

Представление графической информации в компьютере

Разберём, как компьютер кодирует
изображения: пиксели, растр и вектор,
цвет, форматы и сжатие данных.





Почему одной «картинке» недостаточно одного способа хранения

Компьютер не «рисует» глазами: он хранит изображения как числа — и для разных задач нужны разные представления. Фото требуют плавных переходов, а схемы — точных линий и масштабируемости. Компромисс всегда: качество, размер и удобство обработки.



Пиксель как минимальный элемент цифрового изображения

01

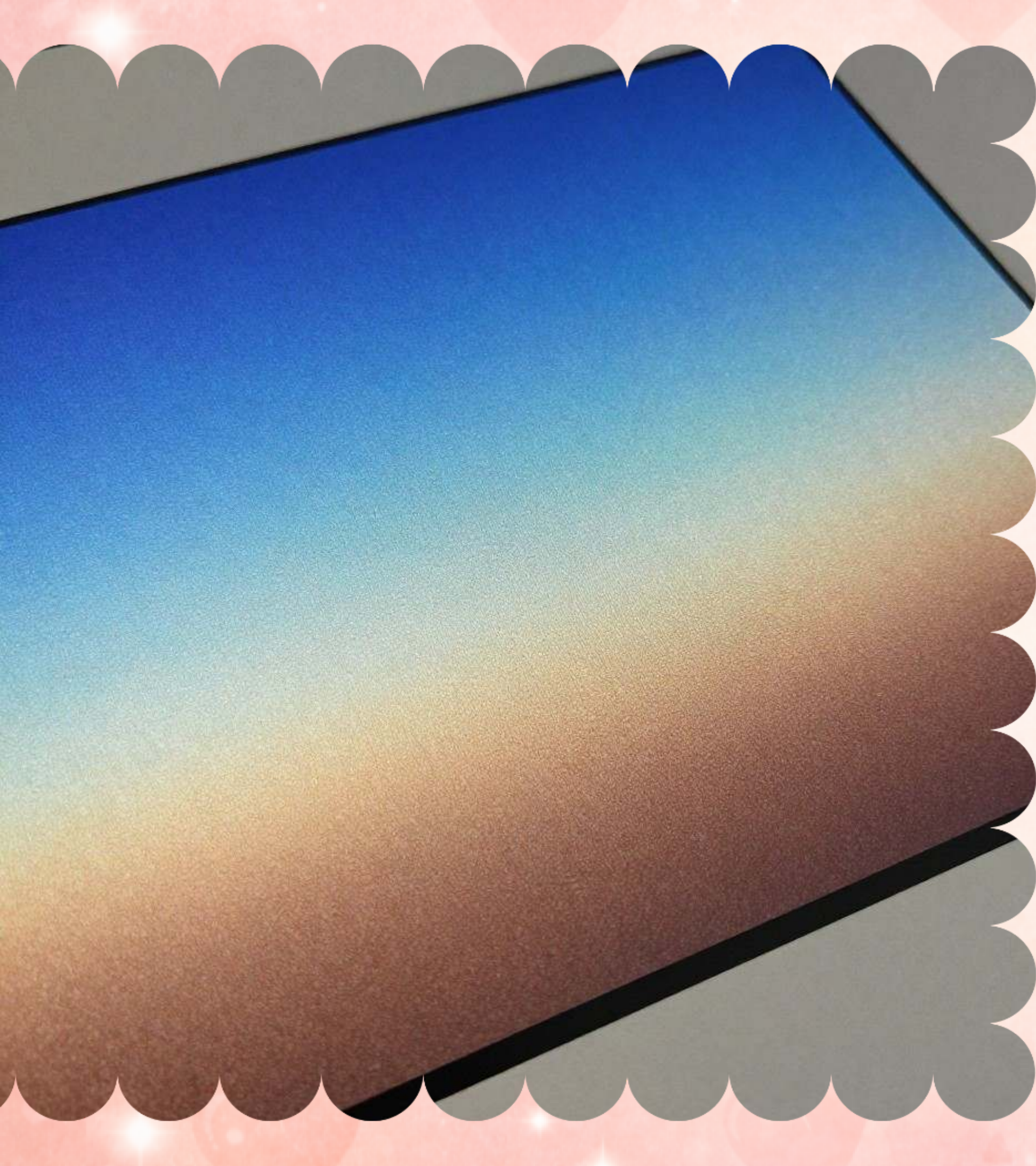
Экранное изображение состоит из прямоугольной сетки пикселей, и каждый пиксель несёт свой код цвета. Он задаёт, какой цвет должен быть в конкретной точке кадра.

02

Размер картинки в пикселях, например 1920×1080 или 1280×720, определяет число точек по горизонтали и вертикали. Чем больше пикселей, тем детальнее становится изображение.

03

При росте количества пикселей увеличивается и объём данных для хранения. Поэтому в реальных задачах всегда приходится балансировать между детализацией и размером файла.



Растровая графика: изображение как массив цветных точек



Растровое изображение хранит цвет каждого пикселя отдельно, поэтому хорошо подходит для фотографий и рисунков с множеством полутонов. Такие изображения естественно передают тени и текстуры.



При сильном увеличении раstra проявляется пикселизация: становятся заметны отдельные квадраты, потому что программа не добавляет новые детали, а растягивает существующие пиксели. Это особенно заметно на градиентах.

Векторная графика: изображение на основе формул и объектов



Векторная графика описывает рисунок геометрическими примитивами: линиями, кривыми, окружностями, многоугольниками и областями заливки. Координаты и параметры задаются математически.



Такой способ хранения удобен для логотипов, иконок, схем, диаграмм, шрифтов и технических чертежей. Важна не «зернистость», а точность контуров и предсказуемый результат при изменениях.



Главное преимущество — отсутствие потери качества при масштабировании: изображение пересчитывается заново под нужный размер, поэтому линии остаются чёткими и пропорции не искажаются.

Сравнение растровой и векторной графики

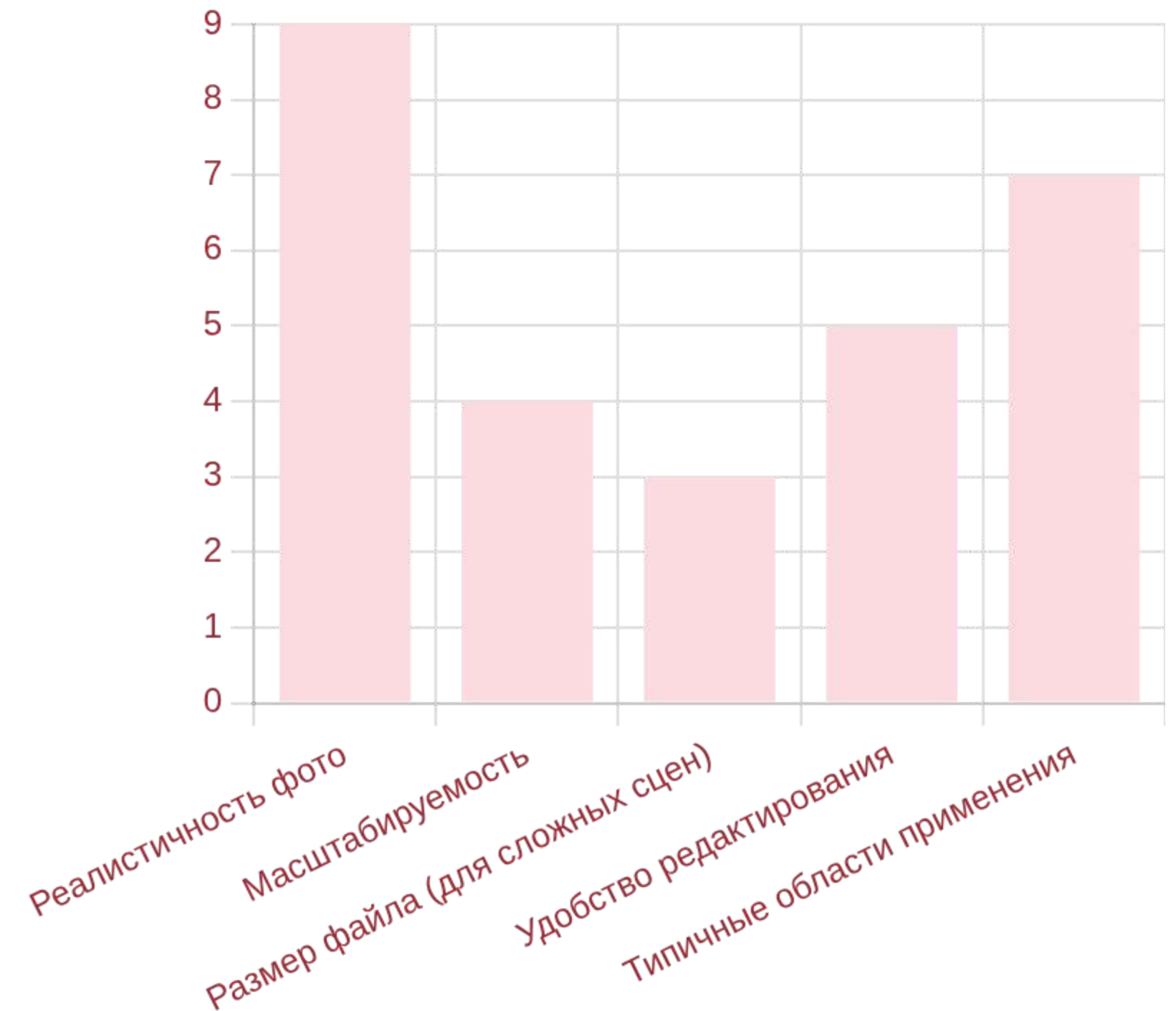


Для растра характерны высокие результаты в передаче фотографических деталей, а для вектора — в операциях с отдельными объектами и в масштабировании без «лесенок».

Диаграмма показывает отсутствие универсального формата: правильный выбор определяется содержанием изображения и тем, как вы планируете его использовать.

Учебные сведения по компьютерной графике (качественное сравнение; численные оценки условны для наглядности).

Сравнение по ключевым параметрам (оценка по 10-балльной шкале)



Разрешение изображения и его роль в детализации

Рост разрешения

Чем выше разрешение, тем больше визуальной информации удаётся сохранить: мелкие элементы, текст и контуры передаются точнее. На практике это влияет на читабельность деталей.

Определение

Разрешение показывает число пикселей в изображении и обычно записывается как ширина×высота (например, 1920×1080). Это базовая «сеточка», на которой строится картинка.

Разные требования

Для фотографии, монитора и печати требования отличаются: одно и то же разрешение по-разному воспринимается в зависимости от способа отображения и масштаба вывода.



Глубина цвета задаёт, сколько бит отводится на кодирование цвета пикселя: чем больше бит, тем больше возможных оттенков и реалистичнее картинка.

Общие сведения из курса по представлению данных и компьютерной графике (расчёт по степеням двойки для битовой глубины: 2^{24}).

16,7 млн ЦВЕТОВ

Такая глубина цвета делает градиенты заметно плавнее и уменьшает риск резких переходов между оттенками.

Как глубина цвета влияет на объем изображения

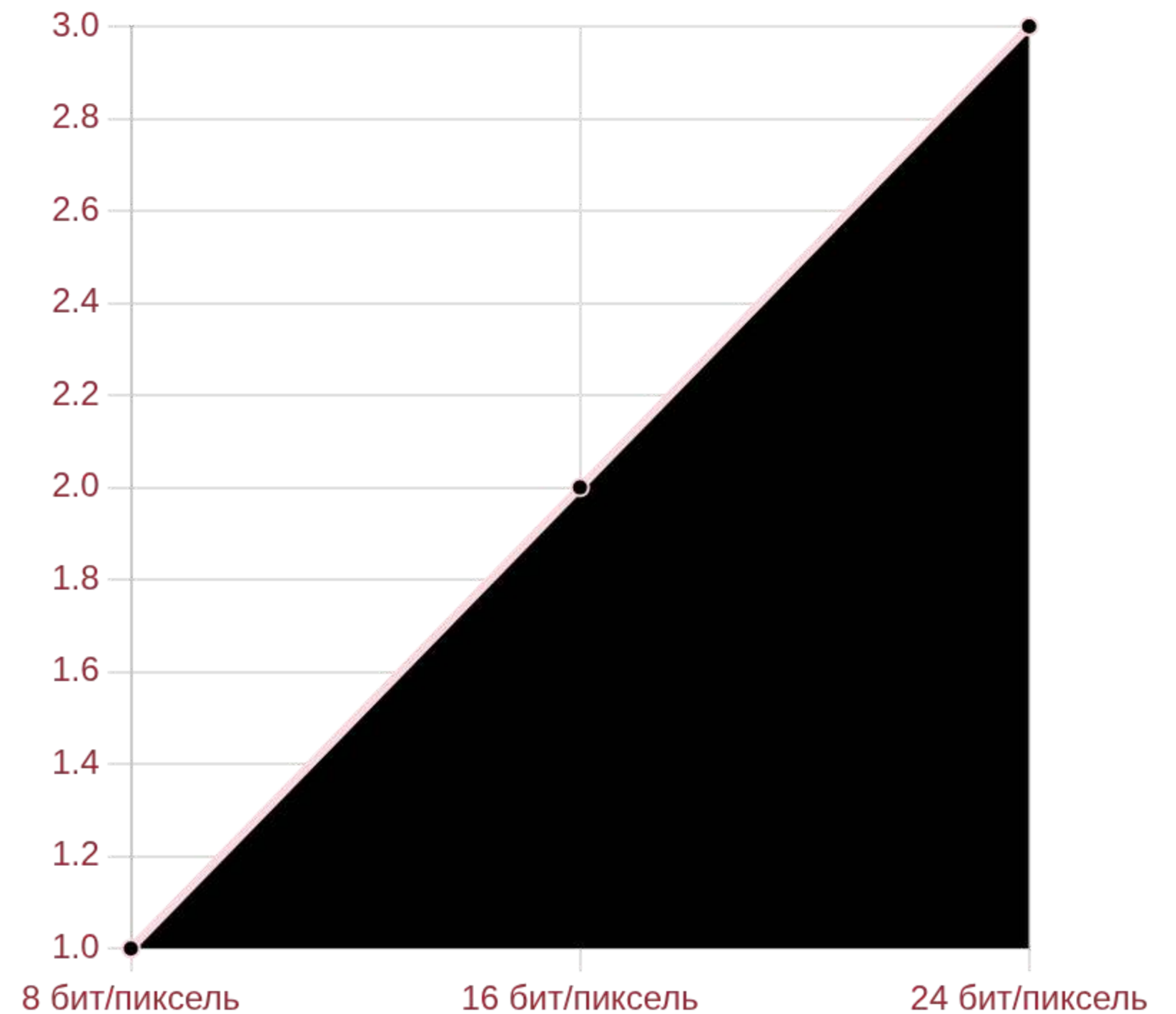


При увеличении глубины цвета растёт количество информации на каждый пиксель, поэтому размер несжатого файла увеличивается почти линейно по числу бит. Это напрямую отражается в хранении и передаче.

Связь «качество — данные» неизбежна: больше оттенков означает больший объём. Практический выбор — под задачу и доступные ресурсы устройства.

Математическая зависимость размера от числа бит на пиксель; числовые значения представлены в относительной шкале.

Почти линейный рост объёма несжатого изображения при фиксированном i



Цветовая модель RGB для экранных устройств

RGB основана на смешении красного, зелёного и синего света. Эта модель подходит для мониторов, телевизоров, смартфонов, камер и веб-графики, потому что такие устройства формируют изображение из излучаемого света.

Промежуточные сочетания значений каналов позволяют получить широкий диапазон оттенков, поскольку человеческий глаз воспринимает итоговый световой сигнал как цветовую композицию.

Если все три канала имеют нулевые значения, получается чёрный цвет; если каналы максимальны – белый. Так строится базовая логика формирования яркости и цвета.

В цифровой технике каждый канал обычно кодируется числом от 0 до 255, то есть цвет записывается как набор трёх параметров. Поэтому RGB удобно представлять в виде двоичных чисел.

Цветовая модель СМΥΚ для печати

- 01** СМΥΚ описывает цвет через четыре канала: голубой, пурпурный, желтый и черный. В полиграфии краски наносятся на бумагу и работают как фильтры света, формируя нужный оттенок при отражении.
- 02** По сравнению с RGB охват СМΥΚ уже, поэтому некоторые яркие цвета с экрана при печати оказываются менее насыщенными. Это важно учитывать при подготовке макетов, особенно для сочных рекламных изображений.
- 03** Черный канал (K) повышает глубину темных участков и помогает точнее передать тени. Дополнительно он снижает расход цветных красок, что делает печать экономичнее и устойчивее по тону.



Формат	Тип графики	Сжатие	Прозрачность	Применение
JPEG	Растр	С потерями	Нет	Фото
PNG	Растр	Без потерь	Да	Интерфейсы
GIF	Растр	Палитра + анимация	Ограниченно	Простая анимация
SVG	Вектор	Без сжатия	Да	Логотипы
BMP	Растр	Почти без	Часто	Файлы большого размера

Популярные форматы графических файлов

Формат определяет, как именно хранится изображение, есть ли прозрачность и как ведет себя размер файла. Выбор стоит делать под задачу: фото, интерфейсы, анимация или графика для печати и масштабирования.

Универсального формата нет: каждый вариант оптимален под свой тип изображения и требования к качеству, прозрачности и размеру файла.

Общие характеристики графических форматов

Сжатие данных: уменьшение объема без потери или с потерями



Сжатие уменьшает объем

Сжатие сокращает размер файлов и ускоряет передачу изображений по сети. Оно также снижает нагрузку на память устройства, что важно при хранении больших медиатек и загрузке на сайты.



Без потерь — сохраняет данные полностью

При сжатии без потерь все исходные данные восстанавливаются точно. Поэтому оно подходит для схем, интерфейсов, текста на изображениях и технической графики, где критичны четкие границы.

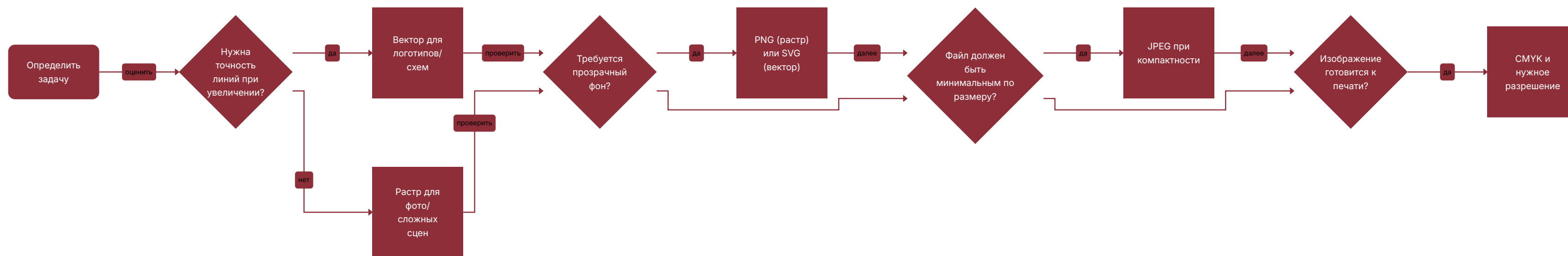


С потерями — больше экономия, но риск артефактов

Сжатие с потерями сильнее уменьшает объем, но часть информации удаляется необратимо. При чрезмерной компрессии в JPEG заметны артефакты и теряются мелкие детали.

Выбор способа хранения изображения по типу задачи

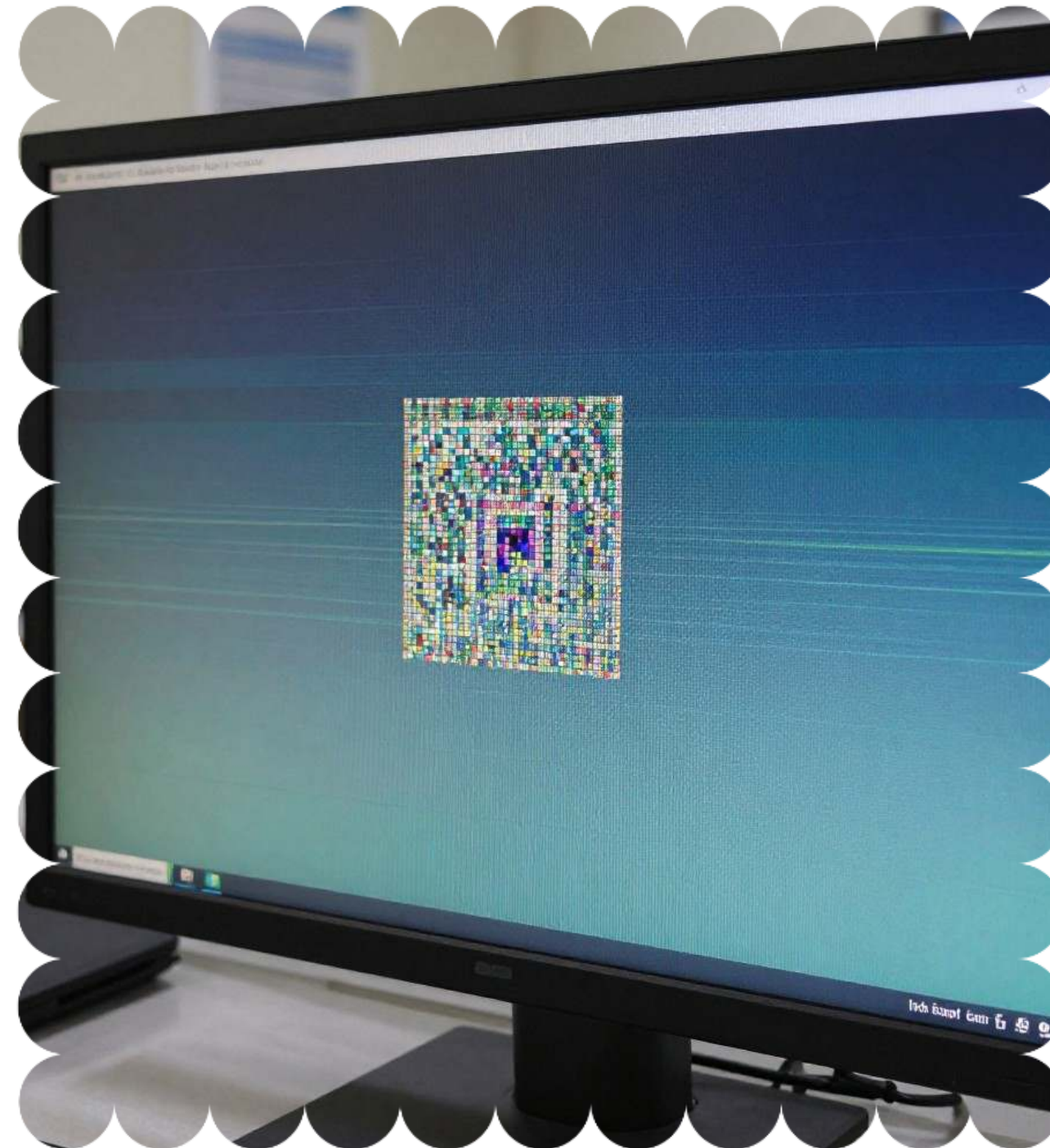
Алгоритм подбора формата: от содержания изображения и требований к качеству — к способу хранения.



Как изображение формируется на экране устройства

Сигнал превращается в массив пикселей

Монитор получает цифровой сигнал от видеокарты, которая формирует кадр как набор пикселей. Каждый пиксель в текущем кадре получает значение цвета и яркости, складывая изображение на экране.



Частота обновления и настройки влияют на качество

Частота обновления измеряется в герцах и показывает, сколько раз в секунду изображение перерисовывается. Качество также зависит от разрешения, цветопередачи и корректного масштабирования.



Пикселизация при увеличении растрового изображения

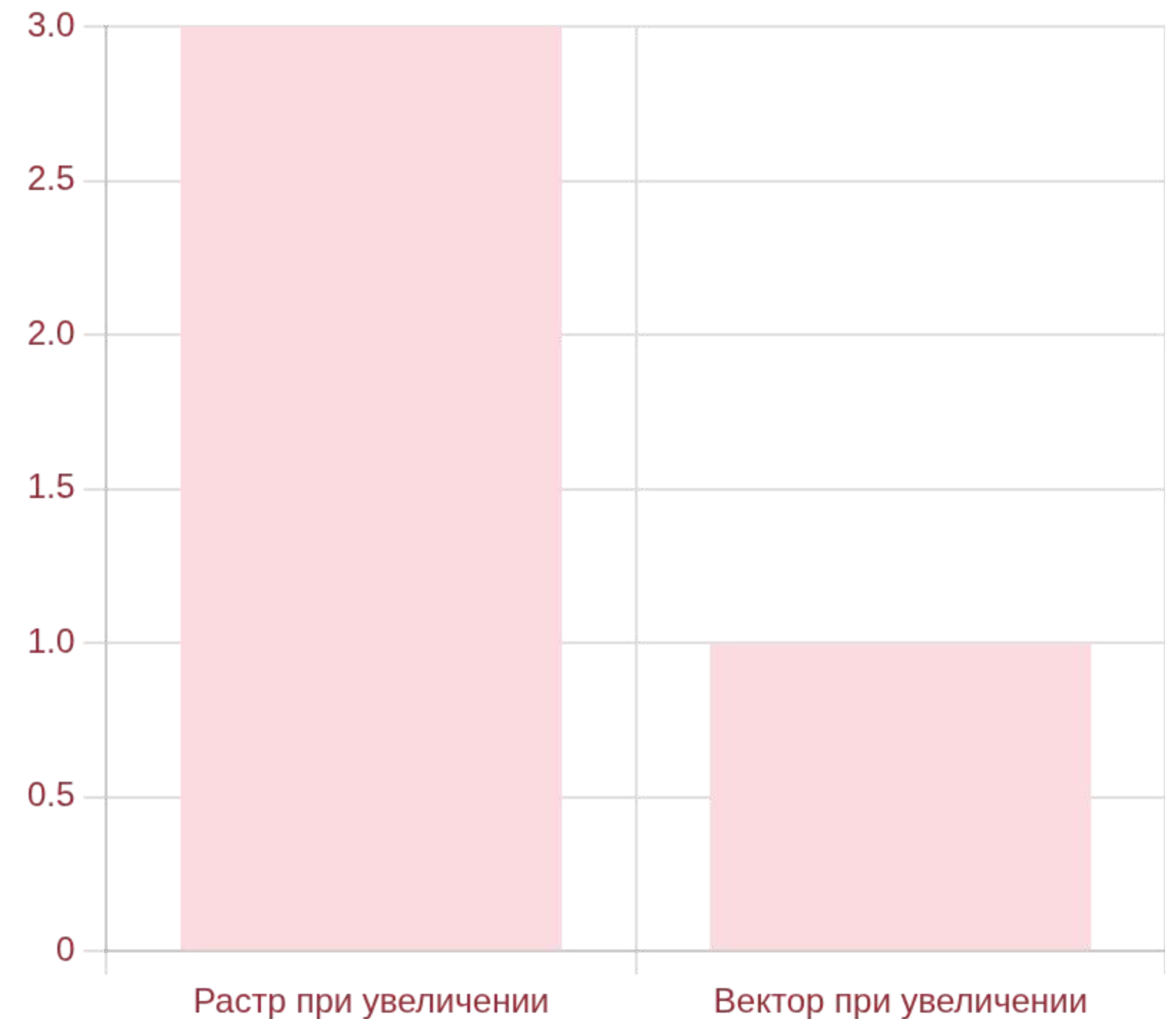


Если исходная картинка растровая, то при увеличении растягиваются существующие пиксели, поэтому границы становятся «ступенчатыми» и контраст мелких деталей снижается.

Сравнение показывает ключевой предел растровой графики: качество при увеличении зависит от исходного разрешения, тогда как вектор пересчитывается и остается четким.

Сопоставление принципов хранения: растр – пиксели, вектор – математические объекты

Как масштабирование выявляет разницу между растром и вектор



Графическая информация в цифровой среде и повседневных технологиях



Игры и 3D-приложения

В играх и трехмерных программах графика должна обновляться быстро. GPU ускоряет обработку кадров, текстур и эффектов, чтобы изображение оставалось плавным и детальным.



Веб-дизайн и интерфейсы

На сайтах и в приложениях используются растровые и векторные элементы: иконки, иллюстрации, схемы. От выбора формата зависит скорость загрузки и качество на разных экранах.



Навигация и визуализация данных

Навигационные сервисы опираются на графические слои: карты, маршруты, отметки. Правильное масштабирование и цветопередача помогают считывать информацию даже при сложном фоне.



Искусственный интеллект для улучшения изображения

Алгоритмы ИИ применяют шумоподавление, улучшение деталей и восстановление изображения. Это повышает полезность графики в обучении, медицине и задачах обработки фото.

Типичные ошибки при работе с графикой

Слишком низкое разрешение приводит к размытию и потере мелких деталей. Тогда даже при аккуратном кадрировании объект выглядит нечетко, а текст и контуры становятся трудно читаемыми.

Неправильный выбор формата может увеличить размер файла или ограничить редактирование. Например, векторные элементы удобнее для масштабирования, а растр лучше подходит для фотографий и сложных сцен.

Перевод из RGB в CMYK без учета цветового охвата меняет оттенки и уменьшает яркость некоторых цветов. В результате макеты могут выглядеть иначе на печати, чем на экране.

Чрезмерное сжатие вызывает артефакты, а чрезмерное увеличение растрового изображения делает пиксели заметными. Качество зависит от набора параметров, а не от одного «внешнего» результата.

Параметры изображения и их влияние на результат

Параметр	Что определяет	Эффект при увеличении	Риск при ошибке	
Разрешение	Детализация	Рост резкости	Размытость	
Глубина цвета	Плавность	Меньше полос	Бэндинг	
Формат	Хранение	Зависит от типа	Несовместимость	
Сжатие	Размер/ качество	Артефакты	Ухудшение деталей	
Цветовая модель	Точность цвета	Разные оттенки	Смещение цветов	

Каждый параметр изображения отвечает за свой аспект качества. Ошибка в одном параметре может проявиться в другом: например, изменение формата влияет на совместимость, а сжатие — на детализацию.

Качество — результат взаимодействия параметров: нужно подбирать их под задачу, а не только «улучшать картинку» на глаз.



Главный вывод: графическое изображение в компьютере — это структурированные данные

Графическая информация в компьютере хранится как кодирование цвета, координат, разрешения, формата и сжатия. Растр и вектор решают разные задачи, а RGB и CMYK обслуживают разные среды — экран и печать. Понимание различий помогает сохранять качество и рационально использовать ресурсы.