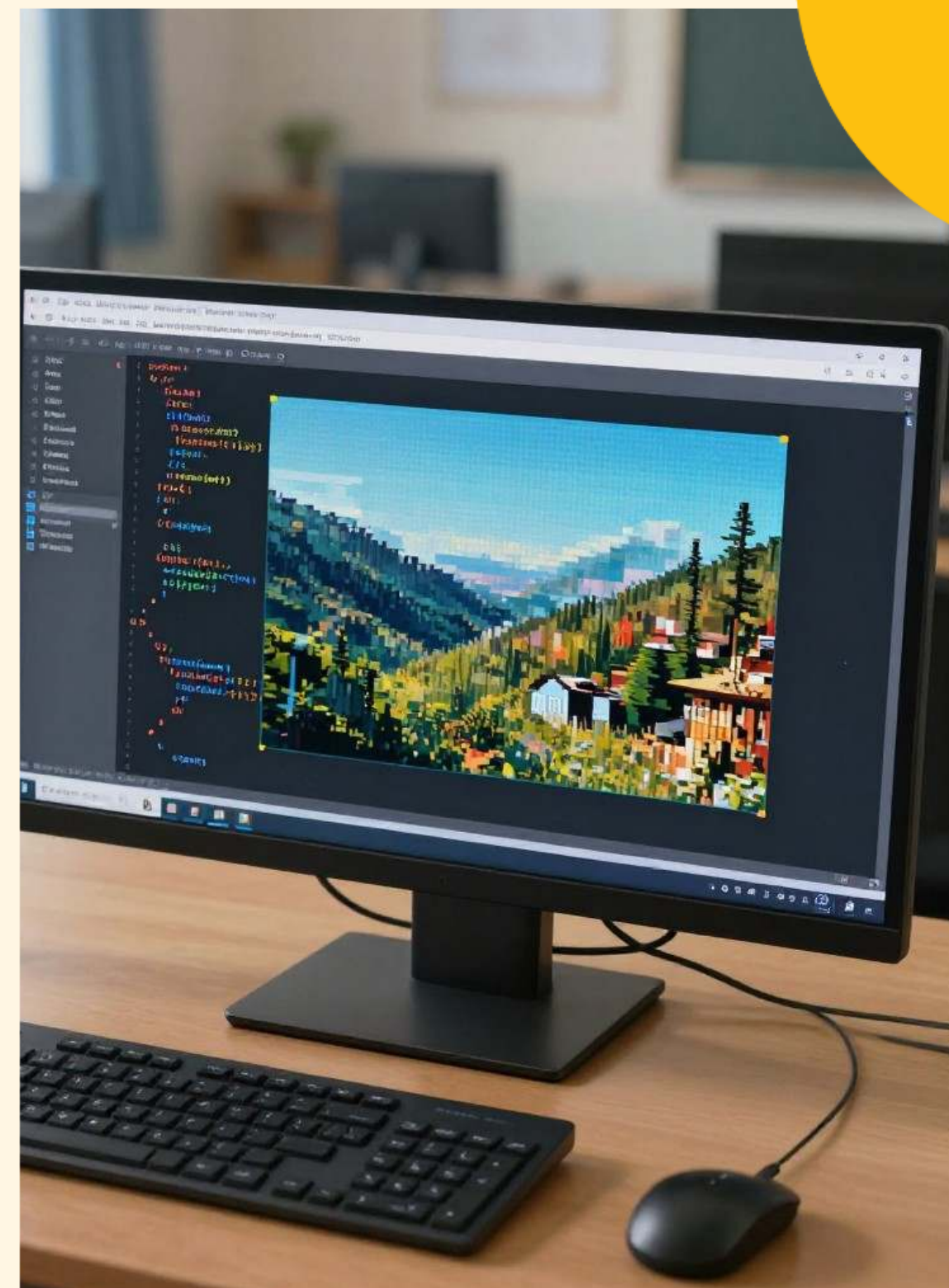


Кодирование графической информации: от пикселя до файла

Разберём, как кодирование задаёт качество изображения, объём файла и удобство работы с ним.



Почему графика «внутри компьютера» всегда становится данными

Компьютерная графика живёт в экранах, интернете, играх, презентациях, печати и цифровых фото. Чтобы изображение можно было сохранить и передать, его кодируют по правилам. Итоговый размер зависит от пикселей, числа цветов и сжатия.



Пиксель как минимальная единица растрового изображения

- Экран собирает картинку из пикселей — небольших точек с определённым цветом. У каждого пикселя есть место в сетке и числовое значение, которое описывает его оттенок.

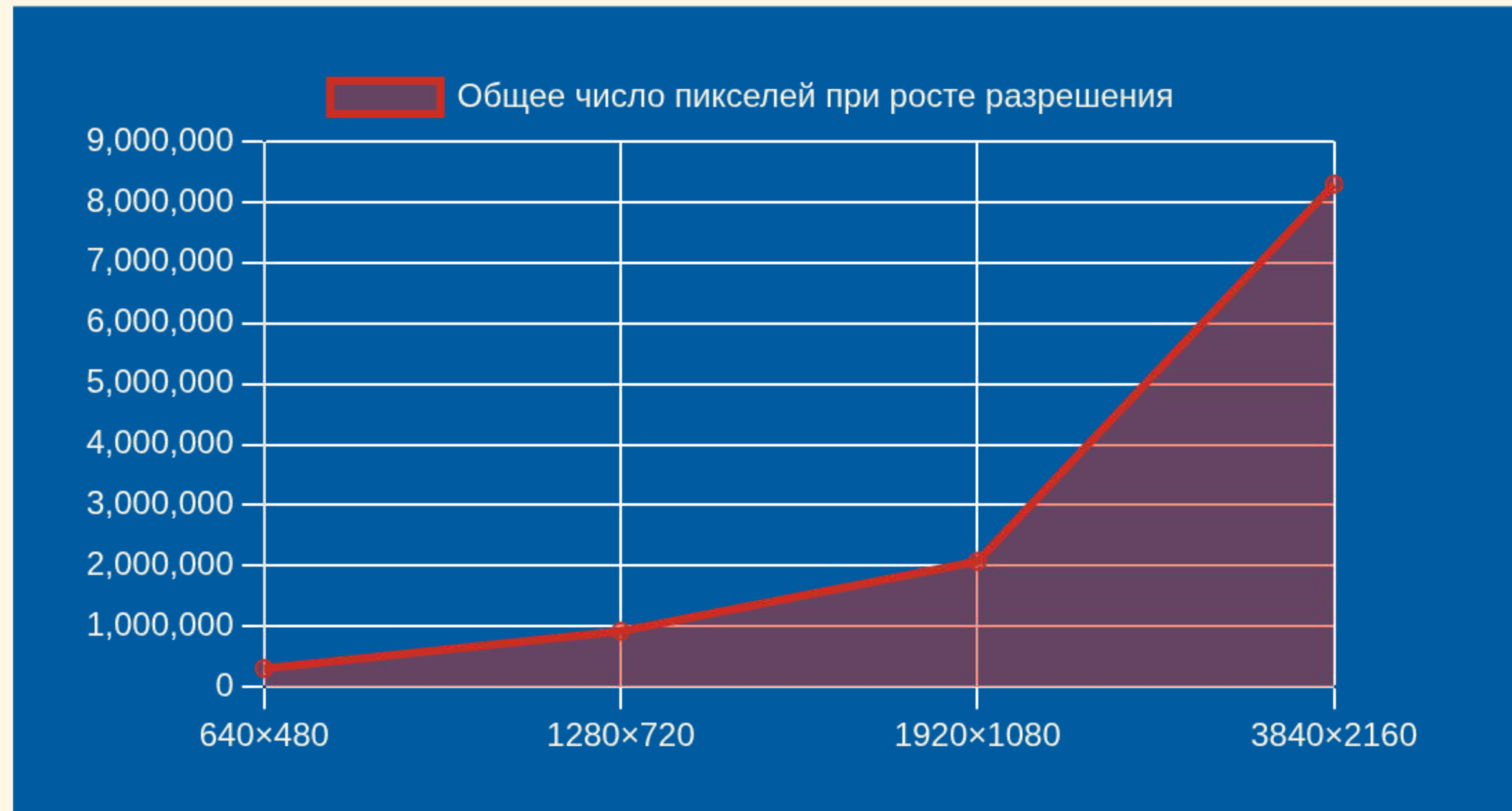
- Разрешение показывает, сколько пикселей размещается по горизонтали и вертикали. Чем их больше, тем выше детализация: мелкие объекты и текст становятся читаемее, а границы — точнее.

- Если пикселей недостаточно, контуры выглядят «ступенчатыми». При сильном увеличении растровое изображение разъезжается по пикселям: вместо гладких линий видны отдельные квадраты и размытие.

Как растёт количество пикселей при увеличении разрешения

При той же глубине цвета объём информации растёт прямо пропорционально числу пикселей: больше точек — больше данных для хранения и передачи.

Переход к более высокому разрешению повышает детализацию, но увеличивает поток данных; поэтому даже «качественная картинка» требует ресурсов.



Расчёт: ширина×высота (значения приведены в условии, без учёта служебных данных файла).

Глубина цвета и число возможных оттенков

Глубина цвета определяет, сколько разных значений может храниться для одного пикселя. Чем больше разрядность, тем плавнее градиенты и выше реалистичность.

С ростом глубины цвета увеличивается количество оттенков, но вместе с ним растёт и объём данных на пиксель, а значит — и размер файла.

Глубина	Возможных цветов
1 бит	2
4 бита	16
8 бит	256
16 бит	65 536
24 бита	16,7 млн
32 бита	24 бита + альфа

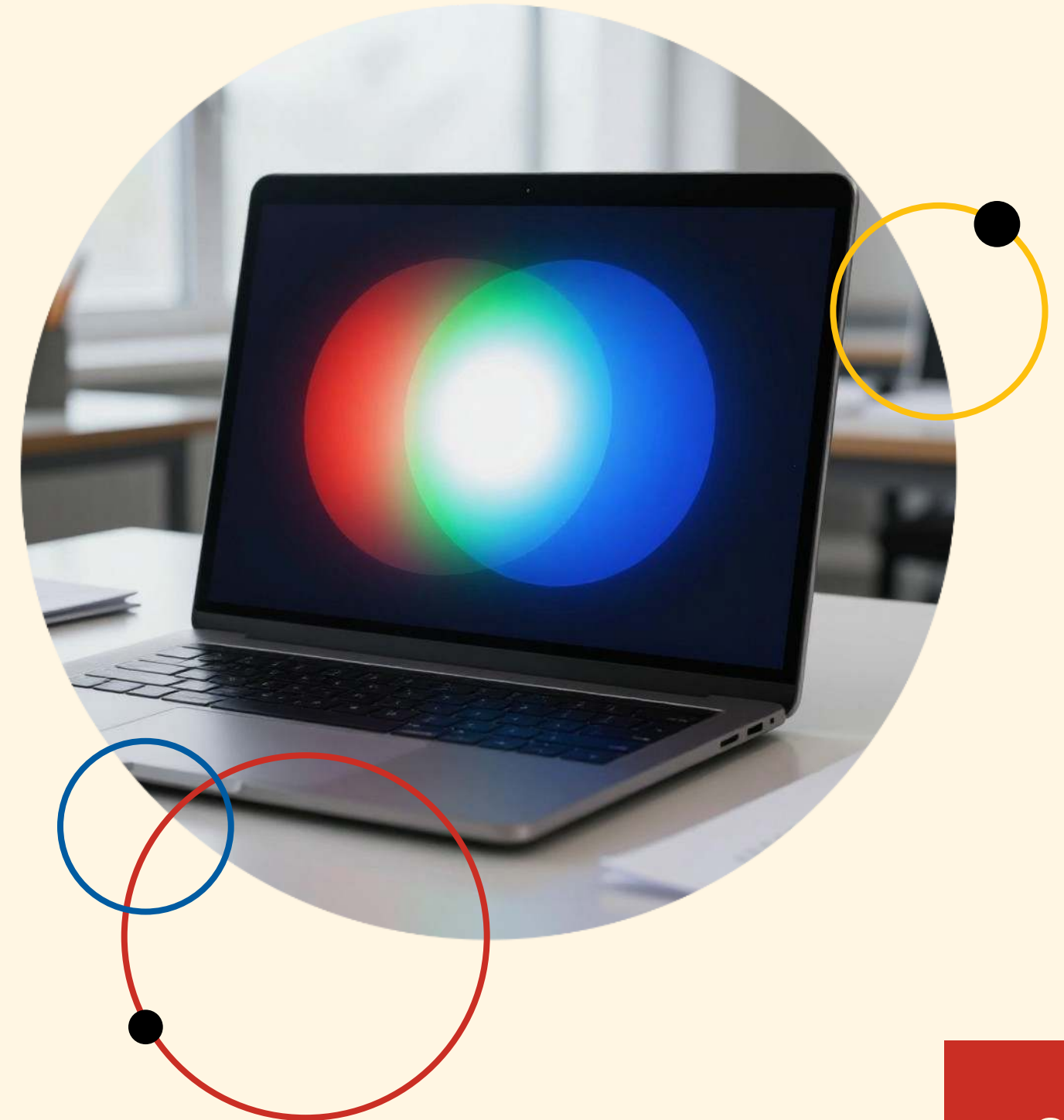
Цветовые модели RGB и CMYK

RGB для экранов

RGB основана на сложении красного, зелёного и синего света. При изменении интенсивности каждого канала формируются разные оттенки; максимум даёт белый цвет. Эта логика подходит дисплеям, потому что они излучают свет.

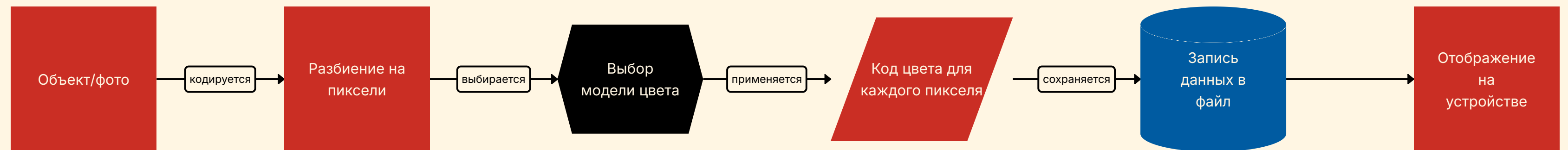
CMYK для полиграфии

CMYK используют в печати: голубой, пурпурный, жёлтый и чёрный пигменты работают с отражённым светом. Поэтому один и тот же исходный цвет на мониторе и на бумаге визуально различается, и требуется настройка цветопередачи.



Как изображение превращается в цифровой файл

Цепочка кодирования растровой картинки: от объекта до данных пикселей в файле.



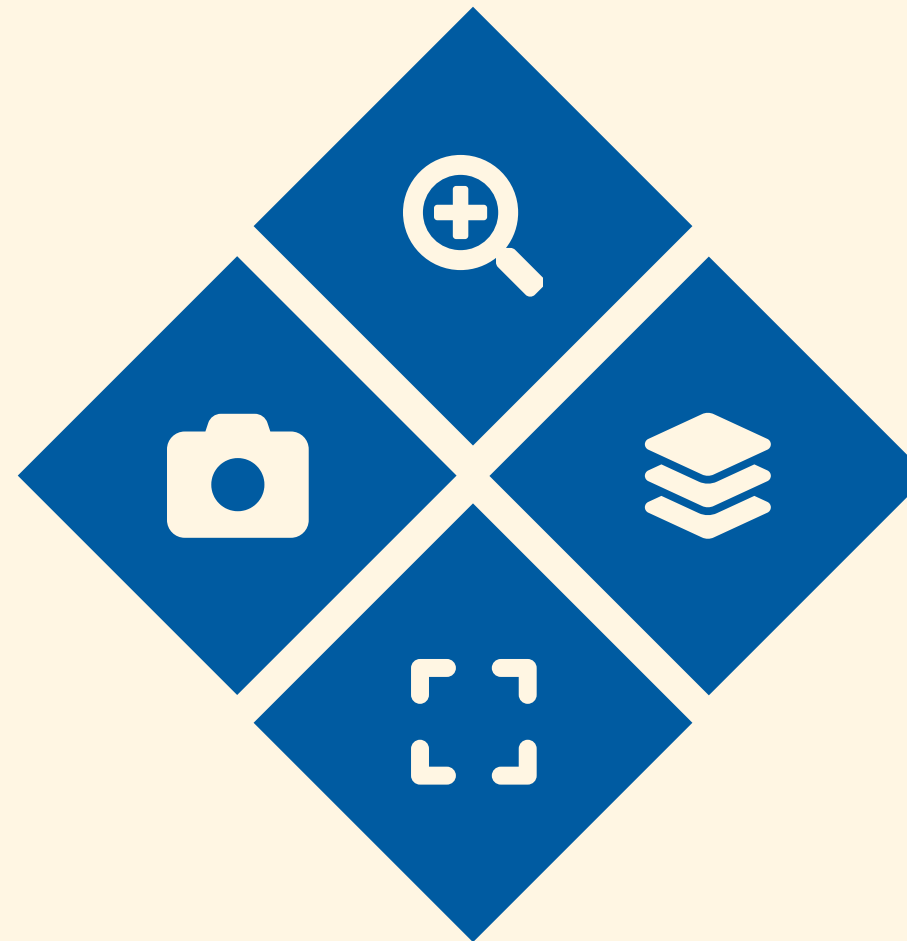
Растровая графика: особенности, преимущества и ограничения

При масштабировании растровая графика пересчитывается «по пикселям», поэтому часть информации усредняется. В итоговой картинке появляются размытость и визуальные артефакты, особенно заметные на резких границах.

Потеря качества при масштабировании

Сильная сторона — фото и полутона

Растровые изображения хорошо передают реальные сцены: тонкие различия цветов, мягкие тени и плавные переходы. Это особенно важно там, где глаз реагирует на мелкие градации оттенков.



Рост объёма при повышении качества

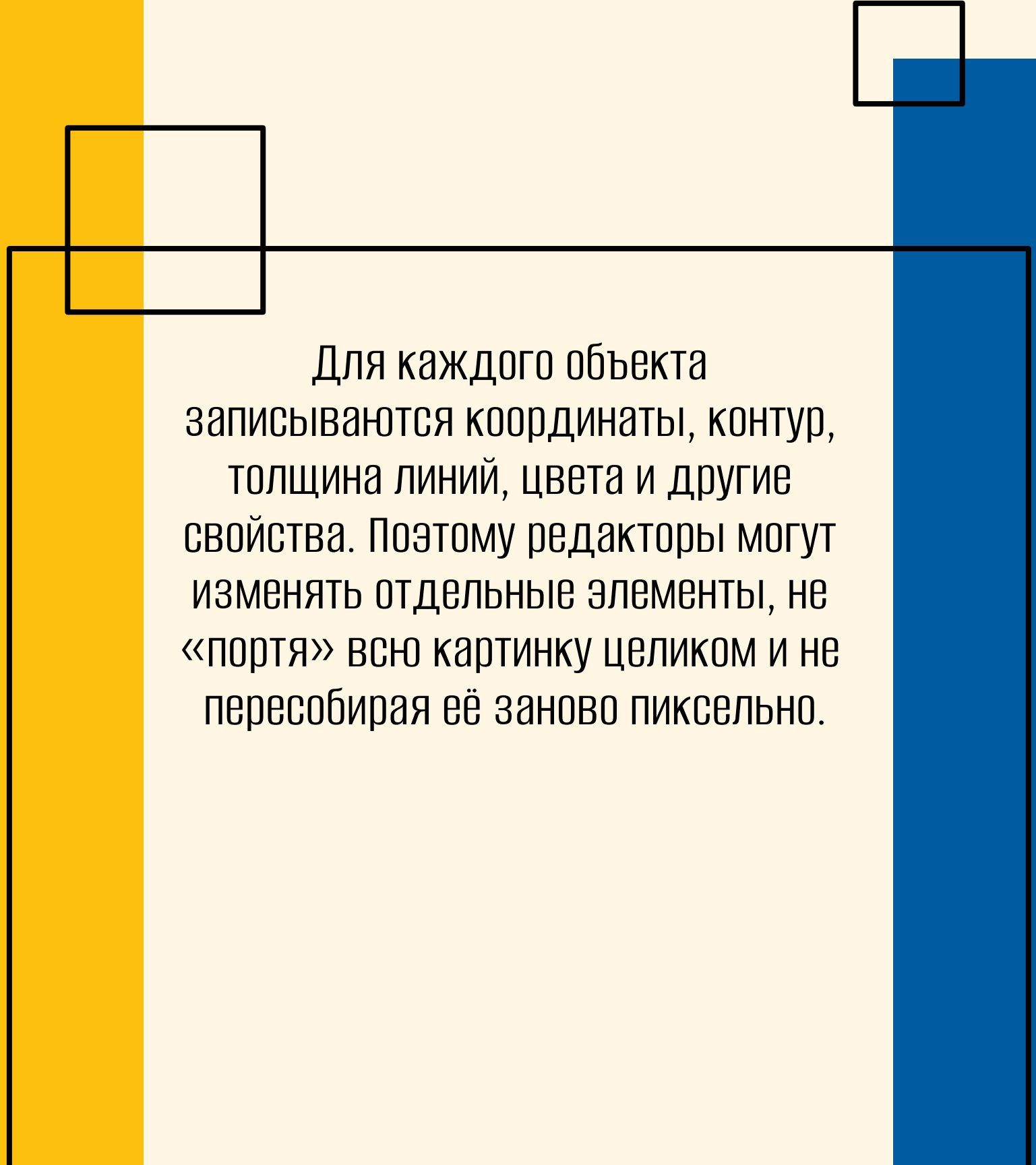
Чтобы улучшить детализацию, нужно добавлять пиксели и часто увеличивать глубину цвета. В результате возрастает количество данных, а вместе с этим — и размер файла, что влияет на хранение и скорость передачи.

Зависимость от разрешения

Чёткость растровой картинки определяется количеством пикселей. Если увеличивать изображение без запаса по разрешению, контуры становятся ступенчатыми, а детальность теряется, даже при аккуратной обработке.

Векторная графика и ее способ кодирования

Векторное изображение описывается геометрией: линиями, отрезками, окружностями, кривыми, многоугольниками и областями заливки. В файле хранятся параметры фигур, а не заранее готовые точки.



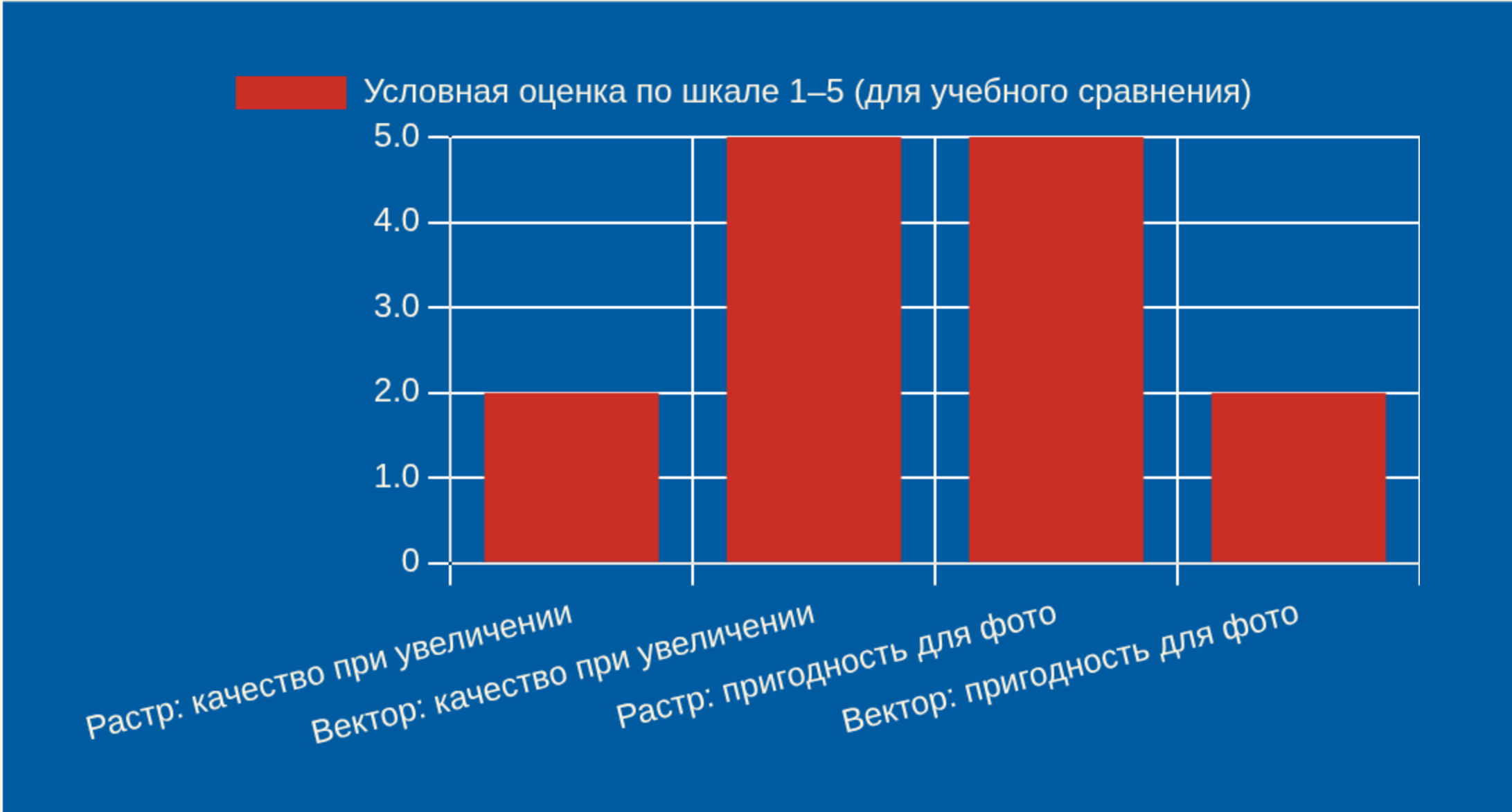
Для каждого объекта записываются координаты, контур, толщина линий, цвета и другие свойства. Поэтому редакторы могут изменять отдельные элементы, не «портя» всю картинку целиком и не пересобирая её заново пиксельно.

При изменении масштаба векторная графика остаётся резкой: её контуры пересчитываются по формулам. Такой подход особенно удобен для логотипов, схем, значков, чертежей и текстовых элементов.

Сравнение растровой и векторной графики

Результат зависит от того, что важнее:
точная передача фотографий или удобная
правка отдельных элементов с сохранением
чёткости при масштабировании.

Если нужно работать с логотипами, схемами и
подписями — чаще выбирают вектор. Для
реалистичных изображений и сложных
полутонов выигрывает растровая графика.



Оценка и интерпретация свойств типов графики на основе их кодирования;
шкала 1–5 не является измерением в конкретных условиях (2026).

Основные форматы графических файлов



JPEG

JPEG использует сжатие с потерями и поэтому особенно эффективен для фотографий. Он хорошо работает, когда важны реальные оттенки и плавные полутона, а небольшой компромисс в точности допустим.



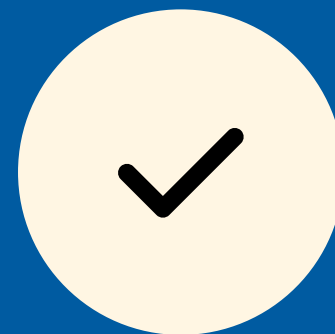
PNG

PNG сохраняет данные без потерь и умеет прозрачность. Формат удобен для схем, скриншотов и элементов интерфейса, где критичны четкие границы, тонкие линии и корректные полупрозрачные области.



GIF

GIF хранит изображение с ограниченной палитрой в 256 цветов и поддерживает простую анимацию. Его выбирают, когда важна «легкость» файлов и анимация не требует большого числа оттенков.



SVG

SVG — это векторный формат, в котором хранятся фигуры и их параметры. Он хорошо подходит для логотипов, диаграмм и значков, потому что масштабирование не превращает линии в «ступеньки» пикселей.



BMP

BMP обычно сохраняет изображение почти без сжатия, поэтому файлы получаются крупными. Формат используют редко, зато он полезен как «эталон» для понимания, сколько места занимают пиксели без экономии.

Сжатие графики: без потерь и с потерями

Без потерь — это восстановление исходных данных полностью. С потерями — уменьшение размера за счет частичного «упрощения» изображения, когда некоторые детали становятся менее точными.

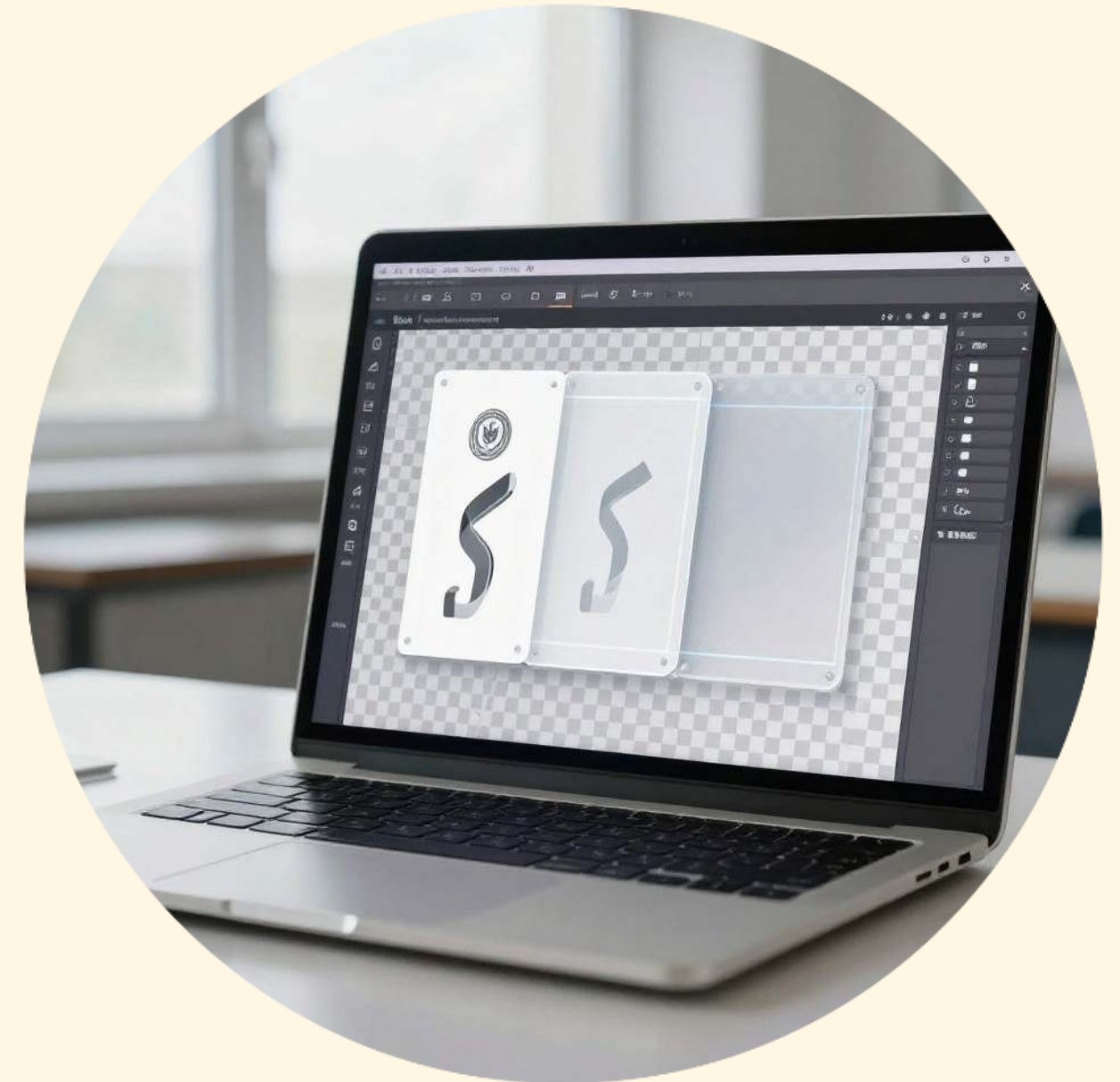
Если важны точность, текст и резкие границы — выбирают без потерь. Если нужен компактный файл для фотографий — чаще подходят потери: экономия заметна, но качество ограничено.

Метод	Форматы	Снижение размера	Качество	Прозрачность
Без потерь	PNG, GIF, TIFF	Среднее	Сохраняется	Обычно поддерживается
С потерями	JPEG, WebP	Больше	Может ухудшаться	Зависит от формата

Обобщение свойств форматов по спецификациям и практическому применению (учебные материалы по компьютерной графике).

Альфа-канал и прозрачность изображения

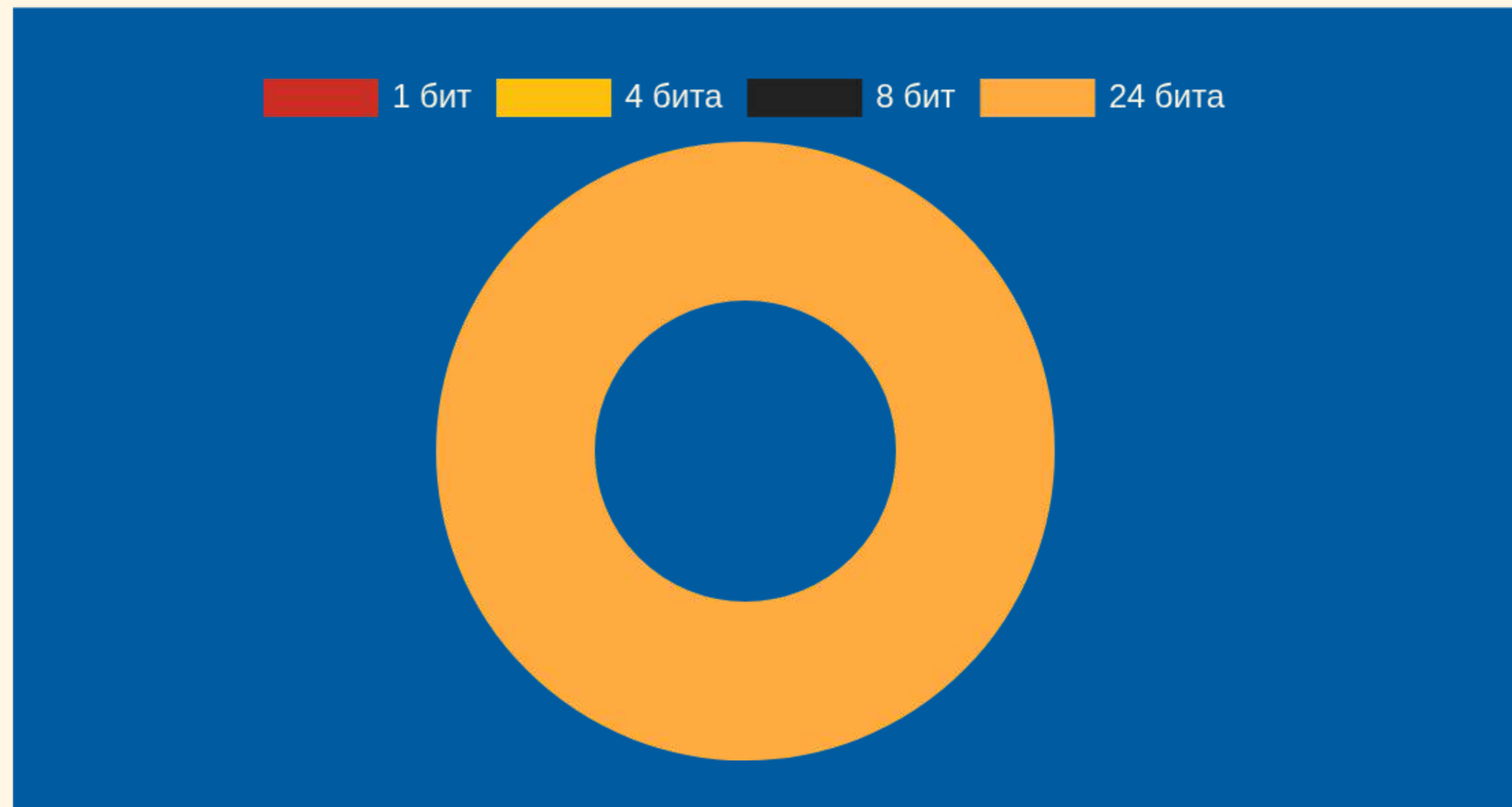
- Альфа-канал добавляет к цвету пикселя дополнительную величину прозрачности: от полностью непрозрачного до полностью прозрачного. Промежуточные значения позволяют делать пиксели частично видимыми, создавая мягкие наложения.
- Альфа-канал применяется в графических редакторах, интерфейсах и при работе со слоями. Он нужен значкам, коллажам и композициям, где важно корректно смешивать объекты друг с другом без «грязных» фонов, поэтому особенно актуален для PNG.



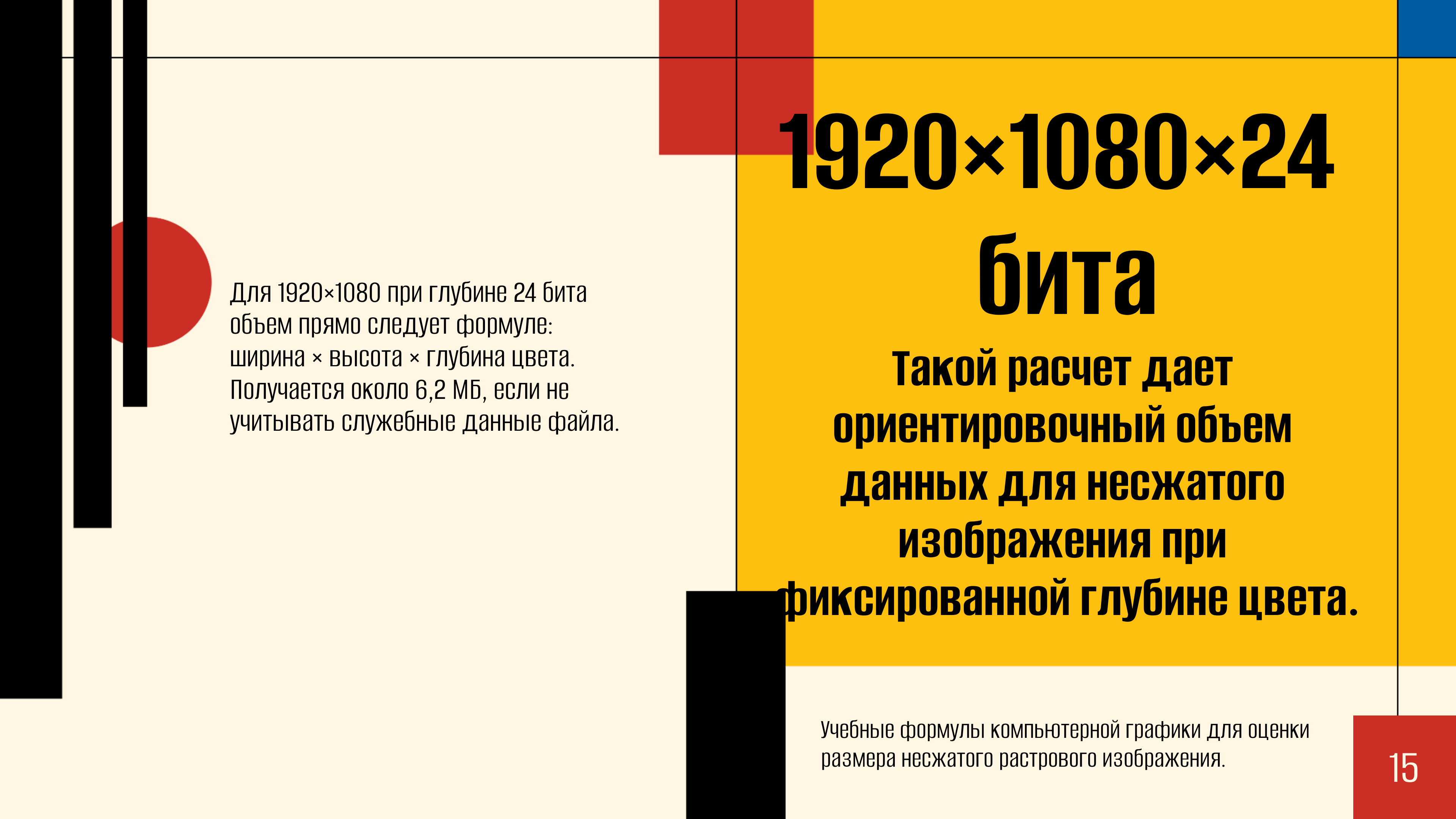
Как глубина цвета влияет на число оттенков

При увеличении разрядности палитра расширяется, а градиенты становятся плавнее. Это помогает уменьшить заметные полосы, появляющиеся при ограниченном числе оттенков.

Данные показывают резкий рост числа вариантов цвета при переходе от 8 бит к 24 битам. Практически это означает более реалистичные плавные переходы, особенно в изображениях с градиентами.



Школьные сведения о глубине цвета и количестве оттенков: 2^n возможных значений.



1920×1080×24 бита

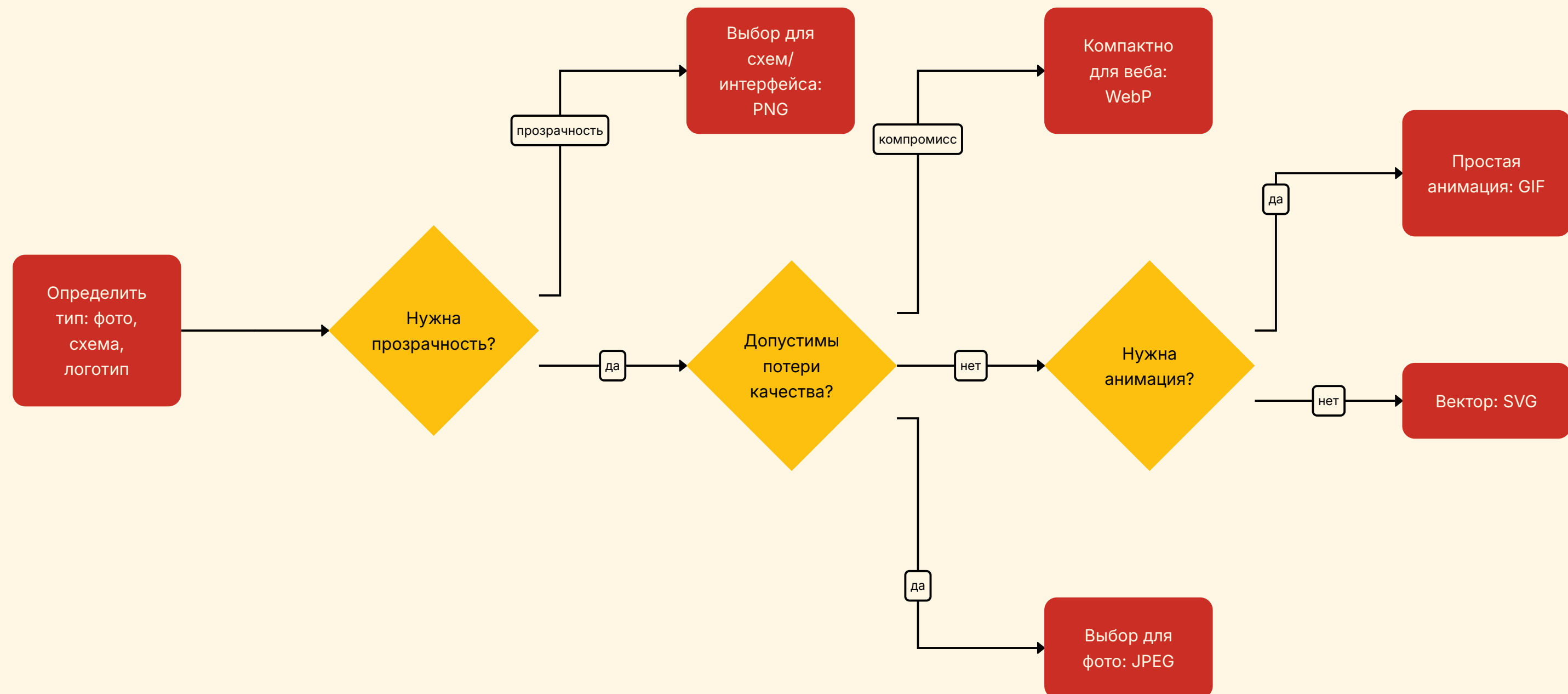
Для 1920×1080 при глубине 24 бита
объем прямо следует формуле:
ширина × высота × глубина цвета.
Получается около 6,2 МБ, если не
учитывать служебные данные файла.

**Такой расчет дает
ориентировочный объем
данных для несжатого
изображения при
фиксированной глубине цвета.**

Учебные формулы компьютерной графики для оценки
размера несжатого растрового изображения.

Как выбирают формат изображения для конкретной задачи

Логика выбора формата: фотография или схема, нужна прозрачность, допустимость потерь, размер и анимация.



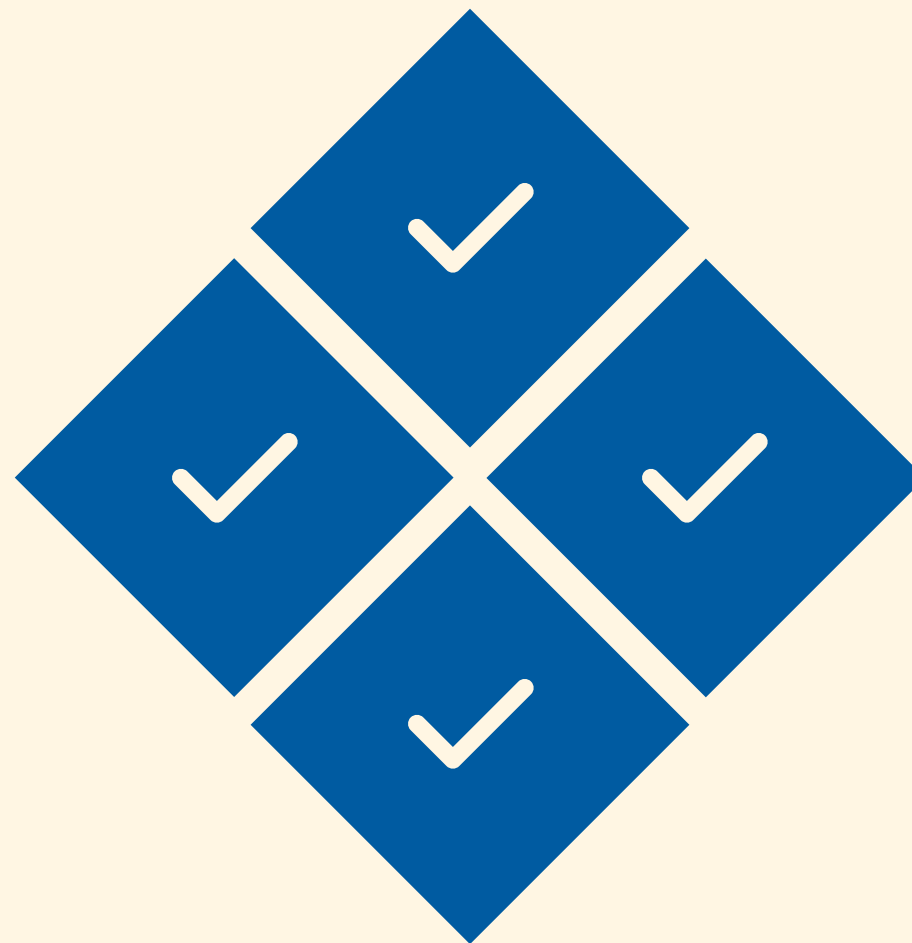
Где применяется кодирование графической информации

В медицинской визуализации важны точные детали и стабильность отображения. В картографии ценятся четкие линии, подписи и удобная масштабируемость, чтобы информация читалась при разных масштабах.

Медицина и картография

Цифровые камеры и смартфоны

Кодирование нужно, чтобы сохранить фото в файл и передать его дальше. Важны детализация и корректная обработка оттенков, ведь от этого зависит воспринимаемое качество снимков.



Компьютерные игры

В играх изображения обрабатываются в реальном времени, поэтому важна скорость декодирования. Кодирование помогает удерживать приемлемый объем памяти и время рендеринга при визуальных эффектах.

Веб-сайты и потоковая передача

В интернете файлы должны быть достаточно компактными, чтобы загружаться быстро. Поэтому выбор формата и сжатия напрямую влияет на скорость просмотра и удобство для пользователей.

Типичные искажения при кодировании и обработке графики

Проблемы масштаба и видимые дефекты

При сильном увеличении появляется пикселизация: изображение становится «квадратным», потому что видны отдельные элементы сетки. При масштабировании в другую сторону возможно размывание, когда детали пересчитываются на основе соседних пикселей.



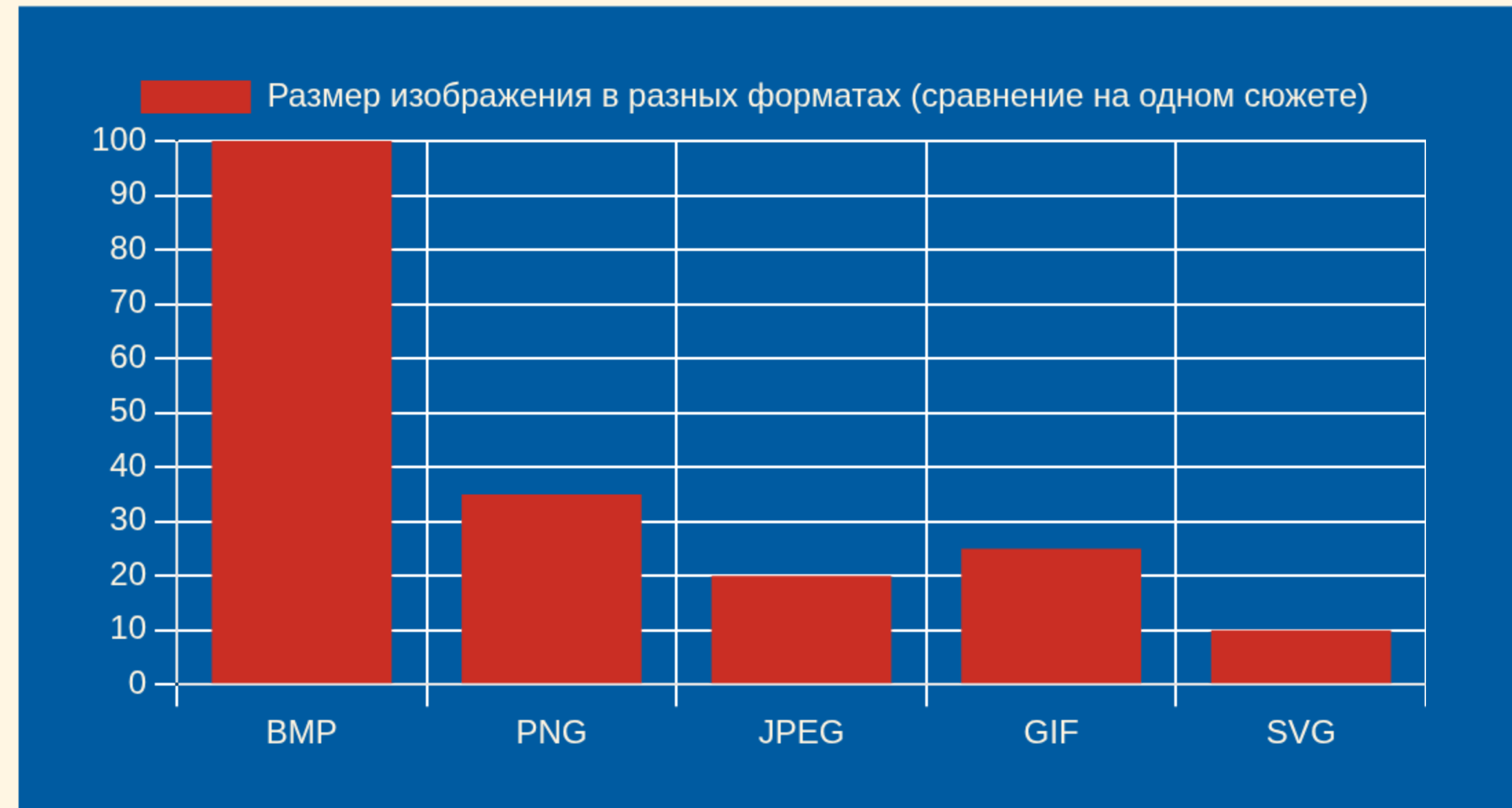
Артефакты и цветовые несовпадения

После интенсивного сжатия JPEG заметны артефакты — особенно на резких границах и мелких текстурах. Кроме того, при разной глубине цвета и различиях цветовых профилей устройства оттенки могут выглядеть иначе, чем ожидалось.

Как меняется размер одного и того же изображения в разных форматах

Сравнение показывает типичную тенденцию: несжатый BMP почти всегда крупнее, а векторный SVG может быть очень компактным для линий и простых фигур.

Компактность определяется не только форматом, но и характером изображения. Фотографии лучше «сжимаются» алгоритмами с потерями, а схемы часто выгоднее хранить как векторные данные.



Сравнение типичных размеров по практике использования форматов; точные значения зависят от конкретного изображения и параметров экспорта.

Главный вывод: кодирование графики определяет качество, размер и удобство использования

Кодирование задает, как именно изображение хранится и передается: пикселями или геометрическими объектами. На результат влияют разрешение, глубина цвета, модель, формат и сжатие, поэтому выбор должен соответствовать задаче.

