Изображения

Итак, начинаю исследование с изучения форматов изображений, используемых в веб-приложениях.

На выбор имеется пять форматов:

* JPEG
* PNG
* WebP
* AVIF
* HEIC
* JPEG 2000
* SVG

Поговорим про каждый из них по отдельности

JPEG

Filename extension: jpg jpeg jpe jif jfif jfi

MIME type: image/jpeg

JPEG (Joint Photographic Expert Group) поддерживается всеми браузерами, один из самых популярных форматов изображений.

Преимущества JPEG:

* Поддержка глубокого цвета: Формат поддерживает до 24-битной цветовой глубины, обеспечивая возможность сохранения изображений с широким спектром цветов, что особенно полезно для фотографий.
* Широкая совместимость: JPEG форматы поддерживаются практически всеми программами для просмотра изображений и веб-браузерами, что делает его использование универсально доступным и простым.
* Регулируемое сжатие: Пользователи могут выбирать уровень сжатия, чтобы достичь оптимального баланса между качеством изображения и его размером.

Недостатки JPEG:

* Потери в качестве: Из-за алгоритма сжатия с потерями, при сохранении изображения в формате JPEG происходит утрата части исходных данных, что может ухудшить качество, особенно после многократного редактирования.
* Неоптимален для текстов и чётких контуров: JPEG не подходит для изображений с резкими контрастными границами, таких как текст или графические схемы, поскольку артефакты сжатия могут размывать линии.
* Отсутствие прозрачности: Формат не поддерживает прозрачные области, что делает его неуместным для графических работ, где необходимы слои прозрачности.
* Неподходящий для многократного редактирования: При каждом сохранении изображения в формате JPEG происходит дополнительная потеря данных, что делает этот формат нежелательным для изображений, которые требуется часто редактировать.

PNG

Filename extension: png

MIME type: image/png

PNG (Portable Network Graphics) — это растровый формат изображения, который предлагает сжатие без потерь. Еще одна особенность PNG формата — изображение может содержать прозрачные области. Поддерживается всеми браузерами

Преимущества PNG:

* Сохранение качества: PNG применяет метод сжатия без потерь, который позволяет сохранить изначальное качество изображения, независимо от количества редактирований и сохранений, в отличие от формата JPEG.
* Поддержка прозрачности: Формат позволяет создавать изображения с прозрачными элементами, что делает его идеальным для использования в логотипах, иконках и дизайне веб-страниц, где требуется наложение слоёв. PNG также позволяет управлять уровнем прозрачности.
* Широкая цветовая палитра: PNG поддерживает глубину цвета от 8 до 48 бит, предлагая от 256 до десятков миллионов оттенков, что делает его универсальным для работы с разнообразными изображениями, включая фотографии и графические элементы.
* Подходит для детализированной графики: Из-за возможности сжать изображения без потерь и удержания чётких контуров, PNG отлично справляется с хранением карт, текстов и иллюстраций высокой детализации.

Недостатки PNG:

* Больший размер файлов: Поскольку PNG использует сжатие без потерь, размер файлов зачастую значительно больше по сравнению с JPEG, что может стать проблемой для больших изображений и привести к замедлению загрузки веб-страниц.
* Не лучший выбор для фотографий: Из-за крупных размеров файлов, PNG может быть менее подходящим для фотографий по сравнению с JPEG, который более эффективно справляется с сжатием фотографических изображений.

WebP

WebP — это формат изображений, разработанный Google, который предлагает сжатие изображений с потерями и без потерь. Он призван обеспечить высокое качество изображений при меньших размерах файлов. Согласно сайту [caniuse.com](https://caniuse.com) поддерживается 95,3% браузеров на 2024 год

Плюсы WebP:

* Поддержка анимации: WebP поддерживает анимированные изображения, поэтому он может использоваться для создания веб‑анимаций.
* Поддержка прозрачности: WebP поддерживает прозрачность, что идеально подходит для логотипов, иконок и слоёв наложения в веб‑дизайне. Так же можно указать интенсивность прозрачности.
* Широкая совместимость: WebP совместим со многими современными браузерами и программами для просмотра изображений, что обеспечивает его удобство использования.

Минусы WebP:

* Степень сжатия несколько ниже, чем у следующий форматов

AVIF

AVIF (AV1 Image File Format) — это современный формат изображений, основанный на технологии сжатия AV1. Он предназначен для обеспечения высокого качества изображений при более низком размере файлов. Согласно сайту [caniuse.com](https://caniuse.com) поддерживается 92,6% браузеров.

Плюсы AVIF:

* Эффективное сжатие с потерями: Формат AVIF применяет современные алгоритмы, которые обеспечивают высокое качество изображения при значительном уменьшении размера файла. Это улучшает скорость загрузки веб-страниц и снижает нагрузку на сеть.
* Высокий динамический диапазон (HDR): AVIF поддерживает изображения с широким диапазоном яркости и более точными цветами, благодаря поддержке HDR.
* Поддержка прозрачности: В формате AVIF возможны как сжатие с потерями, так и без, что делает его подходящим для работы с разными видами изображений, включая те, где есть прозрачные области.
* Совместимость: Формат AVIF поддерживается многими современными браузерами и приложениями для просмотра изображений, что делает его удобным в использовании.

Минусы AVIF:

* Поддержка устаревшими браузерами: Хотя AVIF становится все более популярным, некоторые старые браузеры и программы, такие как Internet Explorer, могут не полностью поддерживать его, что может вызывать проблемы с отображением на некоторых устройствах. Проблема решается с помощью тега picture, он имеет поддержку 96,4%.
* ? Требования к производительности устройств для декодирования: Из-за применения сложных алгоритмов сжатия, для декодирования AVIF может понадобиться больше ресурсов процессора и памяти, что может вызвать трудности на старых устройствах. (В консоли разработчика без замедления 5.57с против 6.40c, при x6 замедлении и slow 4g результат 5.59с Avif против 6.42s WebP)
* Не поддерживает прогрессивную загрузку. (На мой взгляд прогрессивную загрузку лучше заменить фотографией низкого качества, как сделано в Telegram, например)

HEIF/HEIC

Filename extension: heif heic

MIME type: image/heif image/heic

HEIF (High Efficiency Image Format) — это современный формат изображений, который обеспечивает высокую эффективность сжатия и поддержку различных функций, таких как анимация, HDR, прозрачность и многослойность.

HEIC (High Efficiency Image Container) — это контейнерный формат файла, который используется в том числе и для хранения изображений в формате HEIF, как пример можно привести Live Photos сделанные на iPhone.

Плюсы HEIF:

* Эффективное сжатие изображений: HEIF применяет современные методы сжатия, такие как HEVC (High Efficiency Video Coding), которые позволяют значительно уменьшать размер файлов при сохранении высокого качества картинок.
* Разнообразие функций: Этот формат поддерживает не только статические изображения, но и анимации, HDR, прозрачность и многослойные изображения, что делает его универсальным для различных нужд.
* Сохранение высокого качества: Благодаря современным технологиям сжатия, HEIF способен обеспечить высокое качество изображений при уменьшенных размерах файлов.

Минусы HEIF:

* Ограниченная поддержка: Главной проблемой HEIF является его ограниченная поддержка устройствами. Формат широко используется на устройствах Apple, что может затруднить его применение в вебе. Несмотря на это, его знание может быть полезно из-за большого количества пользователей iOS и macOS по всему миру.

Рассмотрим ещё один формат для векторных изображений, по сути у него нет аналогов

SVG

Filename extension: svg

MIME type: image/svg+xml

SVG (Scalable Vector Graphics) — это формат изображений, основанный на XML, который описывает двумерные векторные графики с использованием векторных объектов, таких как линии, кривые, формы и текст.

Плюсы:

* Масштабируемость: SVG изображения могут быть масштабированы без потери качества, что делает их идеальным выбором для различных размеров экранов и разрешений, включая высокоуровневые дисплеи.
* Малый размер файлов: Поскольку SVG использует векторную графику, файлы обычно меньше по размеру, чем растровые изображения, что приводит к более быстрой загрузке веб‑страниц.
* Поддержка текста и шрифтов: SVG позволяет включать текст прямо в изображение, что обеспечивает лучшую читаемость и поддержку различных шрифтов на веб‑страницах.
* Возможность взаимодействия и анимации: SVG поддерживает интерактивные элементы и анимацию, что делает его удобным для создания динамических и анимированных изображений.

Минусы:

* Сложность: SVG формат имеет множество особенностей в использовании по сравнению с растровыми

форматами, поэтому требуется хотя бы базовое понимание того как этот формат работает

* Ограничения в применении: по своей природе, если можно так выразиться, SVG формат подходит для «простых» изображений.

Набор качеств для форматов изображений

Итак, изучив основные форматы, можно определить набор параметров для оценки качества форматов

* Поддерживаемость браузерами
* Степень сжатия
* Сохранение качества
* Скорость декодирования (в случае с браузерами: как быстро браузер отобразит изображение при максимальной скорости интернета)
* Скорость кодирования

Составим таблицу для наглядного сравнения форматов растровых изображений. По каждому критерию я буду выставлять балл от 1 до 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Поддерживаемость браузерами | Степень сжатия | Сохранение качества | Скорость декодирования | Скорость кодирования |
| PNG | 10 | 0 | 10 | 10 | 10 |
| JPEG | 10 | 5 | 7 | 10 | 10 |
| WEBP | 10 | 7 | 7 | 10 | 10 |
| AVIF | 9 | 10 | 7 | 10 | 5 |
| HEIC | 1 | 10 | 7 | 10 | 10 |

В ходе исследования были выделены три фаворита: Webp и AFIV для растровой графики и SVG для векторной. Давайте вкратце разберёмся, как работают эти алгоритмы

Webp

Как устроено сжатие в WebP

Сжатие с потерями основано на утверждении, что яркость и цвет соседних пикселей мало отличаются друг от друга.

Изображение делится на макроблоки. Внутри каждого макроблока декодер предсказывает яркость и цвет следующего пикселя на основе ранее полученных блоков. Он использует уже декодированные пиксели в непосредственной близости от каждого из макроблоков, и пытается закрасить неизвестные фрагменты в них. Таким образом данные о фрагментах, которые можно предсказать могут считаться лишними и удаляются из кода, что обеспечивает более эффективное сжатие.

После математически обратимого преобразования (с помощью ДКП) результат подвергается квантованию и энтропийному кодированию.

Lossy WebP

В lossy WebP используется арифметическое кодирование — один из алгоритмов энтропийного сжатия. В WebP используется блочное квантование и биты распределяются адаптивно между различными фрагментами изображения: меньшее количество бит для фрагментов с низкой энтропией и большее — для фрагментов с высокой. Такое кодирование считается более гибким, чем код Хаффмана (который использует JPEG).

JPEG делит изображение на одинаковые блоки, а технологии кодирования WebP использует умное деление. В тех частях картинки, где есть много мелких и быстро изменяющихся деталей блоки имеют размер 4 х 4 пикселя, а в монотонных областях — 16 х 16.

Как происходит прогнозирование

Декодер VP8 имеет 2 класса прогнозирования:

* Intra — внутрикадровое пространственное предсказание блока на основе значений пикселей из соседних, уже закодированных блоков, слева и сверху.
* Inter — межкадровое временное предсказание (оценка векторов движения).

Intra имеет четыре алгоритма прогнозирования для блоков 16х16 и 8 для детализирующих блоков 4 х 4:

* H\_PRED горизонтальное прогнозирование. Заливает следующую колонку на основе той, что находится слева от нее.
* V\_PRED вертикальное прогнозирование. Заливает следующий ряд на основе предыдущего верхнего.
* DC\_PRED заполняет блок, используя усредненные значения цвета и яркости пикселей строки
* TM\_PRED заполняет блок, используя не только усредненные значения строки A и колонки L, но и пиксель P, который находится сверху и слева от блока. Каждая строка начинается с пикселя в колонке L и заполняется в соответствии с различиями пикселей в колонке, начиная от пикселя P.

Изображение разбивается на сегменты, которые имеют явно схожие характеристики. Для каждого такого сегмента параметры сжатия и способы прогнозирования настраиваются независимо. Таким образом биты перераспределяются туда, где они наиболее полезны.

Сжатие без потерь

При сжатии без потерь используется вариант алгоритма LZ77 — кода Хаффмана. А также пространственное прогнозирование и преобразование цветового пространства.

Не только прогнозирование

Сжатие с альфа-каналом

Формат WebP позволяет получить сжатую картинку с альфа-каналом без потерь. Раньше, чтобы получить прозрачность все изображение должно было быть lossless. А в WebP можно уменьшить вес картинки с прозрачными областями.

Цветовое преобразование

Также в WebP используется методы адаптивного квантования цветовой составляющей, чтобы предотвратить влияние цветовых каналов друг на друга. Изображение делится на блоки и для каждого блока применяется свой режим трансформации green\_to\_red, green\_to\_blue или red\_to\_blue. Цветовое преобразование сохраняет неизменным значение зеленого канала G, преобразует красный R в зависимости от зеленого, и синий В в зависимости от зеленого, а затем в зависимости от красного.

Цветовое кеширование

Сжатие lossless WebP использует уже обработанные фрагменты изображения для работы с новыми пикселями. В случае если подходящие совпадения не найдены, используется локально созданная палитра. Эта палитра постоянно обновляется цветами, найденными при сканировании картинки.

Индексирование палитры

Если в картинке используется менее 256 цветов, алгоритм создает отдельный массив индексов цветов и сохраняет его отдельно, чтобы подменить значение цвета на индекс для каждого пикселя.

Для картинок с небольшим количеством мелких деталей используется технология апскейлинга. Прежде чем кодироваться — изображение ресайзится.

AVIF

AV1 Image File Format (AVIF) (/əˈviːf/) — это спецификация формата файла изображения для хранения изображений или последовательностей изображений, сжатых с помощью AV1, в формате контейнера AVIF. Он конкурирует с HEIC, который использует тот же формат контейнера, что и ISOBMFF[англ.], но HEVC для сжатия. Версия 1.0.0 спецификации AVIF была завершена в феврале 2019 года.

AOMedia Video 1 (AV1) — открытый стандарт сжатия видео, предназначенный для кодирования видео, передаваемого по сети Интернет. Разрабатывается некоммерческой организацией Alliance for Open Media, созданной в 2015 году и состоящей из компаний, занимающихся производством электроники (AMD, Apple, ARM, Broadcom, Intel, Nvidia), распространением видео по запросу (Apple, Amazon, Facebook, Google, Hulu, Netflix), разработкой веб-браузеров (Apple, Google, Mozilla, Microsoft)[3][4][5].

По оценкам инженеров Facebook, AV1 позволяет на 50 % уменьшить битрейт при одинаковом качестве по сравнению с H.264, и на 30 % — по сравнению с VP9, при этом — чем выше разрешение, тем эффект сжатия лучше. Включение аппаратной поддержки AV1 позволяет получить преимущества улучшенного видеокодека, перенеся работу по декодированию с программного обеспечения на GPU (особенно это важно для ноутбуков, так как позволяет снизить энергопотребление)[6].

**Информации про AVIF было найдено очень мало**

SVG

Как упомяналось ранее, SVG использует XML, где описывается какие геометрические фигуры будут использованы для построения изображения. В итоге получается примерно такой файл:

<!DOCTYPE html>

<html>

<body>

<h2>Three ellipses on top of each other</h2>

<svg height="150" width="500" xmlns="http://www.w3.org/2000/svg">

<ellipse cx="240" cy="100" rx="220" ry="30" fill="purple" />

<ellipse cx="220" cy="70" rx="190" ry="20" fill="lime" />

<ellipse cx="210" cy="45" rx="170" ry="15" fill="yellow" />

Sorry, your browser does not support inline SVG.

</svg>

</body>

</html>

Готовые решения для конвертирования изображений в web-разработке

WebP

Для **Node.js** существуют следующие библиотеки:

* webp-converter <https://www.npmjs.com/package/webp-converter>
* imagemin imagemin-webp https://www.coderrocketfuel.com/article/how-to-compress-webp-image-files-using-node-js

Для **Java**

* img-to-webp-spring-service <https://github.com/phip1611/img-to-webp-spring-service>

AVIF

Node.js

* avif-cli <https://www.npmjs.com/package/avif?activeTab=explore>
* bimgc <https://socket.dev/npm/package/bimgc>

Java

* vavi-image-avif - <https://github.com/umjammer/vavi-image-avif>

SVG

Node.js

* @neplex/vectorizer - <https://github.com/neplextech/vectorizer>

Java

* https://github.com/nsclass/ns-svg-converter

Источники

* Устройство WebP - <https://habr.com/ru/companies/io/articles/261651/>
* Форматы изображений в вебе (2024) - https://habr.com/ru/articles/803823/
* Can I use … Support Tables for HTML5, CSS3, etc - https://caniuse.com