



# Основные направления обработки информации, связанной с изображениями

Компьютерная графика  
Обработка изображений



# Изображения

- Непрерывная и дискретная формы изображения.
- Непрерывное изображение представляется как функция яркости, определенная на пространстве  $S(r, t)$ , где  $r$  – пространственный компонент,  $t$  – время.
- Динамическим называется изображение, которое зависит от времени.
- Статическим называется изображение, которое не зависит от времени.

# Непрерывная и дискретная формы изображения

Пиксель - это идеальная точка, для которой функция распределения яркости определяется как

$$S(\mathbf{r}) = \rho \delta(\mathbf{r} - \mathbf{r}_0) \delta(t - t_0)$$

**Дискретное** изображение получается из непрерывного путем дискретизации его непрерывных координат с получением дискретных. Процесс **дискретизации** непрерывного изображения может быть представлен как умножение непрерывного изображения на пространственную дискретизирующую функцию.

После дискретизации выполняется *квантование* для преобразования непрерывного диапазона яркостей  $[0, S]$  в последовательность дискретных величин  $S_i$ , расположенных на равных интервалах  $\Delta S$ . Таким образом, двумерное дискретное изображение может быть представлено в виде **матрицы**, каждому элементу которой соответствует одна величина из диапазона возможных значений  $S_i$ .



# Типы изображений

- одномерные динамические - осциллограммы;
- двумерные статические- рисунки, карты, и др.;
- двумерные динамические - фильмы;
- квазитрехмерные статические - стереофотографии, 3D графика;
- квазитрехмерные динамические - стереофильмы, 3D графика;
- трехмерные динамические - голограммы.



# Задачи компьютерной графики

- Визуализация - создание изображений.
  - Изображения создаются исходя из модели того, что нужно отображать. Принято различать системы...
- Обработка изображений - это преобразование изображений.
  - Входными данными является изображение с его параметрами,
  - результатом обработки - тоже изображение, отличающееся по своим характеристикам.

# Характеристики изображения

- Контраст,
- четкость,
- коррекция цветов,
- сглаживание,
- уменьшение шумов,
- выделение контуров перед распознаванием,
- создание бинарного изображения,
- разделение по цветам

Методы обработки существенно отличаются для изображений, синтезированных системой компьютерной графики и для изображений, полученных в результате оцифровки черно-белых и цветных фотографий или рисунков.

# По области применения различают системы

- Графического дизайна;
- Геоинформационные системы;
- Системы автоматизированного проектирования.
- Системы виртуальной реальности. Тренажеры, игровые системы (шлем-дисплей, сенсоры)

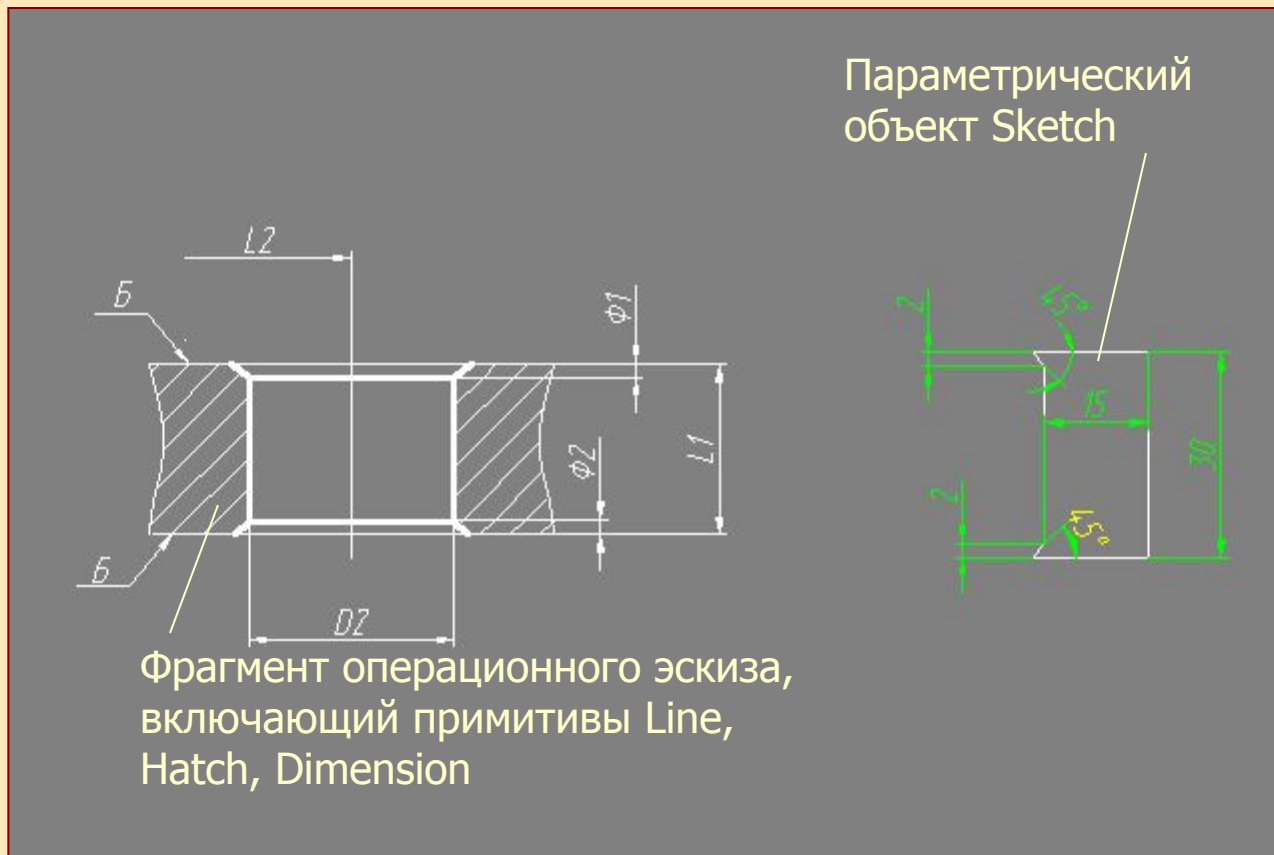
Изображение выполнено  
в системе графического  
дизайна XARA и  
экспортировано в JPEG







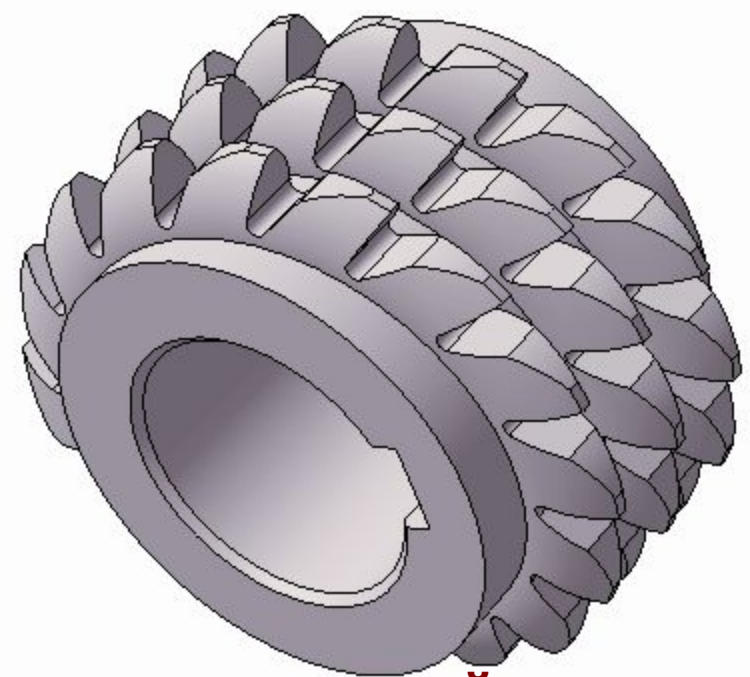
# Изображение создано в CAD-системе MechanicalDesktop





Имя	Псевдоним	Выражен
Шестерня 078.505.0.0102.00		
D2		
Фронтальная плоскость		
Горизонтальная плоскость		
Профильная плоскость		
Начало координат		
Операция выдавливания:1		
Скругление:1		
Фаска:2		
Ось операции:2		
Смещенная плоскость:1		
Спираль цилиндрическая:2		
Приклеить кинематический элемент		
Смещенная плоскость:7		
Эскиз:9		
Вырезать элемент выдавливания		
Копия по концентрической сетке		

Дерево построения	
Шестерня 078.505.0.0102.00	
Фронтальная плоскость	
Горизонтальная плоскость	
Профильная плоскость	
Начало координат	
Операция выдавливания:1	
Скругление:1	
Фаска:2	
Ось операции:2	
Смещенная плоскость:1	
Спираль цилиндрическая:2	
Эскиз:6	
Приклеить кинематический элемент	
Смещенная плоскость:7	
Эскиз:9	
Вырезать элемент выдавливания	
Копия по концентрической сетке	



**Вид**

1.4727

Нормально к...

- Спереди
- Сзади
- Сверху
- Снизу
- Слева
- Справа
- Изометрия XYZ
- Изометрия YZX
- Изометрия ZXY
- Диметрия

**Твердотельная модель червячной шлицевой фрезы создана в САПР КОМПАС ГРАФИК**





Инверсирование  
изображения



Устранение пятен  
реализовано  
с помощью тени и  
осветления







Изменение яркости





# Основные направления обработки изображений:

1. Ввод графических документов.
2. Обработка изображений земной поверхности, медицинских изображений (идентификация), человеческих лиц и пр.
3. Обработка трехмерных сцен. Навигация и определение местоположения подвижного объекта с помощью видеокамеры.
4. Параллельные и нейронные архитектуры для обработки изображений в режиме реального времени.
5. Распознавание образов.





# Важнейшие задачи обработки (1) изображений и их особенности

- Системы распознавания текста и векторизации графических изображений. Последние обычно описывают изображения , в виде отрезков прямых, точек, дуг с недостаточной высокой точностью и не имеют промышленного применения.
- Практическое отсутствие автоматических методов. Интерактивность и зависимость от навыков и опыта оператора.
- Актуальной является задача управления объектами сцены на семантическом уровне, а не пикселями изображения.



# Важнейшие задачи обработки (2) изображений и их особенности

- Большой объем данных и сложность алгоритмов обработки.
- Изначально разрабатывались для обеспечения зрения роботов и систем специального назначения, распознавание печатного и рукописного текста.

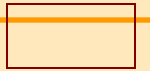
# Графические данные и виды форматов

- **Графический формат** - порядок, согласно которому данные, описывающие изображение, записаны в файле.
- **Графические данные** обычно разделяются на класса:
  - векторные
  - растровые
- Отдельно выделяют метафайловый формат (медиаконтейнер) – он хранит данные и информацию о том, как данные будут сохраняться внутри файла.

# Растровые изображения (1)

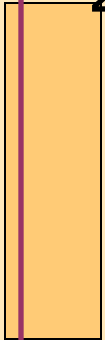
- **Пиксел** (picture element) - минимальный элемент цифрового изображения.
- Растровое изображение строится из большого числа пространственно упорядоченных пикселей. При одинаковых размерах пиксели могут иметь различные оптические характеристики (цвет, плотность и др).
- **Растровое изображение** представляет собой массив (bitmap=битовая карта) пикселей — матрицу определенной размерности, каждому элементу которой ставится в соответствие некоторое число - код яркости (0 - 256).





# Битовая карта изображения

- **Битовая карта** - набор последовательно записанных двоичных разрядов, то есть последовательность (массив) битов. Имеет несколько значений.
  1. Битовая карта в цифровых изображениях — матрица, хранящая значения элементов изображения (пикселов)
  2. Битовая карта в файловых системах — структура, хранящая информацию о наличии и расположении свободного места. Каждому кластеру в файловой системе соответствует один бит: сектор занят/свободен -(1/0)



## Растровые изображения (2)

- **Цветовая (битовая) глубина изображения** - число битов, отведенное каждому пикселу в этом массиве. Одинаково для всех пикселей изображения. Бывает 1, 2, 4, 8, 16, 24, 32 бита. Глубина цвета определяем минимальную разницу между яркостями пикселя.
- **Разрешающая способность** характеризует расстояние между соседними пикселями. Измеряется в количествах пикселей на единицу длины dpi (dots per inch). 1600x1200, 1280x1024, 1152x864, 1024x768, 800x600, 640x480.



# Векторные данные

- **Векторные данные** используются для представления прямых, многоугольников, кривых и пр. с помощью базовых (опорных, контрольных, ключевых) точек.
- Вместе с информацией о *базовых точках* хранятся *атрибуты* (цвет, толщина, тип линий) и набор правил (соглашений) вывода (рисования).



# Тип графического файла

определяется способом хранения и типом графических данных.

- Различают следующие форматы:
  - векторный;
  - растровый;
  - метафайловый,
  - специальные.





# Векторные форматы

- Векторный формат содержит математические описания элементов изображения. Создается и открывается определенным редактором. Предусмотрены форматы для импорта изображений.
- Примеры:
  - \*.ai, \*.svg, \*.xar, \*.eps, \*.fla, \*.tga;
  - \*.dwg, \*.dxf, \*.m3d, \*.frw, \*.igs;
  - \*.pdf
  - \*.vrml



# Растровый формат

используется для хранения растровых данных.

- Растровые файлы содержат
  - битовую карту изображения,
  - ее спецификацию.
- Примеры:
  - \*.bmp, \*.tiff, \*.gif, \*.pcx, \*.jpeg, \*.png
  - \*.raw.

# Медиаконтейнер

- способен хранить любой тип данных.
- На практике для каждого типа данных существуют отдельные группы контейнеров.
  - Эти группы реализованы для специфических требований и информации, которая будет сохраняться в них.
  - Медиаконтейнеры обычно предназначены для сохранения медиаинформации, которая условно делится на изображения, видео и аудио

# Метафайловый и специальные форматы

Метафайловый формат позволяет хранить в одном файле и векторные, растровые и видео данные.

- \*.cdr, \*.wmf, \*.emf, RIFF

Специальные предназначены для хранения

- мультипликации (видеоинформации), \*.exi;
- мультимедиа-форматы - хранят звуковую, видео- и графическую информацию \*.avi, \*.qtif, \*.mov;
- гипертекстовые - хранят текст и связи-переходы внутри него \*.html;
- гипермедиа - гипертекст, графическая, видеоинформация;
- форматы трехмерных сцен (3DStudioMax);



# Специальные форматы

- форматы шрифтов \*.ttf, \*.fon, и др.
- **WebP** - формат сжатия изображений с потерями и без потерь качества [3], Google Inc. Работает не в RGB, а в YUV. Модификации WebP и WebM.
- WebP Lossless, для графики работает в цветовом пространстве ARGB.
- WebM – специальные open source JavaScript-библиотеки позволяют отобразить видео в формате WebM в браузерах, поддерживающих

# Модель YUV

**YUV** — цветовая модель, в которой цвет состоит из трёх компонентов — яркость (Y) и два цветоразностных компонента (U и V).

Компоненты **YUV** определены на основе компонент **RGB** следующим образом:

$$Y = K_R \cdot R + (1 - K_R - K_B) \cdot G + K_B \cdot B$$

$$U = B - Y$$

$$V = R - Y$$

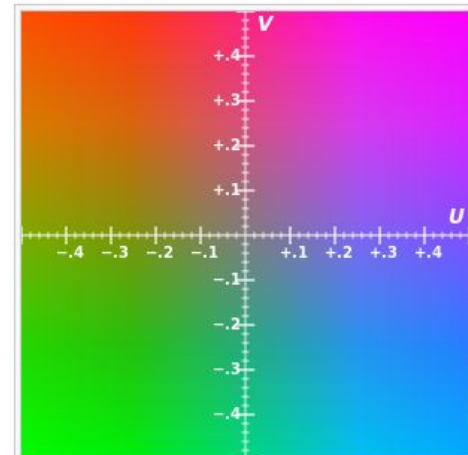
Обратное преобразование в **RGB**

$$R = Y + V$$

$$G = Y - \frac{K_R \cdot V + K_B \cdot U}{1 - K_R - K_B}$$

$$B = Y + U$$

Обратное преобразование сохраняет диапазон изменения компонент **RGB**, но диапазон изменения компонент **U** и **V** больше, чем у **Y**, что не удобно для кодирования и передачи сигнала / данных. Поэтому вводится нормировка.



Цветовая плоскость U-V (при Y=0.5), представленная в цветовой гамме RGB



# Преобразование форматов и сжатие данных



# Преобразование формата данных

- необходимо для **обмена** изображений между различными программами (графическими системами).
- Преобразование **растровых** данных возможно без потери информации.
- Преобразование **векторных** данных в растровые - растеризация - неизбежно связано с потерей информации - переход от *непрерывного* к *дискретному* описанию.
- Обратная задача - **трассировка** изображения - реализуется с тем или иным качеством в различных системах.
- Результат преобразования зависит от *алгоритмов* векторизации и *характера* изображения.





# Сжатие данных

- **Сжатие** - процесс уменьшения физического размера блока данных.
- Необходимо в связи с большим размером файлов..
- Каждый графический формат поддерживает свой (свои) **алгоритмы сжатия**.
- **Физическое сжатие** данных- реализуется без учета содержащейся в них информации.
- **Логическое сжатие** - основано на логическом анализе информации. Для графических данных не применяется.
- **Сжатие с потерями информации** - после распаковки произошло изменение данных. Часть информации была отброшена для достижения большей степени сжатия.
- **Сжатие без потерь информации** - после распаковки данные соответствуют исходной информации.

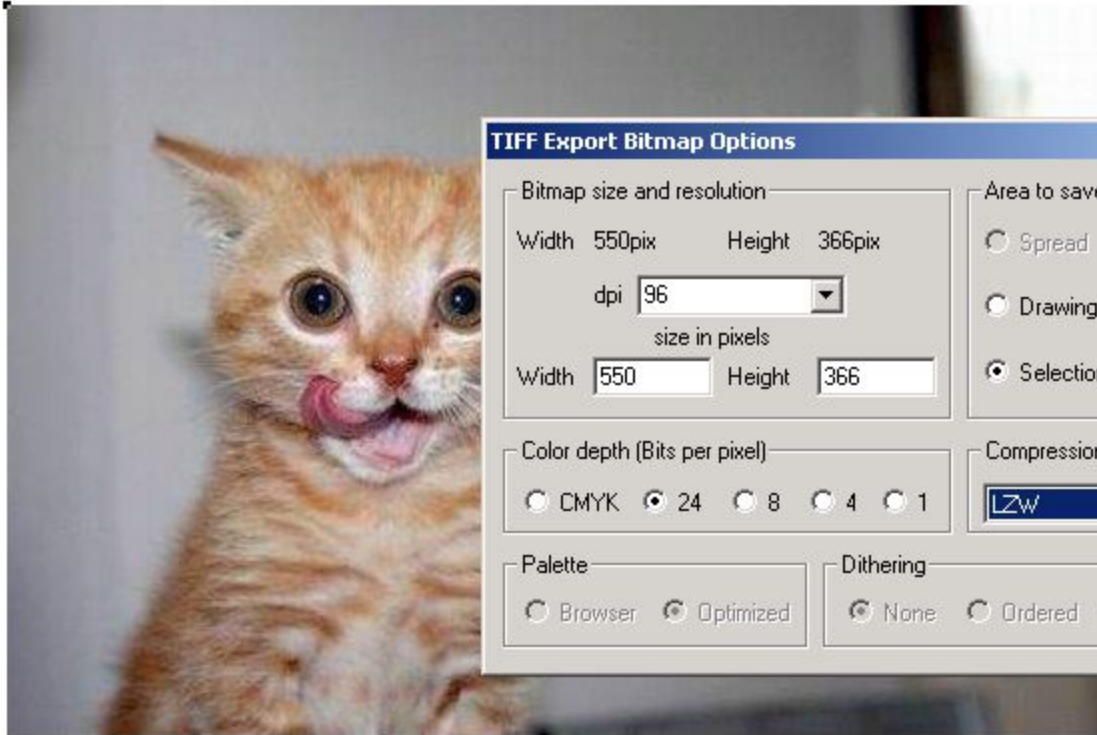


# Методы сжатия

- **Упаковка пикселов** - компактная запись пикселов с глубиной 1, 2, 4 бита в 8-битовые байты, соответственно, по 8, 4, 2 штуки;
- **Групповое кодирование** - (Run-Length Encoding, RLE) - Общий алгоритм кодирования. Применяется в BMP, TIFF, PCX;
- **Алгоритм Lempel-Ziv-Welch (LZW)** - применяется в форматах GIFF и TIFF;
- **Алгоритм JPEG** - разработан для фотографий и включает набор методов сжатия;
- **Фрактальное сжатие** - кодирование растровых изображений в совокупность математических данных, описывающих фрактальные (схожие, повторяющиеся) свойства изображения.
- Обычно сжимают растровые изображения. векторные небольшие и почти несжимаемые.
- Алгоритмы сжатия не задают формат, а определяют способ кодирования данных.



# Алгоритм сжатия Lempel-Ziv-Welch



**TIFF Export Bitmap Options**

Bitmap size and resolution

Width 550pix Height 366pix

dpi 96

size in pixels

Width 550 Height 366

Area to save

Spread

Drawing

Selection

Export

Cancel

Help

Color depth (Bits per pixel)

CMYK  24  8  4  1

Compression

LZW

Palette

Browser  Optimized

Dithering

None  Ordered  Error diffusion



# Список использованных ИСТОЧНИКОВ

1. <https://pechatnick.com/terms/bitmap-bitovaya-karta>
2. Dictionary of Computing, 2-nd Edition, Oxford University Press, 1985.
3. [https://developers.google.com/speed/webp/docs/c\\_s\\_tudy](https://developers.google.com/speed/webp/docs/c_s_tudy) - Comparative Study. Comparative study of WebP, JPEG and JPEG 2000

