода", выполненный в начале 60-х годов.

- 1968, ВЦ АН СССР, машина БЭСМ-6, вероятно, первый отечественный растровый дисплей, видеопамять на магнитном барабане весом 400 кг;
- 1972, Институт автоматики и электрометрии (ИАиЭ), векторный дисплей Символ;
- 1973, ИАиЭ, векторный дисплей Дельта. Малая серия — 1974 Институт прикладной физики (ИПФ), Новосибирск. Серия — 1975, завод "Луч", Новосибирск;
- 1977, ИАиЭ, векторный дисплей ЭПГ-400. Серия ~ 1979;
- 1979, ИПФ, цветной полутонный растровый дисплей Гамма-1 —  $256 \times 256 \times 6$  бит;
- 1981–1982, векторные дисплеи ЭПГ-СМ (ИАиЭ), ЭПГ-2СМ. Серия Серия ~ 1983;
- 1982, Киев, НИИ периферийного оборудования, векторный дисплей СМ-7316 (4000–6000 векторов, 4096 символов, разрешение  $2048 \times 2048$ );
- 1982, ИПФ, растровый цветной полутонный дисплей Гамма-2.  $512 \times 512 \times 8$  бит, таблица цветности, поддержка окон;
- 1983, ИПФ, растровая цветная полутонная дисплейная станция Гамма-4.  $1024 \times 768 \times 8$  бит, таблица цветности, поддержка окон, видеопамять 4 Мбайта, прокрутка изображения, плавное, переменное масштабирование. Серия ~ 1985–1986;
- 1984, ИПФ, растровое расширение символьных терминалов Гамма-5.  $512 \times 256 \times 1$  бит, прокрутка изображения. Серия 1984;

Итак, стартовав в 1950 г., компьютерная графика к настоящему времени прошла путь от экзотических экспериментов до одного из важнейших, всепроникающих инструментов современной цивилизации, начиная от научных исследований, автоматизации проектирования и изготовления, бизнеса, медицины, экологии, средств массовой информации, досуга и кончая бытовым оборудованием. Можно выделить следующие этапы развития:

- 60–70-е годы — научная дисциплина. Бурное развитие методов, алгоритмов — отсечение, генерация примитивных графических элементов, закраска узорами, реалистическое представление сцен (удаление невидимых линий и граней, трассировка лучей, излучающие поверхности);
- 80-е годы — прикладная наука. Отработка методов, средств, аппаратуры в различных сферах приложений;
- • 90-е годы — основное средство общения человека с ЭВМ.

### 0.1.3 Направления компьютерной графики

В нынешнем, устоявшемся состоянии принято разделять компьютерную графику на следующие направления:

- изобразительная компьютерная графика,
- обработка и анализ изображений,
- анализ сцен (перцептивная компьютерная графика),
- компьютерная графика для научных абстракций (когнитивная компьютерная графика — графика, способствующая познанию).

## Изобразительная компьютерная графика

Объекты: синтезированные изображения. Задачи:

- построение модели объекта и генерация изображения,
- преобразование модели и изображения,
- идентификация объекта и получение требуемой информации.

### Обработка и анализ изображений

Объекты: дискретное, числовое представление фотографий. Задачи:

- повышение качества изображения,
- оценка изображения — определение формы, местоположения, размеров и других параметров требуемых объектов,
- распознавание образов — выделение и классификация свойств объектов (обработка аэрокосмических снимков, ввод чертежей, системы навигации, обнаружения и наведения).

Итак, в основе обработки и анализа изображений лежат методы представления, обработки и анализа изображений плюс, естественно, изобразительная компьютерная графика хотя бы для того, чтобы представить результаты.

### Анализ сцен

Предмет: исследование абстрактных моделей графических объектов и взаимосвязей между ними. Объекты могут быть как синтезированными, так и выделенными на фотоснимках.

Первый шаг в анализе сцены — выделение характерных особенностей, формирующих графический объект(ы).

Примеры: машинное зрение (роботы), анализ рентгеновских снимков с выделением и отслеживанием интересующего объекта, например, сердца.

Итак, в основе анализа сцен (перцептивной компьютерной графики) находятся изобразительная графика + анализ изображений + специализированные средства.

### Когнитивная компьютерная графика

Только формирующееся новое направление, пока недостаточно четко очерченное.

Это компьютерная графика для научных абстракций, способствующая рождению нового научного знания. База — мощные ЭВМ и высокопроизводительные средства визуализации.

Общая последовательность познания заключается в, возможно циклическом, продвижении от гипотезы к модели (объекта, явления) и решению, результатом которого является знание (рис. 0.1.1).

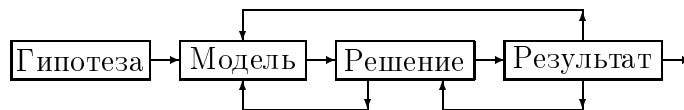


Рис. 0.1.1: Последовательность процесса познания

Человеческое познание использует два основных механизма мышления, за каждым из которых закреплена половина мозга:

- осознанное, логико-вербальное, манипулирует абстрактными последовательностями символов (объектов) + семантика символов + прагматические представления, связанные с символами. Возраст этого механизма, связанного с наличием речи, — до 100 тыс. лет;
- неосознанное, интуитивное, образное, работает с чувственными образами и представлениями о них. Возраст этого механизма — время существования на Земле животного мира.

Первоначально ЭВМ имели малую производительность процессоров и средств компьютерной графики, т.е. по сути дела имели возможность работы только с символами (некоторый упрощенный аналог логического мышления).

С появлением супер-ЭВМ, производительностью в миллиард и более операций в секунду и графических супер-станций, производительностью до сотен миллионов операций в секунду, появилась возможность достаточно эффективного манипулирования образами (картинами).

Важно отметить, что мозг не только умеет работать с двумя способами представления информации, причем с образами он работает иначе и эффективнее чем ЭВМ, но и умеет соотносить эти два способа и совершать (каким-то образом) переходы от одного представления к другому.

В этом контексте основная проблема и задача когнитивной компьютерной графики — создание таких моделей представления знаний, в которых можно было бы однообразно представлять как объекты, характерные для логического (символического, алгебраического) мышления, так и объекты, характерные для образного мышления.

Другие важнейшие задачи:

- визуализация тех знаний, для которых не существует (пока ?) символических описаний,
- поиск путей перехода от образа к формулировке гипотезы о механизмах и процессах, представленных этими (динамическими) образами на экране дисплея.

Появление когнитивной компьютерной графики — сигнал о переходе от эры экстенсивного развития естественного интеллекта к эре интенсивного развития, характеризующегося глубоко проникающей компьютеризацией, рождающей человеко-машинную технологию познания, важным моментом которой является непосредственное, целенаправленное, активирующее воздействие на подсознательные интуитивные механизмы образного мышления.

Одним из ярких и наиболее ранних примеров приложения когнитивной компьютерной графики является работа Ч. Страуса “Неожиданное применение ЭВМ в чистой математике” (ТИИ-ЭР, т. 62, N 4, 1974, с. 96 – 99). В ней показано как для анализа сложных алгебраических кривых используется “n-мерная” доска на основе графического терминала. Пользуясь устройствами ввода математик может легко получать геометрические изображения результатов направленного изменения параметров исследуемой зависимости. Он может также легко управлять текущими значениями параметров, “углубляя тем самым свое понимание роли вариаций этих параметров”. В результате получено “несколько новых теорем и определены направления дальнейших исследований”.

#### 0.1.4 Приложения компьютерной графики

Как уже отмечалось, компьютерная графика стала основным средством взаимодействия человека с ЭВМ. Важнейшими сформировавшимися областями приложений являются:

- компьютерное моделирование, которое явилось исторически первым широким применением компьютерной графики,
- системы автоматизации научных исследований, системы автоматизации проектирования, системы автоматизации конструирования, системы автоматизации производства, автоматизированные системы управления технологическими процессами,
- бизнес,
- искусство,
- средства массовой информации,
- досуг.

В настоящее время появилось новое, очень интересное приложение компьютерной графики — виртуальная реальность.

По телевидению часто можно видеть передачи иллюстрирующие приложения компьютерной графики в автоматизации проектирования (были передачи об автоматизированном проектировании самолетов, автомобилей), много передач об автоматизации производства с различными робототехническими системами.

Передачи о мире бизнеса практически не обходятся без показа различной дисплейной техники и ее использования.

Что касается искусства, то достаточно упомянуть, что один из самых крупных первых суперкомпьютерных центров мира находился на студии Уолта Диснея и использовался для подготовки мультфильмов. Всем известно, что многие “жутики” и боевики также готовились с широким использованием средств компьютерной графики для подготовки высокореалистичных сцен.

Применение компьютерной графики в средствах массовой информации мы видим ежедневно, как в виде различных заставок и телеэффектов на экране, так и в виде газет, при подготовке многих из которых используется электронная верстка на компьютере.

С компьютерными играми, отнимающими не только время досуга, конечно же знаком каждый.

Поэтому здесь мы рассмотрим, в основном, приложения компьютерной графики в компьютерном моделировании, а также немного познакомимся с самым новым приложением — системами виртуальной реальности.

### Компьютерная графика в интегрированной САПР

Общий цикл разработки какого-либо промышленного изделия, будь то радиатор батареи центрального отопления или же самолет можно представить в виде схемы, показанной на рис. 0.1.2.

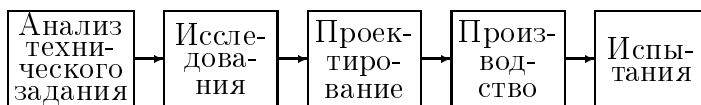


Рис. 0.1.2: Общий цикл разработки изделия

Естественно, что могут выполняться возвраты на повторное выполнение общего цикла, начиная с некоторого этапа.



## Компьютерная графика в компьютерном моделировании

Ясно, что в настоящее время на всех этапах разработки в той или иной мере используются ЭВМ и, следовательно, компьютерная графика. На этапе исследований важную роль, кроме натуральных экспериментов, играет компьютерное моделирование работы изделия и/или его компонент. Рассмотрим в качестве примера задачи моделирования быстропротекающих процессов механики сплошной среды. К таким задачам относятся задачи сварки и упрочнения взрывом, задачи расчета синтеза алмазов взрывом, задачи расчета защиты космических аппаратов от метеоритов и микрометеоритов и т.д.

Следует отметить что реальный физический процесс развивается за единицы или десятки микросекунд, поэтому натуральный эксперимент не только дорогостоящ, но и дает мало информации. Обычно это один или несколько рентгеновских снимков, прямые измерения, как правило, или затруднены, или датчики вносят заметные изменения в процесс.

В этой связи опыт специалистами в этой области набирается довольно долго — порядка десятка и более лет и зачастую носит интуитивный характер. Поэтому одна из важнейших функций компьютерного моделирования — быстрое получение опыта. Ведь по сути дела комплекс компьютерного моделирования при адекватности моделей, заложенных в него, представляет собой уникальную экспериментальную установку, на которой можно измерить (выдать на дисплей) любую величину, что угодно изменить, даже приостановить процесс для подробного анализа и пустить дальше. Попробуйте это сделать со взрывом!

Другая важная функция комплекса компьютерного моделирования, особенно при оснащении его средствами коллективного, управляемого просмотра машинных фильмов, показывающих процесс не отдельными картинками, а в динамике — унификация представлений о процессе у заинтересованных специалистов — исследователей, конструкторов, технологов, испытателей. Так как обычно, при использовании только числовой информации и без образного ее представления у разных специалистов формируются собственные представления (не обязательно у всех одинаковые и правильные).

В настоящее время широко используются двумерные задачи моделирования, когда важны две пространственных координаты. Это достаточно широкий круг реальных процессов, но в ряде случаев обязательно использование трехмерных задач, которые считаются много дольше. Основные проблемы, стоящие на пути широкого распространения трехмерных задач, состоят в том, что, во-первых, при существующих численных методах объем вычислений растет быстрее чем четвертая степень требуемой точности, во-вторых, визуализация результатов существенно усложняется и замедляется. В дальнейшем рассмотрении речь идет о двумерных задачах.

В качестве иллюстрации некоторых явлений, которые необходимо моделировать, на рис. 0.1.3 показан один из этапов решения осесимметричной задачи обжатия капсулы с газом с помощью взрыва.

Рассчитываемая область разбивается сеткой с общим числом узлов до 1–2 тысячи. Количество независимых переменных, рассчитываемых для узла, от 8 до 12 (скорость вдоль осей X и Y, давление, плотность, температура и т.д.).

### Количественные параметры задач

Расчет ведется небольшими шагами по времени. Вначале на одном шаге по времени для всех узлов рассчитываются все величины. Затем время увеличивается и расчет повторяется.

Время расчета одного шага от 3–4 с до 3–7 мин. в зависимости от используемого процессора и задачи. Т.е. время расчета 2048 шагов — от 1.2–1.7 часа до 102–287 часов, т.е. до 4–12 суток



Рис. 0.1.3: Обжатие капсулы с газом.

непрерывной работы процессора. (Для сравнения отметим, что по известным литературным данным процессорное время при моделировании на ЭВМ поляны цветов — около 18 часов, а ядерного устройства — до года. Так что рассматриваемые нами задачи не очень длительные).

Объем данных, перерабатываемых на одном шаге по времени — от 163\_840 чисел (маленькая задача с расчетной сеткой  $256 \times 128$  и 5 рассчитываемыми параметрами) до 3\_932\_160 чисел (большая задача с расчетной сеткой  $1024 \times 256$  и 15 рассчитываемыми параметрами). При использовании 32-х разрядных вещественных чисел это соответствует объему данных на шаге от 640 Кбайт до 15 Мбайт. (Для многих расчетов необходима 64-х разрядная арифметика, что приводит к удвоению объемов данных.)

Суммарный объем перерабатываемых данных при 2048 шагах по времени — от 1.28 до 30 Гбайт.

Представим себе что все эти гигабайты мы напечатали бы на листах стандартной писчей бумаги размером  $210 \times 297$  мм толщиной порядка 0.1 мм через 1.5 интервала, т.е. 40 строк по 62 символа в каждой. На изображение одного 32-х разрядного вещественного числа надо 12 символов:  $\pm.XXXXXXX\pm XX$ .

На маленькую задачу потребовалось бы:

$$\frac{256 \cdot 128 \cdot 5 \cdot 2048 \cdot 12}{40 \cdot 62} \simeq 1\,623\,602$$

Если сложить эти листы, то

высота стопки составила бы  $1\,623\,602 \times 0.1$  мм  $\simeq 162$  метра,  
длина полосы составила бы  $1\,623\,602 \times 297$  мм  $\simeq 482$  километра.

На большую задачу потребовалось бы:

$$\frac{1024 \cdot 256 \cdot 15 \cdot 2048 \cdot 12}{40 \cdot 62} \simeq 38\,966\,437.$$

Если сложить эти листы, то

высота стопки составила бы  $38\,966\,437 \times 0.1$  мм  $\sim 3\,897$  м,  
длина полосы составила бы  $38\,966\,437 \times 297$  мм  $\sim 11\,573$  километра.

Очевидно, что выдача результатов задачи в числовом виде, когда длина выдачи была бы больше, чем расстояние от Москвы до Владивостока (оно равно  $\simeq 11$  тысяч километров), совершенно лишена смысла.

ЭВМ настолько много и быстро продуцирует результатов, что без новой технологии их обработки моделирование становится просто бессмысленным. Можно предложить два способа представления результатов в форме удобной для восприятия и анализа человеком:

- ЭВМ изыскивает простую, физически оправданную аппроксимацию результатов в виде аналитических формул. Этот подход ориентирован на логическое мышление и, вообще говоря, для рассматриваемых задач трудно реализуем за исключением быть может каких-то специальных, модельных постановок;
- ЭВМ в процессе счета формирует машинные фильмы, показывающие изменение расчетных величин в процессе счета. Так как за один шаг по времени изменение, вообще говоря, мало, то вывод делается через некоторое количество шагов, например, 10. Таким образом, при 2048 шагах счета для одной величины надо запомнить около 200 кадров. Для вывода распределений скалярных величин, например, давлений, плотностей и т.д. удобно использовать растровые дисплеи. При этом одному узлу расчетной сетки соответствует 1 точка изображения, а цвет и яркость точки характеризуют значение рассчитываемой величины. При одном байте на точку изображения и с учетом того, что эти синтезированные изображения достаточно легко сжимаются простыми и быстрыми алгоритмами в среднем в 30 раз, получаем, что объем данных для фильма, показывающего изменение одной величины (для большой задачи) равен:

$$\frac{31024 \cdot 256 \cdot 2048}{10 \cdot 30} \simeq 1.7$$

что, вообще говоря, поменьше, чем  $30 \text{ Гбайт}/12 = 2.5 \text{ Гбайт}$

В последнем случае, правда, возникает другая проблема. Она состоит в том, что числовая информация утрачивается и остаются только визуальные образы, а для разработки конструкций конечно же нужна числовая информация. Здесь используются два подхода:

- просмотр и анализ фильма, выяснение интересующих мест и повторный пересчет с требуемого места,
- сброс всей числовой информации на большую внешнюю память, формирование и просмотр фильмов выполняются с использованием результатов из внешней памяти. Повторный пересчет с занятием дорогостоящего времени суперкомпьютера не требуется.

На существующем уровне технологии дисковых накопителей реально сохранять решение не более чем одной задачи, поэтому практически возможно использование только подхода с пересчетом. Кроме этого, обычно расчеты выполняются в режиме диалога с оперативным изменением параметров задачи, поэтому технологических проблем с возвратами не возникает. Конечно, вычислитель используется недостаточно эффективно.

## Виртуальная реальность

Виртуальная реальность — новейшее направление приложений компьютерной графики, позволяющее имитировать окружающую действительность с новым уровнем взаимодействия человек-ЭВМ. Основа систем виртуальной реальности — высокопроизводительная графическая рабочая станция, обладающая достаточным быстродействием и изобразительными возможностями для формирования высокореалистичных цветных полутоновых изображений. Устройства отображения в различных системах могут быть самыми различными — от обычных мониторов высокого разрешения до экранов во всю стену, используемых в имитаторах боевых действий, или же стереоскопических систем отображения, в том числе и в виде специальных очков, смонтированных в шлем, надеваемый на голову.

Одно из важнейших отличий систем виртуальной реальности от других систем отображения — наличие средств воздействия не только на зрение, но и на другие органы чувств. В первую очередь это системы стереозвука, имитирующие требуемое распределение и интенсивности источников звука в пространстве. Наиболее дорогие системы обеспечивают воздействие и на осязание за счет использования специальных шлемов, перчаток и костюмов, которые за счет встроенных в них устройств не только определяют положение головы, направление взгляда, положение рук, пальцев, тела, но и имитируют прикосновения, сопротивление или “податливость” ручек и т.д. Можно почувствовать прикосновение к объекту существующему лишь в памяти компьютера! Осталось симитировать запахи.

В настоящее время системы виртуальной реальности очень дороги. Самые дешевые стоят около 20 тыс долларов, более совершенные системы — около 100 тысяч долларов. Профессиональная перчатка для систем виртуальной реальности “дейта-глав”, оборудованная датчиками положения, стоит \$ 8800, ее упрощенный вариант для видеоигр — “пауэр-глав” стоит \$ 99 и позволяет играющим действовать без мышки и клавиш — просто мановением руки.

Из-за большой сложности и дороговизны разработка систем виртуальной реальности во всем мире финансируется, в основном, по военным программам. Первые шаги в этом направлении были сделаны еще в 60-е годы при разработке компьютерных тренажеров для летчиков. Но в них обеспечивалось воздействие только на зрение. К таким системам, например, принадлежат системы “Аксай” и “Альбатрос”, разработанные Институтом Автоматики и электрометрии. Известны также и т.н. полунатурные системы моделирования, имитирующие вибрации, тряску и эволюции самолета и работу бортового оборудования. Обучаемый при этом находится в кабине самолета, укрепленной на системе перемещающихся направляющих. Конечно, тренажеры очень дороги, так только один специальный шлем в тренажерах для военных летчиков стоит около 1 млн долларов.

На Западе сейчас появилось множество игр, использующих принципы систем виртуальной реальности. В 1992 г. в Чикаго (США) открылся центр виртуальных видеоигр, где клиент, заняв место в модели одноместной кабины космического корабля, играет в звездные войны. Цена билета — 7 долларов. За первые три месяца было продано 300 тысяч билетов!

Сейчас работы по виртуальной реальности на Западе рассматриваются как главное направление развития компьютерной отрасли в будущем, которое окажет важное влияние на конкурентоспособность стран. В целом пик развития этого направления ожидается в начале грядущего тысячелетия.

## 0.2 Зрительный аппарат человека

Системы отображения графической информации воздействуют на зрительный аппарат человека, поэтому с необходимостью должны учитывать как физические, так и психофизиологические особенности зрения.

### 0.2.1 Устройство глаза

На рис. 0.2.1 показан поперечный размер глазного яблока человека. Свет попадает в глаз через роговицу и фокусируется хрусталиком на внутренний слой глаза, называемый сетчаткой. Сетчатка преобразует свет в импульсы в нервных волокнах и состоит из трех слоев клеток. Удивительно то, что светочувствительные клетки, известные как колбочки и палочки, формируют слой клеток в задней части сетчатки. Таким образом, свет должен вначале пройти два слоя клеток, прежде чем он воздействует на колбочки и палочки. Причины для такого обратного устройства сетчатки не полностью поняты, но одно из объяснений состоит в том, что расположение светочувствительных клеток в задней части сетчатки позволяет любому паразитному непоглощенному свету попасть на клетки находящиеся непосредственно позади сетчатки, которые содержат черный пигмент — меланин. Клетки, содержащие меланин, также помогают химически восстанавливать светочувствительный визуальный пигмент в колбочках и палочках после того, как они были отбелены на свету.

Интересно отметить, что природа создала целый ряд конструкций глаза. При этом глаза у всех позвоночных похожи на глаза человека, а глаза у беспозвоночных либо сложные (фасеточные) как у насекомых, либо недоразвитые в виде светочувствительного пятна. Только у осьминогов глаза устроены как у позвоночных, но светочувствительные клетки находятся непосредственно на внутренней поверхности глазного яблока, а не как у нас позади других слоев, занимающихся предварительной обработкой изображения. Поэтому, возможно, особого смысла в обратном расположении клеток в сетчатке нет. А это просто один из экспериментов природы.

Итак, внутренний слой сетчатки глаза содержит два типа светочувствительных рецепторов, занимающих область с раствором около  $170^\circ$  относительно зрительной оси:

- $\simeq 100$  млн. палочек (длинные и тонкие рецепторы ночного зрения),
- $\simeq 6.5$  млн. колбочек (короткие и толстые рецепторы дневного зрения).

Информация от рецепторов передается в мозг по зрительному нерву, содержащему около 800 тысяч волокон.

Колбочки и палочки содержат зрительные пигменты. Зрительные пигменты очень похожи на любые другие пигменты, в том, что они поглощают свет и степень поглощения зависит от длины волны. Важное свойство зрительных пигментов состоит в том, что когда зрительный пигмент поглощает фотон света, то изменяется форма молекулы и в то же самое время происходит переизлучение света. Пигмент при этом изменился, измененная молекула поглощает свет менее хорошо чем прежде, т.е. как часто говорят, “отбеливается”. Изменение формы молекулы и переизлучение энергии некоторым, пока еще не вполне ясным образом, инициируют светочувствительную клетку к выдаче сигнала.

Информация от светочувствительных рецепторов (колбочек и палочек) передается другим типам клеток, которые соединены между собой. Специальные клетки передают информацию в зрительный нерв. Таким образом волокно зрительного нерва обслуживает несколько светочувствительных рецепторов, т.е. некоторая предварительная обработка изображения выполняется непосредственно в глазу, который по сути представляет собой выдвинутую вперед часть мозга.

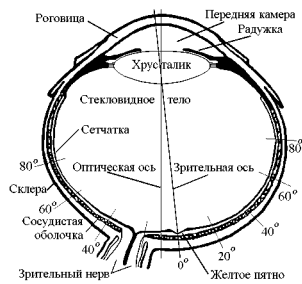


Рис. 0.2.1: Поперечный разрез глаза

Область сетчатки, в которой волокна зрительного нерва собираются вместе и выходят из глаза, лишена светочувствительных рецепторов и называется слепым пятном.

Радужная оболочка (см. рис. 0.2.1) действует как диафрагма, изменяя количество света, проходящего в глаз. Диаметр зрачка меняется от  $\sim 2$  мм (при ярком свете) до  $\sim 8$  мм (при малой освещенности).

За сетчаткой находится сосудистая оболочка, которая содержит капилляры, снабжающие глаз кровью.

Наружняя оболочка глаза — склера, состоит из плотных волокон.

## 0.2.2 Чувствительность глаза

При ярком свете чувствительность палочек мала, но при низких уровнях освещенности их чувствительность возрастает и обеспечивает нашу способность видеть при тусклом свете. Палочки содержат пигмент с максимальной чувствительностью на длине волны около 510 нм (точечная линия на рис. 0.2.2), в зеленой части спектра. Пигмент палочек часто называется зрительным пурпуром из-за его цвета. Максимальная плотность палочек приходится на область с раствором около  $20^\circ$  относительно оси.

Колбочек существует три типа отличающихся фоточувствительным пигментом. Колбочки обычно называют “синими”, “зелеными” и “красными” в соответствии с наименованием цвета, для которого они оптимально чувствительны. Упомянутые три пигмента имеют максимальные поглощения приблизительно на 430, 530 и 560 нм. Этим длинам волн соответствует не синий, зеленый и красный цвета, а фиолетовый, сине-зеленый и желто-зеленый. Более логичным было бы использование терминологии корото-, средне-, длинноволновые колбочки.

Максимальная плотность колбочек достигается в области с раствором порядка  $8^\circ \times 6^\circ$  относительно оси, называемой желтым пятном. В центре этой области (ямке) к каждой колбочке подходит отдельное волокно зрительного нерва. Это область максимальной остроты зрения.

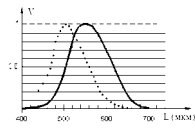


Рис. 0.2.2: Спектральная чувствительность глаза

Суммарная кривая спектральной чувствительности глаза для случая яркого освещения, т.е. цветного зрения, показана на рис. 0.2.2 сплошной линией.

Из этих графиков видно почему “ночью все кошки серы”. В самом деле, например, отклик на красный цвет ( $\lambda = 700$  нм) при низких уровнях освещения (точечная кривая на рис. 0.2.2) практически равен нулю. Поэтому красный цвет ночью будет выглядеть черным.

В силу того, что коэффициент преломления в радужке и хрусталике растет с увеличением частоты света, глаз не избавлен от хроматической аберрации. Т.е. если изображение сфокусировано для одной из частот, то на других частотах изображение расфокусировано. Хрусталик оптимально фокусирует на сетчатке свет с длиной волны около 560 нм. Так как пики чувствительности средне- длинноволновых колбочек (530 и 560 нм, соответственно) близки к друг к другу, поэтому изображения для этих колбочек могут быть одновременно сфокусированными. Изображение же для коротковолновых палочек будет размытым. Так степень фокусировки разная, то не требуется одинаковой разрешающей способности глаза для разных типов колбочек. В глазу человека на одну коротковолновую колбочку приходится 20 средне- и 40 длинноволновых. В этой связи понятно, почему ширина полосы пропускания для “холодных”, коротковолновых цветов в телевидении может быть выбрана существенно меньшей без субъективно заметной потери верности воспроизведения.

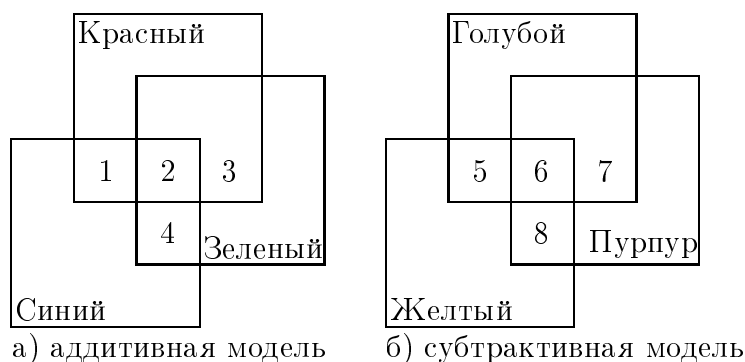
### 0.3 Физические принципы формирования оттенков

В компьютерной графике имеется два типа цветных объектов — самосветящиеся, излучающие объекты, такие как экраны ЭЛТ, плазменные панели, матрицы светодиодов и т.п. и несамосветящиеся объекты, отражающие или преломляющие падающий на них свет, такие как, например, оттиски на бумаге, светофильтры и т.п.

Для самосветящихся объектов используется аддитивное формирование оттенков, когда требуемый цвет формируется за счет смешения трех основных оттенков цветов. В этом случае удобно использование модели смешения RGB (Red, Green, Blue — красный, зеленый, синий).

Для несамосветящихся объектов используется субтрактивное формирование оттенков, основанное на вычитании из падающего света определенных длин волн. В этом случае удобно использование модели смешения CMY (Cyan, Magenta, Yellow — голубой, пурпурный, желтый).

Эти модели смешения показаны на рис. 0.3.1. На рис. 0.3.1а показаны результаты смешения цветов в аддитивной модели для трех самосветящихся площадок чистых цветов (красного, зеленого и синего) и результаты их смешения — площадки 1–4. На рис. 0.3.1б показаны результаты смешения цветов в субтрактивной модели для трех несамосветящихся площадок чистых цветов (голубого, пурпурного и желтого) и результаты их смешения — площадки 5–8.



- 1) Пурпурный=Красный+Синий;
- 2) Белый=Красный+Зеленый+Синий;
- 3) Желтый=Красный+Зеленый;
- 4) Голубой=Зеленый+Синий;
- 5) Зеленый=Голубой+Желтый;
- 6) Черный=Голубой+Пурпурный+Желтый;
- 7) Синий=Голубой+Пурпурный;
- 8) Красный=Желтый+Пурпурный.

Рис. 0.3.1: Модели смешения цветов

Цвета одной модели являются дополнительными к цветам другой модели. Дополнительный цвет — цвет, дополняющий данный до белого. Дополнительный для красного — голубой (зеленый+синий), дополнительный для зеленого — пурпурный (красный+синий), дополнительный для синего — желтый (красный+зеленый) и т.д.

Пример субтрактивного формирования оттенков показан на рис. 0.3.2. При освещении падающим белым светом в слое голубой краски из спектра белого цвета поглощается красная



часть, затем из оставшегося света в слое пурпурной краски поглощается зеленая часть спектра, отраженный от поверхности бумаги свет еще раз подвергается поглощению и в результате мы видим синий цвет.

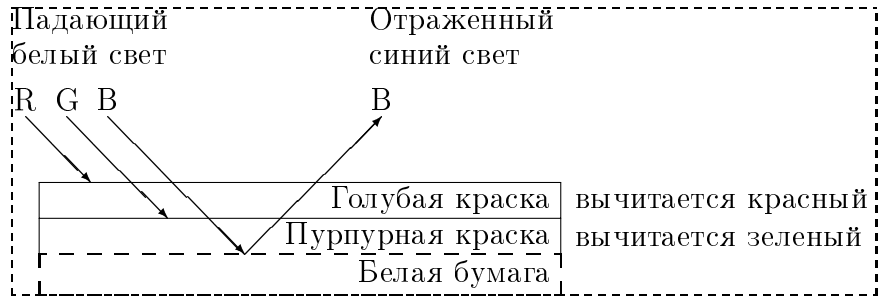


Рис. 0.3.2: Цвет несамосветящегося объекта

## 0.4 Цветовые модели

Назначение цветовой модели — дать средства описания цвета в пределах некоторого цветового охвата, в том числе и для выполнения интерполяции цветов. Наиболее часто в компьютерной графике используются модели RGB, CMY, YIQ, HSV и HLS.

RGB (Red, Green, Blue — красный, зеленый, синий) — аппаратно-ориентированная модель, используемая в дисплеях для аддитивного формирования оттенков самосветящихся объектов (пикселей экрана). Система координат RGB — куб с началом отсчета (0,0,0), соответствующим черному цвету (см. рис. 0.4.1). Максимальное значение RGB — (1,1,1) соответствует белому цвету.

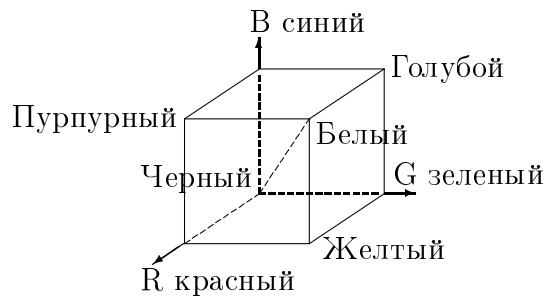


Рис. 0.4.1: Цветовой куб модели RGB

CMY (Cyan, Magenta, Yellow — голубой, пурпурный, желтый) — аппаратно-ориентированная модель, используемая в полиграфии для субтрактивного формирования оттенков, основанного на вычитании слоев краски части падающего светового потока. Цвета модели CMY являются дополнительными к цветам модели RGB, т.е. дополняющими их до белого. Таким образом система координат CMY — тот же куб, что и для RGB, но с началом отсчета в точке с RGB координатами (1,1,1), соответствующей белому цвету. Цветовой куб модели CMY показан на рис. 0.4.2.

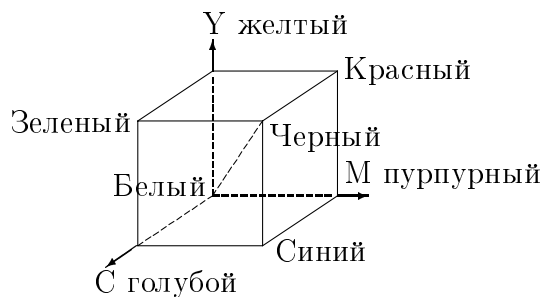


Рис. 0.4.2: Цветовой куб модели CMY

Преобразования между пространствами RGB и CMY определяются следующим образом:

$$[RGB] = [111] - [CMY]$$

Причем единичный вектор-строка в модели RGB — представление белого цвета, а в модели CMY — черного.

YIQ — аппаратно-ориентированная модель, используемая в телевидении и служащая для сокращения передаваемой полосы частот за счет использования психофизиологических особенностей зрения. Преобразования между пространствами RGB и YIQ определяются соотношениями (0.4.1) и (0.4.2).

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.522 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (0.4.1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0.956 & 0.623 \\ 1.0 & -0.272 & -0.648 \\ 1.0 & -1.105 & 0.705 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad (0.4.2)$$

HSV (Hue, Saturation, Value — цветовой тон, насыщенность, количество света или светлота) — модель, ориентированная на человека и обеспечивающая возможность явного задания требуемого оттенка цвета (см. рис. 0.4.3). Подпространство, определяемое данной моделью — перевернутый шестигранный конус.

По вертикальной оси конуса задается  $V$  — светлота, меняющаяся от 0 до 1. Значению  $V=0$  соответствует вершина конуса, значению  $V=1$  — основание конуса; цвета при этом наиболее интенсивны.

Цветовой тон  $H$  задается углом, отсчитываемым вокруг вертикальной оси. В частности,  $0^\circ$  — красный,  $60^\circ$  — желтый,  $120^\circ$  — зеленый,  $180^\circ$  — голубой,  $240^\circ$  — синий,  $300^\circ$  — пурпурный, т.е. дополнительные цвета расположены друг против друга (отличаются на  $180^\circ$ ).

Насыщенность  $S$  определяет насколько близок цвет к “чистому” пигменту и меняется от 0 на вертикальной оси  $V$  до 1 на боковых гранях шестигранного конуса.

Точка  $V=0$ , в которой находится вершина конуса, соответствует черному цвету. Значение  $S$  при этом может быть любым в диапазоне 0–1. Точка с координатами  $V=1, S=0$  — центр основания конуса соответствует белому цвету. Промежуточные значения координаты  $V$  при  $S=0$ , т.е. на оси конуса, соответствуют серым цветам. Если  $S=0$ , то значение оттенка  $H$  считается неопределенным.

Подпрограммы перехода от RGB к HSV и обратно приведены в Приложении 1.

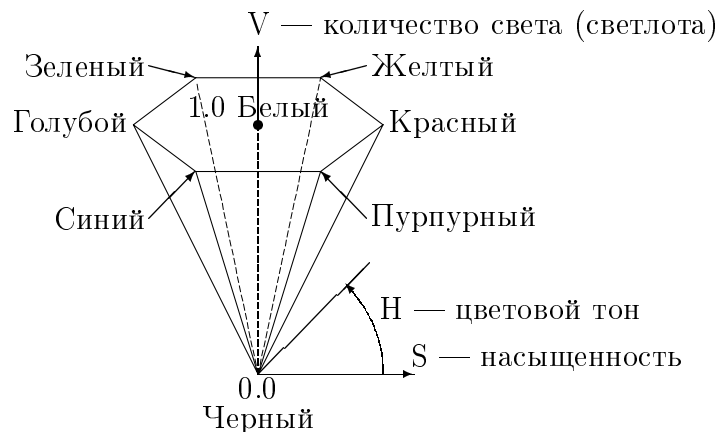


Рис. 0.4.3: Цветовая модель HSV

HLS (Hue, Lightness, Saturation — цветовой тон, светлота, насыщенность) — модель ориентированная на человека и обеспечивающая возможность явного задания требуемого оттенка цвета (см. рис. 0.4.4). Эта модель образует подпространство, представляющее собой двойной конус, в котором черный цвет задается вершиной нижнего конуса и соответствует значению  $L=0$ , белый цвет максимальной интенсивности задается вершиной верхнего конуса и соответствует значению  $L=1$ . Максимально интенсивные цветовые тона соответствуют основанию конусов с  $L=0.5$ , что не совсем удобно.

Цветовой тон  $H$ , аналогично системе HSV, задается углом поворота.

Насыщенность  $S$  меняется в пределах от 0 до 1 и задается расстоянием от вертикальной оси  $L$  до боковой поверхности конуса. Т.е. максимально насыщенные цветовые цвета располагаются при  $L=0.5$ ,  $S=1$ .

В общем, систему HLS можно представить как полученную из HSV “вытягиванием” точки  $V=1$ ,  $S=0$ , задающей белый цвет, вверх для образования верхнего конуса.

Подпрограммы перехода от RGB к HLS и обратно приведены в Приложении 1.

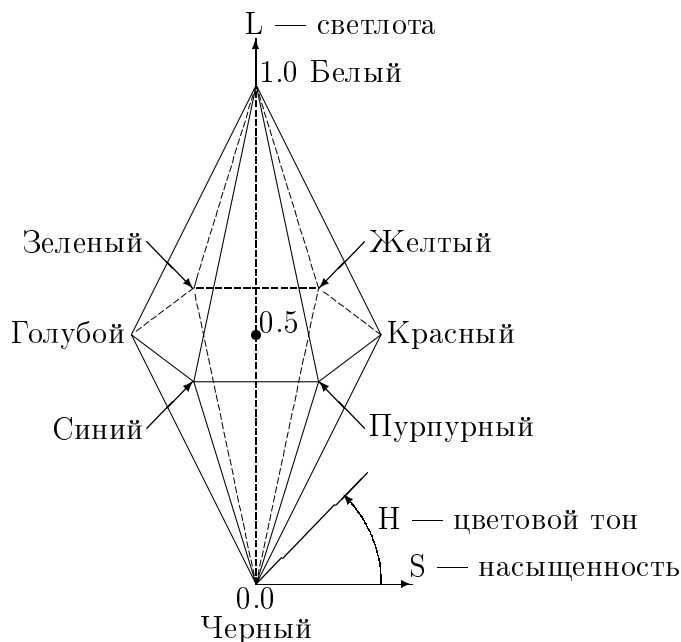


Рис. 0.4.4: Цветовая модель HLS

### 0.4.1 Задание цветов

Растровые дисплеи, как правило, используют аппаратно-ориентированную модель цветов RGB.

В наиболее распространенных растровых дисплеях — дисплеях с таблицей цветности значения кодов пикселей, заносимые в видеопамять, представляют собой индексы элементов таблицы цветности. При необходимости отображения некоторого пиксела на экран по его значению выбирается элемент таблицы цветности, содержащий тройку значений — RGB. Эта тройка и передается на монитор для задания цвета пиксела на экране.

В полноцветных дисплеях для каждого пиксела в видеопамять заносится тройка значений RGB. В этом случае при необходимости отображения пиксела из видеопамяти непосредственно выбирается тройка значений RGB, которая и передается на монитор (но может и передаваться в таблицу цветности).

В модели RGB легко задавать яркости для одного из основных цветов, но по крайней мере затруднительно задать оттенок с требуемым цветовым тоном и насыщенностью. В различного рода графических редакторах эта задача обычно решается с помощью интерактивного выбора из палитры цветов и формированием цветов в палитре путем подбора значений RGB до получения требуемого визуального результата. Более удобно в этом случае использовать модели HVS или HLS, позволяющие непосредственно задать требуемый оттенок. Конечно, при занесении данных в таблицу цветности или для полноцветных дисплеев — в видеопамять требуется перевод значений в систему RGB.

### 0.4.2 Интерполяция цветов

Интерполяция цветов требуется во многих случаях для создания эффектов реалистичности изображения, например, при наложении цветов в технике акварели, т.е. при наложении одного прозрачного цвета на другой, при создании эффектов постепенного изменения цвета в последовательности картин, при построочном заполнении многоугольника методом Гуро и т.д.

Если требуется интерполировать между двумя цветами обладающими одним и тем же цветовым тоном (насыщенностью), так чтобы получаемые цвета имели тот же самый цветовой тон (насыщенность), то необходимо использовать модель или HVS или HLS.

В остальных случаях более удобно пользоваться аппаратно-ориентированной моделью RGB.



Рис. 0.5.1: Классификация механических методов печати

## 0.5 Печатающие устройства

Обычное печатающее устройство помещает отдельные символы друг возле друга на бумагу, как, например, пишущая машинка. Зачастую поэтому говорят также последовательной (посимвольной) печати. В отличие от этого печатающее устройство типографского типа во время процесса печати выдает на бумагу по определенному правилу всю строку (или даже лист) целиком и поэтому может называться параллельным печатающим устройством. Такое устройство требует средств промежуточного хранения для накопления полной строки (листа). В частности, при печати офсетным методом таким средством промежуточного хранения является так называемая фотоформа — алюминиевый лист с вытравленным образом печатаемого листа.

С точки зрения процесса переноса цветов на бумагу устройства печати можно разделить на механические и немеханические. На рис. 0.5.1–0.5.2 приведена классификация механических и немеханических методов документирования. На рис. 0.5.3 приведена классификация, детализирующая различные методы отображения текстов.

Посимвольно-ориентированные методы гарантируют высокое качество текста, но для графического представления они вряд ли применимы. Растрово-ориентированные методы подходят также и для графического вывода.

### 0.5.1 Разрешение устройств

Важнейшей характеристикой растровых устройств, формирующих изображение из отдельных точек, является пространственное разрешение. Оно в первую очередь определяется размером элементарного пятна, а также возможностями размещения пятен. Так как обычно растровое пятно представляет собой некоторую область с убыванием интенсивности от центра к краям, то для различимости отдельных пятен необходимо не слишком большое их перекрытие. На рис. 0.5.4 показаны случаи различимости и неразличимости отдельных пятен. Если пересечение пятен ниже уровня  $\approx 0.6$  от максимальной интенсивности, то пятна различимы (см. рис. 0.5.4а). Если же пересечение пятен выше уровня 0.6, то пятна неразличимы (см. рис. 0.5.4б). Это обстоятельство используется для повышения качества печати (степени гладкости) на растровых принтерах за счет частичного наложения пятен.

Другими факторами, определяющими разрешающую способность, являются:

- форма пятна (круглое, эллипсоидальное или иное);
- распределение яркости по пятну (по гауссу или иное).

Кроме пространственного разрешения существенной характеристикой устройств документирования является адресное разрешение, т.е. точность с которой может задаваться позиция. Как

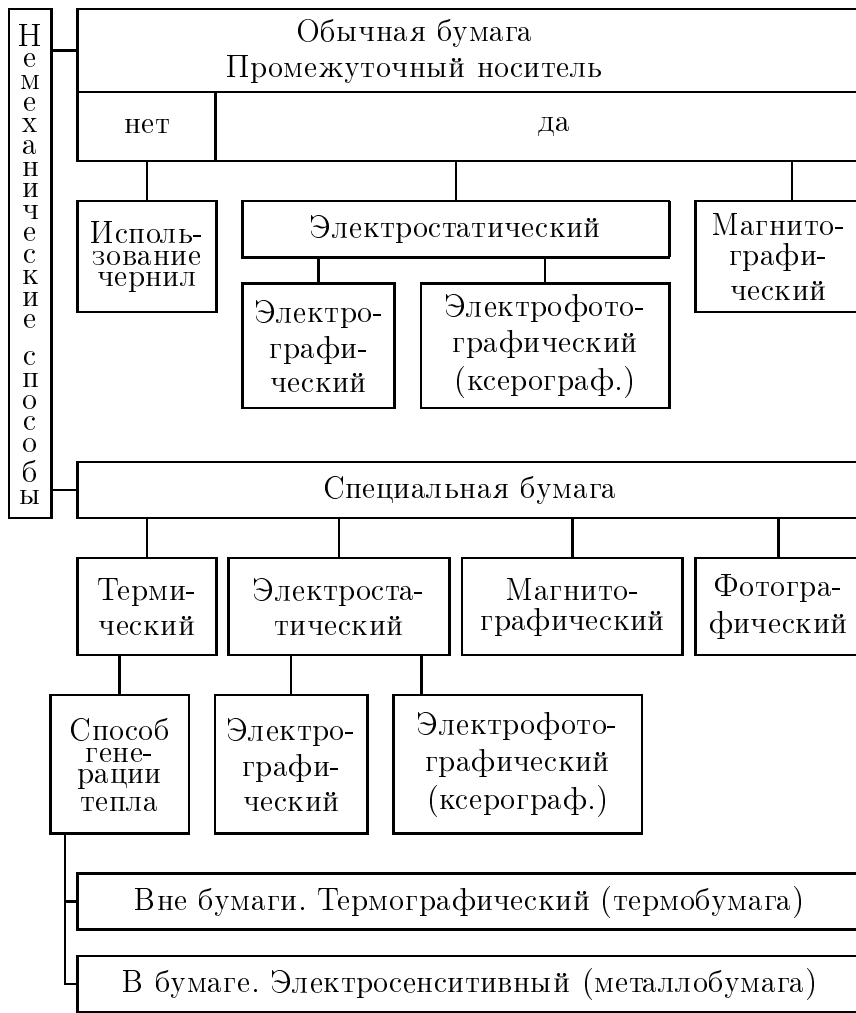


Рис. 0.5.2: Классификация немеханических методов печати

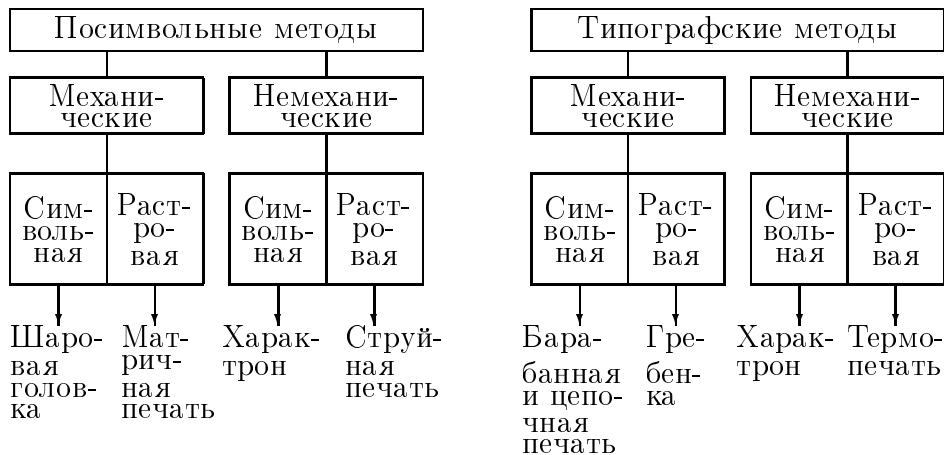


Рис. 0.5.3: Классификация методов печати текстов

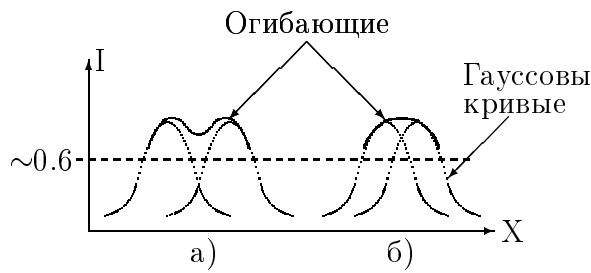


Рис. 0.5.4: Физическое разрешение растровой печати



Рис. 0.5.5: Схема устройства печатающей головки матричного принтера

правило, адресное разрешение превышает пространственное или во всяком случае не меньше последнего.

Адресное разрешение определяет размер памяти, требуемый для хранения отображаемого растрового образа, а также возможности по устранению ступенчатости изображения.

## 0.5.2 Матричный принтер

Широко распространенным устройством является матричный принтер. Он позволяет довольно качественно выводить текст и формировать графические изображения в основном для вспомогательных целей.

Основной элемент матричного принтера — печатающая головка, содержащая от 7 до 48 вертикально расположенных штырьков (игл), с шпилькой, выталкиваемых электромагнитами до соударения с красящей лентой. Принцип работы одной иголки матричного принтера показан на рис. 0.5.5а). Наиболее распространены 9 и 24-х игольчатые принтеры. Важной характеристикой принтера является разрешение, определяемое диаметром иглы. Обычные значения — десятые доли миллиметра (до 3–5 точек/мм, т.е. 85–127 точек/дюйм). Адресное разрешение по оси Y определяется расстоянием между иглами ( $\Delta h$  на рис. 0.5.5б)) и составляет до 170 точек/дюйм. Наиболее современные 48-игольчатые матричные принтеры имеют разрешение приближающееся в 300 точкам/дюйм.

В одноцветных принтерах красящая лента свернута в кольцо Мебиуса, так что обеспечивается использование ленты с двух сторон. Цветные принтеры, в зависимости от используемой бумаги, работают в один или несколько проходов. Если возможно использование обычной бумаги, то все цвета печатаются в каждой строке, прежде чем произойдет перемещение к следующей. Цвет переключается механически вертикальным смещением красящей ленты. В более



скоростных (и дорогостоящих) цветных принтерах, использующих специальную толстую перфорированную бумагу, лента состоит из трех частей: красной, желто-зеленой и синей. Печать происходит постранично в три прохода.

Забота о неналожении точек различных цветов друг на друга возлагается на пользователя. При повторной печати точки другим цветом краситель с бумаги частично переносится на ленту, поэтому ленты быстро мажутся, особенно, светлых оттенков.

Обычно принтеры имеют несколько комплектов внутренних шрифтов и память для загрузки шрифтов, созданных пользователем. В хороших принтерах оба хода головки рабочие и слева-направо и справа-налево.

Как правило, принтеры имеют встроенную память для накопления распечатываемого текста (до нескольких десятков страниц).

Обычно имеется два режима работы — символьный и графический. В символьном режиме принтер, используя внутреннее растровое описание шрифтов, сам управляет печатью строк. В графическом режиме пользователь должен сам подготовить поточечное описание строк.

Скорости печати в символьном режиме порядка первых сотен символов в секунду (120, 120–200, более 200). Скорости печати строк в минуту также первые сотни (до 200, 200–400, более 400).

Лидеры на рынке матричных принтеров — фирмы Epson и NEC (Nippon Electric Company).

В наименовании высококачественных принтеров обычно имеются буквы, определяющие наивысшее качество печати:

NLQ — Near Letter Quality — качество, близкое к качеству пишущей машинки;

LQ — Letter Quality — качество пишущей машинки. В 9-ти игольчатых принтерах обычно реализуется NLQ, а в 24-х игольчатых — LQ.

### Протокол матричного принтера

Стандартами де-факто для знакосинтезирующих принтеров стали три системы команд:

- система команд фирмы Epson;
- система команд фирмы IBM для семейства принтеров Proprinter;
- система команд фирмы IBM для графического принтера (IBM Graphics printer).

### 0.5.3 Струйная печать

Цветная струйная печать хорошо подходит для использования в деловой графике. Набор сопел для чернил размещается в головке печати, с, по крайней, мере одним соплом на один субтрактивный цвет. Нынешние модели базируются на технологиях: “капля по запросу”, “пузырьковой технологии струйной печати” и “Micro Piezo, Micro Dot, Micro Wave”.

Принцип действия струйной печати пояснен на рис. 0.5.6. Цилиндрический пьезоэлектрический кристалл плотно надет на резиновую трубку, заканчивающуюся соплом. При подаче напряжения на кристалл трубка обжимается и выбрасывает каплю чернил в сопло. Дроссель служит для того, чтобы при обжатии трубки чернила выбрасывались только в сопло, а не в резервуар с чернилами. Частота работы сопел составляет до 900 герц.

Количество сопел одного цвета, требуемое разрешением и скоростью печати, вертикально размещается в печатающей головке. Для цветной печати обычно используется три цвета — желтый, голубой, малиновый. Часто добавляется дополнительный черный цвет.

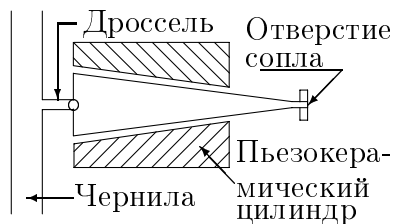


Рис. 0.5.6: Схема устройства отдельного сопла в струйном принтере

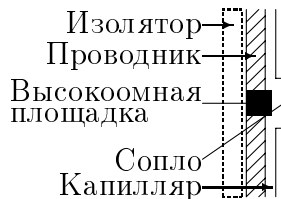


Рис. 0.5.7: Схема образования капель в струйном принтере фирмы HP

Обычная разрешающая способность по горизонтали до 150 точек на дюйм (6 точек/мм) по горизонтали и до 100 точек на дюйм по вертикали (4 точки/мм). Достижения современной технологии изготовления головок позволяют разместить до 50 сопел на 1/6 дюйма, чем обеспечивается вертикальное разрешение до 300 точек на дюйм (12 точек/мм).

Суммарная скорость печати в целом невысока — от 20 до 50 символов в секунду и порядка 90 секунд на лист формата А4 в графическом режиме.

Достоинством устройств струйной печати является малое энергопотребление и практически бесшумная работа.

В струйных черно/белых принтерах фирмы HP используется и другой способ формирования капелек, показанный на рис. 0.5.7. На изолирующую подложку нанесены токоподводящие проводники. На небольшом расстоянии от подложки находится пленка с отверстиями сопел. Напротив каждого сопла в разрыве токоподводящего проводника размещена высокоомная площадка. Между подложкой и пленкой с соплами образован капилляр для подвода специальных чернил. При пропускании импульса тока около 1 А высокоомная площадка быстро разогревается, под действием теплового удара формируется волна, выбрасывающая капельку чернил из сопла.

Фирма Seiko Epson Corporation разработала новую технологию струйной печати (Micro Piezo, Micro Dot и Micro Wave), отличительным свойством которой является управление мениском чернил в сопле. Технология позволяет управлять размером и формой чернильных пятен, повысить скорость выстреливания капель, увеличить количество оттенков до шести, включая полутона, и устранить зернистость. Технология позволяет получить разрешение до 1440 точек на дюйм ( $\approx 57$  точек на мм). Принцип работы струйных принтеров, использующих новую технологию, представлен на рис. 0.5.8. На рис. 0.5.9 показаны результаты печати по технологиям Micro Dot и традиционной.

### Протокол устройств струйной печати

Стандартом де-факто для струйных принтеров является протокол принтеров фирмы Epson,

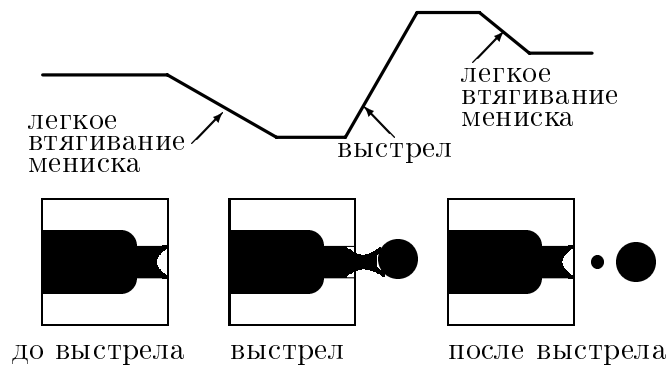


Рис. 0.5.8: Управление формой и полетом капель в новых принтерах Stylus

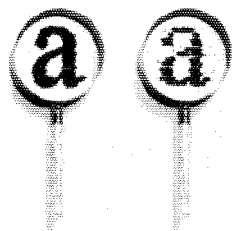


Рис. 0.5.9: Струйная печать по технологиям Micro Dot и традиционной

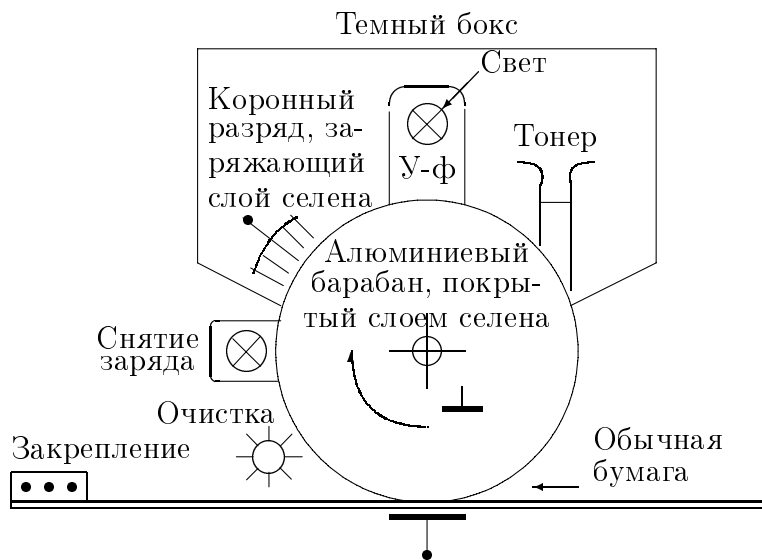


Рис. 0.5.10: Схема устройства черно-белого лазерного принтера

ряд принтеров эмулирует язык HPGL, разработанный фирмой Хьюлетт-Паккард для графопостроителей.

### 0.5.4 Лазерный принтер

Лазерные принтеры используют ксерографический (электрофотографический) метод печати, который также применяется в большинстве аппаратов копирования. В целом лазерный принтер — монохромное устройство. В настоящее время имеются и цветные лазерные принтеры, по сути представляющие собой конструктивное объединение нескольких лазерных принтеров.

#### Черно-белый лазерный принтер

Схема устройства монохромного лазерного принтера показана на рис. 0.5.10. Слой фоточувствительного селена, нанесенный на алюминиевый барабан, в темном боксе аппарата получает равномерный положительный поверхностный заряд с помощью коронного разряда. Этот фоточувствительный слой является изолятором в темноте и полупроводником при освещении. Заряженный слой облучается источником света с целью создания на нем скрытого изображения в виде распределения заряда. Скрытое изображение делается видимым с помощью мелкодисперсного порошка положительного тонера. Синхронно с вращением барабана перемещается обычная бумага. Частички тонера под действием электростатического поля переносятся на бумагу. Полученное изображение фиксируется термическим способом. Перед следующим заряданием фоточувствительный слой очищается от оставшихся частиц тонера и разряжается.

Картинка формируется лазерным лучом на фоточувствительном слое в виде узора точек. Типовая разрешающая способность современных лазерных принтеров 600 точек/дюйм (24 точки/мм, точнее 23.6 точек/мм). Это обеспечивает очень высокое качество для текста и любой графики. Трудности возникают лишь при выводе больших черных поверхностей. В современных принтерах плотность печати доведена до 1200 точек/дюйм, что превышает качество типо-

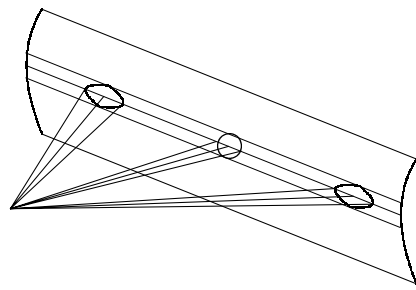


Рис. 0.5.11: Изменение формы пятна в лазерном принтере

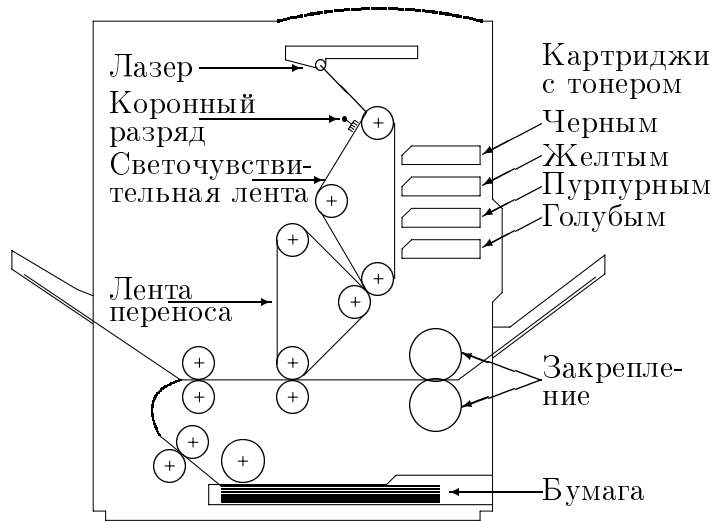


Рис. 0.5.12: Схема устройства цветного лазерного принтера

графской печати. Ограничение в разрешающей способности в обычных лазерных принтерах с одним отклоняющимся лучом связано с различной формой пятна в центре барабана (круг) и на краях (эллипс) (рис .0.5.11).

Лазерный принтер — постраничное устройство. Максимальная скорость составляет порядка 150 страниц в минуту.

### Цветной лазерный принтер

Схема устройства цветного лазерного принтера показана на рис. 0.5.12. Как видно из рисунка цветной лазерный принтер подобен черно-белому.

Для печати за один проход по бумаге используются гибкая светочувствительная лента, покрытая слоем селена, и лента переноса, на которой формируются изображения для всех четырех цветов.

Изображение, сформированное на ленте переноса, переносится на бумагу и затем закрепляется нагреванием.

### Протоколы лазерных принтеров

Стандартом де-факто для лазерных принтеров является система команд PCL (язык команд

принтера) лазерных принтеров HP LaserJet Plus фирмы Hewlett Packard.

Широкое распространение получают различные языки описания страниц (PDL — Page Description Language). Каждый язык описания страниц — это “настоящий” язык программирования, с переменными, подпрограммами, операторами, управляющими структурами. Языки включают графические операции, обеспечивающие создание изображений, и каждый имеет возможности по работе со шрифтами. Наиболее развитыми PDL являются Interpress фирмы Xerox, DDL фирмы Imagen, PostScript фирмы Adobe. Хотя эти языки и очень похожи, стандартом де-факто стал язык PostScript.

Основные отличия, с точки зрения подготовки шрифтов, между PCL и PostScript состоят в том, что шрифты в PCL готовятся в виде битовой карты, поэтому для разных масштабов нужны различные версии одного и того же шрифта. В языке же PostScript описание шрифта — схема его символов, изменяемая программным путем, поэтому легко реализуются не только различные масштабы, но печать по различным направлениям, например, вдоль контура фигуры, изменение наклона букв и т.д.

Многие лазерные принтеры имеют как систему команд PCL, так и аппаратные интерпретаторы PostScript. В целом данные для печати в PCL во много раз короче, чем данные, представленные в PostScript, соответственно несравненно меньше и время передачи данных на принтер. Особенно эти проблемы заметны при программной обработке, например, на принтер, поддерживающий только PostScript, выдаются данные из системы TeX для авторской подготовки статей. Зачастую при этом менее 10 Кбайт текста преобразуется более чем в 100 Кбайт данных на языке PostScript.

Основная область использования PostScript — настольные издательские системы. Такого сорта системы, без его использования требуют подготовки нескольких сотен шрифтов.

### 0.5.5 Принтеры на твердых красителях

Цветной лазерный принтер по сути представляет собой конструктивно объединенные четыре монохромных принтера, работающие последовательно. Для большинства цветных лазерных принтеров скорость черно-белой (монохромной) печати в четыре раза превышает скорость цветной печати.

В отличие от цветных лазерных принтеров, принтеры на твердых красителях (solid ink printers) исходно разрабатывались как цветные устройства. Перед началом работы твердые красители расплавляются нагревательным элементом и готовы к работе в течение рабочего дня (рис. 0.5.13).

Расплавленные красители поступают в печатающую головку, обеспечивающую плотность печати 1200 точек/дюйм (47 точек/мм) по горизонтали и 600 точек/дюйм (24 точки/мм) по вертикали. Скорость печати — 6 страниц/мин в обычном режиме, до 10 страниц/мин в быстром режиме и 5 страниц/мин при печати прозрачных слайдов. Важно отметить, что время начала печати первой страницы составляет менее 0.1 с против  $\approx 1$  с для цветных лазерных принтеров. Это связано с тем, что в лазерном принтере должен нагреваться ролик для закрепления изображения.

Существенными преимуществами принтеров на твердых красителях являются простота смены красителей — стержни красителей добавляются по мере надобности, малое количество (две) типов расходимых компонент, против 5–12 для цветных лазерных принтеров и возможность работы с высоким качеством цветопередачи на самых различных носителях, в том числе и на обычных слайдах для слайд-проекторов.

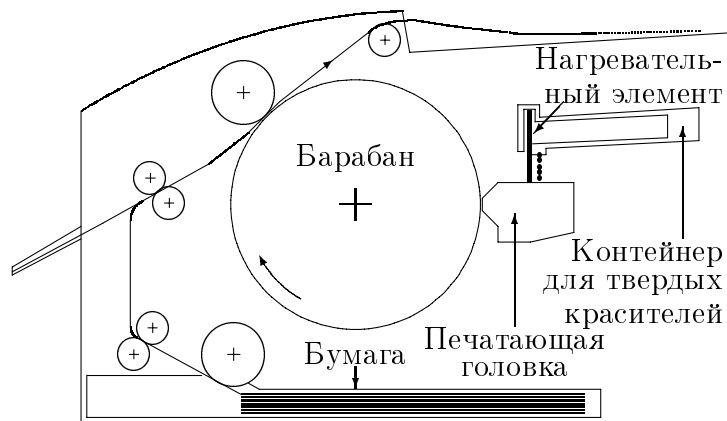


Рис. 0.5.13: Схема устройства принтера на твердых красителях

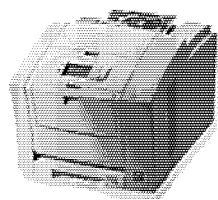


Рис. 0.5.14: Принтер Phaser 840 на твердых красителях

В лаборатории фирмы Тектроникс, разработавшей эти принтеры, достигнута скорость печати до 100 страниц/мин. На рис. 0.5.14 показан внешний вид принтера Phaser 840, выпускаемого фирмой Тектроникс.

### **Протоколы принтеров на твердых красителях**

Рассмотренными принтерами поддерживаются языки описания страниц Adobe Postscript Level 2, HPGL, HCL-5.

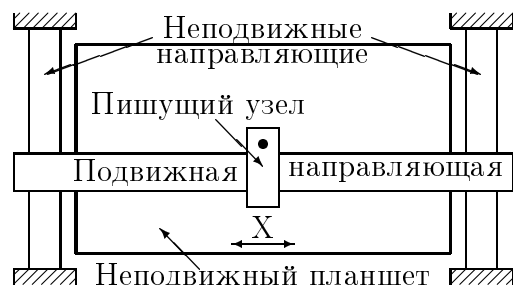


Рис. 0.6.1: Схема планшетного графопостроителя

## 0.6 Графопостроители

Назначение графопостроителей — высококачественное документирование чертежно-графической информации.

В разделе рассмотрены назначение, классификация, принципы действия и устройство графопостроителей.

Графопостроители можно классифицировать следующим образом:

- по способу формирования чертежа — с произвольным сканированием и растровые;
- по способу перемещения носителя — планшетные, барабанные и смешанные (фрикционные, с абразивной головкой).
- по используемому инструменту (типу чертежной головки) — перьевые, фотопостроители, со скрайбирующей головкой, с фрезерной головкой.

Результат работы графопостроителей состоит в установлении головки в требуемую позицию, маркировании позиций и/или их соединение линиями.

Требуемые координаты задаются в 2D координатной системе, большей частью это XY-система.

### 0.6.1 Планшетные графопостроители

В планшетных графопостроителях (рис. 0.6.1) носитель неподвижно закреплен на плоском столе. Закрепление либо электростатическое, либо вакуумное, либо механическое за счет притягивания, прижимающих бумагу пластинок, к (электро)магнитам, вмонтированным в поверхность стола. При документировании на бумаге специальной бумаги не требуется. Головка перемещается по двум перпендикулярным направлениям. Размер носителя ограничен размером планшета.

В некоторых устройствах небольших размеров головка закреплена неподвижно, а перемещается стол с закрепленным на нем носителем, как это сделано во фрезерных станках с числовым программным управлением.

### 0.6.2 Графопостроители с перемещающимся носителем

Имеются три разновидности графопостроителей с перемещающимся носителем:

- барабанные графопостроители, в которых носитель фиксированного размера укреплен на вращающемся барабане;



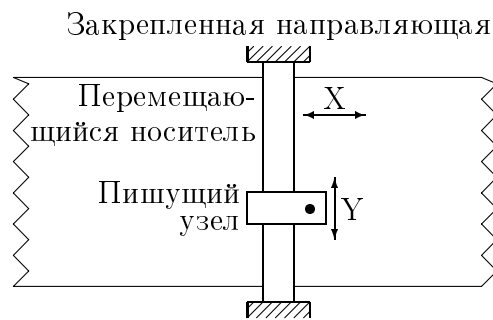


Рис. 0.6.2: Схема графопостроителя с перемещающимся носителем

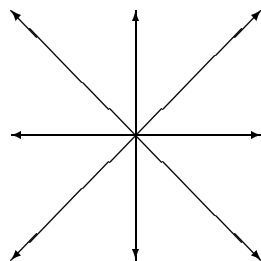


Рис. 0.6.3: Возможные направления перемещения при использовании шаговых двигателей

- фрикционные графопостроители, в которых носитель перемещается с помощью фрикционных роликов. Эти графопостроители (при равных размерах чертежа) много меньших габаритов, чем барабанные. Одна из новых разновидностей фрикционного графопостроителя, появившаяся благодаря технологическим достижениям в металлообработке, — графопостроитель с т.н. абразивной головкой, в которых валики привода бумаги — стальные со специальной насечкой, не забивающейся волокнами бумаги;
- рулонные графопостроители, которые подобны фрикционным, но используют специальный носитель с краевой перфорацией.

Общая схема графопостроителя с перемещающимся носителем приведена на рис. 0.6.2.

Вне зависимости от способа перемещения носителя, система привода графопостроителей с произвольным сканированием использует либо шаговые двигатели, поворачивающиеся на фиксированный угол при подаче одного импульса, либо исполнительную систему с обратной связью, содержащую двигатели привода и датчики положения.

Перемещения с шаговыми двигателями обычно выполняются на 1 шаг по одному из 8 направлений (рис. 0.6.3).

Поэтому требуется аппроксимация вычерчиваемой кривой штрихами основных направлений. Повышение точности аппроксимации достигается как уменьшением шага, так и путем увеличения числа направлений перемещения за счет использования дополнительных пар моторов или за счет изменения передаточного числа.

### 0.6.3 Электростатические графопостроители

Электростатические графопостроители работают на безударном электрографическом растровом принципе (рис. 0.6.4). Специальная диэлектрическая бумага перемещается под элек-

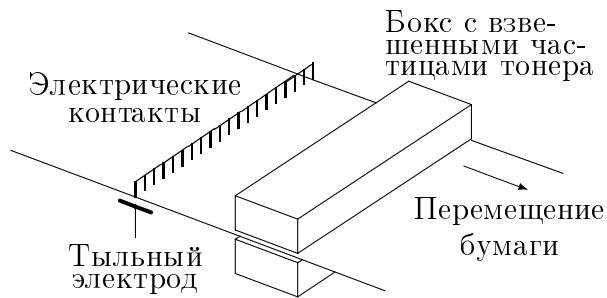


Рис. 0.6.4: Схема электростатического графопостроителя

тростатической головкой, содержащей иголки с плотностью 40–100 на 1 см. К иголкам прикладывается отрицательное напряжение в результате чего диэлектрическая бумага заряжается и на ней создается скрытое изображение. Затем бумага проходит через бокс, в котором над ней распыляется положительно заряженный тонер. Заряженные области притягивают частицы тонера. В цветных системах этот процесс повторяется для каждого их основных субтрактивных цветов — голубого, пурпурного и желтого, а также черного.

Электростатические графопостроители быстрее перьевых графопостроителей, но медленнее лазерных печатающих устройств. Их скорость составляет от 500 до 1000 линий, наносимых на бумагу в 1 мин. Они работают с разрешением 200–400 точек на дюйм. Электростатические графопостроители необходимы, если требуется высококачественный цветной вывод для САД-системы. Такой графопостроитель в 10–20 раз быстрее перьевого. Среди лидеров на рынке этих устройств фирмы Versatec, Calcomp и Venson. Эти графопостроители весьма дорогие их цена 30–150 тысяч долларов.

Основные параметры графопостроителей

	Планшетные	Барабанные
Формат черчения	Ширина 210 – 840 мм Длина 297 – 1188мм	Ширина 210 – 1140 мм Длина 297мм – не ограничена
Скорость черчения	80 – 1140 мм/с	30–300 мм/с
Точность	0.8 – 0.0025 мм	0.7–0.0025 мм
Разрешение	0.4 – 0.0025 мм	0.1–0.0025 мм

Точность определяется минимально возможным значением приращения координаты. Обычные значения десятки микрон.

Разрешение определяется фактическими возможностями исполнительной системы и чертежной головки. Для перьевых графопостроителей обычные значения — доли миллиметра. Для фотопостроителей — менее 10 микрон.

Уникальные высокоточные графопостроители имеют зачастую и уникальные протоколы управления.

Графопостроители широкого распространения, как правило, поддерживают протокол графопостроителей фирмы Хьюлетт-Паккард HPGL (Hewlett Packard Graphics Language). Он содержит небольшое количество графических функций, легко читается и интерпретируется.

Некоторые графопостроители интерпретируют протокол REGIS, разработанный для терминалов VT 240 (и более мощных).

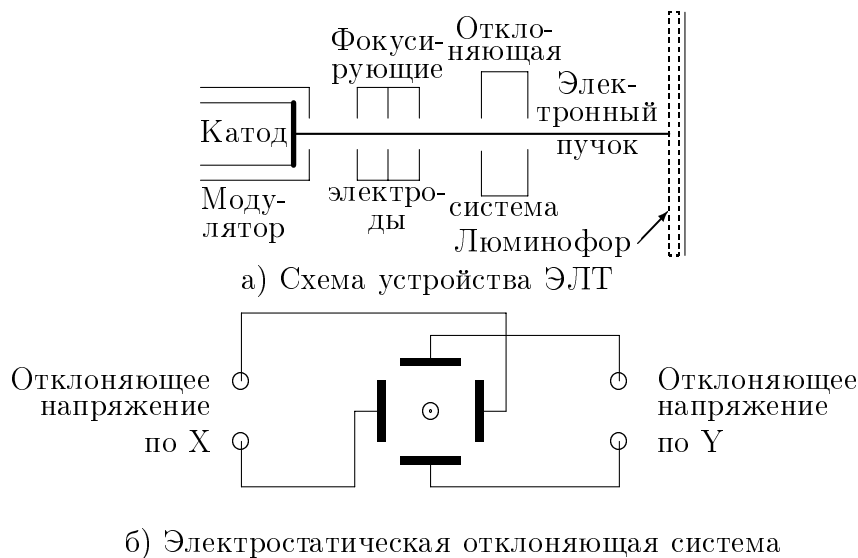


Рис. 0.7.1: Устройство черно-белого кинескопа

## 0.7 Электронно-лучевые трубки

В разделе рассмотрены электронно-лучевые трубки, наиболее широко используемые для отображения в компьютерной графике — черно-белые и цветные электронно-лучевые трубки (кинескопы).

### 0.7.1 Черно-белые кинескопы

Наиболее просто устроена черно-белая электронно-лучевая трубка (рис. 0.7.1а). Нагретый катод испускает электроны, которые проходят через модулятор, управляющий яркостью и фокусируются фокусирующими электродами, так, чтобы пятно на экране было порядка 0.1 — 0.3 мм. Далее электронный пучок отклоняется отклоняющей системой. Для черно-белых трубок дисплеев обычно используется электромагнитное отклонение с помощью катушек, надетых на горловину кинескопа, но используется и электростатическое отклонение с помощью отклоняющих пластин, как это показано на рис. 0.7.1б. Отклоненный пучок попадает на внутреннюю поверхность колбы, покрытую сплошным слоем люминофора, и вызывает его свечение. Ускоряющее напряжение для таких кинескопов  $\approx 10$  Кв, ток пучка  $\approx 10$  мка. При диаметре пятна  $\approx 0.25$  мм выделяемая мощность  $\approx 150$  Вт/см<sup>2</sup> (электроплитка  $\approx 7$  Вт/см<sup>2</sup>).

Пространственное разрешение таких ЭЛТ определяется диаметром пятна и составляет 3–10 точек/мм (85–250 точек/дюйм).

Адресное разрешение определяется уже схемотехническими решениями разработчиков дисплея и составляет для пользователей обычно 1024 пиксела по горизонтали (10 разрядов). Для уменьшения эффектов ступенчатости внутри цифровых систем разверток векторных дисплеев используется 11 или 12 разрядов.

### 0.7.2 Цветные кинескопы

Устройство цветных кинескопов подобно устройству черно-белых с тем отличием, что обязательно имеется три типа люминофора для красного, зеленого и синего цветов, а также средств

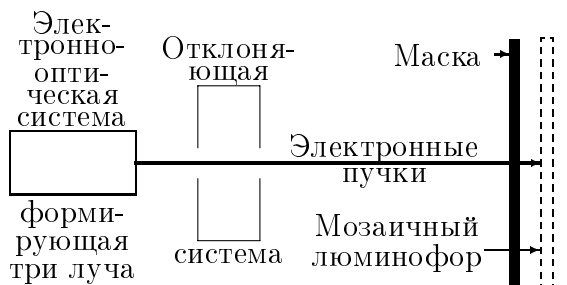


Рис. 0.7.2: Общая схема цветного кинескопа

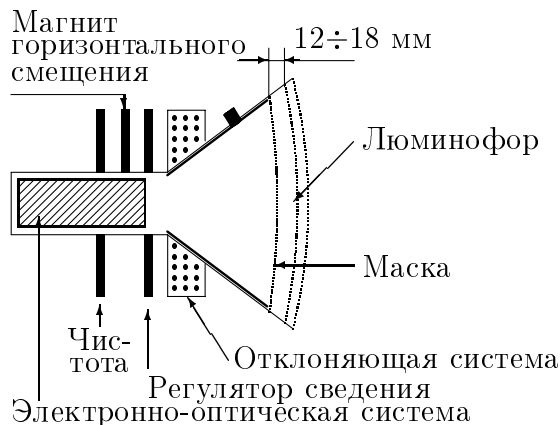


Рис. 0.7.3: Устройство цветного кинескопа

для формирования и управления тремя отдельными электронными лучами (в некоторых, скорее экспериментальных цветных кинескопах используется единственный электронный луч).

Первый цветной масочный кинескоп с тремя отдельными электронными пушками, установленными друг относительно друга под  $120^\circ$  был разработан в США в 1950 г (кинескоп с дельта-образным расположением пушек). На рис. 0.7.2–0.7.5 иллюстрируется устройство такого кинескопа.

Основные параметры таких кинескопов следующие:  $U_{\text{анода}} \approx 20 \div 25$  Кв,  $I_{\Sigma} \approx 1.5 \div 2$  ма диаметр пятна  $\approx 0.25$  мм  $P \approx 2000$  Вт/см<sup>2</sup>.

### Проблемы цветных масочных кинескопов

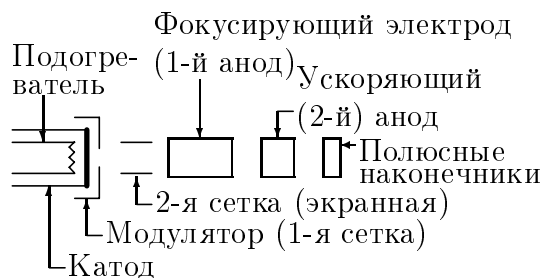


Рис. 0.7.4: Электронно-оптическая система цветного кинескопа

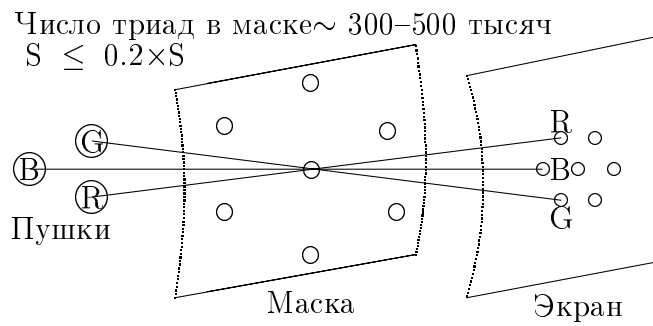


Рис. 0.7.5: Схема формирования отдельных цветов в масочном кинескопе

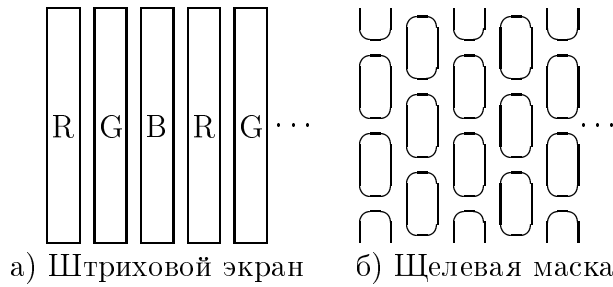


Рис. 0.7.6: Структура экрана и маски планарного кинескопа

Основными проблемами цветных масочных кинескопов, определившими необходимость разработки альтернативных конструкций кинескопов, являются следующие:

- большие напряжения и токи,
- малая яркость (на экран из-за маски попадает  $\sim 20\%$  электронов),
- малый срок службы,
- недостаточно высокая чистота цвета — однородность свечения экрана по каждому цвету в отдельности,
- статическое сведение лучей,
- динамическое сведение лучей,
- статический и динамический балансы белого.

### Кинескопы с планарным расположением пушек

В настоящее время наиболее массовым типом цветного кинескопа является т.н. планарный кинескоп (рис. 0.7.6), называемый также кинескопом со штриховым люминофором, или кинескопом с щелевой маской (PIL — Precision In Line). У таких кинескопов все электроды электронно-оптической системы кроме катодов — общие. Требуется прецизионное изготовление и сборка всех узлов трубки.

Достоинства планарных кинескопов:

- бóльшая прозрачность маски,
- бóльшее заполнение экрана люминофором,
- бóльшая яркость и меньшее энергопотребление,
- до углов отклонения в  $90^\circ$  не нужно динамическое сведение лучей.

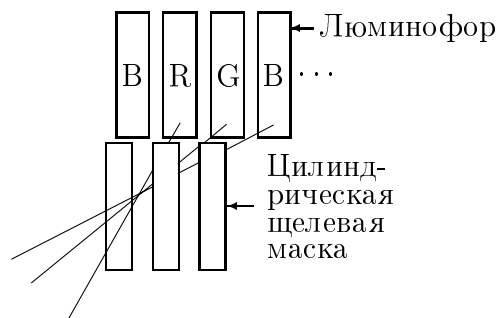


Рис. 0.7.7: Структура экрана и маски трехлучевого тринитрона

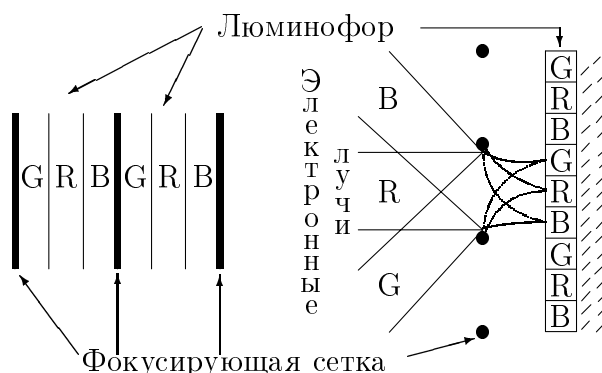


Рис. 0.7.8: Принцип работы трехлучевого хроматрона

### Трехлучевой тринитрон

Подобен планарному кинескопу (рис. 0.7.7), но имеется один, а не три катода. Отдельные лучи формируются диафрагмами. Используется цилиндрическая щелевая маска. Применяется в малогабаритных устройствах. Характерная особенность изображений на тринитронах — тонкий горизонтальный штрих на примерно  $1/3$  высоты экрана, вызванный конструктивными особенностями.

### Трехлучевой хроматрон — США, 1951 г.

Основные проблемы любого масочного кинескопа — потеря части электронов на маске в результате чего понижается яркость свечения. Этот недостаток практически полностью устранен в трех- и однолучевых хроматронах и индексных кинескопах. В трехлучевом хроматроне (рис. 0.7.8) лучи отклоняются тонкой фокусирующей сеткой.

Одной триаде из RGB полосок люминофора соответствует одна проволока фокусирующей сетки.

Для поддержания высокой чистоты цвета требуется пять электромагнитных катушек:

- один охватывает весь экран,
- четыре — по углам экрана.

Яркость хроматрона выше чем у любого масочного кинескопа.

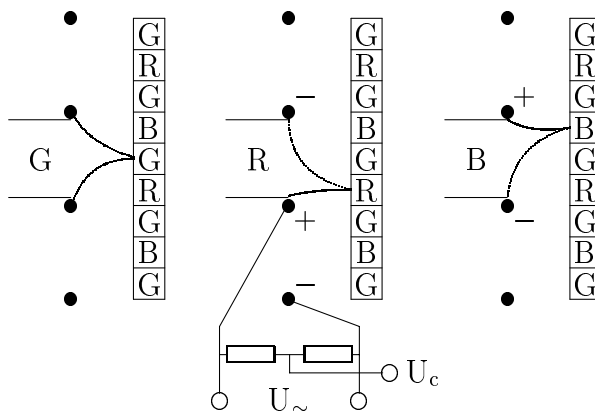


Рис. 0.7.9: Принцип работы однолучевого хроматрона

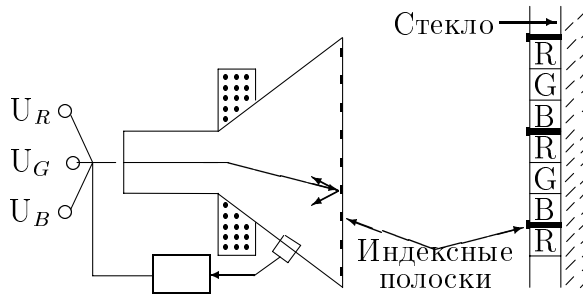


Рис. 0.7.10: Принцип работы индексного кинескопа

Недостатки: сложная система поддержания чистоты цвета, меньшее разрешение, требуется высокая стабилизация питания.

### Однолучевой хроматрон

Несколько более простую конструкцию по сравнению с трехлучевым хроматроном имеет однолучевой хроматрон (рис. 0.7.9). Но как видно из рисунка требуется втрое более скоростная система управления отклоняющими напряжениями на сетке.

Все остальные конструктивные особенности, достоинства и недостатки у однолучевого хроматрона соответствуют таковым для трехлучевого.

### Индексный кинескоп

Наиболее точное управление лучом обеспечивается в индексном кинескопе (рис. 0.7.10).

На внутреннюю поверхность экрана колбы (см. рис. 0.7.10) нанесены вертикальные полоски триад люминофора, разделенные индексными полосками, излучающими в ультрафиолетовом диапазоне внутрь колбы. Излучение принимается датчиком. Таким образом точно известно положение луча по горизонтали.

В зависимости от положения луча его ток модулируется напряжением для красного, зеленого или синего цветов ( $U_R$ ,  $U_G$  или  $U_B$ ).



Недостатки индексного кинескопа следующие:

- сложность поэлементной коммутации цветов,
- высокая линейность строчной развертки,
- более низкие яркость, контрастность, четкость.

### 0.7.3 Разрешение мониторов

Число строк на мониторе с растровым сканированием зависит от расстояния наблюдения и остроты зрения человека. Для получения хорошего качества и меньшего утомления глаза человека число строк должно быть таким, чтобы при наблюдении с выбранного расстояния отдельные строки не были различимы.

Пусть с расстояния  $L$  две смежных строки видны под углом  $\phi$  (рис. 0.7.11), в этом случае выполняется соотношение:

$$\tan\left(\frac{\phi}{2}\right) = \frac{h/2}{L} = \frac{(H/N)/2}{L},$$

где  $h$  — расстояние между строками,  
 $L$  — расстояние от глаза до экрана,  
 $H$  — высота экрана,  
 $N$  — число строк разложения.

Так как угол  $\phi$  очень мал, то  $\tan(\frac{\phi}{2}) \simeq \frac{\phi}{2}$ , отсюда следует:

$$N = \frac{1}{\phi} \cdot \frac{H}{L}$$

Для неразличимости двух строк угловое расстояние между ними должно составлять величину порядка  $1''$  (на самом деле это значение варьируется возле  $4/10000$  радиана  $\simeq 1.38''$ ).

Отсюда получаем:

$$N = 2500 \cdot \frac{H}{L}$$

Важным параметром в определении числа строк является отношение высоты экрана  $H$  к расстоянию до глаза  $L$ . Имеется оптимальное отношение. Если расстояние мало, глаз не видит весь объект. Напротив, при большом расстоянии становятся неразличимыми детали. При формате кадра 4:3<sup>1</sup> оптимальное расстояние наблюдения составляет 4...6 высот экрана. Таким образом получаем, что оптимальное число строк составляет от 416 до 625.

Кроме психофизиологических условий разрешение мониторов определяется технологией изготовления. Для цветного кинескопа разрешение определяется расстоянием  $h$  между пятнами люминофора одного и того же цвета (рис. 0.7.12). Для современных кинескопов это расстояние равно 0.26 мм.

### 0.7.4 Люминофоры

Экран электронно-лучевых трубок покрывается изнутри люминофором, который излучает свет при падении на него электронного луча (флуоресценция). После выключения луча излу-

---

Формат кадра 4:3 выбран из тех соображений, что зона действия желтого пятна глаза составляет 8 по горизонтали и 6 по вертикали. Объект, попадающий на желтое пятно, виден наиболее ясно и целиком, т.е. соотношение его сторон также должно быть порядка 4:3.

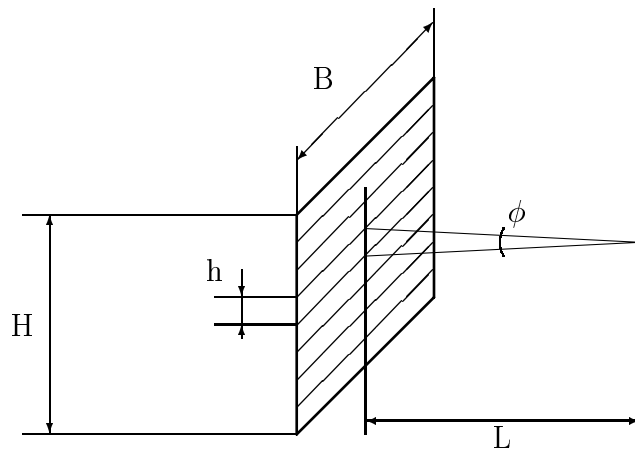


Рис. 0.7.11: Определение количества строк разложения

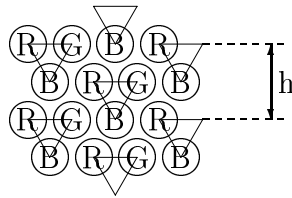


Рис. 0.7.12: Разрешение цветного кинескопа

чение продолжается еще некоторое время (фосфоресценция) (рис. 0.7.13). Время этого послесвечения является одним из важных параметров трубки, так как определяет как часто будет требоваться регенерация изображения, т.е. его повторный вывод для того чтобы пользователь видел немерцающую картину. Параметры некоторых люминофоров приведены в таблице 0.7.1.

### 0.7.5 Гамма-коррекция

Электронно-лучевые трубки мониторов обладают нелинейной зависимостью интенсивности свечения люминофора  $I$  от числа электронов в луче  $N$ . Упрощенно эта зависимость имеет вид:

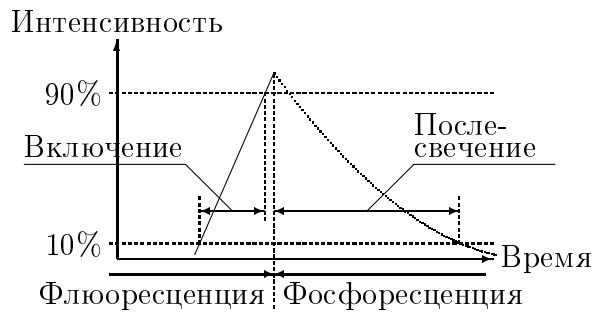


Рис. 0.7.13: Зависимость световой эмиссии люминофора от времени

Таблица 0.7.1: Параметры некоторых люминофоров

Тип	Состав	Длина волны (нм)	Цвет	Послесвеч. 10%	Применение
P1	Zn <sub>2</sub> SiO <sub>4</sub> :Mn	525	YG	24	Радар, осцилл.
P2	ZnS:Cu	543	YG	35–100	Осцилл.
P4	ZnS-CdS:Ag	460/560	W	25/60	TV
P7	ZnS-CdS:Cu	440/560	B/YG	.05/300	Радар, осцилл.
P11	ZnS:Ag	460	B	.025/.08	Фотозап.
P12		Оранж.	Or	210	
P13		Зелен.	G	0.038	
P16	Ca <sub>2</sub> Mg <sub>2</sub> Si <sub>2</sub> O <sub>7</sub> :Ge	385	UV	0.12 мкс	Фотозап., скан.
P18		YG	YG	600	
P20	ZnS-CdS:Ag	560	YG	0.05–2	
P22-B	ZnS:Ag	440	B	22мкс	TV
P22-G	ZnS-CdS:Ag	535	YG	60мкс	TV
P22-R	Y <sub>2</sub> O <sub>2</sub> S:Eu	635	R	1	TV
P22-G <sub>LP</sub>	Zn <sub>2</sub> SiO <sub>4</sub> : Mn : As	525	YG	150	Дисплеи
P31	ZnS:Cu	522	G	40мкс	Осцилл.
P39	Zn <sub>2</sub> SiO <sub>4</sub> : Mn : As	525	YG	150	Больш. послесв.
P42	Zn <sub>2</sub> SiO <sub>4</sub> : Mn : As	520	YG	10	Высок. яркость
P43	Cd <sub>2</sub> O <sub>2</sub> S : Tb	544	YG	1	Дисплеи
P44	La <sub>2</sub> O <sub>2</sub> S : Tb	540	YG	1	Дисплеи
P45	Y <sub>2</sub> O <sub>2</sub> S : Tb	420/540	W	2	Дисплеи
P49	Zn <sub>2</sub> SiO <sub>4</sub> : Mn	525/615	YG/R	30/1.2	Дисплеи на пентронах

$$I = a \times N^\gamma, \quad (0.7.1)$$

где  $a$  и  $\gamma$  — константы. Число электронов пропорционально напряжению  $U$  на управляющей сетке монитора, определяемому значением кода пиксела  $V$ , поэтому соотношение 0.7.1 может быть переписано в виде:

$$I = k \times U^\gamma = c \times V^\gamma, \quad (0.7.2)$$

Более точно это соотношение имеет вид:

$$I = k \times U^{\gamma(U)} + I_0, \quad (0.7.3)$$

где  $k$  — константа,  $\gamma$  зависит от  $U$ ,  $I_0$  — паразитная засветка. Для упрощения полагают  $I_0 = 0$ , а  $\gamma$  — константой. У различных типов мониторов  $\gamma$  меняется от 1.4 до 3.0. Для компьютерных мониторов значение  $\gamma$  обычно принимают равным 2.5. Таким образом, если мы имеем соотношение между яркостями двух вычисленных пикселов  $V_1/V_2 = 0.5$ , то на мониторе соотношение их яркостей будет равно 0.177.

Чтобы избежать искажений, используется гамма-коррекция входного сигнала. Для этого входное значение возводится в степень  $1/\gamma$  и затем передается монитору.

На самом деле зависимости более сложные. Необходимо учитывать уровень освещения в комнате, где находится компьютер, яркость и контрастность, установленные на мониторе, и наконец субъективное восприятие пользователя.

### Встроенные гамма-коррекции

Некоторые графические системы имеют встроенную аппаратную гамма-коррекцию, которая может регулироваться. Как правило, встроенная гамма-коррекция отличается от усредненного значения гаммы мониторов, равного 2.5 и близка к минимальному значению гаммы мониторов (см. выше). Дополнительная гамма-коррекция, требуемая для обеспечения верности воспроизведения, называется “системной”.

Файл изображения может иметь свою собственную гамму, равную величине гамма-коррекции, использованной при формировании файла. Эта гамма-коррекция называется “файловой”. Большинство растровых графических файлов, за исключением файлов TGA и PNG, не предусматривают сохранения “файловой гаммы”, поэтому при воспроизведении может потребоваться ее подбор.

IBM PC и графические станции фирмы SUN не имеют встроенной гамма-коррекции, т.е. их системная гамма примерно равна 2.5. Поэтому для верного воспроизведения сигнал (значение кода пиксела) следует возводить в степень  $1/2.5$ .

Графические станции фирмы Silicon Graphics имеют встроенную аппаратную гамма-коррекцию, равную 1.7, т.е. при посылке пиксела в буфер кадра его значение возводится в степень  $1/1.7$ . Таким образом программное обеспечение должно само вначале отрегулировать сигнал возведением в степень  $1/1.5$  ( $2.5/1.7 \sim 1.5$ ). Т.е. системная гамма для этих станций составляет 1.5.

Графические станции Макинтош имеют встроенную аппаратную гамма-коррекцию, равную 1.4. Таким образом для полной гамма-коррекции программное обеспечение должно само вначале отрегулировать сигнал возведением в степень  $1/1.8$  ( $2.5/1.4 \sim 1.8$ ). Т.е. системная гамма для этих станций составляет 1.8.

Таблица 0.7.2: Зависимость моделей цветов от системной гаммы

Зависимые от системной гаммы	Независимые от системной гаммы
RGB	CIE XYZ
CMY	CIE xyY
CMYK	Lab
HSV	Luv
HLS	Munsell

### Влияние гамма-коррекции на передачу цветов

Значения гаммы для различных мониторов могут быть различны для красного, зеленого и синего каналов. Кроме этого мониторы могут отличаться и по спектральным характеристикам люминофоров. Поэтому для обеспечения верности воспроизведения изображения, построенного на другом мониторе, может потребоваться подбор гаммы. Обычно это делается после оценки гаммы монитора с помощью калибровочных тестовых изображений.

Влияние значения гаммы на цветопередачу очевидно. Пусть, например мы имеем изображение цветного квадрата, построенного с использованием модели RGB (см. раздел 0.4) и вычисленным соотношением яркостей цветов равным 8:2:2. После воспроизведения на графической системе без гамма-коррекции мы получим соотношение, равное 32:1:1.

Все цветовые модели можно разделить на два типа — зависящие и не зависящие от системной гаммы. В таблице 0.7.2 приведена классификация некоторых из моделей цветов.

## 0.8 Дисплеи с произвольным сканированием луча (каллиграфические или векторные дисплеи)

В данном разделе рассматривается архитектура дисплеев с произвольным и растровым сканированием луча, использующих ЭЛТ в качестве устройства отображения.

Вводятся понятия дисплейного файла, линейного дисплейного файла и структурированного дисплейного файла, дисплейного процессора.

Показывается, что дисплейный процессор состоит из дисплейного контроллера, блока преобразований и дисплейного генератора.

Рассматриваются два алгоритма работы генератора векторов — цифрового дифференциального анализатора и Брезенхема.

Анализируется процесс преобразования модели объекта в изображение, рассматриваются целесообразные границы разделения функций между центральным процессором и графической системой, описываются альтернативные архитектуры дисплеев.

В настоящее время подавляющее распространение получили растровые дисплеи. Рассмотрение векторных дисплеев производится, в основном, из-за того что построение изображения на них осуществляется с использованием привычных понятий позиционирования и вычерчивания. На примере таких устройств формулируются общие понятия и выделяются принципиальные компоненты любой графической системы, назначением которой является представление модели объекта в изображение.

В векторных дисплеях изображение строится в виде совокупности отдельно и достаточно точно выдаваемых отрезков. Основная проблема большинства дисплеев, в частности дисплеев на обычных электронно-лучевых трубках состоит в том, что если линия прочерчивается один раз, то за время послесвечения она пропадет с экрана. Решение этой проблемы заключается в том, что построение изображения циклически повторяется (регенерируется) с требуемой частотой (обычно с частотой сети — 50 Гц). Для такой регенерации используется дисплейный файл, представляющий собой описание изображения.

### Дисплейный файл

Дисплейный файл — набор команд дисплейного процессора. В очередном цикле регенерации команды последовательно выбираются из дисплейного файла и исполняются дисплейным процессором. Наряду с командами позиционирования луча и построения отрезков, обычно имеются команды построения символов, команды задания атрибутов построений таких как тип линии (сплошная, штриховая и т.п.), яркость подсветки, мерцание.

### Линейный и сегментированный дисплейные файлы

Линейный дисплейный файл — линейная последовательность команд описания изображения, может быть содержащая команды перехода для повторения процесса отображения файла. Линейный дисплейный файл может быть разбит на отдельные, независимо модифицируемые части. С использованием сегментов модификации изображения могут заключаться в изменении только требуемых частей. Такого сорта линейный дисплейный файл называется сегментированным.

## Структурированный дисплейный файл

Для сокращения объема дисплейного файла в составе команд дисплейного процессора могут предусматриваться не только графические команды формирования изображений и перехода, но и команды организации циклов и обращения к подпрограммам. Если описание изображения построено с использованием вложенных вызовов подпрограмм, то такого сорта дисплейный файл называется структурированным.

Легко видеть, что наличие механизма вложенных вызовов подпрограмм, в общем случае, требует наличия у дисплейного процессора средств геометрических преобразований и отсечения частей изображения, выходящих за пределы экрана. (Представьте себе изображение в перспективе улицы из типовых домов. Можно обойтись одной подпрограммой для построения дома, но каждый дом должен подвергаться индивидуальному преобразованию, а для некоторых домов потребуется отсечение).

### Архитектура

Первые графические векторные дисплеи с регенерацией появились на зарубежном рынке в конце 60-х годов. В нашей стране серийные векторные дисплеи появились в середине 70-х годов.

Практически с самого начала векторные дисплеи имели разрешение  $1024 \times 1024$  точки и могли без мерцания отображать от 1500 до 3000 векторов. Векторы могли быть нескольких типов (сплошные, штриховые, точечные, пунктирные и т.д.) с несколькими градациями яркости. Встроенные средства генерации символов обычно имели возможность строить символы двух фиксированных размеров нескольких ориентаций. В качестве средства диалога использовались буквенно-цифровая и символьная клавиатуры и световое перо.

Рассмотрим архитектуру простой графической системы. Далее будет использоваться понятие “дисплейный процессор” широко распространенное в литературе, но не имеющее точного определения. Здесь под ним мы будем понимать (рис. 0.8.1) аппаратную часть графической системы, которая:

- формирует изображение на дисплее из описания картины;
- обрабатывает графический ввод от пользователя.

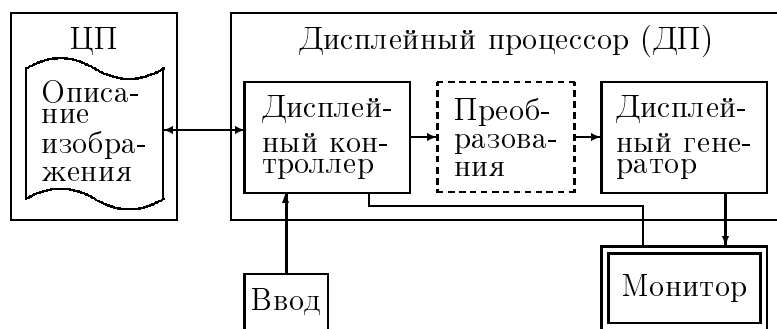


Рис. 0.8.1: Общая архитектура простой графической системы

Обе задачи весьма специальные и вычислительно интенсивны и могли бы значительно затруднить обычную работу ЦП.

В зависимости от требований приложений и используемых приборов вывода дисплейные процессоры имеют различную сложность и работают на различных принципах. По принципу формирования изображения различаются два основных класса дисплейных процессоров (см. рис. 0.8.1):

- каллиграфические или векторные устройства, называемые еще устройствами с произвольным сканированием луча;
- растровые устройства.

Дисплейный процессор обязательно состоит из дисплейного контроллера и дисплейного генератора. Более мощные дисплейные процессоры оснащаются и собственными средствами преобразования и сохранения геометрических данных.

Рассмотрим назначение модулей дисплейного процессора справа-налево: дисплейный генератор, преобразования, дисплейный контроллер.

### 0.8.1 Дисплейный генератор

Задача дисплейного генератора — управление X,Y-отклонением и интенсивностью луча. Как уже отмечалось, дисплейный файл, хранящий описание изображения, представляет собой программу, циклически исполняемую дисплейным процессором. При частоте регенерации в 50 Гц одно исполнение программы должно занимать 20 мс. Если изображение слишком сложное, то дисплейный процессор не успевает его построить за 20 мсек и становится заметным мерцание. Для того, чтобы сложное изображение не мерцало требуется высокая скорость записи изображения дисплейным генератором, который работает на аналоговых, цифровых или смешанном принципах. В настоящее время используются чисто цифровые генераторы. Их основные преимущества: большая точность повторения построений (на аналоговых устройствах бывает заметен мелкомасштабный дребезг отрезков из-за недостаточной точности повторения со временем), температурная стабильность, легкая адаптируемость к разрешению и скорости прибора отображения.

#### Генератор векторов

Назначение генератора векторов — соединение двух точек отрезком прямой. Все вектора картинки должны быть одинаково яркими, если иное не задано специально. Это достигается (при неизменном токе луча) за счет одинаковой скорости вычерчивания. Кроме задания X,Y-координат должна управляться и интенсивность луча для построения видимого отрезка при включенном луче и позиционирования при выключенном. Рассмотрим генератор векторов, работающий на принципе ЦДА — цифрового дифференциального анализатора (DDA — Digital Differential Analyzer).

#### Цифровой дифференциальный анализатор

Так как для управления трубкой отклоняющие по X и Y напряжения должны формироваться независимо друг от друга, запишем выражение для координат вектора в следующей общей форме:

$$V(U) = P_n + (P_k - P_n) \cdot U = P_n + dP \cdot U$$



где  $V = [X, Y]$  и  $P_n, P_k$  - точки начала и конца вектора.

Параметр  $U$  имеет  $k + 1$  дискретное значение  $U_i = i/k; i = 0, 1, \dots, k$ . В этом случае точка  $V_i = V(U_i)$  есть:

$$V_i = P_n + dP \cdot (i/k) = P_n + i \cdot (dP/k)$$

Умножение, путем  $i$ -кратного сложения даст:

$$V_i = P_n + \underbrace{(dP/k) + \dots + (dP/k)}_i$$

Начальная точка  $P_n = V_0$ , конечная точка  $P_k = V_k$ .

Очередная точка вычисляется из соотношения:

$$V_{i+1} = V_i + (dP/k)$$

Результат деления  $dP/k$  определяется, при представлении  $k$  в виде целой степени двойки  $k = 2^p$ , путем  $p$ -кратного сдвига  $dP$ . Вектор получается путем  $2^p$  кратного прибавления приращения  $dP/2^p$  к исходной точке. Так как все вектора на экране должны иметь одинаковую яркость, то скорость построения должна быть одинаковой. Вследствие того что ЦДА работает с определенной тактовой частотой, количество точек должно быть пропорционально проекции длины вектора на экран. Для определения длины вектора достаточно соотношения:

$$L = \max(|dX|, |dY|)$$

Число адресуемых точек обычно выбирается так, что диаметр точки  $D = 1.41 \times R$ , где  $R$  точно единица раstra. Это определяется необходимостью смыкания точек, стоящих в противоположных углах по диагонали квадратной растровой сетки.

Так как вектора на дисплее должны выглядеть непрерывными, то приращение  $|dP|/2^p$  должно быть меньше чем элемент раstra:

$$\left. \begin{array}{l} |dX|/2^p < 1 \\ |dY|/2^p < 1 \end{array} \right\} \Rightarrow 2^p > L$$

Чтобы по возможности уменьшить число сложений,  $p$  вычисляется таким образом, чтобы выполнялось:

$$2^p > L \geq 2^{p-1}$$

При положительных относительных координатах  $dP$  из этого соотношения легко аппаратно определяется  $p$ . Для этого сдвигают значения  $dP$  в сторону уменьшения, пока в обоих регистрах не останутся нули перед двоичной точкой. При этом одновременно вычисляются и значения  $dP/2^p$ . При отрицательных значениях вначале находится двоичнодополнительное число, а затем выполняется определение  $p$ .

Для того чтобы можно было отобразить больше векторов без мерцания одновременно выполняют построение вектора на экране и определение инкремента  $dP/2^p$  для следующего. Этим

способом можно выполнять в сумматоре без помех следующие сложения при начале нового вектора. При обработке коротких векторов можно, для повышения скорости вычислений, находить инкремент  $dP/2^p$  в “барабанном сдвигателе” (это сдвиговый регистр, в котором решена проблема перехода через границы слов). При использовании быстрых переключающих схем удастся достичь времени прорисовки диагонали дисплея за 20 мкс при разрешении  $1024 \times 1024$  (тактовая частота 50 МГц).

На рис. 0.8.2 приведена структурная схема 10-ти разрядного генератора векторов ( $1024 \times 1024$  точки). Определенные вычислителем 10-ти разрядные значения  $2^p$ , используются затем для вычисления 10-ти разрядных же дробных значений  $dX/2^p, dY/2^p$ . Эти значения прибавляются к дробной части 20-ти разрядных сумматоров координат, в целую часть которых в начале построения загружаются начальные координаты точки. Результаты суммирования передаются на цифро-аналоговые преобразователи (ЦАП’ы) для формирования аналоговых значений отклоняющих напряжений.

Ясно, что при длине слова ЦАП’а равной разрешению дисплея (здесь  $10 \times 10$  бит) отдельные фрагменты вектора будут отделены явно видимыми ступеньками в 1 дискрету. При увеличении разрядности ЦАП’а с захватом разрядов дробной части сумматора позиционирование будет более точным. Так при 11-битовых ЦАП’ах позиционирование будет с точностью до  $1/2$  пиксела, а при 12 разрядах — уточнение еще на  $1/4$ . Это поясняется в таблицах на рис. 0.8.2, в которых:

$$\begin{array}{ll} X_n = Y_n = 0, & X_k = 6, Y_k = 3 \quad \text{в десятичной системе} \\ dX/2^p = 0.11 & \text{в двоичной системе} \\ dY/2^p = 0.011 & \text{в двоичной системе} \end{array}$$

Другое улучшение может быть получено использованием округления. Для этого перед началом генерации вектора в оба сумматора надо загрузить 0.5. Отклонение вычерченной точки от точной позиции будет по максимуму равно половине растрового расстояния.

В растровых дисплеях ступеньки в одну дискрету возникают обязательно. Их устранение возможно не за счет уточнения позиционирования, как в векторных, а за счет корректировки изображений пикселов, что приводит к понижению пространственного разрешения. Более подробно борьба с лестничным эффектом в растровых дисплеях освещена во второй части курса.

### Другие алгоритмы генерации векторов

Описанный выше генератор векторов имеет тот недостаток, что точки могут прописываться дважды и поэтому на дисплее с произвольным сканированием могут вызывать нежелательные колебания яркости. Кроме того из-за независимого вычисления обеих координат нет предпочтительных направлений.

Субъективно, однако, лучше смотрятся вектора с единичным шагом по большей относительной координате (несимметричный ЦДА). Для  $dX > dY$  (при  $dX, dY > 0$ ) это означает, что координата по X направлению должна увеличиться на  $1 dX$  раз, а координата по Y-направлению должна также  $dX$  раз увеличиться, но на  $dY/dX$ . Так как, как правило,  $dX$  не является степенью двойки, то требуется выполнения деления, что затруднено при аппаратной реализации.

Брезенхем предложил алгоритм, исключаящий операцию деления, имеющуюся в ЦДА. Основная идея алгоритма состоит в том, что если угловой коэффициент прямой больше  $1/2$ , то естественно точку, следующую за точкой  $(0,0)$ , поставить в позицию  $(1,0)$  (рис. 0.8.3а), а если угловой коэффициент больше  $1/2$ , то — в позицию  $(1,1)$  (рис. 0.8.3б). Построение отрезка из точки  $(0,0)$  в точку  $(4.2,1.7)$  показано на рис. 0.8.3в).

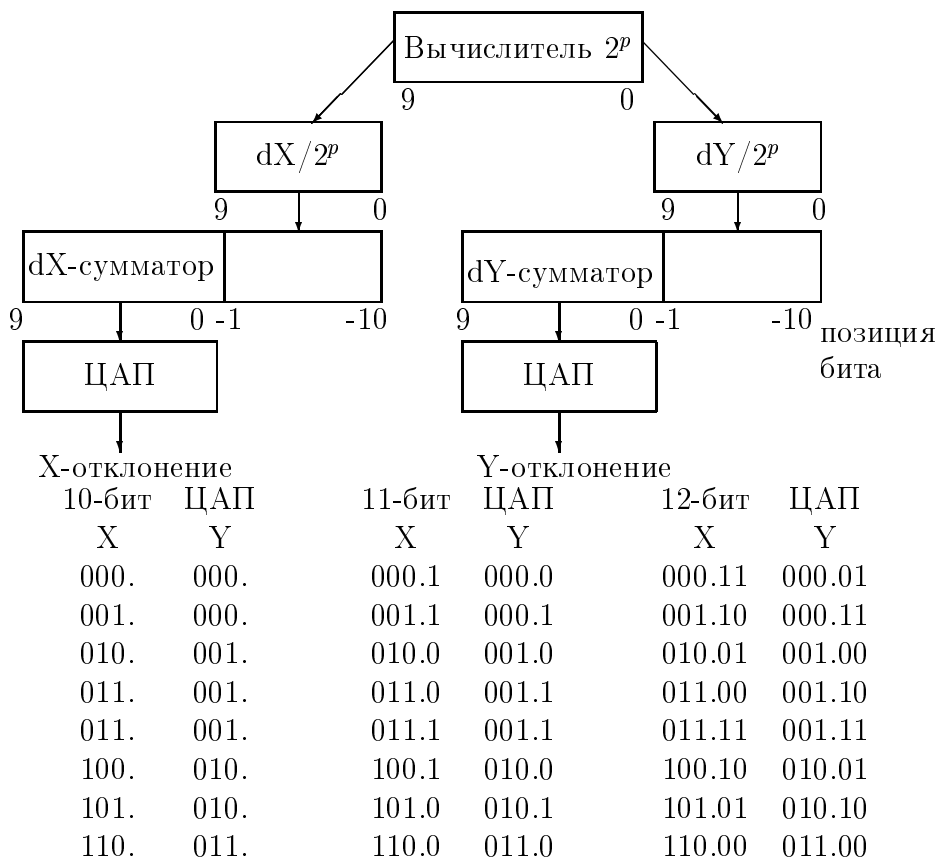


Рис. 0.8.2: Структура цифрового генератора векторов

Программные реализации алгоритмов ЦДА и несимметричного ЦДА, а также алгоритма Брезенхема для генерации отрезков рассмотрен во второй части курса — “Основные алгоритмы”.

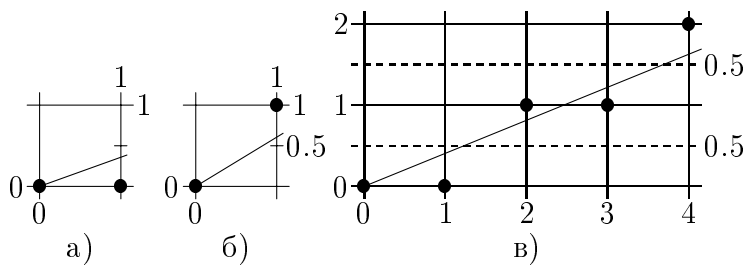


Рис. 0.8.3: Генерация вектора по алгоритму Брезенхема

Разработаны алгоритмы цифрового генератора векторов для окружностей и конических сечений.

Цифровой генератор векторов представляет собой специализированный, легко применимый процессор, встраиваемый в современные растровые системы как один из функционально-ориентированных процессоров. Его интеграция с приборами вывода различной точности и скорости показана на рис. 0.8.4.

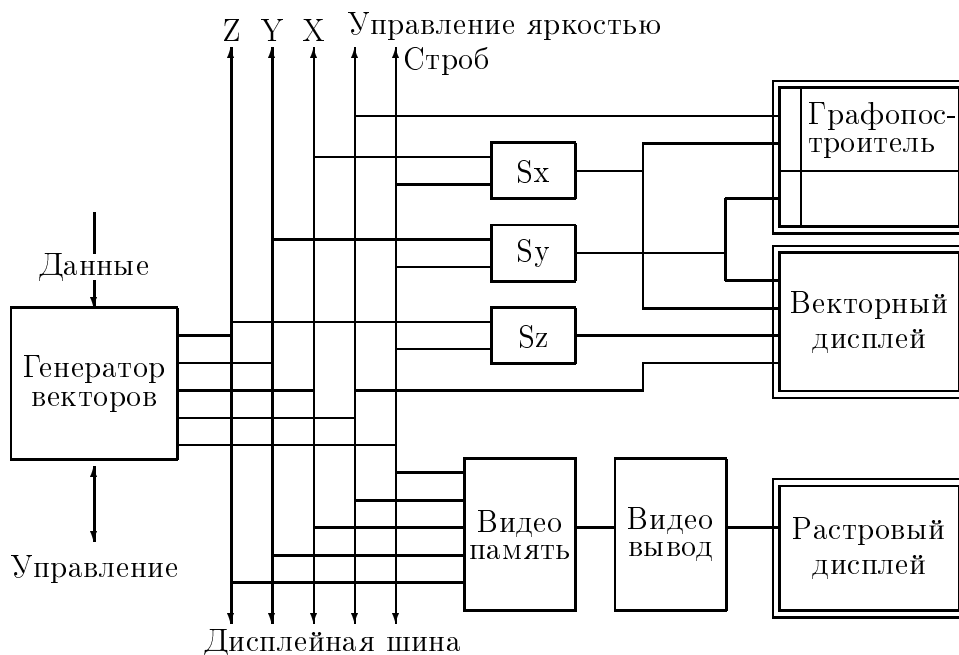


Рис. 0.8.4: Варианты использования генератора векторов

### Знакогенератор

Знаки (буквы, цифры, значки), наряду с векторами, являются наиболее широко употребляемыми графическими примитивами. Они в общем представлены в закодированной форме, например, ASCII, КОИ-8 и т.д. Эти коды преобразуются знакогенератором в координатные значения для представления соответствующих знаков.

Используются два основных метода — векторный метод и метод матрицы точек.

#### Векторный метод

Знаки запомнены в генераторе символов в виде приращений координат. Для формирования изображения знака вызывается генератор векторов для построения отрезков, составляющих знак. Для повышения эффективности кодирования используются короткие векторы в виде отрезков, соединяющих точки специального символического растра. Знак кодируется как последовательность номеров растровых точек. Фактические относительные координаты, которые передаются генератору векторов, должны перед этим преобразоваться, чтобы, например, сохранить масштаб или направление вычерчивания знаков. При этом такое преобразование не влияет на качество знака. Генератор знаков, работающий таким образом, имеет высокое качество генерации знаков и пригоден для генерации определенных пользователем последовательностей символов.

#### Метод матрицы точек

При использовании такого метода знаки заданы как последовательности точек внутри точечной матрицы (например,  $5 \times 7$  или  $16 \times 32$  точек) и запомнены в генераторе знаков. При выводе знака матрица знака сканируется по строкам или по столбцам и используется для управления

включением/выключением луча. Качество символов приемлемое только при отображении 1:1. Координатные преобразования, за исключением сдвига и поворота на углы, кратные 90, как правило, невозможны. Так как тогда каждая отдельная точка знака должна бы была преобразовываться и соответствующим образом выводиться, что приводило бы к затратам времени, неприемлемым для большинства приложений.

Наряду с управлением отклонением включением/выключением луча в некоторых реализациях знакогенератор формирует сигналы для управления межсимвольным расстоянием в строке. Знакогенераторы с матрицей точек обычно используются в дисплеях с растровым сканированием луча.

### **Генераторы кривых и плоскостей**

Наряду с генераторами векторов и знаков, многие дисплеи имеют генераторы кривых и плоскостей. Преимущество их использования в первую очередь состоит в большей скорости вычерчивания, а, кроме того, экономится память для хранения картины и требуется меньшее время для ее передачи. В основном, используются генераторы дуг окружностей и эллипсов.

## **0.8.2 Преобразования**

Преобразование изображения заключается либо в его геометрическом, координатном преобразовании, либо в преобразовании видимости, заключающемся, в частности, в отсечении частей полной сцены, невидимых в заданном окне просмотра.

### **Координатные преобразования**

Дисплеи большой мощности обеспечивают выполнение геометрических преобразований (сдвиг, вращение, масштабирование, перспектива) в реальном времени с помощью аппаратуры перемножения матриц. При этом используются однородные координаты, позволяющие любое преобразование представить как умножение координат вектора на матрицу  $4 \times 4$ . На преобразование требуется 16 умножений и 12 сложений. При перспективных преобразованиях дополнительно требуются 2 деления. Аппаратура преобразований, в основном, реализуется на СБИС. На обычных стандартных чипах достигнуты времена преобразования в 3 мкс (без делений).

### **Отсечение векторов**

Если изображение выходит за пределы экрана, то на части дисплеев увеличивается время построения за счет того, что изображение строится в "уме". В некоторых дисплеях выход за пределы экрана приводит к искажению картины, так как координаты просто ограничиваются при достижении ими граничных значений, а не выполняется точный расчет координат пересечения (эффект "стягивания" изображения). Некоторые, в основном, простые дисплеи просто не допускают выхода за пределы экрана. Все это, особенно в связи с широким использованием технологии просмотра окнами, требует выполнения отсечения сцены по границам окна видимости.

Программное исполнение отсечения достаточно медленный процесс, поэтому, естественно, в мощные дисплеи встраивается соответствующее оборудование.

Первое сообщение об аппаратуре отсечения, реализующей алгоритм Коэна-Сазерленда для отсечения отрезка на прямоугольном окне в устройстве Clipping Diviger, появилось в 1968 г.

Имеются реализации в СВИС варианта алгоритма Сазерленда-Ходгмана для отсечения многоугольника на прямоугольном окне. В этом алгоритме многоугольник последовательно отсекается всеми четырьмя сторонами окна.

Различные алгоритмы отсечения отрезков и многоугольников, в том числе и уже упомянутые алгоритмы Коэна-Сазерленда и Сазерленда-Ходгмана, подробно рассмотрены во второй части курса — “Основные алгоритмы”.

### 0.8.3 Дисплейный контроллер

Из блок-схемы простой графической системы (см. рис. 0.8.1), после рассмотрения работы дисплейного генератора и блока преобразований, ясно назначение дисплейного контроллера:

- связь с компьютером;
- управление формированием изображения;
- обработка графического ввода.

Специального пояснения обеспечения связи с компьютером повидимому не требует.

В чем заключается обработка графического ввода будет в основном выяснено при рассмотрении различных устройств ввода.

Здесь требует пояснений лишь функция управления формированием изображений, которая зависит от выбранной архитектуры. Если, например, дисплейный файл находится в ЦП, то ЦП выставляет стартовый адрес и иницирует контроллер. Когда контроллеру требуется следующая команда, то он приостанавливает процессор и выполняет доступ к памяти. Полученная команда декодируется и исполняется. Это могут быть команды построения, управления атрибутами построения, выполнения преобразований, отсечения, задания следующего адреса доступа и т.д. Понятно, что в этом случае контроллер отнимает время ЦП, которое могло бы быть использовано более полезным образом. С этой целью проанализируем различные построения графических систем.

### 0.8.4 Варианты архитектуры графических систем

Здесь мы рассмотрим самые общие соображения по построению графической системы, не зависящие от метода формирования изображения (произвольное или растровое сканирование луча).

В общем, цель графической системы — визуализация модели объекта, обрабатываемой компьютером.

Можно представить себе два основных способа представления модели объекта — алгоритмический, когда модель объекта вместе с описанием изображения и процесса его генерации “размазаны” по некоторому алгоритму. Другой способ — представление модели объекта в структуре данных. В этом случае достигается разделение по сути дела различных функций — манипулирование моделью (моделирование объекта, сцены и т.п.) со стороны прикладной системы и отображение модели в виде картин.

Ясно, что отображение, хотя временами и очень трудная, но стабильная проблема, в то время как моделирование определяется потенциально бесконечным количеством приложений. Поэтому, оговорив интерфейс, есть смысл разделять моделирование и отображение так чтобы было возможно из независимое развитие.

Принципиальным моментом в разработке распределенной системы является выбор оптимального разделения функций между подсистемами моделирования и отображения. Естественно предполагать, что какая-либо граница раздела должна находиться в месте полного завершения выполнения некоторой функции. Кроме того, вряд ли целесообразно разделять между различными подсистемами некоторый файл данных и программу (процессор) его “интенсивной” обработки. Для уяснения задачи распределения функций рассмотрим модель графической системы с точки зрения процессов преобразования информации при формировании изображения.

Процесс вывода может быть представлен следующей схемой:

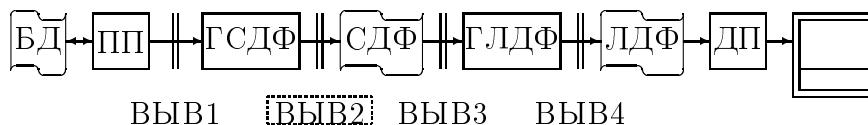


Рис. 0.8.5: Модель процесса вывода в графической системе. БД — база данных прикладной программы; ПП — прикладная программа; ГСДФ — генератор структурированного дисплейного файла; СДФ — структурированный дисплейный файл; ГЛДФ — генератор линейного дисплейного файла; ЛДФ — линейный дисплейный файл; ДП — дисплейный процессор; ВЫВ1, ВЫВ2, ВЫВ3, ВЫВ4 — возможные границы разделения функций

В этой модели отдана дань традиционному соображению о полезности структурированного дисплейного файла, представляющего собой иерархию вложенных объектов. Каждый объект, кроме примитивных элементов, может содержать экземпляры объектов нижнего уровня. Экземпляр объекта может сопровождаться набором совершаемых над ним преобразований. Использование экземпляров объекта выполняется либо его копированием, либо ссылкой (аналогично вызову подпрограммы). На самом же деле такое описание изображения подразумевает иерархичность его структуры, которая действительно имеет место в только ряде задач проектирования.

Итак: ВЫВ1 — подсистема моделирования готовит высокоуровневую информацию для построения структурированного дисплейного файла. Подсистема отображения формирует его и поддерживает работу с таким файлом, т.е. обмен информацией производится на наиболее высоком семантическом уровне.

ВЫВ2 — подсистема моделирования формирует структурированный дисплейный файл и передает его на хранение и интерпретацию в подсистему отображения. Понятно, что в этом случае возрастает разнообразие и сложность передаваемых данных, так как кроме информации по построению изображения должна передаваться информация, связанная с управлением структурой данных.

ВЫВ3 — подсистема моделирования формирует информацию для построения линейного дисплейного файла, возможно сегментированного. Линейный дисплейный файл формируется и поддерживается подсистемой отображения. Он является результатом выполнения геометрических преобразований, преобразований окно — порт и отсечения над полным изображением.

ВЫВ4 — подсистема моделирования формирует линейный (сегментированный) дисплейный файл и в аппаратно-ориентированной форме передает его подсистеме отображения, в задачи которой входит лишь поддержка функций приема и интерпретации файла в изображение дисплейным контроллером. Требования к вычислительной мощности подсистемы отображения и ее программному обеспечению в данном случае минимальны.

Следует отметить, что, так называемая, аппаратно-ориентированная форма для современных дисплеев, как правило, весьма высокоуровневая и включает в себя типичные наборы примитивов, атрибутов, средств сегментирования и даже структурирования. Аппаратность же в основном заключается в использовании некоторой вполне определенной формы кодирования, причем обычно наиболее компактной.

Из рассмотренного ясно, что наиболее целесообразны границы раздела ВВВ1, ВВВ3, ВВВ4. Причем первые две обеспечивают аппаратную независимость, а последняя соответствует минимальным требованиям к подсистеме отображения.

Естественно полагать, что подсистема моделирования находится в ЦП, а подсистема отображения представлена дисплейным процессором.

Для повышения быстродействия при выполнении преобразований используются архитектуры с двойной буферизацией, использующие два дисплейных контроллера и два буфера для сохранения изображения. Дисплейный контроллер 1 выполняет преобразования и формирует линейный дисплейный файл в одном из буферов. Параллельно с этим более простым дисплейный контроллером 2 ведется отображение из второго буфера. Затем буфера переключаются.

### 0.8.5 Области применения векторных дисплеев

Основными качествами векторных дисплеев являются:

- векторное представление с высоким разрешением,
- динамическое представление с преобразованиями в реальном масштабе времени,
- высокая интерактивность.

Параметры векторных дисплеев по разрешению, достигнутые уже более десятилетия назад, все еще проблематичны для растровых устройств. Так, например, дисплей СМ 7316 (НИИ периферийного оборудования, Киев, 1982 г.) мог отображать без мерцания до 4000–6000 векторов и 4096 символов при разрешении  $2048 \times 2048$ .

Поэтому такие дисплеи хорошо подходят для задач САПР, в которых требуется быстрое представление большого числа отрезков и малое время ответа на воздействие пользователя.

Кроме этого векторные дисплеи (их геометрическая часть) могут рассматриваться как геометрические процессоры для растровых систем.



## 0.9 Растровые дисплеи

Растровый (телевизионный) принцип формирования изображения заключается в разложении изображения на горизонтальные строки, состоящие из отдельных элементов. Вывод такого изображения осуществляется независимо от процесса построения с одинаковой скоростью последовательным сканированием по строкам в направлении сверху-вниз от 25 до 80 раз в секунду. В отличие от векторных дисплеев, в силу отделения процесса формирования картины от процесса ее вывода, сложность немерцающего изображения не ограничена.

Хотя телевидение (аналоговое) появилось еще в 30-е годы, использование телевизионного построения представления изображений в компьютерной графике стало реально возможным с появлением дешевой полупроводниковой памяти для хранения растрового представления и прогрессом в технологии мониторов.

Растровые дисплеи имеют наиболее широкое распространение, что связано со следующими основными свойствами устройств этого класса:

- обеспечивается наивысшее качество при меньшей стоимости,
- полные цветовые возможности и легкость представления закрашенных поверхностей,
- возможность совместимости с телевидением позволяет смешивать синтезированные и естественные изображения и поддерживать новые технологии в телекоммуникациях (видеотекст, системы Multimedia),
- интерактивная компьютерная графика и обработка изображений могут выполняться в рамках одной системы,
- сложность немерцающего изображения практически неограничена,
- телевизор хорошо знаком каждому как привычный предмет обихода.

Основные компоненты растрового дисплея показаны на рис. 0.9.1. Подсистема создания изображения формирует растровое представление сцены из описания, переданного от компьютера. Это поточечное представление в виде отдельных пикселей (от словосочетания picture element, иногда используется “pel” — “пэл”) заносится видеопамять (кадровый буфер, буфер регенерации, битовая карта). Видеопамять непрерывно и независимо от процесса занесения поточно сканируется видеоконтроллером, формирующим сигналы для монитора, выдающего картинку.

Сравнивая с векторными дисплеями (см. рис. 0.8.1, 0.8.5 – ??), видим что перечисленные компоненты соответствуют позиции дисплейного генератора. Назначение и архитектура других компонент графической системы, занимающихся геометрическими преобразованиями (если они есть), конечно, подобна, но эти компоненты не являются принципиальными при рассмотрении растрового дисплея.

### 0.9.1 Видеоконтроллер

Видеоконтроллер формирует изображение на экране монитора путем его поточечного вывода. При этом точно соблюдает время развертки строки, либо соответствующее телевизионным нормам, либо predetermined параметрам (ширина полосы, частота строк и кадров и т.д.). Некоторые мониторы для сокращения полосы пропускания используют т.н. чересстрочную развертку (Interlacing), при которой две картинки имеют половинное вертикальное разрешение и выдаются на экран по двум вложенным в друг друга гребенкам строк (рис. 0.9.2). Вначале выдаются строки 1-го полукадра, затем 2-го. Таким образом, при частоте кадров 50 Гц полная

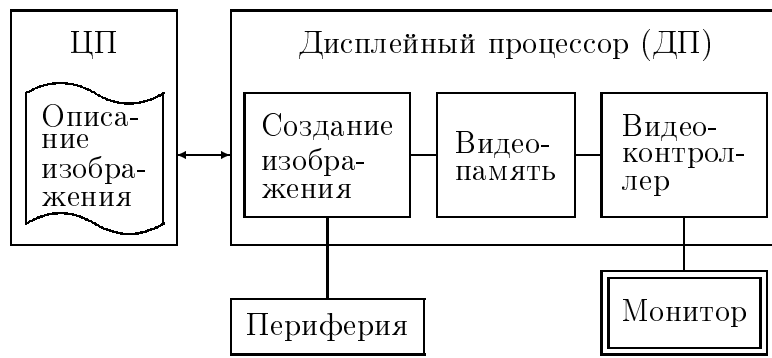


Рис. 0.9.1: Основные компоненты растрового дисплея

смена изображения на экране выполняется с частотой 25 Гц. Такой способ используется в телевидении. Для синтезированных изображений он мало пригоден (при частоте кадров 50 Гц), так как одиночные горизонтальные линии, занимающие ровно одну строку, неприятно и очень утомляюще мерцают.

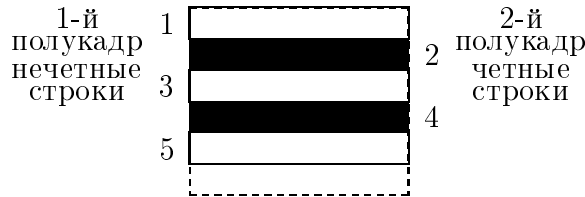


Рис. 0.9.2: Чересстрочная развертка

Время, требуемое для вывода картинка на монитор, определяется частотой строк и кадров, а также временами обратного хода строчной и кадровой разверток (рис. 0.9.3). Например, для телевизионного стандарта в 625 строк длительность строки составляет 64 мкс, из которых 12 мкс (почти 20%) — время обратного хода строчной развертки. Из времени картинка должно забираться время на обратный ход кадровой развертки. В этом случае длительностью в 25 строк.

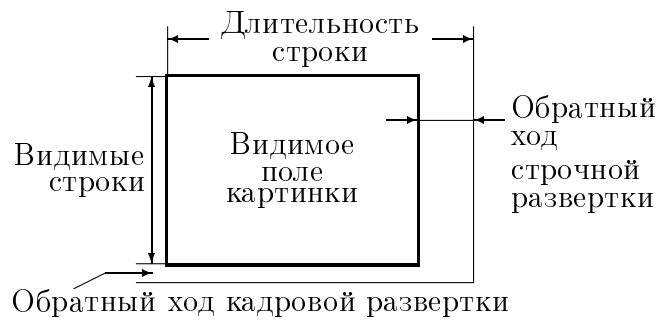


Рис. 0.9.3: Потери на обратный ход строчной и кадровой разверток

Важнейшим параметром не только для монитора, но и для растровой системы в целом является время, требуемое на обработку одного пиксела. Время, требуемое на обработку строки

аппаратурой дисплея, составляет:

$$t_s = \frac{1/\text{частота кадров} - t_{\text{обратного хода кадровой развертки}}}{\text{число видимых строк в картине}}$$

Причем для чересстрочной развертки обратный ход кадровой развертки за полную прорисовку картинку выполняется, естественно, дважды.

Так как часть времени строчной развертки тратится на перемещение луча к началу следующей строки (обратный ход строчной развертки, horizontal retrace), то время на обработку одного пиксела составит:

$$t_n = \frac{t_s - t_{\text{обратного хода строчной развертки}}}{\text{число пикселей в строке}}$$

В таблице 0.9.1. приведены времена для некоторых мониторов.

Таблица 0.9.1: Характерные времена для некоторых мониторов

Разрешение X×Y	f кадр. Гц	t полн. строки мкс	t обр. строки мкс	t обр. кадров мкс	Черес- строч- ность	тп на пиксел нс
512×512	30	60.4	10.9	1203	да	96.7
768×576	50	64	12	1612	да	70.3
1024×768	60	20.92	4	600	нет	16.52
1024×1024	60	15.69	4	600	нет	11.42
1280×1024	60	15.69	4	600	нет	9.13

В мониторах используются кинескопы с прямоугольной областью отображения с соотношением сторон обычно 4:3, что соответствует особенностям зрения человека. Если же для дисплея отношение числа пикселей в строке к числу строк отличается от 4:3, например, 512:512, то для занятия под изображение всей поверхности экрана потребуется сформировать не квадратные, а прямоугольные пиксели. Чтобы картинка выглядела на экране правильно (круг как круг и т.д.) ее потребуется предварительно исказить. Поэтому все современные дисплеи либо имеют соотношение 4:3 (640:480, 768:576, 1280:960 и т.д.), либо занимают под изображение не всю поверхность экрана.

После этих замечаний относительно представления растровых изображений легко сформулировать назначение видеоконтроллера:

- адресация и чтение данных из видеопамати;
- формирование синхроимпульсов разверток по горизонтали и вертикали, соответствующих формату изображения. Эти синхроимпульсы используются монитором для формирования отклоняющих напряжений;
- управление монитором для задания требуемых цветов и интенсивностей, цифро-аналоговое преобразование.

Кроме этого, из таблицы 0.9.1. видно, что времена на обработку пиксела существенно меньше, чем времена доступа к динамической полупроводниковой памяти, составляющие 100 нс и более, поэтому видеопамать обычно организована специальным образом, так что из нее читается сразу не один пиксел.

Дополнительно заметим, что пиксели строки отображаются на экран строго последовательно. Поэтому обычным техническим решением является перепись пикселей в быстрый сдвиговый регистр с целью параллельно-последовательного преобразования и формирования сигналов управления цветами и интенсивностями по данным из этого регистра.

Таким образом блок-схема простого видеоконтроллера имеет вид:

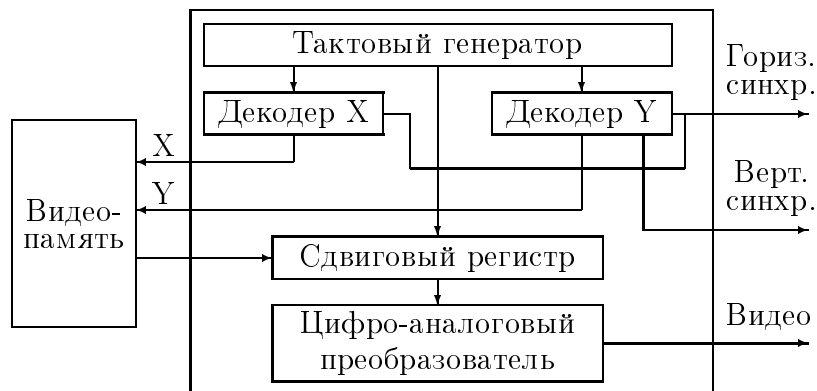


Рис. 0.9.4: Блок-схема простого видеоконтроллера

### Полноцветные дисплеи и дисплеи с таблицей цветности (ТЦ)

Мы, по сути дела, рассмотрели однобитовый растровый дисплей (дисплей с одной битовой плоскостью, одним битовым планом). Увеличивая количество плоскостей мы, естественно, увеличиваем количество отображаемых цветов. Можно выделить два класса цветных растровых дисплеев:

- полноцветные дисплеи, в которых для каждого пиксела сразу хранятся значения R, G, B. Обычно полноцветные дисплеи имеют по 1 байту на цвет. Т.е. объем видеопамати составит почти 4 Мбайта при полной картинке  $1280 \times 1024$  специально организованной, быстрой памяти. В таком дисплее на экране одновременно формально может присутствовать до  $2^{24}$  цветов (более 16 миллионов);
- дисплеи с таблицей цветности (ТЦ) (Look up table — LUT), в которых значение считанного пиксела не сразу передается на цифро-аналоговое преобразование, а используется как адрес в таблице цветности. По этому адресу выбираются значения яркостей по R, G, B и уже они определяют цвет точки. Таким образом, имея видеопамать в 1 байт на пиксел и таблицу цветности с тремя байтами на элемент, мы можем одновременно отобразить 256 оттенков из  $2^{24}$ . Основное свойство этого дисплея, однако, состоит не в том, что экономится память, а в том, что изменяя всего 1–3 байта в таблице цветности, можно одновременно изменить цвет у всех точек изображения с одинаковым кодом пиксела, не изменяя содержимого видеопамати. С помощью таблицы цветности реализуются не только различные динамические визуальные эффекты, но может быть реализована произвольная фильтрация, получены любые битовые срезы, выполнены преобразования гистограммы без изменения данных в видеопамати, а также гамма-коррекция (см. п. 0.7.5)

Использование таблицы цветности поясняется на рис. 0.9.5.

#### Управление плоскостями изображения

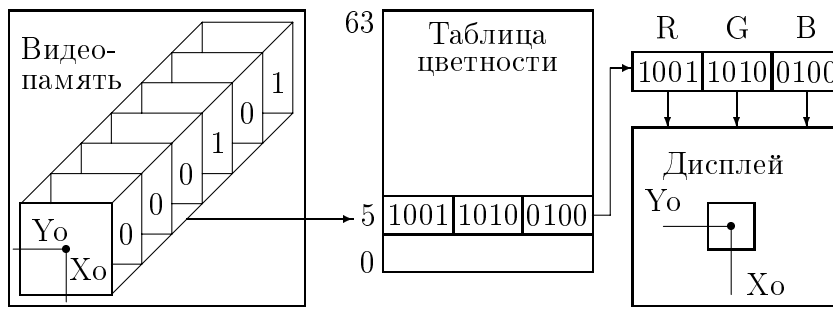


Рис. 0.9.5: Использование таблицы цветности для дисплея с 6-ти битным пикселом. В данном примере на экране одновременно может присутствовать 64 оттенка из полной палитры в 4096 оттенков

В дисплее в  $N$  битами на пиксел можно запомнить  $N$  черно-белых, бинарных или же  $N/3$  RGB картин. Каждая такая картина может отображаться с помощью соответствующей загрузки ТЦ. Если требуется, то их можно даже переключать с тактовой частотой! В случае черно-белого бинарного изображения ТЦ требуется загружать таким образом, чтобы ее элемент, соответствующий 1, содержал значения, дающее либо белый, либо черный оттенок.

Пусть, например, имеем дисплей с шестью битами на пиксел, с нумерацией плоскостей от нуля до пяти. Таблица цветности тогда должна иметь 64 элемента. Предположим, что надо включить изображение с зеленым цветом для плоскости номер ноль, а все остальные выключить. А затем включить красным цветом изображение в битовой плоскости номер один. Соответствующие этому раскраски таблицы цветности будут иметь вид:

Раскраска ТЦ для включения зеленым цветом изображения в плоскости номер 0

Код пиксела		Значение элемента ТЦ
дес.	двоичный	
0	000000	0 0 0
1	000001	0 G 0
2	000010	0 0 0
3	000011	0 G 0
4	000100	0 0 0
...	...	...
62	111110	0 0 0
63	111111	0 G 0

Раскраска ТЦ для включения красным цветом изображения в плоскости номер 1

Код пиксела		Значение элемента ТЦ
дес.	двоичный	
0	000000	0 0 0
1	000001	0 0 0
2	000010	R 0 0
3	000011	R 0 0
4	000100	0 0 0
...	...	...
62	111110	R 0 0
63	111111	R 0 0

### Приоритет изображений

Во многих приложениях требуются изображения, объекты которых хотя и размещены в пространстве, но не имеют глубины. В этом случае не требуется решение общих задач пересечения поверхностей и удаления невидимых граней. Достаточно просто упорядочить объекты по глубине. К этому классу относятся, например, изображения слоев печатной платы, изображения слоев чертежа, окна системы управления окнами и т.д. Это упорядочение или приоритет

может быть реализовано с использованием битовых плоскостей, закрашенных требуемым образом. Так как это не полное решение 3D задачи, то часто говорят об 2.5D изображениях. Если хотим получить эффект перекрытия отдельных прозрачных изображений, то заданному приоритету следует сопоставить соответствующий смешанный цвет в таблице цветности.

Для простоты рассмотрим трехбитовый дисплей. В старшей битовой плоскости разместим изображение А с наибольшим приоритетом, в младшей битовой плоскости — изображение В с наименьшим приоритетом. Возможные значения раскраски таблицы цветности в зависимости от значений пикселей для слоев, перекрывающихся в соответствии с приоритетом и для случая прозрачных слоев приведены ниже:

Значение пиксела	Содержимое таблицы цветности при перекрытии слоев	Содержимое таблицы цветности при прозрачных слоях
0 0 0	Цвет фона	черный
0 0 1	Цвет картинки В	синий
0 1 0	Цвет картинки Б	зеленый
0 1 1	Цвет картинки Б	изумрудный
1 0 0	Цвет картинки А	красный
1 0 1	Цвет картинки А	пурпурный
1 1 0	Цвет картинки А	коричневый
1 1 1	Цвет картинки А	белый
Картинка: А В В	Приоритет А > Б > В	Прозрачные слои

### Эффекты перемещений с помощью ТЦ

Динамическое изменение соотношений между кодами пикселей и соответствующих им цветом с помощью таблицы цветности позволяет моделировать эффекты движения, причем выполняемые очень быстро, в темпе разверток, а не в темпе генерации примитивов. Например, построим кривую с распределенными вдоль нее значениями кодов пикселей. Меняя элементы ТЦ, соответствующие этим кодам мы можем создавать эффекты ее постепенного построения, движения вдоль нее, рассыпания и т.д.

Другой способ — построим несколько раз кривую. Каждую реализацию своим кодом пикселей. Меняя содержимое таблицы цветности, может создать впечатление перемещения кривой по экрану, в том числе и с оставляемым ею следом.

### Преобразования изображения “на лету”

На уровне видеовывода, кроме цветовых преобразований, могут быть реализованы несложные геометрические преобразования в реальном времени причем также без изменения данных в видеопамати.

При этом жесткое соответствие 1:1 между адресами пикселей в видеопамати и их координатами на экране утрачивается и выполняется преобразование окно просмотра в видеопамати — окно видимости на экране (Widow — Viewport transformation). Обычно это преобразование включает в себя масштабирование, сдвиг и иногда поворот. Масштабирование обычно выполняется в целое число раз (Zoom) и заключается в соответствующем дублировании пикселей в X и Y направлении.

Для реализации сдвига нужны регистр координат и быстрые компараторы, которые сравнивают текущую позицию луча с желаемой новой позицией объекта и управляют адресацией для видеопамяти. Аналогичным образом может выполняться отсечение и вращение на 90.

Иное техническое решение для выполнения плавного масштабирования, перемещения, переверотов и даже произвольных перестановок строк изображения относительно горизонтальной оси и зеркального отображения относительно вертикальной оси предложено и реализовано В.Г. Сизыхом в станциях Гамма-4 и Гамма-Т [10]. Идея состоит в использовании программно загружаемой таблицы, устанавливающей соответствие между строками изображения и адресами видеопамяти, — т.н. таблицы адресов (ТА). Для каждой строки экрана имеется один элемент таблицы, задающий, в частности, адрес видеопамяти, с которого надо начать отображение, и масштаб по X. Увеличение по оси Y формируется повторением в очередной строке ТА предыдущего адреса видеопамяти. Уменьшение — пропуском очередного адреса. На рис. 0.9.6 приведен пример ТА для видеопамяти 1024×1024 при экранном разрешении 768×576 и выводе нижних 576 строк видеопамяти “вверх ногами” со смещением на 10 пикселей по оси X и единичными масштабами по осям. Начало видеопамяти — левый верхний угол.

Плавное, не целочисленное масштабирование выполняется путем выбрасывания/дублирования не всех, а части пикселей в строке и строк.

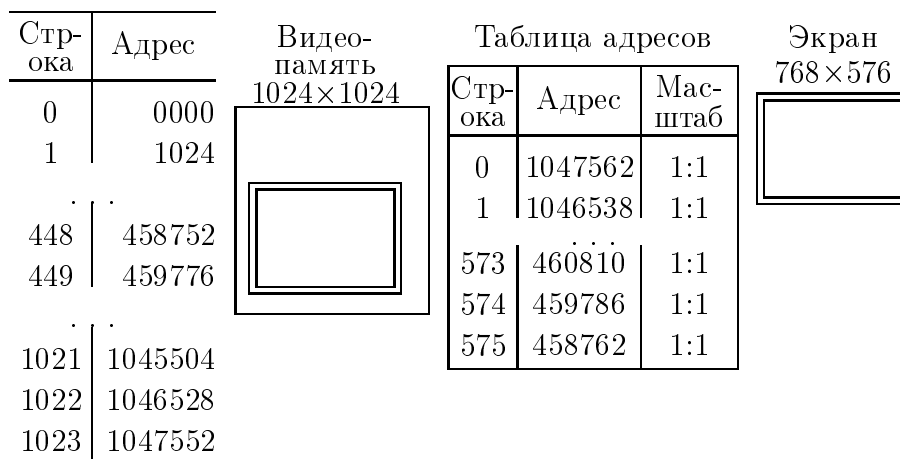
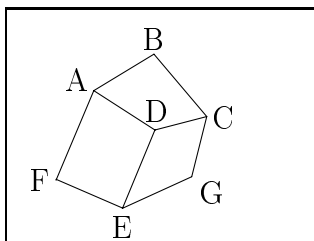


Рис. 0.9.6: Преобразование изображений в темпе разверток с использованием таблицы адресов

Также В.Г. Сизыхом предложен и реализован метод однотонной заливки многоугольников в темпе видеовывода, позволяющий повысить скорость раскраски по максимуму в 768 раз. Для обычных синтезированных сцен повышение скорости более чем в 100 раз. Идея состоит в том, что при выполнении видеовывода строки отличные от нуля коды пикселей “защелкиваются” на регистре и передаются на вывод до тех пока не придет код, отличный от установленного. Последовательность закраски следующая:

- кадровый буфер очищается (все пиксели получают значение 0);
- у однотонно закрасенных многоугольников кодом с требуемым цветом строятся только левые ребра;
- крайние правые ребра прорисовываются кодом отличным от 0, но цвет, соответствующий данному коду, должен совпадать с цветом фона.

На рис. 0.9.7 показано применение данного метода для закраски на белом фоне многоугольника ADCD красным цветом, ADEF — зеленым и EDCG — синим.



Записать элементы ТЦ				Пояснение	Построить отрезки
Но- мер	Байты ТЦ				
	R	G	B		
0	255	255	255	белый	AB кодом 2
1	255	255	255	белый	BC кодом 1
2	255	000	000	красный	CD кодом 4
3	000	255	000	зеленый	DA кодом 2
4	000	000	255	синий	DE кодом 4
					EF кодом 3
					FA кодом 3
					CG кодом 1
					GE кодом 1

Рис. 0.9.7: Закраска многоугольников “на лету”. ADCD — красным, ADEF — зеленым и EDCG — синим.

## 0.9.2 Создание изображения

Модуль генерации изображений растрового дисплея (см. рис. 0.9.1), часто называемый в литературе “image creation system” (ICS), имеет то же назначение что и дисплейный генератор для векторного дисплея. Важнейшее отличие от генератора изображений для векторного дисплея состоит в том, что в растровом дисплее генератор изображения в произвольное время заносит его представление в видеопамять (т.н. функционально-растровое преобразование).

Вывод изображения на экран осуществляется независимо работающей подсистемой отображения, которая синхронно с разверткой читает строки пикселей и выдает их на экран.

В результате такого разделения сложность изображения на растровом дисплее в принципе не ограничена. Фактическое же ограничение — количество пикселей на экране составляет для обычных дисплеев сотни тысяч элементов по сравнению с первыми тысячами для векторных дисплеев.

С этим несомненным достоинством связаны два основных недостатка растровых дисплеев по сравнению с векторными:

- непосредственное взаимодействие с изображением затруднено, так как временного соответствия между “указанием” объекта на экране и его генерацией не существует, поэтому в растровых системах используется программное моделирование сцены для определения совпадения координат объекта и положения того или иного устройства указания;
- динамическое представление “не дергающихся” перемещений требует, чтобы скорости генерации и воспроизведения совпадали. Это в общем случае возможно только на векторных дисплеях.

Прогресс в области вычислительных технологий, по-видимому, позволит найти приемлемое решение этих проблем, например с использованием рассмотренного ниже растрирования в реальном времени.



Набор команд для растровых дисплеев, соответственно дополнительным возможностям, шире. Он включает команды задания яркостей или цвета, определения закрашенных поверхностей, команды копирования блоков пикселей, известные как Bitblt (Bit Boundary Block Transfer) или RasterOp (растровые операции) и выполнения логических операций над кодами пикселей.

В общем, команда пересылки блока пикселей имеет вид:

приемник:= источник **ОП** приемник,

где **ОП** — арифметическая или логическая операция над кодами пикселей.

Практически используются следующие операции:

Очистка	приемник:= константа
Узор	приемник:= постоянный узор
Копирование	приемник:= источник
Инвертирование	приемник:= <b>NOT</b> приемник
Логическое И	приемник:= источник <b>AND</b> приемник
Логическое ИЛИ	приемник:= источник <b>OR</b> приемник
Исключающее ИЛИ	приемник:= источник <b>XOR</b> приемник

Особенный интерес представляет операция **XOR** (Exclusive OR), позволяющая после двойного ее применения с одним и тем же источником восстановить исходное значение пикселя. В частности, эта операция может использоваться для генерации курсоров графического ввода:

(Приемник **XOR** Источник) **XOR** Источник =

Приемник **XOR** (Источник **XOR** Источник) =

Приемник **XOR** 0 = Приемник

### Кодирование данных в видеопамяти

До сих пор рассматривались растровые дисплеи, у которых один пиксел соответствует одному элементу видеопамяти (т.н. Bit-Map дисплей). Очевидно, что при этом требуются большие размеры видеопамяти, так для изображения 1024×1024 пикселей при одном байте на пиксел требуется видеопамять в 1 Мбайт. Во многих случаях отличие двух соседних пикселей в строке и двух соседних строк мало, что в принципе позволяет сэкономить память. Мы здесь рассмотрим два наиболее простых способа кодирования — кодирование длин серий, заключающееся в учете повторения кодов пикселей вдоль строки и клеточное кодирование, представляющее изображение разбитым на отдельные фрагменты.

### Кодирование длин серий

К одному из простейших методов кодирования относится кодирование длин серий (run length encoding), заключающееся в том, что группа повторяющихся значений пикселей кодируется в виде пар из значения пикселя и числа его повторений.

С помощью такого кодирования достигается сокращение занимаемого объема в 10 и более раз. Так в задачах численного моделирования (см. п. ) средняя степень сжатия составляла 29 раз. Но для некоторых случаев возможно и увеличение объема, например, если изображение имеет вид шахматного поля из отдельных пикселей.

Ясно, что при таком способе кодирования утрачивается соответствие между координатой пикселя на экране и адресом в памяти изображения. Доступ к строкам возможен путем их по-

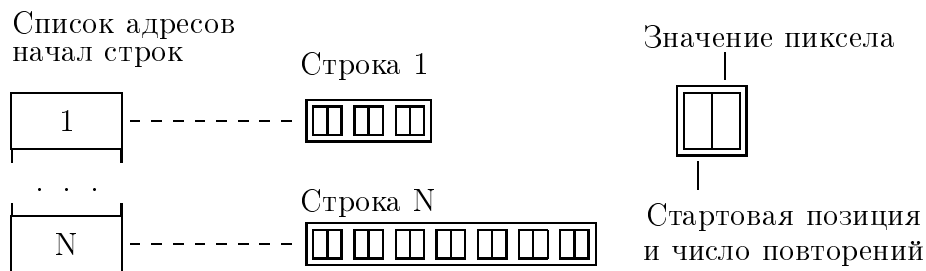


Рис. 0.9.8: Кодирование изображения длинами серий в виде списка

последовательного просмотра. Для сокращения времени доступа можно использовать списочную структуру, как это показано на рис. 0.9.8.

Легко видеть, что у метода кодирования длинами серий, особенно при интерактивной работе, имеются следующие недостатки:

- объем занимаемой памяти сильно зависит от изображения и резко возрастает при сокращении длины повторения;
- изменение изображения требует исключения и создания элементов списка;
- определение длины серии требует просмотра. Для каждой строки нужно все описание картины для определения возможного вклада.

В целом, этот метод кодирования используется не для интерактивной работы, а для хранения изображения на диске или для передачи. В частности, кодирование длин серий используется в описании изображения для растровых дисплеев фирмы Tektronix, и в графическом редакторе Paintbrush на IBM PC для оперативного хранения изображения.

Понятно, что искажение значения кодового слова при кодировании длин серий может привести к распространению искажения до конца строки или даже картины. Для защиты от этого в реально используемых схемах кодирования применяется периодическая вставка т.н. абсолютных кодов — незакодированных значений пикселей.

### Клеточное кодирование

В таком дисплее экран разбит на прямоугольные ячейки (знакоместа). Содержимое ячейки кодируется и размещается в видеопамати по адресу знакоместа. Сокращение занимаемого объема памяти достигается за счет меньшей длины кодового слова, по сравнению с количеством пикселей в ячейке. Этот метод пригоден для алфавитно-цифровых дисплеев и примитивной графики. Так, например, при размере ячейки  $8 \times 8$  и 8-ми битовом кодовом слове для дисплея со знакоместами достигается восьмикратная экономия памяти по сравнению с Bit-Map дисплеем. Наряду с символами и знаками в таком дисплее можно представлять и элементарные вектора. Функционально-растровое преобразование для векторов в таком дисплее выполняется подбором подходящей комбинации знакомест.

Такое представление используется в настоящее время только для алфавитно-цифровых режимов, так как в одном знакоместе может быть представлен либо один знак, либо отрезок и так как цвет и другие параметры отображения задаются сразу для всех пикселей знакоместа.

Использование знакомест для простого текстового дисплея разрешением  $512 \times 512$  точек при размере знакоместа  $8 \times 8$  и размере матрицы символа  $5 \times 7$  показано на рис. 0.9.9. Такая матрица символов позволяет представить три вида шрифта:

- большие + маленькие символы латиницы,
- большие + маленькие символы кириллицы,
- большие символы латиницы + кириллицы.

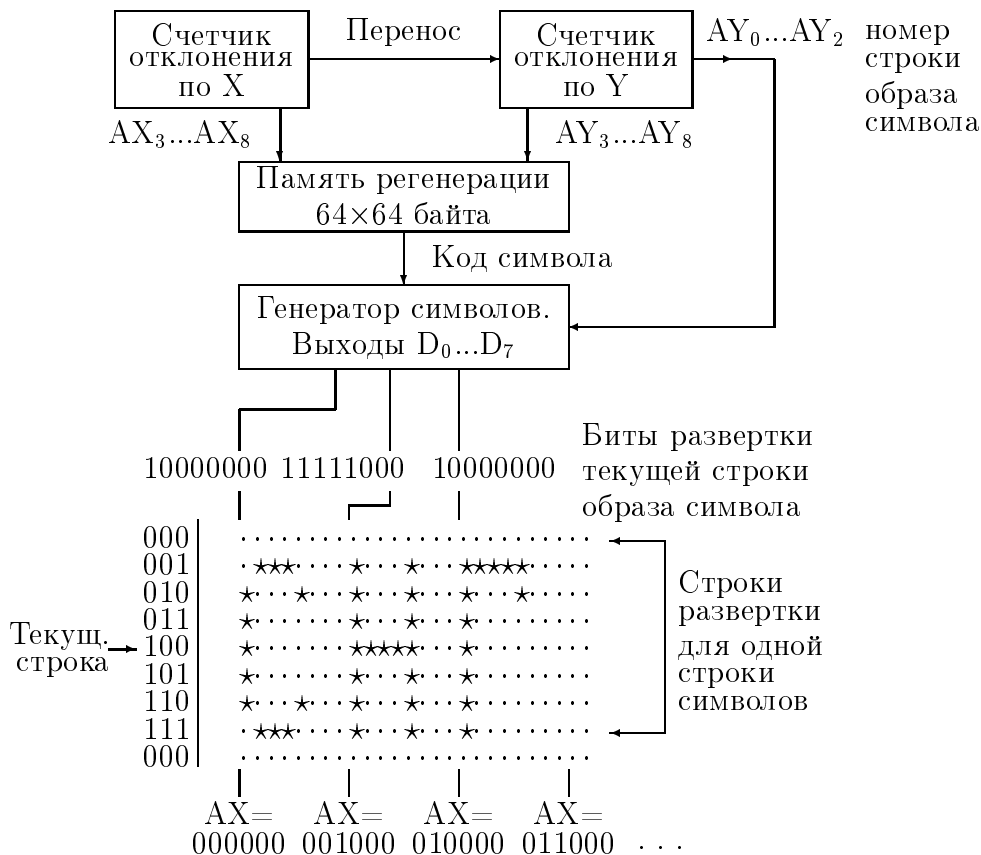


Рис. 0.9.9: Простой текстовый дисплей. Выдается строка из трех символов СНГ. Луч находится на 5-й строке развертки очередной строки символов. Значки  $\cdot$  отмечают возможные позиции пикселей. Значки  $\star$  отмечают позиции требуемых пикселей.

Кроме аппаратного использования для алфавитно-цифровых режимов, этот метод кодирования используется и в программной реализации для сокращения объема изображения при передаче или запоминании на диске. В частности, клеточное представление изображения используется в растровых дисплеях фирмы Tektronix и графических стандартах GKS (Graphical Kernel System) и CGI (Computer Graphics Interface).

### Повышение быстродействия подсистемы визуализации

Одним из возможных методов повышения быстродействия системы визуализации является растривание в реальном времени, при котором, как и в векторных дисплеях, промежуточное хранение изображения в видеопамати отсутствует. В этом случае для каждого пиксела экрана требуется просканировать описание всего изображения, чтобы определить вклад различных объектов картины. Насколько трудна эта задача видно из таблицы 0.9.1, когда для дисплеев с разрешением  $1024 \times 768$  это надо проделать за 16.5 нс! При введении буфера строки на обработку описания сцены требуется уже 20.9 мкс, что все равно слишком мало.

В какой-то мере решить эту задачу можно распараллеливанием. Можно усмотреть два способа распараллеливания:

- “процессор на геометрический элемент” — распараллеливание на уровне объектов, когда каждому примитиву описания изображения соответствует свой генератор (генераторы для векторов, кривых, поверхностей и т.д.),
- “процессор на пиксел” — распараллеливание на уровне пикселей, когда описание сцены “пропускается” сквозь пиксели и они решают что и как видимо.

В высококлассных, дорогостоящих графических системах в определенной мере используются оба способа, причем “процессор на пиксел” использовался только в экспериментальных дисплеях. Обычно же специализированный растровый процессор обслуживает полосу пикселей.

На рис. 0.9.10 приведен сравнительный анализ методов распараллеливания.

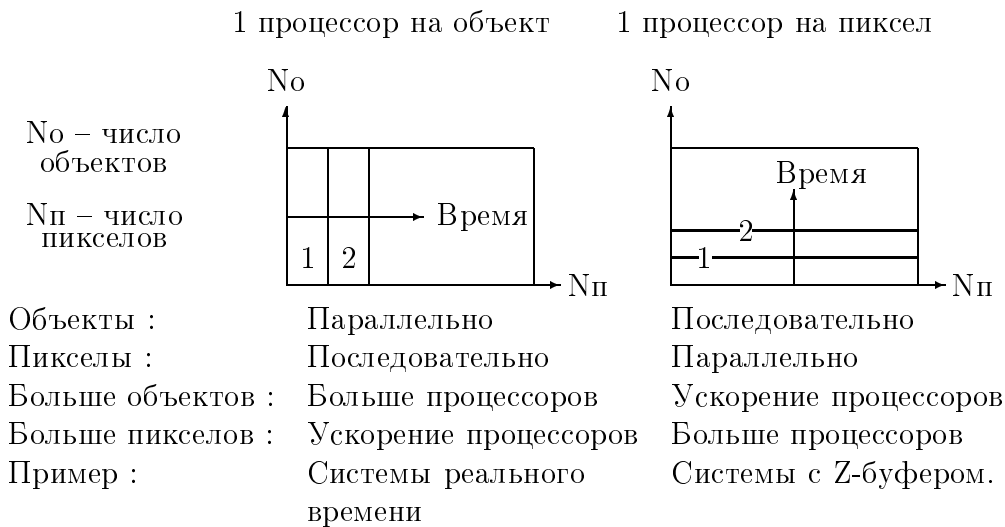


Рис. 0.9.10: Сравнение двух подходов к мультипроцессированию

При использовании мультипроцессоров главная проблема возникает не при преобразовании и разложении картины и не при вычислении пикселей, а при формировании полной картины из пикселей, которые сформированы из отдельных частей картины. Она должна решаться таким же образом, как и проблема перекрытия фрагментов, возникающая при реалистичном представлении трехмерных сцен.

### 0.9.3 Видеопамять

В ныне существующих растровых дисплеях одной из важнейших компонент, обеспечивающих взаимодействие подсистем генерации и отображения, является видеопамять. Каждая из подсистем должна иметь свой доступ к видеопамяти и большой ширины. Причем доступ от подсистемы отображения для обеспечения немерцающего изображения жестко синхронизован с развертками монитора. А доступ со стороны подсистемы генерации для достижения высокой интерактивности и больших скоростей перемещения объектов желательно исполнять в произвольный момент времени.

Важность специального внимания к разработке видеопамяти проиллюстрируем несложным примером. Очевидно, что при попытке одновременного доступа к видеопамяти со стороны подсистемы генерации изображения и со стороны видеоконтроллера возникает конфликт. В ранних дисплеях, таких как, например, система Pericolor 2000, конфликт разрешался простейшим образом: цикл памяти отдавался генератору изображения, так как если отдать его видеоконтроллеру, то информация в память не будет занесена и картинка исказится. При этом, естественно, некоторые точки изображения видеоконтроллером пропускались, так как развертка идет непрерывно. Это приводило к характерным помехам в виде “снега” на экране. После завершения

записи в память снег, конечно, пропадал. Один из простых способов решения этой проблемы — разрешение доступа генератору изображений во времена обратного хода разверток, когда изображение на экране не формируется. Это резко снижало быстродействие системы.

Видеопамять, как правило, строится на микросхемах динамической памяти с произвольным доступом (DRAM — Dinamic Random Access Memory), обладающих большим объемом и обычно имеющих организацию  $N \times 1$  разряд (однобитовая память). В дальнейшем изложении для упрощения анализа и почти без уменьшения общности, если иное не будет оговорено специально, будем рассматривать видеопамять наиболее распространенной, т.н. послойной архитектуры. В этом случае видеопамять организована в виде индивидуально управляемых битовых слоев с 1 битом на пиксел. Увеличение числа бит на пиксел достигается аппаратным комплексированием отдельных слоев. Для некоторых областей приложений, таких как, например, обработка изображений, более естественна архитектура “в глубину”, когда видеопамять организована с требуемым числом байт на пиксел.

Обычные микросхемы памяти имеют время цикла порядка первых сотен наносекунд. Из сравнения с таблицей 0.9.1 видно, что это заметно превышает время на пиксел, требуемое разверткой. В этой связи память организуется таким образом, чтобы за один доступ одновременно считывается многобитное слово памяти (16, 32, 64 бита). Следует отметить, что число микросхем памяти, требуемое для организации широкого доступа, обычно дает суммарный объем памяти больший, чем требуется из соображений разрешения. Рассмотрим, например, память для дисплея  $1024 \times 768$  пикселов в суммарном объеме в 786 432 бит. Требуемое время доступа к пикселу составляет 16.52 нс (см. таблицу 0.9.1). Используя микросхемы памяти организации  $64K \times 1$  бит с временем доступа 340 нс, получаем, что для достижения требуемого времени доступа надо одновременно запустить не менее 21 микросхемы. Суммарный объем памяти составит  $21 \times 64 \times 1024 = 1\,376\,256$  бит, что почти в два раза больше чем надо. На самом деле еще хуже, так как обычно ширину доступа выбирают их ряда 8, 16, 24 и т.д. Тогда получим суммарный объем памяти  $24 \times 64 \times 1024 = 1\,572\,864$  бита, что ровно в 2 раза больше.

На рис. 0.9.11 показан простой вариант для дисплея  $512 \times 512 \times 1$  бит с 16 корпусами  $16K \times 1$  400 нс динамической памяти с произвольным доступом (см. таблицу 0.9.1). Под управлением тактового генератора все микросхемы одновременно читаются и со скоростью  $\approx 10.3$  мегапикселов/с загружают данные в 16-ти разрядный сдвиговый регистр. Видно что память двух-портовая, т.е. шины данных, адреса и управления подключаются и от видеоконтроллера и от генератора изображений.

При одновременном доступе ко всем 16 микросхемам выдаются 16 пикселов, чем обеспечивается время отображения порядка 1600 нс. Так как цикл памяти в 400 нс, то видеоконтроллеру требуется только около 25% от имеющегося в наличии суммарного времени доступа. Остаток, включая и время обратного хода, может быть использован подсистемой генерации изображения для записи в видеопамять.

Замена этих 16 микросхем памяти на 4 микросхемы  $64K \times 1$  не улучшит, а ухудшит показатели системы. Ширина доступа станет 4 пиксела, т.е. обеспечится время отображения в 400 нс, а время цикла памяти остается порядка 400 нс, так что на доступ от генератора изображений останется только время обратного хода.

Таким образом, задачу увеличения разрешения дисплея применением микросхем большего объема решить нельзя. Будет требоваться все увеличивающаяся ширина доступа, так как цикл динамической памяти становится много большим времени отображения пиксела (см. таблицу 0.9.1). С другой стороны, как было ранее показано в примере для дисплея  $1024 \times 768$ , объем па-

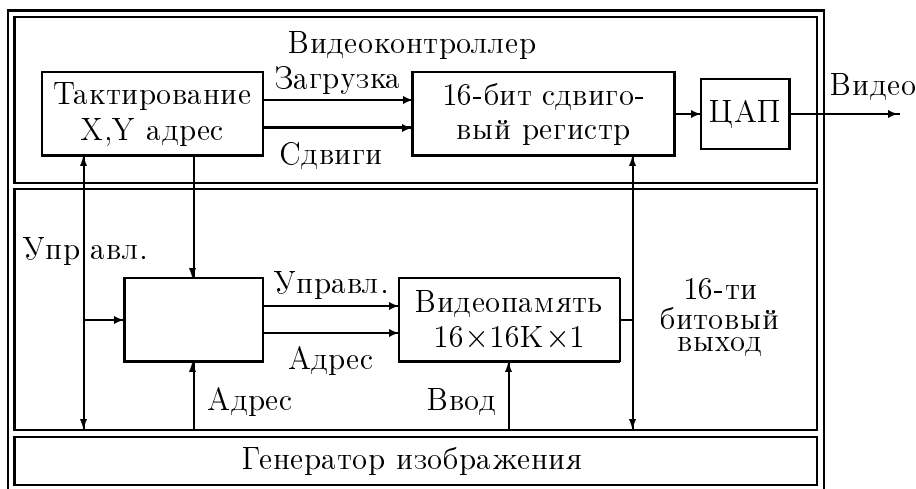


Рис. 0.9.11: Простейший растровый дисплей

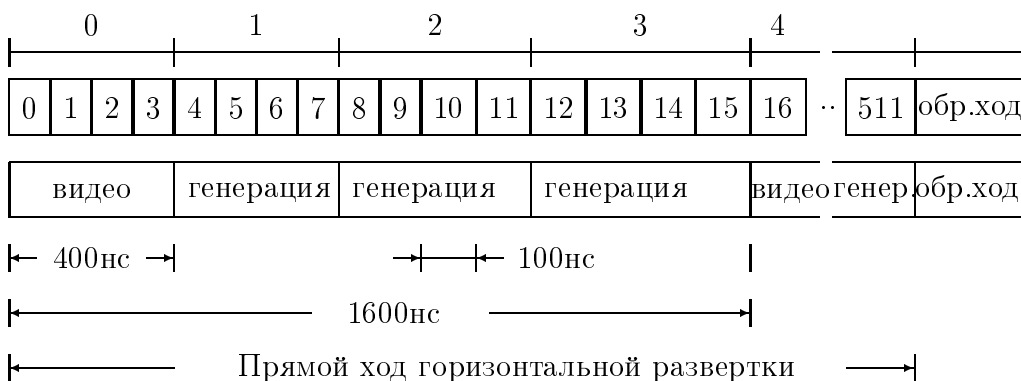


Рис. 0.9.12: Распределение цикла памяти между видеоконтроллером и генератором изображения для системы  $512 \times 512 \times 1$ , с  $16 \times 16K \times 1$  DRAM

мяти, определяемый шириной доступа становится все больше объема, требуемого разрешением. Требуется иные технические решения.

Радикальное и дорогостоящее решение. Времена доступа к статической памяти в 3–7 раз меньше, чем у динамической, энергопотребление в 1.3–5.5 раз больше, корпуса больших размеров, стоимость выше. Не требуется регенерации памяти. При некоторых эксплуатационных требованиях, например, повышенной радиационной стойкости, это единственное решение. Для иллюстрации в таблице 0.9.2 приведены некоторые параметры нескольких микросхем памяти одинакового объема.

Также недешевое, технически сложное решение. Используется два комплекта видеопамяти. В то время как идет отображение из одного буфера, второй буфер целиком отдается генератору изображений. Во время обратного хода кадровой развертки буфера меняются местами.

Взамен микросхем побитовой организации  $nK \times 1$ , используются микросхемы  $nK \times t$ , например,  $16K \times 4$ ,  $32K \times 8$ . Эти микросхемы можно использовать двояко:

Таблица 0.9.2: Параметры некоторых микросхем памяти

	Тип	Объем бит	Время нс	Мощн. мвт	Кор- пус	Цена
K565PY5	Динамич.	64Кх1	240-450	360/30	16	
K132PY10A	Статич.	64Кх1	75	460/165	22	
K537PY17	Статич.	8Кх8	200	470/22	18	

1. В одном слове микросхемы размещаются  $t$  пикселей. Очевидно, что сокращение времени доступа для видеовывода пропорционально длине слова. Недостаток — отдельная точка не может адресоваться индивидуально. Запись точки будет заключаться в вычислении адреса слова памяти, в которой она содержится; в определении позиции точки в слове; чтении слова; модификации требуемых разрядов; записи слова.

2. Одно слово микросхемы отводится под 1 пиксел, соответствующей глубины. До сих пор рассматривали дисплеи “послойной” организации, в которых для получения требуемой глубины пиксела набирается требуемое число слоев. Этот же поход — построение видеопамяти “в глубину”. Рассмотрим реализацию видеопамяти  $1024 \times 1024 \times 4$ . Для такого объема требуется 64 микросхемы  $64К \times 1$ . Полагая, что из соотношения между длительностью обработки пиксела и временем цикла микросхемы следует одновременно запускать 16 микросхем получим, что при использовании микросхем  $64К \times 1$  за один доступ к памяти запрашивается 16 пикселей по 4 бита каждый, в то время как при использовании микросхем  $16К \times 4$  того же самого объема в 64 Кбит будет запрошено уже 64 пиксела по 4 бита. Таким образом будет достигнуто лучшее соотношение между временами доступа видеоконтроллера и генератора изображений. Относительным недостатком такого подхода можно считать невозможность индивидуальной адресации отдельной битовой плоскости. Но такая возможность и не нужна для многих приложений. Для тех же приложений где это требуется отдельные слои могут защищаться с помощью программно загружаемой маски, как это, например, было сделано в дисплеях Гамма-4 и Гамма-Т [10]. Если разряды, разрешенные к модификации в регистре маски отмечаются 1, то его использование можно представить следующим образом:

$$V := ((V \text{ OR } M) \text{ XOR } M) \text{ OR } (N \text{ AND } M)$$

|

Зачистка разрядов в  
пикселе видеопамяти

|

Вырезание по  
маске разрядов в  
заносимом пикселе

где  $V$  — значение пиксела в видеопамяти,  $N$  — заносимый пиксел,  
 $M$  — регистр маски.

VRAM (Video Random Access Memory) [15] была разработана для решения проблемы конкуренции за циклы памяти между видеорегенерацией и модификацией изображения при их одновременном выполнении.

VRAM представляет собой двухпортовую динамическую память, имеющую в своем составе:

- собственно массив памяти  $256 \times 256 \times 1$  ( $64К \times 1$ ) или  $256 \times 256 \times 4$ ;
- 256 или 1024-битовый сдвиговый регистр.

Каждая из этих двух частей имеет собственные сигналы ввода/вывода и управления. Сдвиговый регистр загружается строкой из массива памяти за один цикл. После загрузки сдвиго-



вый регистр независим от массива памяти и может тактироваться с максимальной частотой до 25 МГц. После требования всего одного цикла для записи 256 или 1024 бит VRAM готова одновременно и для видеорегенерации и для обновления изображения. Время на доступ для отображения составляет менее 1.5% времени доступа. В системах же базирующихся на DRAM эти затраты достигают до 60%.

На рис. 0.9.13 представлена структура VRAM микросхемы TMS4161 фирмы Texas Instruments.



Рис. 0.9.13: Структура VRAM

## 0.10 Другие типы дисплеев

В данном разделе рассмотрено некоторые другие, менее распространенные типы дисплеев — дисплеи на запоминающей трубке, дисплеи на плазменной панели, дисплеи с жидкокристаллическим индикатором, электролюминисцентные дисплеи и дисплеи с эмиссией полем.

В дальнейшем планируется включить разделы по новым, уже появляющимся на рынке технологиям, таким как органические электролюминисцентные дисплеи и дисплеи с поверхностным эммитированием заряда (surface conduction emitter displays).

### 0.10.1 Дисплеи на запоминающей трубке

Как было отмечено во введении, запоминающие электронно-лучевые трубки — ЗЭЛТ (рис. 0.10.1) (Direct View Storage Tube — DVST) появились в конце 60-х годов. Основное их свойство — возможность сохранения изображения прямо на экране достаточно длительное время (до одного часа). Электронный луч от записывающего прожектора строит изображение на запоминающей сетке, представляющей собой мелкую проволочную сетку, покрытую диэлектриком.

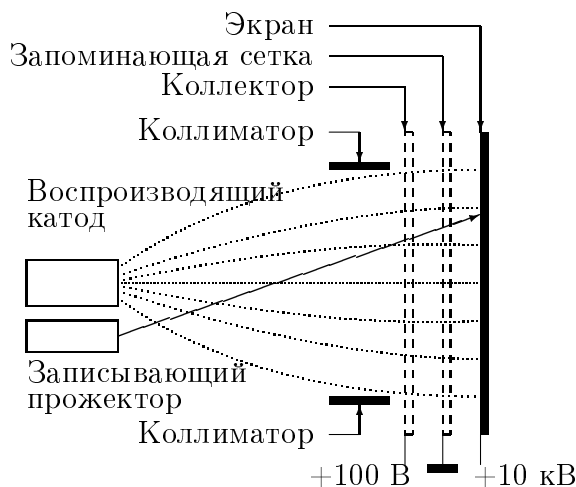


Рис. 0.10.1: Схема запоминающей электронно-лучевой трубки

Перед началом нового построения старое изображение стирается для чего запоминающая сетка равномерно отрицательно заряжается подачей на нее импульса длительностью около 0.5 с. При этом экран неприятно вспыхивает.

Изображение записывается высокоскоростными электронами записывающего прожектора, которые выбивают из запоминающей сетки электроны, захватываемые положительно заряженным коллектором. В результате на запоминающей сетке создается скрытое изображение в виде областей относительного положительного заряда. Скорость перемещения записывающего луча с сохранением эффекта запоминания не превышает 200 м/с. Т.е. диагональ обычного дисплея прорисовывается за время  $\approx 2$  мс, что в 100 медленнее чем на векторных дисплеях.

Воспроизведение запомненного изображения осуществляется с помощью потока несфокусированных низкоскоростных проявляющих электронов. Пластины коллиматора изгибают траектории проявляющих электронов таким образом, чтобы они подходили к запоминающей сетке под прямым углом. Попадая на участки относительного положительного заряда сетки, проявляющие электроны проходят сквозь сетку и вызывают свечение люминофора. Остальные проявляющие электроны отталкиваются и попадают на коллектор.

При уменьшении энергии записывающего луча возможно отображение без запоминания, когда луч возбуждает люминофор, но не изменяет распределение заряда на запоминающей сетке. Это используется, в основном, с вспомогательными целями — для вывода перемещающегося курсора, для отображения поясняющих надписей и рисунков.

Итак, основные виды работ ЗЭЛТ следующие:

1. Стирание изображения.
2. Запись изображения.
3. Воспроизведение запомненного изображения.
4. Отображение без запоминания.

Достоинства ЗЭЛТ:

- большое разрешение (до  $4096 \times 4096$  при 19 дюймах),
- отсутствие мерцания,
- низкие требования к скорости отклоняющей системы,
- малая стоимость.

Недостатки ЗЭЛТ:

- обычно невозможно выборочное стирание,
- мал контраст изображения,
- обычно черно-белое изображение,
- небольшая скорость работы.

Дисплеи на ЗЭЛТ представляют собой упрощенную модель векторного дисплея. Так как запоминающая трубка — устройство непосредственного хранения изображения, то не требуется периодического воспроизведения (регенерации) изображения. Достаточно однократного построения картинка. Вместе с тем скорости построения существенно ниже, однако следует заметить, что фактическое разрешение дисплеев на запоминающей трубке до  $4096 \times 4096$  не достигнуто пока ни каким другим дисплеем.

## 0.10.2 Дисплеи с плазменной панелью

Дисплеи с плазменной панелью (Plasma Display Panels — PDP) используют явление свечения при разряде в газе. Как показано на рис. 0.10.2, первые плазменные панели состояли из параллельных покрывающих прозрачных пластин, пространство между которыми заполнено газом (на основе неона) под атмосферным давлением. Между покрывающими пластинами находится стеклянная пластина с отверстиями (маска), разделяющая газ на растр из маленьких ячеек. На покрывающих пластинах с их внутренних сторон нанесены группы прозрачных полосок параллельных проводников, находящихся напротив отверстий в маске. Проводники на одной пластине перпендикулярны проводникам на другой пластине. Таким образом каждая ячейка представляет собой нечто вроде маленькой неоновой лампочки (на самом деле — тира-трона).

Если для незажженной ячейки между проводниками нарастает напряжение, то при достижении напряжения зажигания  $U$  (рис. 0.10.3) ячейка загорается. Газ в ячейке ионизируется, в ячейке возникает плазменный столбик, ток через нее заметно увеличивается, а напряжение падает до напряжения горения  $U$ , заметно меньшего, чем напряжение загорания. При уменьшении приложенного напряжения до напряжения потухания  $U$  разряд в ячейке прекращается

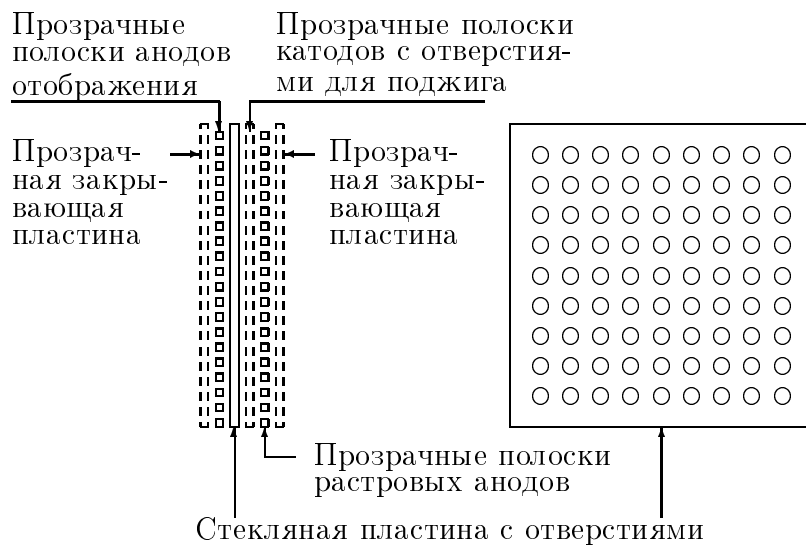


Рис. 0.10.2: Схема плазменной панели

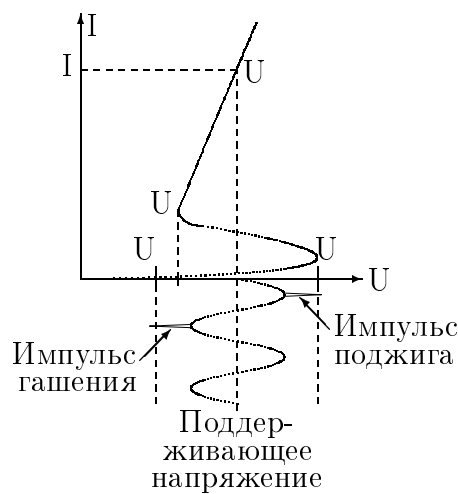


Рис. 0.10.3: Вольтамперная характеристика разряда в газе

и она гаснет за время порядка 20 мкс, за которое все ионы попадут на отрицательный электрод.

В зависимости от приложенного напряжения, ячейки панели могут находиться в трех состояниях:

- отображение содержимого панели. Поддерживающее напряжение между проводниками покрывающих пластин равно среднему напряжению горения  $U_c$  (см. рис 0.10.3). В этом случае горящие ячейки продолжают гореть и образуют светящиеся точки на изображении, а не горящие не зажигаются;
- включение точки. К поддерживающему напряжению между проводниками, пересекающимися над требуемой ячейкой, добавляется такой импульс поджига, что достигается напряжение зажигания  $U$  и ячейка загорается (если она еще не горела);

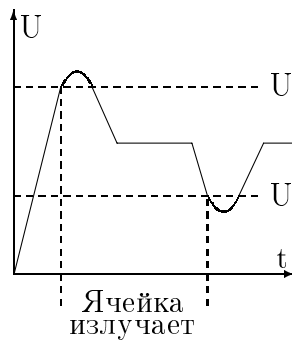


Рис. 0.10.4: Временная диаграмма работы плазменной ячейки

- стирание точки. К поддерживающему напряжению между проводниками, пересекающимися над требуемой ячейкой, добавляется такой импульс гашения, что достигается напряжение потухания  $U$  и ячейка выключается (если она горела).

Более поздние плазменные панели не содержат маски и растровых анодов. Она также состоит из двух параллельных стеклянных пластин, на внутренних поверхностях которых нанесены взаимно перпендикулярные группы проводников, формирующие растровые ячейки панели. Проводники покрыты диэлектрической пленкой и между ними приложено высокочастотное напряжение, недостаточное для зажигания, но достаточное для горения (поддерживающее напряжение). При высокочастотном питании газ ведет себя так, как будто он разделен на отдельные ячейки. Частота питающего напряжения такова, что хотя разряд при малых напряжениях и прекращается, но ионы не успевают рекомбинировать, поэтому при возрастании напряжения разряд возобновляется и создается впечатление непрерывного свечения точки.

Для включения негорящей ячейки к поддерживающему напряжению добавляется импульс поджига, достаточный для зажигания.

Для выключения горящей ячейки к поддерживающему напряжению добавляется такой импульс, что до достижения очередного максимума напряжение будет таково, что ионы успеют рекомбинировать и ячейка не загорится.

Известны и другие конструкции плазменных панелей, например, для вывода цветных полутоновых изображений. В таких панелях отдельный пиксел представляет собой триаду газоразрядных тиратронов, имеющих общий электрод для поддержания разряда в смеси редких газов типа He и Xe (рис. 0.10.5). Возникающее при этом ультрафиолетовое (UV) излучение возбуждает люминофоры, которыми покрыты торцы тиратронов. Эти люминофоры имеют соответствующие фотолюминесцентные характеристики и излучают красный, зеленый и синий цвета. Эффект изменения яркости моделируется за счет зажигания/поджигания разряда с требуемой (не слишком большой) частотой.

Дисплеи на плазменной панели в некотором смысле идеальные устройства. Имеются все важнейшие качества, вплоть до цветовых возможностей:

- большой угол наблюдения (до  $160^\circ$ ), так как свет излучается во всех направлениях,
- изображение может запоминаться, выборочно стираться и строиться снова,
- поточечная адресация позволяет использовать как векторные, так и растровые принципы построения изображения,

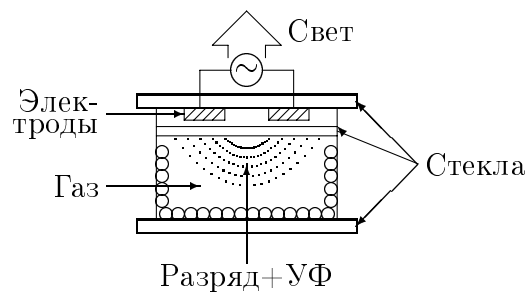


Рис. 0.10.5: Ячейка цветного дисплея с плазменной панелью

- панель плоская, поэтому дисплей может иметь много меньшие размеры, чем дисплей на электронно-лучевой трубке,
- картинка, независимо от ее сложности и структуры полностью лишена мерцания,
- информация от внешних источников изображений, например, слайдов или фильмов может проецироваться сквозь этот дисплей, обеспечивая таким способом простое смешение с картинками, сформированными компьютером.

К сожалению соотношение цена/возможности хуже, чем у дисплеев на электронно-лучевых трубках. Это в особенности касается разрешения, так как разрядные ячейки не могут быть сделаны особенно малыми, что обычно приводит к шагу пиксела порядка 1 мм. Предельное значение шага — до 25 ячеек на сантиметр. Это создает проблемы при использовании PDP в ТВВЧ и в качестве мониторов рабочих станций. В отличие от этого ЭЛТ, жидкокристаллические дисплеи и дисплеи с эмиссией полем могут иметь шаг пиксела всего 0.2 мм. Другие недостатки — относительно большое время включения/выключения — порядка 20 мкс/точку, относительно высокое напряжения питания — десятки вольт и эффективность, так 40-дюймовый PDP обычно потребляет приблизительно 300 Вт, в то время как пиковая яркость — только 1/3 таковой от ЭЛТ, потребляющей около 150 Вт.

В настоящее время дисплеи на плазменной панели в основном используются либо в особых условиях применения (высокие вибрации, низкие температуры) либо в качестве экранов коллективного просмотра. Лидерами в серийном производстве PDP являются фирмы Фуджицу и Хитачи параметры некоторых дисплеев этих фирм приведены в табл. 0.10.1.

### 0.10.3 Дисплей с жидкокристаллическим индикатором

Австрийский ботаник Friedrich Reinitzer открыл жидкие кристаллы в 1888 г. В 1963 г. Williams в фирме RCA исследовал поляризационные эффекты в жидких кристаллах. В 1973 г. был разработан первый дисплей на жидких кристаллах (EL 8025) для переносной ЭВМ.

Жидкие кристаллы находятся в некотором числе фаз, промежуточных между твердым и жидким состояниями. Молекулы ЖК являются стрежнеобразными органическими соединениями (рис. 0.10.6) и находятся в различных ориентациях в этих фазах (рис. 0.10.7).

В изотропической фазе, которая является жидкой фазой при повышении температуры, и позиция и ориентация молекул случайны.

Если температура понижается, то в ЖК совершаются переходы через различные фазы, одна из которых — так называемая нематическая фаза используется в дисплеях (twisted nematic

Таблица 0.10.1: Параметры дисплеев с плазменной панелью

Параметр	Хитачи		Фуджицу	
Диагональ	25"	41"	21"	42"
Разрешение	1024×768	1024×768	640×480	852×480
Шаг пикселей, мм		0.27 (гориз.) 0.81 (верт.)		
Оттенков Бит R×G×B		262 144 6×6×6	262 144 6×6×6	16.7млн 8×8×8
Контраст		300:1		
Яркость, кд/м <sup>2</sup>			180	300
Угол обзора, °	160°	160°	160°	160°
Стандарты	XGA, SVGA, VGA, видео	XGA	VGA, видео	видео и данные
Экран, мм	508×381	976×796	422×314	920×518
Толщина, мм	80	150	32	75
Вес	15 кг	37 кг	10.6 lbs	40 lbs
Электропитание		100 Вт		

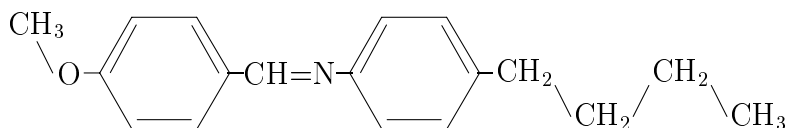


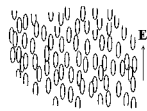
Рис. 0.10.6: Структурная формула жидкого кристалла

liquid crystal displays — дисплеи на закрученных нематических жидких кристаллах). В этой фазе позиции молекул все еще случайны, но все они ориентированы в одном направлении.

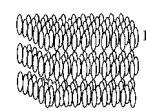
Если температура понижается далее, то молекулы получают периодическую упорядоченность в слоях (смектическая фаза). Таким образом при понижении температуры в ЖК увеличивается упорядоченность и в конце концов наступает твердое состояние.



Изотропическая  
фаза



Нематическая  
фаза



Смектическая  
фаза

Рис. 0.10.7: Фазы жидкого кристалла

Вследствие оптической и электрической анизотропии ЖК-молекул коэффициент преломления зависит от направления поляризации света относительно оси молекулы. Это свойство используется для поворота поляризации при прохождении света через закрученную ЖК-структуру.

Известно, что свет не проходит через два скрещенных поляризатора (рис. 0.10.8).

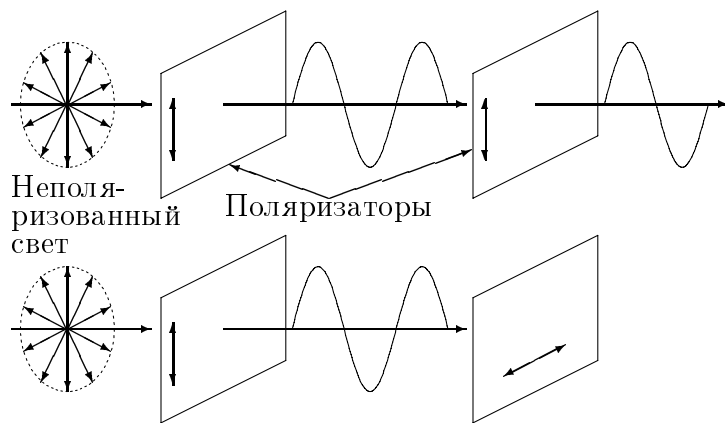


Рис. 0.10.8: Прохождение неполяризованного света через поляризаторы

Молекулы в нематической фазе принудительно закручиваются за счет их помещения между двумя стеклянными пластинами, каждая из которых имеет линейчатую гравировку, перпендикулярную к другой. На поверхности стекла молекулы вынужденно размещаются вдоль гравировки, а так как гравировки взаимно перпендикулярны, то между пластинами формируются перекрученные цепочки ЖК-молекул (рис. 0.10.9).

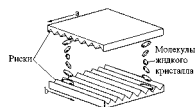


Рис. 0.10.9: Закрученный нематический жидкий кристалл

Расстояние между пластинами составляет порядка 10 мкм. В зависимости от расстояния между пластинами и типа ЖК-кристалла закрученность составляет обычно  $90^\circ$  или  $270^\circ$  (twisted nematics и super-twisted nematics, TN и STN).

При приложении электрического поля молекулы, вследствие их анизотропии, ориентируются вдоль поля. В этом случае цепочки раскручиваются и пропадает возможность поворота плоскости поляризации (рис. 0.10.10).

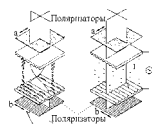


Рис. 0.10.10: Электронно-оптическое переключение жидким кристаллом



ЖК-дисплеи имеют два таких перекрещенных поляризатора с перекрученным жидким кристаллом между ними (рис. 0.10.11). Благодаря вращению плоскости поляризации ЖК-цепочками свет проходит и дисплей становится ярким. При приложении электрического поля к взаимноперпендикулярным прозрачным электродам, нанесенных на внутренние стороны пластин, эффект поворота плоскости поляризации пропадает и соответствующий пиксел становится темным.

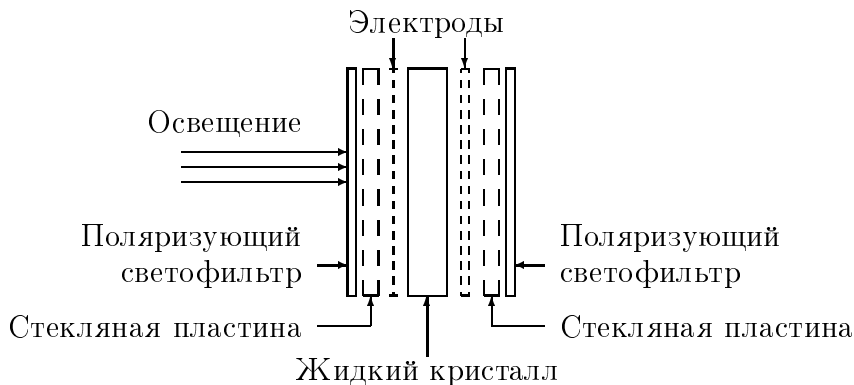


Рис. 0.10.11: Схема жидкокристаллического индикатора

При выключении напряжения кристалл за время порядка от первых десятков до сотен миллисекунд возвращается в исходное состояние.

Важной особенностью жидких кристаллов является то, что при протекании постоянного тока кристалл подвергается электролитической диссоциации и теряет свои свойства, поэтому жидкокристаллические индикаторы запитываются переменным напряжением, с постоянной составляющей не более десятков милливольт.

В простых индикаторах (с пассивной матрицей) ячейки раstra, составляющие изображение, запитываются последовательно. Для этого на проводники, пересекающиеся над нужной точкой, подают напряжение. В результате точка подсвечивается. Благодаря большому времени релаксации и достаточно высокой частоте сканирования ( $\approx 1$  мс на строку) изображение не мерцает. Естественно, что такие индикаторы медленны. Цветные ЖК-индикаторы используют три ячейки раstra для формирования пиксела. Яркость свечения для каждой из компонент определяет цветовой оттенок.

Для решения проблемы быстродействия были разработаны ЖК-дисплеи с активной матрицей, в которых каждый пиксел снабжается независимо управляемым тонкопленочным транзистором (thin-film transistor, TFT). Такие дисплеи значительно более быстродействующие, но и имеют большую стоимость, так как для цветного дисплея  $800 \times 600$  надо иметь 1 440 000 бездефектных транзисторов. Однако современные достижения в технологии позволили довести цену ЖК-дисплея с активной матрицей до \$600.

В целом ЖК-индикаторы имеют следующие основные характеристики:

- толщина  $\sim 1/6$  ЭЛТ,
- вес  $\sim 1/5$  ЭЛТ,
- энергопотребление  $< 1/4$  ЭЛТ,
- отсутствует мерцание,
- отсутствуют геометрические искажения,
- отсутствует паразитное излучение,

- цена  $\sim 3 \times$ ,
- небольшая контрастность изображения  $\sim 1:100$ ),
- небольшая яркость  $\sim 200 \text{cd/m}^2$ ,
- малый угол просмотра  $\sim 50^\circ$ ,
- небольшая скорость работы,
- ограниченный температурный диапазон работы.

Следует ожидать, что с развитием технологии недостатки будут устраняться.

#### 0.10.4 Дисплей с электролюминисцентным индикатором

В 1937 г. был открыт эффект электролюминесценции, заключающийся в излучении света легированного марганцем ZnS под воздействием электрического поля высокой напряженности ( $\sim 10^6/\text{см}$ ).

В 1981 г. началось практическое использование электролюминисцентных индикаторов. Они обеспечивают очень высокую контрастность и яркость при малых инерционности и толщине и высокой прочности. Срок их службы — более 120 000 часов, диапазон рабочих температур от  $-45^\circ\text{C}$  до  $+65^\circ\text{C}$ , угол обзора до  $160^\circ$ .

Панель электролюминисцентного индикатора состоит из стеклянной пластины, прозрачных проводников из окиси олова, чередующихся слоев изолятора и люминофора и темных проводников, перпендикулярных прозрачным (рис. 0.10.12).

Люминофор светится под воздействием переменного напряжения, прикладываемого к электродам строк и столбцов. Это переменное напряжение возбуждает свободные электроны в кристаллической структуре люминофора. Возбужденные свободные электроны, сталкиваясь с атомами примеси, переводят их электроны на более высокие энергетические уровни. При возврате в свое обычное состояние они испускают фотоны в видимом спектре.

Люминофор, применяемый в электролюминисцентных индикаторах, требует довольно высокого напряжения (170–210 В), что требует применения более дорогостоящих микросхем управления, чем для жидкокристаллических индикаторов.

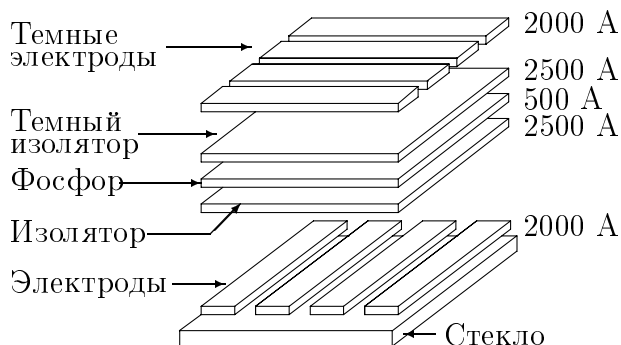


Рис. 0.10.12: Схема электролюминисцентного индикатора

### 0.10.5 Дисплеи с эмиссией полем

Дисплеи с эмиссией полем (FED) — плоские дисплеи, которые по принципу работы подобны обычным ЭЛТ. Но в отличие от обычной ЭЛТ, в которой имеется одна (три) электронных пушки, в данных дисплеях каждый пиксел имеет собственный, независимо адресуемый источник электронов. Электроны генерируются из холодных катодов, имеющих форму очень острых микроигл, которых на каждый пиксел может иметься до нескольких тысяч (рис. 0.10.13). Приложение электрического поля между прозрачным анодом экрана и катодом приводит к очень высоким напряженностям электрического поля из-за заостренности микроигл. Микроиглы изготавливаются из высокотемпературного материала, типа молибдена для того, чтобы при очень большой плотности тока, характерной для центров эмиссии, не возникало разбрызгивание металла и затупление игл. Электроны затем ускоряются и фокусируются на люминофор и свет испускается таким же образом как в ЭЛТ.

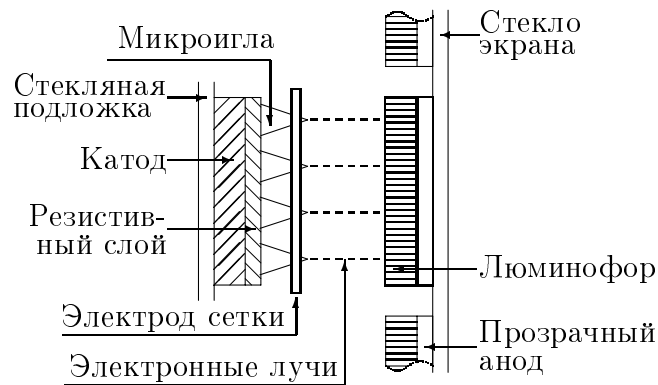


Рис. 0.10.13: Схема индикатора с эмиссией полем

Дисплеи с эмиссией полем представляются весьма перспективными из-за небольших габаритов, широкого угла наблюдения (почти  $180^\circ$ ), малого энергопотребления (несколько ватт для дисплея размером с записную книжку), хорошего воспроизведения цветов (люминофор ЭЛТ), высокой скорости работы (в принципе той же что и ЭЛТ).

Применение люминофоров ЭЛТ требует использования высокого ускоряющего напряжения, что заставляет электроны расходиться от прямой линии. При этом пикселы становятся менее точно определенными, т.е. изображение размывается. Для компенсации дефокусировки требуются дополнительные электроды, усложняющие конструкцию. Кроме этого, так как данный дисплей — вакуумное устройство, то наличие остаточных газов в сочетании с высоким напряжением вызывает разрушительную ионную бомбардировку микроигл катодов. По этим причинам в настоящее время среди разработчиков ведется дискуссия о целесообразности использования люминофоров ЭЛТ.

Дисплеи данного типа (в основном пятидюймовые одноцветные дисплеи для приборов) в настоящее время серийно в промышленных масштабах производит PixTech (консорциум компаний в Европе, США и Японии).

Компания Candescent Сан-Хосе, Калифорния объявила о планируемом выпуске таких дисплеев размером с записную книжку.

Однако успехи в крупносерийном производстве ЖК индикаторов с активной матрицей, по-видимому, не дадут возможности дисплеям с эмиссией полем получить широкий рынок.

## 0.11 Дисплейные подсистемы IBM PC

Дисплеи для IBM PC относятся к растровым дисплеям. Их специальное рассмотрение делается в связи с тем, что в настоящее время они наиболее распространены, хотя вряд ли могут служить примером изящных технических решений.

Дисплейная подсистема IBM PC состоит из двух основных компонент:

- видеоадаптера,
- монитора.

В свою очередь и видеоадаптеры и мониторы разделяются на несколько различных типов. Допускаются определенные (не произвольные) их комбинации.

### 0.11.1 Видеоадаптеры

Первоначально IBM PC выпускались с черно-белым адаптером MDA (Monochrome Display Adapter). Максимальное разрешение составляло  $640 \times 350$  точек, графические возможности отсутствовали. Следующим был также черно-белый видеоадаптер “Геркулес”, выпущенный фирмой Hercules Computer Technology, Inc. Этот адаптер обеспечивает разрешение  $720 \times 350$  с графическими возможностями.

Первым цветным видеоадаптером фирмы IBM стал CGA (Color Graphics Adapter). Разрешение его мало ( $320 \times 200$ ), цветов мало (до 4), программирование противоестественное. Максимальное разрешение составляет  $640 \times 200$ .

Затем был выпущен видеоадаптер EGA — (Enhanced Graphics Adapter). Он обеспечивает разрешение  $640 \times 200$  при 16 цветах из 64. Максимальное разрешение  $640 \times 350$ .

Первым видеоадаптером со сравнительно приемлемыми характеристиками стал VGA (Video Graphics Array) с максимальным разрешением до  $800 \times 600$  при 256 цветах.

Затем фирма IBM разработала видеоадаптер 8514/A, имевшего параметры более соответствующие сложившимся потребностям —  $1024 \times 768$  при 256 цветах.

Последняя разработка фирмы IBM — видеоадаптер XGA (eXtended Graphics Array) с не самыми современными возможностями  $1024 \times 768$  при 256 цветах. Он на уровне регистров совместим с VGA.

Многие фирмы выпускают улучшенные версии VGA под названиями Super VGA и Ultra VGA, но общий стандарт отсутствует.

Фирма Texas Instruments предложила стандарт на программный интерфейс с интеллектуальными видеоадаптерами, использующими графические процессоры TMS 340xx (TIGA-стандарт, Texas Instruments Graphics Architecture). В настоящее это самые мощные видеоадаптеры для IBM PC.

Сводка основных параметров упомянутых видеоадаптеров приведена в таблице 0.11.1.

### 0.11.2 Мониторы

Для отображения текста и графики используется несколько типов мониторов:

- композитный монитор, на вход которого подается композитный сигнал системы NTSC (National Television System Committee), применяемой в телевидении. Используется с видеоадаптером CGA;

Таблица 0.11.1: Характеристики видеоадаптеров IBM PC

Адаптер	Подключение	Разрешение	Развертки		Цветов
			Кадр., Гц	Строчн., кГц	
MDA	TTL	720×350	50	18.43	—
HGC	TTL	720×348	50	18.43	—
CGA	RGB/TTL	640×200	60	15.75	4
EGA	RGB/TTL	640×350	60	21.85	16
EGA+	RGB/TTL	640×480	60	30.50	16
VGA	RGB аналог	640×480	60/70	31.47	16
8514/A	—  —	1024×768	87 черес.	35.50	256
XGA	—  —	1024×768	72	37.50	256
SVGA-1	—  —	800×600	56	35.20	256
SVGA-2	—  —	800×600	60	37.80	256
SVGA-3	—  —	800×600	72	48.30	256
UVGA-1	—  —	1024×768	60	48.40	256
UVGA-2	—  —	1024×768	70	56.50	256
UVGA-3	—  —	1280×960	60	64.00	256
TIGA	—  —	1280×1024	60	64.00	64 млн

- цифровой монитор, на вход которого по  $N$  проводам подается сигнал в цифровой форме. Цифро-аналоговое преобразование выполняется монитором. На таком дисплее можно получить до  $2^N$  оттенков. Используется с видеоадаптерами CGA, EGA. Первоначально выпускался CD (Color Display) — простой 16-ти цветный (4 провода), затем ECD (улучшенный цветной дисплей) с 16-ю оттенками из 64, затем многочастотный цифровой, позволяющий работать с различными частотами кадров, и отображающий 16 оттенков из 64. Используется с видеоадаптерами EGA, VGA;
- аналоговый дисплей, на вход которого подаются готовые RGB-сигналы, сформированные видеоадаптером. Используется с видеоадаптерами VGA, SVGA, XGA.

### 0.11.3 Замечание об адресации в IBM PC

Это замечание предназначено для тех, кто не знаком с обозначением адресов для IBM PC, так как при рассмотрении видеоадаптеров будет применяться именно оно.

Микропроцессор, используемый в IBM PC, может адресовать до 1 Мбайта памяти ( $2^{20}$  байт). Вместе с тем 16-ю разрядами можно адресовать только 64 Кбайт ( $2^{16}$ ), поэтому используются два типа адресов: физический и логический. Физический адрес — 20-битное значение от 0h до FFFFh. Аппаратура при передаче информации между процессором и памятью использует физический адрес. Память разбита на логические сегменты, имеющими длину до 64 Кбайт (FFFFh байт). Каждому сегменту программно назначается базовый адрес, являющийся адресом первого байта в сегменте. Эти адреса кратны 16. Других ограничений на размещение сегментов нет. Логический адрес байта состоит из сегментного значения и значения относительного адреса в сегменте. Записывается логический адрес в виде: BBBB:OOOO. Где BBBB — базовый адрес сегмента, деленный на 16, а OOOO — относительный адрес в сегменте. Физический адрес генерируется сдвигом влево на 4 разряда базового адреса и сложением его со значением относительного адреса.

### 0.11.4 Архитектура видеоадаптеров

Все видеоадаптеры содержат в себе видеопамять и видеоконтроллер (см. рис. 0.9.1). Видеопамять доступна процессору как обычная оперативная память и, в зависимости от типа видеоадаптера, занимает или часть или всю память, отводимую в адресном пространстве IBM PC под экранный буфер (128 Кбайт от адреса A000:0000 до B000:FFFF).

Предусмотрено две группы взаимоисключающих режимов работы видеоадаптеров — символьные и графические.

В символьных режимах видеопамять интерпретируется как последовательность байт, содержащая байты коды отображаемых знаков и байты атрибутов, задающих цвета.

В графических режимах видеопамять интерпретируется самым различным образом, в основном, как некоторым образом упорядоченная последовательность битовых значений. Начало экрана — точка (0,0) расположена в левом верхнем углу. X-координата нарастает слева-направо, а Y-координата — сверху-вниз.

### 0.11.5 Видеоадаптер CGA

Содержит 16 Кбайт видеопамати, начинающейся с адреса B800:0000, и контроллер электронно-лучевой трубки.

В символьных режимах отображается 25 строк по 40 или 80 символов. Для кодирования каждого знакоместа экрана используется 2 байта. 1-й байт — код отображаемого знака, второй байт — байт атрибута, задающий цвета символа и фона, мерцание. Размер знакоместа  $8 \times 8$ , таблица знакогенератора “защита” в ПЗУ и, следовательно, не может быть изменена.

В графических режимах видеопамять интерпретируется или как последовательность двух-битовых значений при разрешении  $320 \times 200$  с четырьмя цветами, или как последовательность однобитовых значений при разрешении  $640 \times 200$  и черно-белом изображении. Видеопамять делится на две части: в первой половине, начинающейся с адреса  $B800:0000$ , содержатся описания всех нечетных строк, во второй, начинающейся с адреса  $B800:2000$ . Левому верхнему углу экрана отвечают два старших бита начального байта видеопамети при разрешении  $320 \times 200$  и старший бит — при разрешении  $640 \times 200$ . Структура видеопамети CGA проиллюстрирована на рис. 0.11.1.

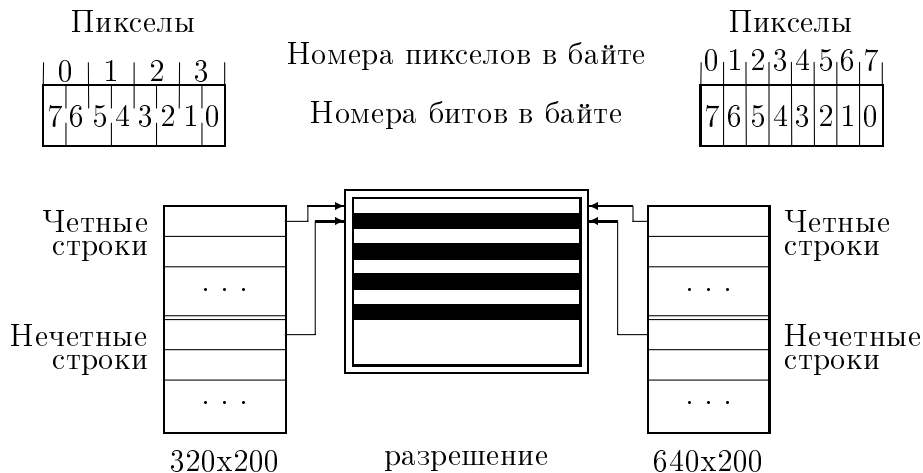


Рис. 0.11.1: Структура видеопамети для CGA

### 0.11.6 Видеоадаптеры EGA/VGA

Видеоадаптеры EGA и VGA — более совершенные устройства, более напоминающие уже рассмотренные нами растровые дисплеи. На рис. 0.11.2 показана блок-схема, общая для этих двух адаптеров, на рис. 0.11.3 — блок-схема, демонстрирующая отличия для VGA, связанные с использованием не цифрового, а аналогового дисплея.

Графический контроллер обеспечивает обмен между центральным процессором и видеопамятью, которая так же как и в CGA находится в адресном пространстве для экранного буфера.

Видеопамять имеет объем, как правило 256 Кбайт, но для EGA может быть и 64 Кбайт. Видеопамять разбита на 4 банка (цветовых слоя). Банки занимают одно адресное пространство таким образом, что по каждому адресу расположено сразу 4 байта по одному байту в каждом слое. Можно запрещать или разрешать запись в отдельные слои памяти при помощи регистра разрешения записи цветового слоя. Для операции чтения в каждый данный момент времени может быть доступен только один слой, задаваемый с помощью регистра выбора читаемого слоя. В большинстве режимов работы видеопамять разделена несколько страниц. При этом одна из них активна и отображается на экране. Запись информации возможна как на активную, так и на неактивные страницы.

Преобразователь последовательности выбирает данные из памяти, требуемым для того или иного режима образом, и формирует последовательный поток бит, передаваемый контроллеру атрибутов.

Контроллер атрибутов, используя таблицу цветности, преобразует информацию о цветах, полученную из видеопамати, в информацию для ЭЛТ.

Контроллер ЭЛТ генерирует синхриимпульсы управления ЭЛТ.

Тактовый генератор управляет временными параметрами видеоадаптера и доступом от процессора в слоям видеопамати.

Показанный на рис. 0.11.3 ЦАП преобразует цифровые сигналы яркостей в аналоговые значения, необходимые аналоговому монитору.

Большое количество различных регистров у адаптеров позволяет гибко регулировать режимы работы, управлять сигналами синхронизации, определять формат экрана и т.д.

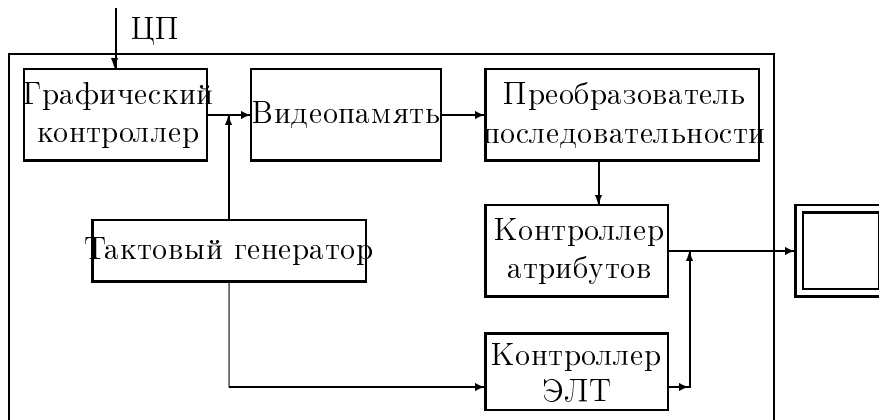


Рис. 0.11.2: Блок-схема видеоадаптеров EGA/VGA

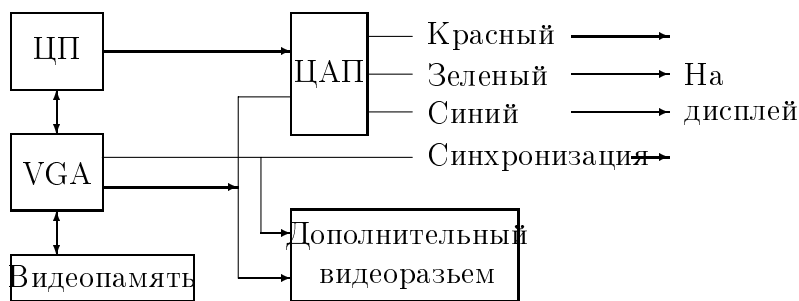


Рис. 0.11.3: Блок-схема видеоадаптера VGA

### Текстовый режим

В текстовом режиме можно вывести на экран 25 строк по 40 или 80 символов. Перепрограммировав некоторые регистры адаптера можно получить для EGA до 43 строк, а для VGA — до 50.



Также как для CGA для кодирования символа используется два байта. Первый из них содержит код символа и заносится в нулевой цветовой слой, второй байт содержит атрибут символа и заносится во второй цветовой слой.

Таблицы знакогенератора, задающие внешний вид символов, размещаются во втором слое видеопамяти. Всего может быть загружено 4 таблицы для EGA и 8 — для VGA. Одновременно могут быть активными две таблицы, что дает возможность отобразить на экране до 512 различных конфигураций знакомест одновременно. Возможные размеры символов приведены в таблице 0.11.1.

Байт атрибута задает цвета символа и фона, мерцание и одну из двух активных таблиц знакогенератора.

### Графические режимы

Имеется большой набор графических режимов. В режимах, совместимых с CGA используется только 0-й слой памяти, распределение которого по адресному пространству и интерпретация значений бит такая же как и для CGA (см. рис. 0.11.1).

В части графических режимов память интерпретируется как послойная с использованием одного, двух или всех четырех слоев. Если используется, например, 4 слоя, то изображение будет с 4-мя битами на пиксел — по биту из каждого слоя. В этом случае имеется 16 оттенков одновременно. При использовании одного слоя, естественно, получаем черно-белое изображение. В режиме с двумя слоями поддерживается три уровня черно-белого (черный, белый, ярко-белый) и мерцание.

В одном из режимов, применимом только для VGA, память интерпретируется как линейная последовательность всех четырех слоев (см. рис. 0.11.4). Это режим с разрешением  $320 \times 200$  с байтом на пиксел, т.е. с 256 градациями.

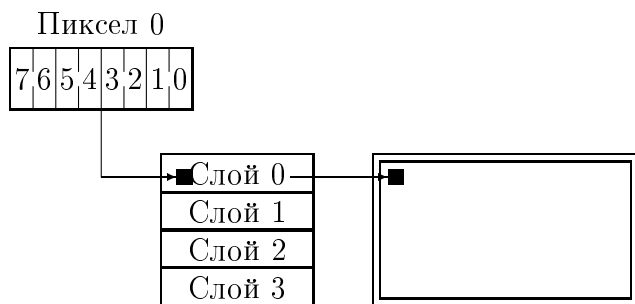


Рис. 0.11.4: Структура видеопамяти для VGA-режима  $320 \times 200 \times 256$

### Графические акселераторы

Для повышения быстродействия графических подсистем IBM PC выпускаются специальные типы адаптеров — графические акселераторы. Графические акселераторы содержат собственные процессоры, которые специализированы для выполнения графических преобразований, поэтому изображения обрабатываются быстрее, чем с использованием универсального ЦП ПЭВМ.

Акселераторы, кроме типа и возможностей графического процессора, различаются по следующим основным параметрам:

- памятью для сохранения изображений. В некоторых случаях используется обычная динамическая память DRAM, но обычно используется специализированная видеопамять VRAM;
- используемой шиной. В настоящее время обычно используется PCI;
- шириной регистров. Чем шире регистр, тем большее число пикселей можно обработать за одну команду. В настоящее время обычная ширина — 64 бита. В ближайшее время на рынке появятся акселераторы с шириной регистров в 128 бит.

## 0.12 Устройства ввода

### 0.12.1 Клавиатуры (Keyboards)

Естественным и единственным устройством ввода текста является клавиатура. Для обнаружения нажатия клавиши используется несколько различных способов: механическое замыкание контактов, изменение емкости, изменение магнитного поля, прерывание луча света и т.д. Важными свойствами клавиатуры являются кодировка, используемая для идентификации нажатия клавиши — ASCII (American Standard Code for Information Interchange), КОИ-7, КОИ-8 (ГОСТ), МПС (Болгария) и т.д., количество клавиш редактирования текста, возможности расширения кодировки за счет нажатия дополнительных клавиш — верхнего и нижнего регистров (Shift), клавиш для задания управляющих символов и кодов (Ctrl, Alt). Существенными являются и эргономические свойства — размеры и расположение клавиш, наличие тактильной обратной связи при нажатии и ощущение контакта при полностью нажатой клавише.

### 0.12.2 Кнопки (Buttons)

Программируемые функциональные кнопки обычно служат для задания какого-либо определенного воздействия, связанного с нажатием, отпусканием или удержанием кнопки (ввод команды, выбор параметра). Конструктивно кнопки устроены аналогично кнопкам текстовой клавиатуры. Чаще всего они объединены с текстовой клавиатурой, но иногда выделены в виде отдельного блока. Другой разновидностью функциональной клавиатуры являются кнопки на зондах планшетов или на “мышке”.

Основное отличие кнопочного устройства от текстового состоит в том, что клавиш на клавиатуре всегда ограниченное количество и с ними жестко связаны определенные метки (коды клавиш). Клавиши же кнопочного устройства не имеют заранее определенного значения и их число может меняться от одной до нескольких десятков.

По сути дела текстовая клавиатура просто разновидность кнопочной, но так как она нашла большое распространение, то обычно выделяется в отдельный класс устройств.

Во многих компьютерных системах за счет написания собственного драйвера можно превратить текстовую клавиатуру в программируемую функциональную и наоборот.

### 0.12.3 Световое перо (Lightpen)

Световое перо (рис. 0.12.1) служит для непосредственного указания элементов изображения на экране. При аналоговой генерации векторов (символов) можно идентифицировать конкретный вектор (символ). При цифровой генерации векторов можно идентифицировать “рассматриваемую” точку изображения. Из-за нечеткой оптики светового пера и трудностей позиционирования в точку на экране эта возможность практически не нужна. Чаще световое перо используется для указания всего элемента изображения. В каждом случае используется временное соответствие светового сигнала и сиюминутного состояния обработки команд построения. На векторном дисплее таким образом устанавливается однозначное соответствие между текущей командой, обрабатываемой дисплейным генератором, и элементом изображения, от которого пришел световой импульс.

Для идентификации некоторого сложного элемента картинки как целого, например, изображения микросхемы, используется техника “идентификаторов указания”. Идентификатор указания представляет собой неграфический, комментирующий элемент дисплейного файла. При аппаратной обработке идентификаторов указания дисплейный процессор запоминает их последовательность, в общем случае иерархическую. Для идентификации объекта при поступлении импульса от светового пера достаточно просмотреть запомненные идентификаторы указания.

Пример дисплейного файла с двумя указуемыми объектами

Команда	Смысл	Стек pīcid
. . . . .	любые дислейные команды	пусто
pīcid 001	идентификатор указания 001	001
. . . . .	графические команды	001
. . . . .	построения элемента 001	001
end_pīcid	конец указуемой группы	пусто
. . . . .	любые дислейные команды	пусто
pīcid 002	идентификатор указания 002	002
. . . . .	графические команды	002
. . . . .	построения элемента 002	002
end_pīcid	конец указуемой группы	пусто
. . . . .	любые дислейные команды	пусто
endpīc	конец дисплейного файла	пусто

В данном примере имеется два не вложенных указуемых элемента. До отработки команды pīcid 001 стек идентификаторов указания пуст. После ее отработки и до первой команды end\_pīcid в стеке будет находиться 001. После отработки end\_pīcid стек идентификаторов указания очищается и до команды pīcid 002 остается пустым.

В растровых дисплеях отклонение луча не зависит от обрабатываемого изображения. В этом случае информация о указании дает позицию на экране. Для идентификации элемента картинки с помощью светового пера в растровых дисплеях требуется “прокрутка в уме” всего видимого изображения для определения принадлежности точки некоторому объекту (объектам). При этом также может использоваться программно реализованная техника идентификаторов указания. Для ускорения интерпретации изображения часто используются так называемые графические оболочки — обычно прямоугольники, иногда окружности, объемлющие элементы изображения. Если точка, указанная световым пером, находится вне графической оболочки, то интерпретацию объекта можно пропустить.

В качестве эха светового пера обычно используют сверхподсветку выделенного элемента изображения. В векторных дисплеях сверхподсветка примитивного элемента (вектор, символ) реализуется аппаратно.

В целом это устройство применяется все реже не только из-за того, что оно более применимо на векторных дисплеях, но, в основном, из-за того что длительная работа с ним весьма утомительна — рука всегда “на весу”.

#### 0.12.4 Планшеты (Tablets)

Планшеты являются устройствами ввода с непосредственным заданием координат (локаторы). Это одно из важнейших устройств ввода. Пользователь может вводить информацию в компьютер привычным образом, как при использовании карандаша и бумаги.

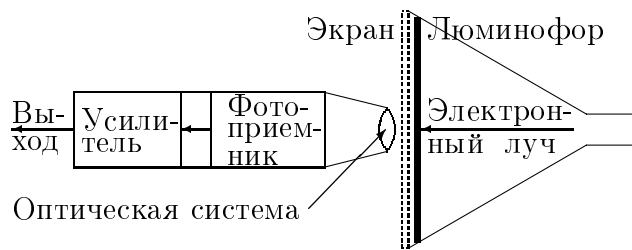


Рис. 0.12.1: Схема работы светового пера. 1 — усилитель, 2 — фотоприемник, 3 — оптическая система

Позиции задаются перемещением зонда планшета (визира или карандаша) по рабочей поверхности. Координата текущего положения зонда определяется с частотой от 200 до 500 раз в секунду. Этим обеспечивается что даже при быстрых перемещениях зонда вдоль какой-нибудь кривой она будет вводиться достаточно гладко. Из-за большой частоты опроса генерируется много данных, поэтому в большинстве случаев они подвергаются дальнейшей обработке для сокращения объема. Обычно используемый способ — выдача новой координатной пары при достижении заданного отклонения от последней зафиксированной.

В общем планшеты работают в различных режимах:

- точечном, когда генерируется координата при нажатии кнопки зонда;
- непрерывном, когда последовательность координат генерируется непрерывно при нахождении зонда в рабочей области планшета (при этом может производиться сокращение объема передаваемых данных так, как это описано выше);
- переключаемом непрерывном, когда генерируется непрерывная последовательность координат при нажатии кнопки зонда;
- приращений, когда формируются приращения к последней выданной позиции.

Имеется много различных способов определения координат зонда на планшете. Далее будут рассмотрены основные из них.

### Потенциометрический (градиентный) планшет

Поверхность планшета представляет собой резистивное покрытие (рис. 0.12.2). По границам к этому покрытию подводится ток попеременно по X и Y направлениям. Зонд планшета имеет гальванический контакт с резистивным покрытием. Координата определяется по падению напряжения в точке контакта. Для устранения искажений поля, вызываемого взаимным влиянием контактных шин, питающие напряжения подводят через диоды. Основные проблемы при разработке таких планшетов:

- материал покрытия должен быть достаточно высокоомным, чтобы обеспечить простоту и точность измерения;
- материал покрытия должен обладать однородной проводимостью, чтобы обеспечить линейность ввода координат;
- материал покрытия должен быть достаточно прочным, чтобы выдержать постоянный контакт с наконечником планшета.

Предлагаются и иные конструкции градиентных планшетов, не требующие гальванического контакта зонда с поверхностью. Например, планшет в котором резистивный слой из окиси олова находится между двух стеклянных пластин. Высокочастотное питание подводится через

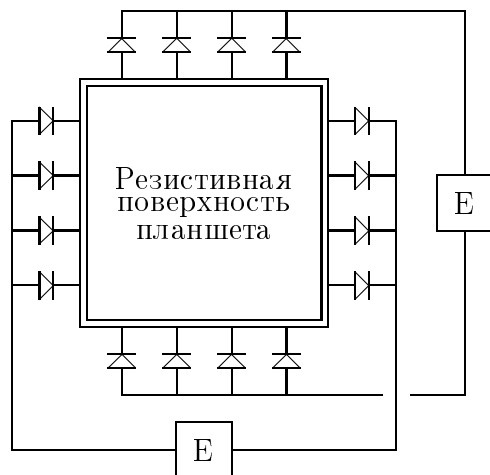


Рис. 0.12.2: Устройство потенциметрического планшета

контактные шины по краям планшета. Используются различные частоты для X и Y напряжений питания. Зонд планшета воспринимает фазы сигналов, различные для различных точек на планшете. Так как используется высокочастотный сигнал, то возможно значительное удаление зонда от поверхности — более 10 мм. Измерение амплитуды сигнала позволяет получить информацию о высоте подъема зонда.

### Акустический планшет

В таком планшете разрядник в зонде излучает ультразвуковой сигнал, который принимается ленточными микрофонами, расположенными на двух смежных сторонах планшета (рис. 0.12.3а). Акустические планшеты с тремя группами микрофонов (рис. 0.12.3б) могут выдавать трехмерную координатную информацию.

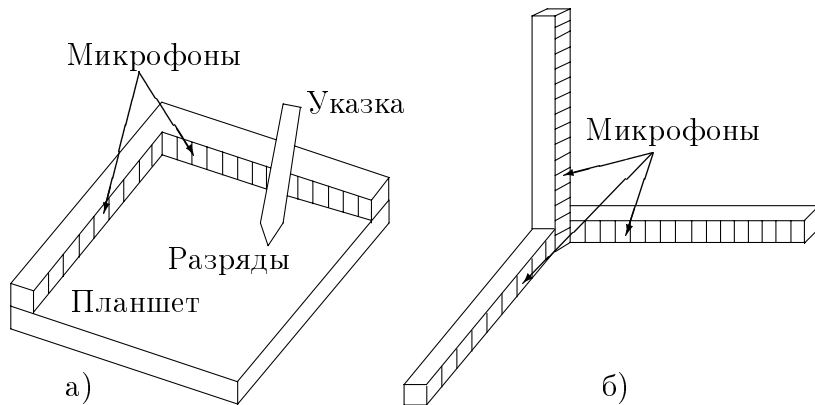


Рис. 0.12.3: Устройство акустического планшета

По времени прихода звука к микрофонам определяется точное положение зонда. В связи с большой скоростью распространения звука в воздухе  $\approx 330$  м/с эти измерения не представля-

ют трудностей. Акустический планшет позволяет определять координаты и при значительном удалении от поверхности планшета, но с ошибкой, как это пояснено на рис. 0.12.4а).

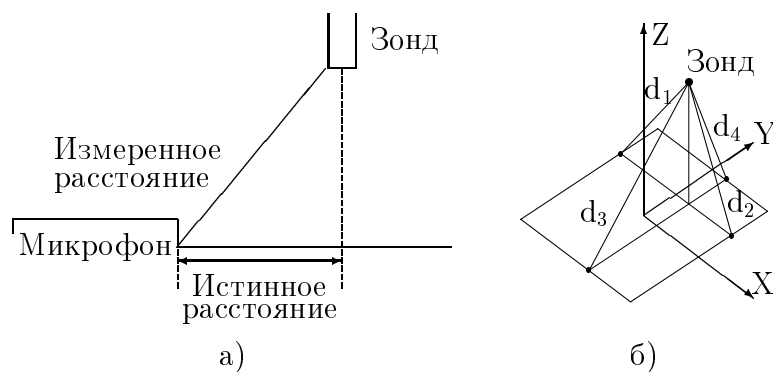


Рис. 0.12.4: Причина ошибки в акустическом планшете

Это свойство планшета позволяет определять 3D координаты с помощью плоского планшета, имеющего по микрофону на каждой стороне. При стороне планшета =  $2a$  и измерении координат  $x$  и  $y$  относительно центра планшета из рис. 0.12.4б) можно написать следующие выражения для  $z$ :

$$\begin{aligned} z^2 &= d_1^2 - (x + a)^2, \\ &= d_2^2 - (x - a)^2, \\ &= d_3^2 - (y + a)^2, \\ &= d_4^2 - (y - a)^2. \end{aligned}$$

Из этого легко находятся:

$$\begin{aligned} x &= (d_1^2 - d_2^2) / 4a; \\ y &= (d_3^2 - d_4^2) / 4a; \\ 4z^2 &= (d_1^2 + d_2^2 + d_3^2 + d_4^2) + 2x^2 + 2y^2 + 4a^2. \end{aligned}$$

Основным недостатком акустического планшета является сильная зависимость от окружающей среды, особенно влажности.

### Емкостные планшеты

В таких планшетах под непроводящей рабочей поверхностью генерируется электромагнитное поле с помощью взаимно перпендикулярных групп проводников (рис. 0.12.5). Проводники в каждой группе должны быть точно параллельны и находиться на одинаковых расстояниях друг от друга. Эти проводники служат передающими антеннами. На передающие антенны поочередно подается высокочастотное напряжение. Сигнал принимается емкостным датчиком зонда. Пусть датчик находится между двумя проводниками, тогда вначале будет получен сигнал от одной антенны, затем от другой. По соотношению амплитуд сигналов можно узнать точное расположение между антеннами.

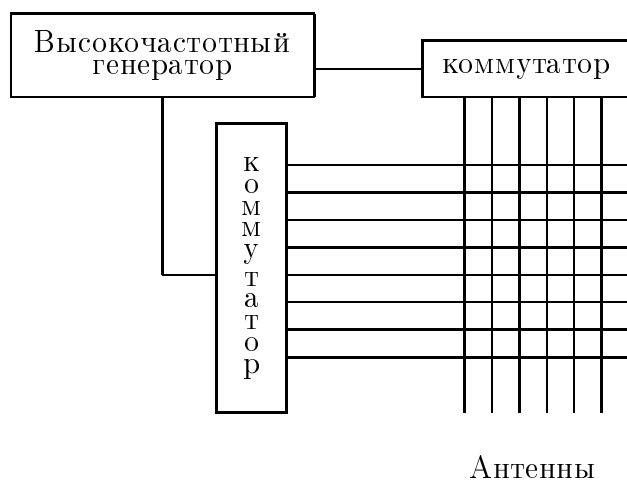


Рис. 0.12.5: Устройство планшета с емкостным съемом информации

Недостатком этого планшета является то, что зонд должен находиться либо непосредственно на рабочей поверхности, либо на малых расстояниях от нее (на толщину нескольких листов бумаги).

Примером такого устройства может служить планшет, разработанный в ИЯФ'е и выпускавшийся рядом организаций. Рабочее поле планшета  $380 \times 380$  мм, проводники расположены на двухсторонней печатной плате с довольно большим шагом — 5мм. Частота высокочастотного генератора — 625 КГц. Время коммутации 200 мкс. Датчиком служит незамкнутая круглая проволочная петля. Встроенный в планшет микропроцессор за 7 мс рассчитывает и выдает координату с довольно высокой точностью — 0.1 мм.

### Магнитоэлектрические планшеты

В таких планшетах катушка в зонде и проводники под рабочей поверхностью планшета могут рассматриваться как первичная и вторичная обмотки трансформатора. Если приемная катушка находится на зонде, то конструктивно этот планшет подобен емкостному планшету. Существенно большее разрешение достигается при использовании обмотки зонда как передатчика, но в этом случае катушка зонда должна иметь много витков, чтобы сгенерировать достаточно мощное поле. Большинство далее упоминаемых кодировщиков, используемых для ввода чертежей, работает на этом принципе.

### Магнестрикционные планшеты

Магнестрикционные планшеты используют магнестрикционные проволоки как носители сигнала, которые под воздействием внешнего магнитного поля незначительно изменяют свою форму (рис. 0.12.6). Магнитное поле, вызываемое передающими катушками на краю планшета и перпендикулярное магнестрикционным проволокам, генерирует изменение их длин. Это изменение длины распространяется вдоль проволоки как волна механического напряжения со скоростью около 5000 м/с. Волна, попадая в приемную катушку, расположенную в зонде планшета, из-за изменения потока формирует в катушке импульс напряжения. Время прихода



волны пропорционально расстоянию от передающей катушки на краю планшета до зонда. Так как расстояние всегда измеряется вдоль проволоки, то не требуется, чтобы проволоки были абсолютно параллельны. Не требуется располагать проволоки так часто, как это следовало бы из разрешения прибора. Более того достаточно их расположить на расстоянии в 2–3 мм все равно при этом на планшете гарантируется достаточное изменение потока. Этот принцип имеет относительно высокую точность (0.01 мм), широко используется в робототехнике и в большинстве планшетов.

Bit Pad One фирмы Summagraphics работает на магнитострикционном принципе. Размер его рабочей области 300×300 мм. На подложке под рабочей поверхностью планшета с шагом ≈ 3 мм расположены по 96 проволок для каждой из осей. Разрешение по координате составляет 0.1 мм.

Стандартом на рынке для РС является планшет Bit Pad Two, имеющий разрешение в 0.05 мм и точность в 0.5 мм.

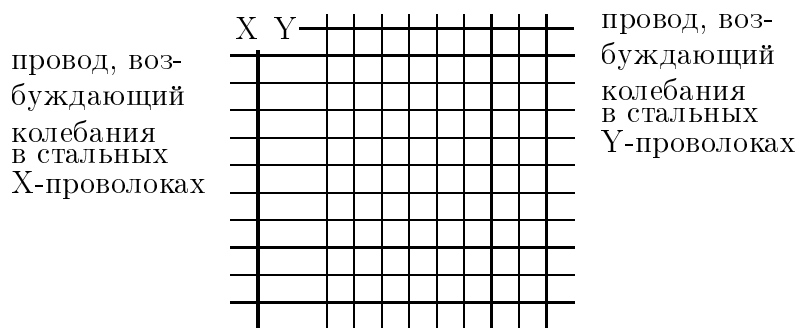


Рис. 0.12.6: Устройство магнитострикционного планшета

### 0.12.5 “Мышь” (Mouse), трекбол (Treckball), джойстик (Joystick)

Не прямой ввод графической информации, как правило, основан на перемещении пользователем по экрану графического курсора.

Имеется три основных метода позиционирования курсора: статический абсолютный, статический относительный и динамический. В первом случае устройство ввода выдает координаты точек, которые определяются текущей позицией устройства ввода. Во втором случае используются приращения координат, соответствующие изменению позиции устройства ввода. В третьем случае выдается последовательность координат для перемещения курсора, направление и скорость перемещения которого задаются установкой устройства ввода, например, направление перемещения задается направлением отклонения рычага, а скорость перемещения — величиной угла отклонения.

#### “Мышь”

Под “мышкой” понимается небольшое ручное устройство ввода, выдающее приращения координат при перемещении “мышки” по рабочей поверхности (по рабочему столу для механических “мышек” и по специальной пластине для оптических “мышек”). При перемещении механической

“мышки” по столу движение передается одним (двумя) шарами на пару потенциометров или датчиков угла поворота, соответствующих перемещениям по двум взаимно-перпендикулярным направлениям. Обычно в качестве датчика угла используется диск с нанесенными непрозрачными радиальными полосками (рис. 0.12.7). Комплект из двух пар светодиод-фотодиод позволяет определить направление поворота. Вследствие произвольного и не слишком точного перемещения “мышки” наилучшим методом ее использования является статический относительный.

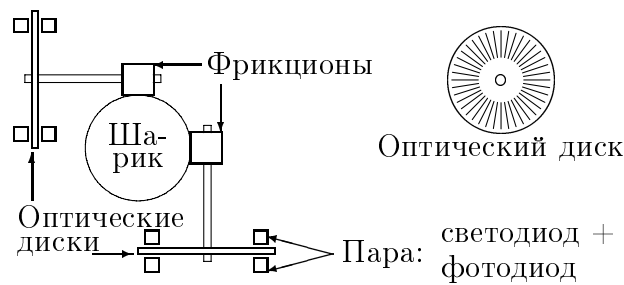


Рис. 0.12.7: Устройство механической мышки

### Трекбол

Представляет собой перевернутую “мышку” с одним большим шаром, приводимым в действие рукой. Для обеспечения медленных перемещений масса шара должна быть сравнительно большой. Это затрудняет большие и быстрые перемещения. Для облегчения задания больших перемещений используют различные сложные конструкции вплоть до подвески шара на воздушной подушке. Так как имеется хорошая тактильная обратная связь, то для управления курсором может использоваться не только координата, но и угол поворота шара, кроме того, так как шар достаточно тяжелый, то можно использовать и величину начального импульса шара. Т.е. трекбол, в отличие от “мышки”, применим для всех трех способов позиционирования курсора — статического абсолютного, статического относительного и динамического. В практическом использовании трекбол заменяется “мышкой”.

### Джойстик

Джойстик представляет собой вертикально стоящий рычаг, который на нижнем конце установлен в кардане и удерживается пальцами в среднем — начальном состоянии. Джойстик является идеальным “рычагом управления курсором”, так как он может отклоняться требуемым образом одновременно по двум осям. Основная область применений — динамический метод позиционирования, однако он применим и для статического абсолютного очень быстрого, но неточного позиционирования. Перемещения джойстика передаются на два потенциометра, соответствующих  $X$  и  $Y$  направлениям и выдающих напряжения для каждой их координат. Очень простая и дешевая конструкция джойстика использует жестко закрепленный жезл, к которому прикреплены датчики растяжения, например, пьезоэлектрические.

## 0.12.6 Потенциометр

Потенциометр с аналого-цифровым преобразователем обеспечивает быстрый ввод точных числовых значений с контролем значения выводом на дисплей. Используются потенциомет-

ры вращения и ползунковые. Для задания значений углов наиболее подходят потенциометры вращения.

## 0.12.7 Растровый сканер

Сканеры используются для растрового ввода изображений с последующей их обработкой и/или документированием.

Одна из важных областей применения сканеров — ввод текстов. При этом обработка введенного изображения выполняется с программного обеспечения распознавания текстов (Optical Character Recognition — OCR).

В САПР сканеры используются для автоматизации ввода ранее подготовленной конструкторской документации. В этом случае проблема заключается в том, что данные от сканера представлены в растровой, а не векторной форме и требуется выполнение обратного преобразования растр-вектор. Эта задача сложна и далека от решения (необходимо распознавать различные изображения и тексты в том числе рукописные, учитывать, что изображение представляется поточно, причем одна и та же линия может получить при сканировании не только различную ширину, но и дырки и т.д.).

Сканеры работают в один или три прохода, большинство сканеров работает в один проход.

Простейшие сканеры — ручные с шириной сканирования до 127 мм (5 дюймов). Например, сканер швейцарской фирмы Logitech International S.A. Ширина сканирования 106 мм, разрешение 100, 200, 300 или 400 точек на дюйм. Встроенный источник подсветки оригинала — красный или желто-зеленый.

Более точные сканеры — стационарные. Имеется три основных варианта стационарных сканеров:

1. Оригинал перемещается относительно неподвижной линейки фотоприемников (сканеры с полистовой подачей и барабанные сканеры для больших форматов).
2. Линейка фотоприемников перемещается относительно оригинала (планшетные сканеры).
3. Проекционные сканеры, в которых изображение неподвижного оригинала проецируется на матрицу фотоприемников, установленных в фокусе объектива. Объектив же перемещается для выбора нужного фрагмента с нужным увеличением.

Цвет подсветки в стационарных сканерах обычно белый. Сканирование цветных изображений обеспечивается сменой светофильтров. В некоторых сканерах смена производится вручную.

Аппаратное разрешение стационарных сканеров от  $300 \times 300$  точек на дюйм (dot per inch — dpi) до  $1200 \times 1200$  dpi. С использованием интерполяции разрешение достигает от  $4800 \times 4800$  dpi до  $10000 \times 10000$  dpi.

Поддержка цветов в стационарных сканерах — либо серая шкала, либо 24 бита/пиксел (> 16 миллионов), либо 30 бит/пиксел (> 1 миллиарда) и до 36 бит/пиксел (> 68 миллиардов).

Выходные форматы растровых файлов — TIF, GIF, BMP, PCX и т.д. Интерфейс для подключения к ПЭВМ — либо параллельный порт, либо SCSI. Обычно сканеры имеют встроенный буфер для сохранения изображения объемом от 32 К до 2 М. Сканеры с полистовой подачей обеспечивают ввод и сканирование от 3 до 8 листов в минуту (при одном проходе).

### 0.12.8 Кодировщик

Кодировщик (по ГОСТ'у — графоповторитель) — устройство автоматического и/или полуавтоматического ввода. В автоматических кодировщиках выполняется сканирование вводимого документа (чертежа, графика, карты), определение и считывание элементов документа.

В более распространенных полуавтоматических кодировщиках оператор вручную перемещает рабочий орган по поверхности, распознает элементы документа (линии, отметки, знаки и т.п.) и идентифицирует их с помощью клавиатуры. Небольшая функциональная клавиатура обычно размещается непосредственно на рабочем органе, а дополнительные — на рабочей поверхности. Определение координат происходит автоматически по положению рабочего органа. Для повышения точности ввода координат рабочие органы (визеры) изготавливаются так, что допускают прецизионную установку в заданную точку. Если координаты расположены в узлах некоторой сетки, то точность обеспечивается округлением до ближайшего узла сетки.

Обычно в полуавтоматических кодировщиках используется два основных режима работы:

- дискретный, когда оператор устанавливает зонд в требуемую точку и выдает команду определения координаты;
- непрерывный, когда оператор перемещает зонд вдоль некоторой линии, а устройство автоматически генерирует последовательность координат. Координаты генерируются либо через заданные интервалы времени, либо по удовлетворению некоторого критерия, например, при достижении заданной разности координат текущей и последней зафиксированной точки.

По сути дела полуавтоматический кодировщик — просто большой планшет с достаточными размерами и разрешением, оснащенный так или иначе выполненной клавиатурой (в простейших случаях в качестве клавиатуры используется часть рабочей поверхности).

Автоматические кодировщики — комбинация планшетного графопостроителя и планшета. Известны и реализации, использующие сканер с последующей программной в той или иной мере интерактивной обработкой документа. Основные проблемы в этом случае — преобразование растр — вектор и недостаточные форматы вводимых документов, что требует решения задачи “сшивки” отдельно введенных фрагментов.

## 0.13 Нетрадиционные устройства

Имеется много разновидностей нетрадиционных устройств, например, “аккордная” клавиатура, представляющая собой комплект программируемых кнопок в виде тонких рояльных клавиш. Основное свойство такой клавиатуры — возможность одновременного нажатия сразу нескольких клавиш. Если клавиш 5 шт. (как и пальцев на руке), то можно получить 32 различных комбинации.

Более распространенным устройством указания является сенсорная панель, позволяющая рукой указывать объекты на экране.

Прогресс в полупроводниковой индустрии сделал реально применимыми и устройства речевого диалога.

С точки зрения средств вывода в системах виртуальной реальности используются по сути дела обычные растровые дисплеи в сочетании с механизмами отслеживания положения головы и направления взгляда для управления формированием сцены.

Кроме уже рассмотренных обычных устройств ввода в системах виртуальной реальности используются устройства ввода с тремя и шестью степенями свободы (перемещения вдоль трех осей и вращения вокруг них), а также устройства для обеспечения тактильной и силовой обратной связи, имитирующие ощущения прикосновения и “податливость” в ответ на воздействие.

### 0.13.1 Сенсорная панель (Touch Screen)

Сенсорная панель — устройство ввода невысокой точности, используемое, в основном, для ручного (пальцем) указания объектов на экране дисплея. Простейшая сенсорная панель представляет собой рамку, размещаемую между экраном, на который выдаются объекты указания, и пользователем. На двух сторонах рамки размещается некоторое количество светодиодов, а на противоположных к ним — фотоприемники. Координаты указания определяются по перекрытию лучей от светодиодов.

Более сложные сенсорные панели используют прозрачную (стеклянную) поверхность, покрытую прозрачным проводящим слоем окиси олова. Факт указания определяется по изменению сопротивления.

Известны сенсорные панели и с другими принципами определения координаты прикосновения пальца, например, с использованием поверхностных акустических волн.

### 0.13.2 Речевой диалог

В общем случае система речевого диалога осуществляет распознавание, смысловую интерпретацию и синтез речевых сигналов и обеспечивает устный диалог человека с ЭВМ либо на формализованных, либо на усеченных естественных языках. В такого сорта системах можно выделить две основные компоненты — средства распознавания речи и средства синтеза речи.

Для надежного распознавания как отдельно произносимых слов так и слитной речи в реальном масштабе времени требуется производительность порядка 10 млн оп/с.

Появились коммерческие системы речевого ввода, распознающие порядка 1000 слов. Слова определяются как отдельные высказывания, отделенные достаточно длительными промежутками от других слов во фразе. Для целей диалога распознаватели речи используются в составе вспомогательных средств подачи отдельных простых команд. На графических рабочих местах такие системы могут освобождать пользователя от необходимости задания команд с помощью

клавиатуры или меню, например, при вводе графической информации с помощью планшетов, когда подача команд с помощью обычных средств диалога нарушает так называемую тактильную непрерывность диалога.

Синтез речи обычно производится по формантному методу. Сначала текст переводится в последовательность фонем, затем по ней рассчитывается динамика управления формантным синтезатором, который генерирует и выдает сообщение.

### 0.13.3 Средства диалога для систем виртуальной реальности

В системах виртуальной реальности, в отличие от обычных приложений компьютерной графики, как правило, требуется вывод и ввод трехмерной координатной информации, как для управления положениями синтезируемых объектов, так и для определения координат частей тела оператора и направления его взгляда.

**Space ball.** Одним из первых появилось устройство спейсбол (space ball), представляющее собой конструктивное объединение мышки и небольшого трекбола. Мышка перемещается оператором по столу и обеспечивает ввод двух координат. Ввод третьей координаты обеспечивается вращением шарика трекбола большим пальцем руки.

Для манипулирования объектами в трехмерном пространстве часто используется техника виртуальной сферы. Управляемый объект окружается (воображаемой) сферой. Для перемещения сферы используется мышь, а вращение сферы и заключенного в нее объекта обеспечивается вращением шарика трекбола.

**Head Mounted Display.** В системах виртуальной реальности используются устройства вывода в виде монтируемых на голове дисплеев (Head Mounted Display — HMD) с бинокулярным всенаправленным монитором (Binocular Omni-Orientation Monitor — BOOM) со средствами отслеживания положения головы (head tracking) и даже отслеживанием положения глаза (eye tracking). Это требуется для создания эффекта “погружения” со стереоскопическим изображением и оперативным изменением сцены при поворотах головы и/или глаз.

Используемые в HMD жидкокристаллические дисплеи обычно невысокого разрешения (до  $417 \times 277$  пикселей). Сравните это с  $1280 \times 1024$  и  $1600 \times 1200$  для настольных систем или с разрешением мониторов для телевидения высокой четкости (ТВВЧ) —  $1920 \times 1035$  и  $1920 \times 1135$ . Поэтому ведутся интенсивные исследования по созданию средств отображения для систем виртуальной реальности, обладающих высоким разрешением при приемлемых значениях электромагнитных наводок. Одна из таких систем, использующих миниатюрные монохромные прецизионные электронные трубки и жидкокристаллические затворы, обеспечивает разрешение до  $2000 \times 2000$ . Интересное решение заключается в формировании изображения лазером непосредственно на сетчатке, но эти предложения пока далеки от коммерческой реализации.

Отслеживание положения головы обеспечивается либо механическими рычажными системами, либо комплектом инфракрасных или электромагнитных датчиков.

**Power Glove, Data Glove, Data Suit.** Непосредственный ввод геометрической информации о положении частей тела с поддержкой тактильной и даже силовой обратной связи обеспечивается перчатками и костюмами данных.

Дешевая перчатка данных — Power Glove, используемая для игр, обеспечивает только четыре уровня данных.

В более продвинутой перчатке данных Data Glove фирмы VPL (рис. 0.13.1) для определения углов сгибания пальцев используются оптические волокна. Для обеспечения тактильной обратной связи используются пневматические активаторы.



Рис. 0.13.1: Перчатка данных фирмы VPL

Были эксперименты обеспечения тактильной обратной связи за счет вибрации пьезокристаллов.

Более точный ввод координатной информации обеспечивают системы с использованием механического рычажного экзоскелета руки (Exos Dexterous Handmaster) и датчиками углов сгибания пальцев на основе эффекта Холла. Это позволяет достигнуть точности  $\pm 0.5^\circ$ . Системы с экзоскелетом позволяют обеспечить и силовую обратную связь.

Более простой прибор, использующий силовую обратную связь, был разработан фирмой Digital и представляет собой рукоятку, подобную рукоятке газа в мотоцикле, которая может менять свое сопротивление скручиванию.

Проблема в обеспечении тактильной и силовой обратной связи состоит в том, что пользователь реагирует на воздействия и вносит изменения быстрее, чем система сможет среагировать. Для хорошего ощущения объекта система тактильной обратной связи должна обеспечивать скорость порядка 100–300 Гц, что почти на порядок выше обычной скорости перезаписи визуальной информации.

Пиджак данных (Data Suit) по принципам работы подобен перчатке данных и отличается только количеством датчиков.



# Литература

- [1] Ньюмен У., Спрулл Р. Основы интерактивной машинной графики. Пер. с англ. М.: Мир, 1976.
- [2] Роджерс Д. Алгоритмические основы машинной графики. Пер. с англ. М.: Мир, 1989.
- [3] Фоли Дж., вэн Дэм А. Основы интерактивной машинной графики: В 2-х книгах. Пер. с англ. М.: Мир, 1985.
- [4] Encarnacao J. Einfuerung in die Graphische Datenverarbeitung// Eurographics '89. Tutorial Notes 1. Hamburg, FRG, September 4–8, 1989. 122 s.

# Литература

- [1] Антонофф Майкл, Линдерхолм Оуэн. Лазерные принтеры// Компьютер Пресс, сборник N 1, 1989, с. 3–8.
- [2] Винцок Т.К. Системы речевого диалога// Материалы пятой школы-семинара "Интерактивные системы" (Кутаиси, 2–10 апреля 1983 г.). Тбилиси: Мецниереба, 1983, с. 16–22.
- [3] Печатающие устройства персональных ЭВМ: Справочник// Под редакцией проф. И.М.Витенберга. М.: Радио и связь, 1992.
- [4] Лисицин Б.Л. Низковольтные индикаторы: Справочник. М.: Радио и связь, 1985.
- [5] Справочник по машинной графике в проектировании/ В.Е.Михайленко, В.А.Анпилогова, Л.А.Кириевский и др.: Под ред. В.Е.Михайленко. А.А.Лященко. Киев: Будівельник, 1984. 184 с.
- [6] Мячев А.А, Степанов В.Н. Персональные ЭВМ и микроЭВМ. Основы организации: Справочник/ Под ред. А.А.Мячева. М.: Радио и связь, 1991.
- [7] Новаковский С.В. Цвет в цветном телевидении. М.: Радио и связь, 1988. 288 с.
- [8] Павлидис Т. Алгоритмы машинной графики и обработки изображений. Пер. с англ. М.: Радио и связь, 1986.
- [9] Роджерс Д., Адамс Дж. Математические основы машинной графики Пер. с англ. М.: Машиностроение, 1980.
- [10] Сизых В.Г. Растровые дисплеи ряда Гамма. Новосибирск, 1985. 26 с. (Препринт ВЦ СО АН СССР; ь 607).
- [11] Ткаченко А.П. Цветное телевидение. Минск: Беларусь, 1981. 253 с.
- [12] Фролов А.В., Фролов Г.В. Программирование видеоадаптеров CGA, EGA, VGA. М.: Диалог-МИФИ, 1992.
- [13] Joseph Hans, Mehl Max. Computer Graphics Hardware: Introduction and State of the Art// Eurographics '91. Tutorial Note 9. Viena, 2.–6. September 1991. Austria, Viena. 29 p.
- [14] Fontenier Guy and Gros Pascal. Architectures of Graphic Processors for Interactive 2D Graphics// Computer Graphics Forum 7 (1988) 78–89.

- [15] Pinkman R., Novak M., Guttag K. Video-RAM exels at fast graphics // Electronics Design, pp. 161–171 (August 18 1983).
- [16] Seidel H.-P. PC Graphics Hardware // Eurographics '88. Tutorial/Cours 8.-Nice, 12.–16. September 1988. France, Nice. 44 p.
- [17] Stralunsfreier Flachbildschirm. MC, Die MikrocomputerZeitschrift. Ъ 8, 1989, s. 66.

## 0.14 Приложение 1. Преобразование моделей цветов

В данном приложении приведены процедуры перевода из RGB в HSV и наоборот и процедуры перевода из RGB в HLS и наоборот, соответствующие процедурам из [Род89] и [Фол85]. В процедурах используется вещественная арифметика. Переход на целочисленную арифметику, более подходящую для реальных применений, может быть выполнен определением констант для максимальных значений координат цвета — R,G,B, оттенка (hue), насыщенности (saturation), светлоты (value или lightness) и соответствующей модификацией текстов процедур.

```
/*===== RGB_HSV
 * Преобразования из модели HSV в RGB и наоборот
 */
```

```
#include <stdio.h>
```

```
#define UNDEFINED -1.0 /* Неопределенное значение для Hue */
```

### 0.14.1 V\_HSVRGB — перевод из HSV в RGB

```
/*----- V_HSVRGB
 * Переводит из модели HSV в модель RGB
 *
 * void V_HSVRGB (float *r, float *g, float *b,
 *               float hue, float sat, float val)
 *
 * Вход:
 * 0 <= hue <= 360 градусов - оттенок. Основные цвета:
 *           0 - красный, 60 - желтый,
 *           120 - зеленый, 180 - голубой
 *           240 - синий, 300 - пурпурный
 *           Остальные цвета между ними
 * 0.0 <= sat <= 1.0 - Saturation - насыщенность
 * 0.0 <= val <= 1.0 - Value - светлота
 *
 * Выход:
 * 0.0 <= r,g,b <= 1.0 - значения красного, зеленого, синего
 * V_HSVRGB= 0 - норма
 *           1/2/3 - ошибка в hue/ошибка в sat/ошибка в val
 */
```

```
int V_HSVRGB (r, g, b, hue, sat, val)
float *r, *g, *b, hue, sat, val;
{ int ii, otw;
  float c1, c2, c3, fr;

  otw= 0;
```

```

if (sat == 0.0) {
    /* Ахроматический цвет */
    *r= val; *g= val; *b= val;
    if (hue != UNDEFINED) ++otw;
} else {
    /* Хроматический цвет */
    hue-= (ii= (int)(hue/360.0)) * 360.0;
    if (ii < 0) hue= -hue;
    ii= (int)(hue /= 60.0);
    fr= hue - ii;
    c1= val*(1.0 - sat);
    c2= val*(1.0 - sat*fr);
    c3= val*(1.0 - sat*(1.0 - fr));
    switch (ii) {
        case 0: *r= val; *g= c3; *b= c1; break;
        case 1: *r= c2; *g= val; *b= c1; break;
        case 2: *r= c1; *g= val; *b= c3; break;
        case 3: *r= c1; *g= c2; *b= val; break;
        case 4: *r= c3; *g= c1; *b= val; break;
        case 5: *r= val; *g= c1; *b= c2; break;
    }
}
return (otw);
} /* V_HSVRGB */

```

## 0.14.2 V\_RGBHSV — перевод из RGB в HSV

```

/*----- V_RGBHSV
* Переводит из модели RGB в модель HSV
*
* void V_RGBHSV (float r, float g, float b,
*               float *hue, float *sat, float *val)
*
* Вход:
* 0.0 <= r,g,b <= 1.0 - значения красного, зеленого, синего
*
*
* Выход:
* 0 <= hue <= 360 градусов - оттенок. Основные цвета:
*           0 - красный, 60 - желтый,
*           120 - зеленый, 180 - голубой
*           240 - синий, 300 - пурпурный
*           Остальные цвета между ними
* 0.0 <= sat <= 1.0 - Saturation - насыщенность
* 0.0 <= val <= 1.0 - Value - светлота
*
* V_RGBHSV= 0 - норма
*           1/2/3 - ошибка в r/ошибка в g/ошибка в b

```

```

*/

int V_RGBHSV (r, g, b, hue, sat, val)
float r, g, b, *hue, *sat, *val;
{ int otw;
  float minc, maxc, h, s, v, dmax, rc, gc, bc;

  otw= 0;
  if (r < 0.0 || r > 1.0) ++otw;    /* Проверки значений */
  if (g < 0.0 || g > 1.0) otw= 2;
  if (b < 0.0 || b > 1.0) otw= 3;
  if (!otw) {
    if ((maxc= r) < b) maxc= b;    /* Поиск макс значения */
    if (maxc < g) maxc= g;
    if ((minc= r) > b) minc= b;    /* Поиск мин значения */
    if (minc > g) minc= g;
    s= 0.0;                        /* Насыщенность */
    if (maxc != 0.0) s= (maxc-minc)/maxc;
    if (s == 0.0) h= UNDEFINED;    /* Ахроматический цвет */
    else {                          /* Хроматический цвет */
      dmax= maxc-minc;
      rc= (maxc-r)/dmax;            /* rc - удаленность */
      gc= (maxc-g)/dmax;            /* цвета от красного */
      bc= (maxc-b)/dmax;
      if (r == maxc) h= bc-gc; else /* Цвет между жел- */
                                          /* тым и пурпурным */
      if (g == maxc) h= 2+rc-bc; else /* Цвет между голу-*/
                                          /* бым и желтым */
      h= 4+gc-rc;                    /* Цвет между пур- */
                                          /* пурным и голубым*/
      if ((h*= 60.0) < 0.0) h+= 360.0;
    }
    *hue= h; *sat= s; *val= maxc;
  }
  return (otw);
} /* V_RGBHSV */

```

### 0.14.3 Тест процедур V\_HSVRGB и V\_RGBHSV

```

/*----- MAIN RGB_HSV
* Тестовая программа проверки прдпрограмм
* преобразований из системы HSV в RGB и наоборот.
*/

void main(void)
{ int ii;

```

```

float r, g, b, r1, g1, b1, hue, sat, val;

r= 0.0;  g= 0.0;  b= 0.0;  hue= 0.0;

m0:printf ("HSV r,g,b= (%f %f %f) ? ", r,g,b);
scanf ("%f%f%f", &r, &g, &b);
ii= V_RGBHSV (r, g, b, &hue, &sat, &val);
printf ("V_RGBHSV: otw=%d hue=%f sat=%f val=%f\n",
        ii, hue, sat, val);
ii= V_HSVRGB (&r1, &g1, &b1, hue, sat, val);
printf ("V_HSVRGB: otw=%d r=%f g=%f b=%f\n",
        ii, r1, g1, b1);

goto m0;
}

/*===== RGB_HLS
* Преобразования из модели HLS в RGB и наоборот
*/

#include <stdio.h>

#define UNDEFINED -1.0 /* Неопределенное значение для Hue */

```

#### 0.14.4 V\_HLSRGB — перевод из HLS в RGB

```

/*----- V_HLSRGB
* Переводит из модели HLS в модель RGB
*
* void V_HLSRGB (float *r, float *g, float *b,
*               float hue, float sat, float lig)
*
* Вход:
* 0 <= hue <= 360 градусов - оттенок. Основные цвета:
*           0 - красный, 60 - желтый,
*           120 - зеленый, 180 - голубой
*           240 - синий, 300 - пурпурный
*           Остальные цвета между ними
* 0.0 <= sat <= 1.0 - Saturation - насыщенность
* 0.0 <= lig <= 1.0 - Value - светлота
*
* Выход:
* 0.0 <= r,g,b <= 1.0 - значения красного, зеленого, синего
* V_HLSRGB= 0 - норма
*           1/2/3 - ошибка в hue/ошибка в sat/ошибка в lig
*/

```

```

static float VAL_RGB (n1, n2, hue)
float n1, n2, hue;
{ float otw;
  if (hue < 0)      hue+= 360.0;
  hue= hue - ((int)(hue/360.0)) * 360.0;
  if (hue < 60.0)  otw= n1+(n2-n1)*hue/60.0; else
  if (hue < 180.0) otw= n2;                      else
  if (hue < 240.0) otw= n1+(n2-n1)*(240.0-hue)/60.0; else
  otw= n1;
  return (otw);
} /* VAL_RGB */

```

```

int V_HLSRGB (r, g, b, hue, sat, lig)
float *r, *g, *b, hue, sat, lig;
{ int otw;
  float m1, m2;

  otw= 0;
  if (sat == 0.0) { /* Ахроматический случай */
    if (hue == UNDEFINED) {
      *r= lig; *b= lig; *g= lig;
    } else ++otw;
  } else { /* Хроматический случай */
    if ((m2= lig) > 0.5) m2= 1.0 - lig;
    m2= lig + sat*m2;
    m1= 2*lig - m2;
    *r= VAL_RGB (m1, m2, hue+120.0);
    *g= VAL_RGB (m1, m2, hue);
    *b= VAL_RGB (m1, m2, hue-120.0);
  }
  return (otw);
} /* V_HLSRGB */

```

### 0.14.5 V\_RGBHLS — перевод из RGB в HLS

```

/*----- V_RGBHLS
* Переводит из модели RGB в модель HLS
*
* void V_RGBHLS (float r, float g, float b,
*               float *hue, float *sat, float *lig)
*
* Вход:
* 0.0 <= r,g,b <= 1.0 - значения красного, зеленого, синего
*
*

```



```

* Выход:
* 0 <= hue <= 360 градусов - оттенок. Основные цвета:
*           0 - красный, 60 - желтый,
*           120 - зеленый, 180 - голубой
*           240 - синий, 300 - пурпурный
*           Остальные цвета между ними
* 0.0 <= sat <= 1.0 - Saturation - насыщенность
* 0.0 <= lig <= 1.0 - Value - светлота
*
* V_RGBHLS= 0 - норма
*           1/2/3 - ошибка в r/ошибка в g/ошибка в b
*/

```

```

int V_RGBHLS (r, g, b, hue, sat, lig)
float r, g, b, *hue, *sat, *lig;
{ int otw;
  float minc, maxc, h, s, l, dm, rc, gc, bc;

  otw= 0;
  if (r < 0.0 || r > 1.0) ++otw;
  if (g < 0.0 || g > 1.0) otw= 2;
  if (b < 0.0 || b > 1.0) otw= 3;
  if (!otw) {
    if ((maxc= r) < b) maxc= b;
    if (maxc < g) maxc= g;
    if ((minc= r) > b) minc= b;
    if (minc > g) minc= g;
    l= (maxc + minc) / 2;      /* Светлота */
    if (maxc == minc) {      /* r=g=b - ахроматический */
      s= 0.0;                /* случай */
      h= UNDEFINED;
    } else {                /* Хроматический случай */
      dm= maxc-minc;
      s= (l < 0.5) ? dm/(maxc+minc) : dm/(2-maxc-minc);
      rc= (maxc-r)/dm;       /* Вычисление тона */
      gc= (maxc-g)/dm;       /* gc задает удаленность */
      bc= (maxc-b)/dm;       /* цвета от красного */
      if (r == maxc) h= bc-gc; else /* Цвет между жел- */
                                          /* тым и пурпурным */
      if (g == maxc) h= 2+rc-bc; else /* Цвет между голу-*/
                                          /* бым и желтым */
      h= 4+gc-rc;           /* Цвет между пур- */
                                          /* пурным и голубым*/
      if ((h*= 60.0) < 0.0) h+= 360.0;
    }
  }
}

```

```

    *hue= h; *sat= s; *lig= l;
}
return (otw);
} /* V_RGBHLS */

```

### 0.14.6 Тест процедур V\_HLSRGB и V\_RGBHLS

```

/*----- MAIN RGB_HLS
* Тестовая программа проверки подпрограмм
* преобразований из системы HLS в RGB и наоборот.
*/

void main(void)
{ int ii;
  float r, g, b, r1, g1, b1, hue, sat, lig;

  r= 0.0; g= 0.0; b= 0.0; hue= 0.0;

m1:printf ("HLS r,g,b= (%f %f %f) ? ", r,g,b);
  scanf ("%f%f%f", &r, &g, &b);
  ii= V_RGBHLS (r, g, b, &hue, &sat, &lig);
  printf ("V_RGBHLS: otw=%d hue=%f sat=%f lig=%f\n",
          ii, hue, sat, lig);
  ii= V_HLSRGB (&r1, &g1, &b1, hue, sat, lig);
  printf ("V_HLSRGB: otw=%d r=%f g=%f b=%f\n",
          ii, r1, g1, b1);

  goto m1;
}

```

