

Учитель Начальной

Курс компьютерной графики

<https://litres.ru/74569473>

SelfPub; 2026

Аннотация

Учебное пособие представляет собой систематическое изложение основ и современных методов компьютерной графики в двадцати главах. Книга последовательно знакомит читателя с растровой и векторной графикой, цветовыми моделями, системами координат и однородными преобразованиями. Подробно рассматриваются алгоритмы растеризации, отсечения, двумерные и трёхмерные геометрические преобразования, проекции, представление объектов, удаление невидимых поверхностей, модели освещения и затенения, текстурирование, параметрические кривые и поверхности. Отдельные главы посвящены трассировке лучей, глобальному освещению, компьютерной анимации, архитектуре GPU и современным графическим API (OpenGL, Vulkan, DirectX, Metal, WebGPU). Заключительная часть освещает практические приложения: игры, кино, VR/AR, научную визуализацию и CAD. Издание содержит формулы, псевдокод и практические рекомендации. Предназначено для студентов, преподавателей и специалистов, изучающих компьютерную графику.

Содержание

ГЛАВА 1. ВВЕДЕНИЕ В КОМПЬЮТЕРНУЮ ГРАФИКУ	5
1.1. Что такое компьютерная графика	5
1.2. Краткая история	7
1.3. Основные направления	9
1.4. Области применения	10
1.5. Базовые понятия	11
1.6. Программное и аппаратное обеспечение	12
ГЛАВА 2. РАСТРОВАЯ И ВЕКТОРНАЯ ГРАФИКА	13
2.1. Растровая графика	13
2.2. Векторная графика	15
2.3. Сравнение и гибридный подход	17
2.4. Практические рекомендации	18
ГЛАВА 3. ЦВЕТОВЫЕ МОДЕЛИ И ВОСПРИЯТИЕ ЦВЕТА	19
3.1. Физика и физиология цвета	19
3.2. Модель RGB	20
3.3. Модель CMY и CMYK	21
3.4. Модели HSV и HSL	22
3.5. CIE Lab и перцепционные пространства	23
3.6. Гамма-коррекция	24
3.7. Практические советы	25

ГЛАВА 4. СИСТЕМЫ КООРДИНАТ И ОДНОРОДНЫЕ КООРДИНАТЫ	26
4.1. Декартовы системы координат	26
4.2. Однородные координаты	27
4.3. Системы координат в графическом конвейере	28
Конец ознакомительного фрагмента.	29

Учитель Начальной Курс компьютерной графики

ГЛАВА 1. ВВЕДЕНИЕ В КОМПЬЮТЕРНУЮ ГРАФИКУ

1.1. Что такое компьютерная графика

Компьютерная графика - это область информатики, занимающаяся созданием, обработкой, анализом и визуализацией изображений с помощью вычислительных систем. Она находится на стыке математики, физики, алгоритмов и аппаратного обеспечения.

Основные задачи компьютерной графики:

- Синтез изображений (rendering) - создание картинки по описанию сцены
- Обработка изображений - фильтрация, улучшение, сжатие, анализ

- Геометрическое моделирование - описание формы объектов
- Анимация - создание движения
- Визуализация данных - представление научной и инженерной информации

1.2. Краткая история

1950-1960-е годы

- 1950 - Ben Laposky создаёт первые "осциллограммы" абстрактные изображения
- 1961 - Spacewar! - одна из первых компьютерных игр (MIT)
- 1963 - Sketchpad Айвена Сазерленда: интерактивное черчение световым пером.

Эта работа считается рождением интерактивной компьютерной графики.

- 1968 - Сазерленд демонстрирует первый HMD (шлем виртуальной реальности)

1970-е годы

- Алгоритмы затенения Гуро (1971) и Фонга (1975)
- Z-буфер Эдвина Катмулла (1974)
- Первые системы CAD
- Появление персональных компьютеров с графическими возможностями

1980-1990-е

- Появление графических ускорителей (SGI, позже 3dfx Voodoo, NVIDIA, ATI)
- OpenGL (1992) и DirectX
- Фотореалистичный оффлайн-рендеринг в кино (Pixar, Industrial Light & Magic)

- Расцвет 3D-игр (Quake, Unreal)

2000-2020-е

- Программируемые шейдеры (DirectX 8/9, OpenGL 2.0)

- Физически корректный рендеринг (PBR)

- Аппаратный ray tracing (NVIDIA RTX, 2018)

- Нейросетевой рендеринг, Gaussian Splatting, генератив-

ный ИИ

1.3. Основные направления

1. Синтез изображений (Computer-Generated Imagery, CGI)
Создание изображений "с нуля" по геометрическому и физическому описанию сцены.
2. Обработка и анализ изображений
Работа с уже существующими растрами: фильтрация, сегментация, распознавание, компьютерное зрение.
3. Геометрическое моделирование
Способы представления формы: полигональные сетки, NURBS, неявные поверхности, воксели, облака точек.
4. Компьютерная анимация
Ключевые кадры, скелетная анимация, физическая симуляция, motion capture.
5. Научная и информационная визуализация
Отображение многомерных данных, потоков, объёмных сканов (КТ, МРТ).
6. Виртуальная, дополненная и смешанная реальность (VR / AR / MR)

1.4. Области применения

- Кино и телевидение (визуальные эффекты, полная CGI-анимация)
- Компьютерные игры и интерактивные развлечения
- Промышленный дизайн, CAD/CAM/CAE
- Архитектура и строительство (BIM)
- Медицина (визуализация томографии, хирургические симуляторы)
- Научные исследования (молекулярная графика, астрофизика, CFD)
- Симуляторы (авиационные, автомобильные, военные)
- Реклама, дизайн интерфейсов, картография
- Образование и тренажёры

1.5. Базовые понятия

Пиксель (pixel) - наименьший элемент растрового изображения.

Разрешение - количество пикселей (например, 1920x1080).

Глубина цвета - сколько бит отведено на цвет одного пикселя (8, 24, 32 бит).

Кадр (frame) - одно полное изображение.

Частота кадров (FPS) - количество кадров в секунду.

Буфер кадра (framebuffer) - область памяти, в которой формируется изображение.

Конвейер рендеринга (graphics pipeline) - последовательность этапов

преобразования данных сцены в пиксели на экране.

Латентность (latency) - задержка между действием пользователя и обновлением

картинки (критично для VR и игр).

1.6. Программное и аппаратное обеспечение

Программное:

- Графические API: OpenGL, Vulkan, DirectX, Metal, WebGPU
- Движки: Unreal Engine, Unity, Godot, CryEngine
- Пакеты моделирования: Blender, Maya, 3ds Max, Cinema 4D
- CAD: SolidWorks, AutoCAD, FreeCAD

Аппаратное:

- GPU (Graphics Processing Unit) - специализированный процессор
- Видеопамять (VRAM)
- Дисплеи (LCD, OLED, мини-LED), VR-шлемы, графические планшеты

ГЛАВА 2. РАСТРОВАЯ И ВЕКТОРНАЯ ГРАФИКА

2.1. Растровая графика

Растровое изображение - это прямоугольная матрица пикселей. Каждый пиксель хранит информацию о цвете (и иногда о прозрачности - альфа-канал).

Характеристики:

- Ширина и высота в пикселях
- Глубина цвета (bits per pixel, bpp)
- Цветовое пространство (sRGB, Adobe RGB, Display P3 и др.)

- Наличие альфа-канала

Преимущества:

- + Естественность для фотографий и сложных текстур
- + Простота отображения на растровых экранах
- + Поддержка сложных эффектов и фильтров
- + Хорошая поддержка прозрачности и полупрозрачности

Недостатки:

- Потеря качества при увеличении (пикселизация)
- Большой объём данных (особенно без сжатия)

- Сложность редактирования отдельных объектов сцены
- Зависимость от разрешения

Пример объёма данных:

Изображение 1920x1080, 24 бита на пиксель без сжатия:
1920 x 1080 x 3 байта ~ 6,2 МБ на один кадр.

При 60 кадрах в секунду - уже около 372 МБ/с (отсюда необходимость сжатия и аппаратного ускорения).

Популярные форматы:

- BMP - простой, без сжатия (или RLE)
- PNG - сжатие без потерь + альфа-канал, идеален для графики с резкими краями
- JPEG - сжатие с потерями, отлично для фотографий
- GIF - до 256 цветов, поддержка простой анимации
- WebP / AVIF - современные форматы с лучшим сжатием
- EXR - HDR-изображения, используется в кино и VFX
- TGA, TIFF - часто применяются в промежуточных пайплайнах

2.2. Векторная графика

Векторное изображение описывается математическими примитивами:

- Точки
- Отрезки и ломаные
- Кривые Безье
- Многоугольники
- Текст (как набор контуров)
- Заливки и обводки

Преимущества:

- + Идеальная масштабируемость без потери качества
- + Компактность для схем, логотипов, иллюстраций
- + Лёгкость редактирования отдельных объектов
- + Возможность точного позиционирования

Недостатки:

- Сложность представления фотореалистичных изображений
- Более высокая вычислительная нагрузка при растеризации сложных сцен
- Ограниченная естественность для органических форм (хотя современные инструменты сильно продвинулись)

Форматы:

- SVG - стандарт для веба, текстовый XML-формат

- AI - Adobe Illustrator
- EPS - Encapsulated PostScript
- PDF - может содержать как вектор, так и растр
- DXF - обмен данными CAD

Пример SVG (простой круг):

```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="black" stroke-  
width="2" fill="red" />  
</svg>
```

2.3. Сравнение и гибридный подход

В реальных системах почти всегда используется гибрид:

1. Геометрия и большинство объектов хранятся в векторном (или полигональном) виде

2. На этапе вывода происходит растеризация - превращение в пиксели

3. Текстуры и фотографии остаются растровыми

Современные GPU выполняют растеризацию аппаратно с огромной скоростью

(миллиарды пикселей в секунду).

Векторные шрифты (TrueType, OpenType) растеризуются на лету. При этом

применяется хинтинг - специальные инструкции, улучшающие читаемость

на малых размерах.

2.4. Практические рекомендации

- Для логотипов, иконок, схем, UI-элементов - вектор (SVG, PDF)
- Для фотографий и сложных текстур - растр (PNG/WebP без потерь или JPEG)
- В играх и 3D: геометрия - полигоны (векторная по сути), поверхности -
 - текстуры (растр)
- Всегда храните исходники в максимальном качестве и с возможностью редактирования

ГЛАВА 3. ЦВЕТОВЫЕ МОДЕЛИ И ВОСПРИЯТИЕ ЦВЕТА

3.1. Физика и физиология цвета

Цвет - это ощущение, возникающее при воздействии электромагнитного излучения видимого диапазона (примерно 380-780 нм) на сетчатку глаза.

В глазу человека:

- Палочки - отвечают за сумеречное зрение (низкая освещённость, нет цвета)
- Колбочки - три типа (L, M, S), чувствительные к длинноволновому (красному), средневолновому (зелёному) и коротковолновому (синему) свету.

Именно наличие трёх типов колбочек лежит в основе трёхкомпонентной теории цвета и большинства цветовых моделей.

3.2. Модель RGB

Аддитивная модель. Цвета получаются сложением красного, зелёного и синего излучений.

Применяется: мониторы, телевизоры, проекторы, сканеры, цифровые камеры.

Цифровое представление:

- 8 бит на канал (0-255) -> 24 бита = 16 777 216 цветов (True Color)

- С альфа-каналом: RGBA, 32 бита

- HDR: 10, 12 или 16 бит на канал (например, RGB10_A2, RGBA16F)

Пример:

Красный = (255, 0, 0)

Зелёный = (0, 255, 0)

Синий = (0, 0, 255)

Белый = (255, 255, 255)

Чёрный = (0, 0, 0)

Жёлтый = (255, 255, 0)

Серый 50% = (128, 128, 128)

Важно: в вычислениях освещения нужно работать в линейном пространстве,

а гамма-коррекцию применять только в конце.

3.3. Модель СМУ и СМУК

Субтрактивная модель. Используется в печати.

С (Cyan) - голубой

М (Magenta) - пурпурный

У (Yellow) - жёлтый

К (Key) - чёрный

Почему добавляют чёрный:

- Смешение СМУ на практике не даёт глубокого чёрного
- Чёрная краска дешевле
- Улучшается чёткость текста

Пример преобразования (упрощённо):

$$C = 1 - R$$

$$M = 1 - G$$

$$Y = 1 - B$$

(затем выполняется undercolor removal и добавление К)

3.4. Модели HSV и HSL

Более удобны для человека при выборе цвета.

HSV:

- H (Hue) - тон (0-360 deg)
- S (Saturation) - насыщенность (0-1)
- V (Value) - значение/яркость (0-1)

HSL:

- H - тон
- S - насыщенность
- L - светлота

Эти модели широко используются в палитрах графических редакторов.

3.5. CIE Lab и перцепционные пространства

CIE XYZ - стандартное колориметрическое пространство.

CIE Lab - перцепционно более равномерное:

- L - светлота
- a - ось от зелёного к красному
- b - ось от синего к жёлтому

Используется в профессиональной цветокоррекции и при вычислении разницы между цветами (ΔE).

3.6. Гамма-коррекция

Человеческое зрение и старые ЭЛТ-мониторы имеют нелинейную характеристику.

Стандарт sRGB использует гамму $\sim 2,2$.

Правильный пайплайн:

1. Текстуры и цвета обычно хранятся в sRGB (с гаммой)
2. Перед вычислениями освещения - перевод в линейное пространство
3. Все расчёты освещения - в линейном пространстве
4. Перед выводом на экран - обратная гамма-коррекция (и тональная компрессия для HDR)

Ошибка многих начинающих: считать освещение прямо в sRGB - картинка получается слишком тёмной в полутонах.

3.7. Практические советы

- Для веба и большинства приложений - sRGB
- Для печати - работа в CMYK или преобразование через правильные ICC-профили
- Для кино и VFX - линейные пространства + ACES
- Всегда учитывайте цветовой профиль монитора при критичной работе с цветом

ГЛАВА 4. СИСТЕМЫ КООРДИНАТ И ОДНОРОДНЫЕ КООРДИНАТЫ

4.1. Декартовы системы координат

В компьютерной графике встречаются:

- Правая система координат (Right-handed)

- Левая система координат (Left-handed)

В правой системе: если большой палец - X, указательный

- Y, то средний - Z.

OpenGL (классический): правая система.

DirectX: чаще левая.

Важно всюду придерживаться одной системы, иначе модели будут "вывернуты".

4.2. Однородные координаты

Одно из ключевых изобретений компьютерной графики.

Точка (x, y, z) представляется как (x, y, z, w) , где обычно $w = 1$.

Вектор направления - $(x, y, z, 0)$.

Зачем это нужно:

- Все аффинные преобразования (включая перенос) записываются одной матрицей 4×4

- Перспективное проецирование становится матричным умножением + делением на w

- Удобно работать с бесконечно удалёнными точками ($w = 0$)

Преобразование:

$$(x', y', z', w') = (x, y, z, w) \times M$$

Декартовы координаты после преобразования:

$$X = x' / w'$$

$$Y = y' / w'$$

$$Z = z' / w'$$

4.3. Системы координат в графическом конвейере

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.