Николай Козлов · 6.12.2019

badoo by MagicLab

РЕАЛЬНАЯ ОПТИМИЗАЦИЯ ИЗОБРАЖЕНИЙ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Николай Козлов

Android QA Lead


badoo


455 000 000+
пользователей
по всему миру

MagicLab

badoo

bumble

Lumen

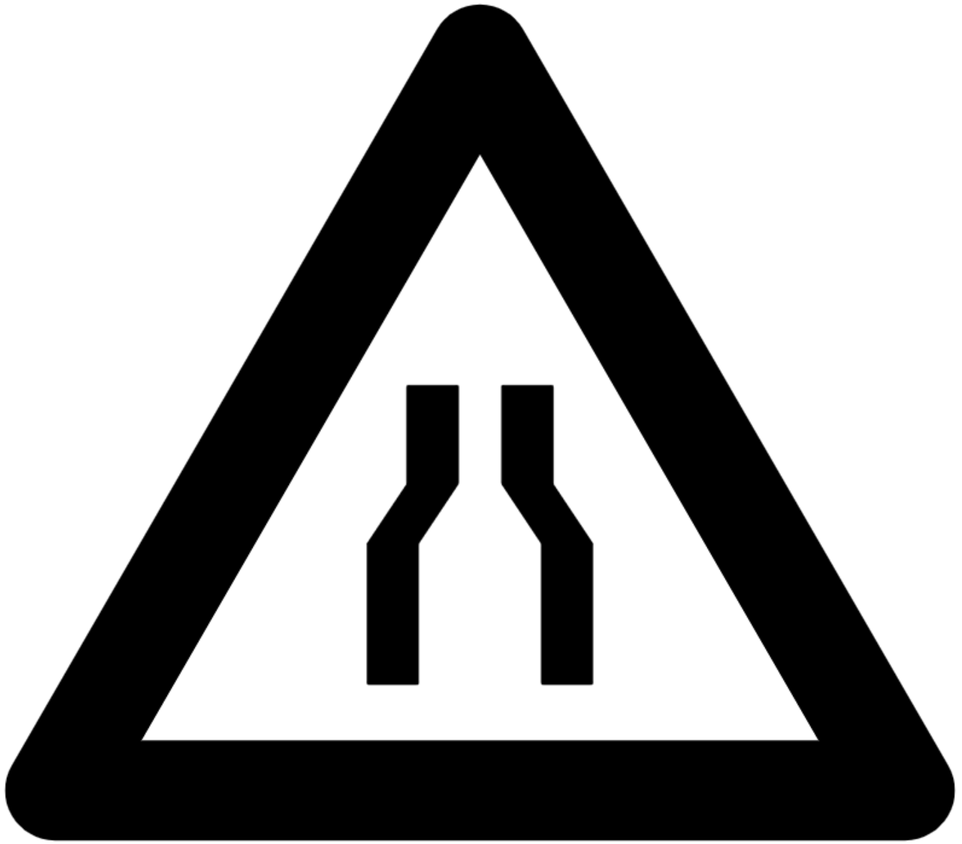
CHAPPY

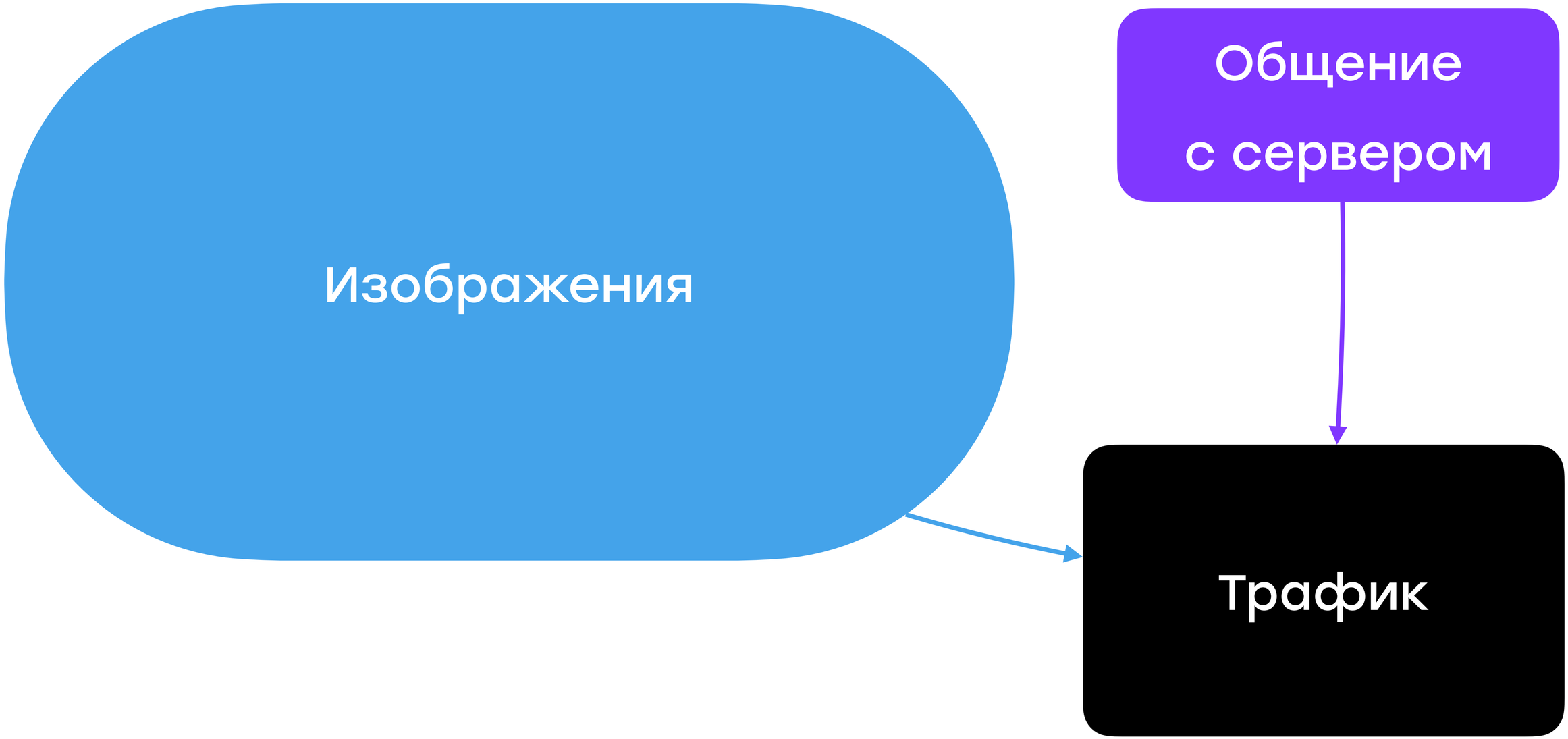
Изображения 83% трафика

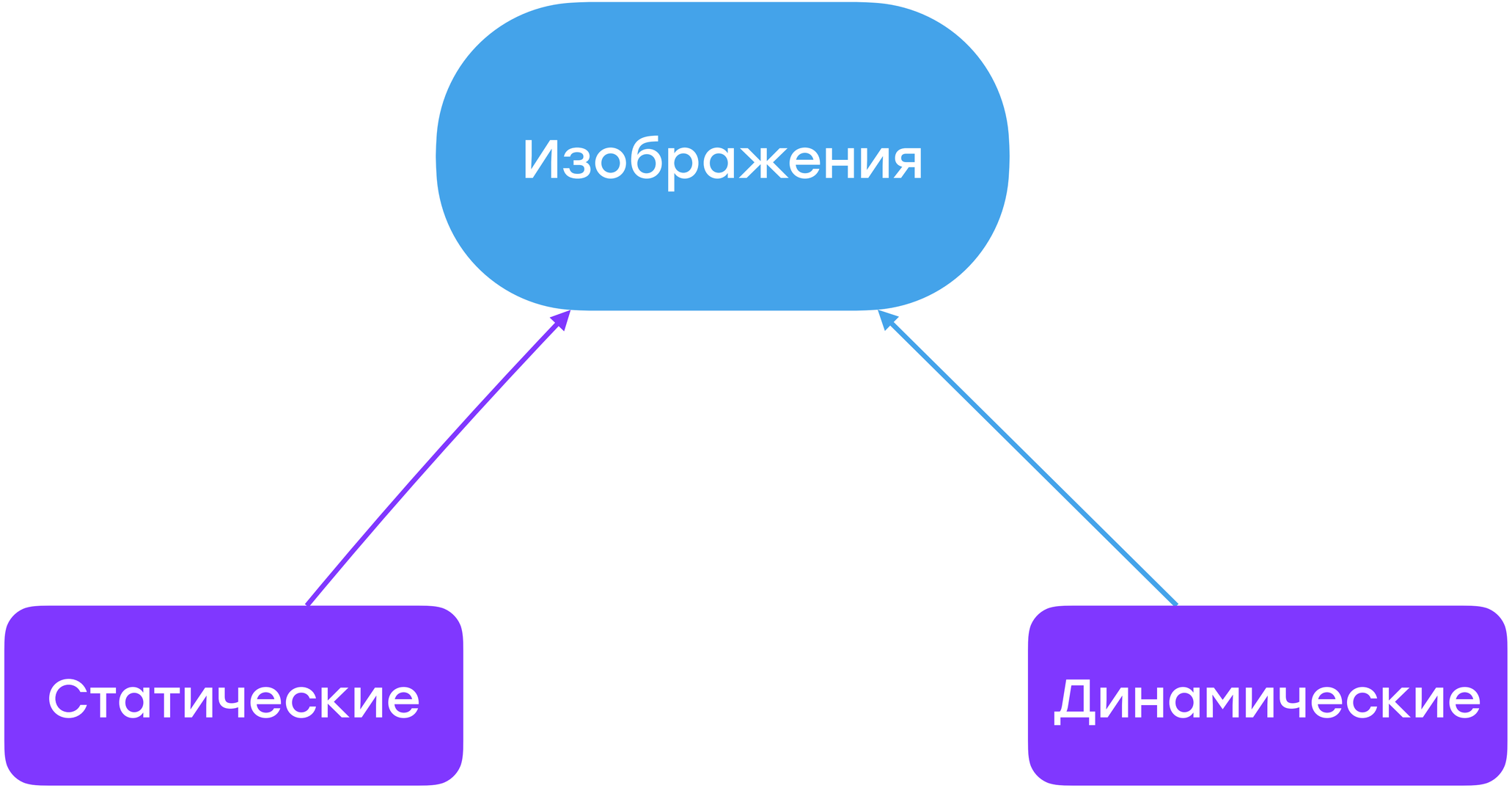
Почему я хочу поделиться с вами этими знаниями?

Введение

Трафик и канал




Изображения
Общение
с сервером
Трафик


Изображения
Статические
Динамические

Задавались ли вы вопросом, какого размера изображение передаете на клиент?

Оптимизация изображений

1. Битва форматов
2. Режем разрешение
3. Как работает кодирование (и где можно сэкономить)
4. Цифровая субдискретизация
5. Качество изображений
6. Прогрессив или бэйзлайн?
7. Удаление лишней информации

Исходник

Исходник

1. Хорошее разрешение
2. Не сильно сжат (при формате с потерями)
3. Должен быть поддерживаемого формата

Исходник

Битва форматов

Форматы

Формат / Кодек	Кодирование	Декодирование	Детали	Тип файла / расширение
BMP		•		BMP (.bmp)
GIF		•		GIF (.gif)
JPEG	•	•	Base+progressive	JPEG (.jpg)
PNG	•	•		PNG (.png)
WebP	• (Android 4.0+)	• (Android 4.0+)		WebP (.webp)
HEIF		• (Android 8.0+)		HEIF (.heic; .heif)

BMP

Сжатие минимальное (только кодирование длин серий).

Поддерживается огромным количеством устройств, но из-за больших размеров не получил распространения в сети.

52.5 MB

GIF

Способен хранить сжатые данные **без потери качества**, но не более 256 цветов.

Главное в формате — несколько картинок в контейнере (анимации).

Поддерживается огромным количеством устройств, но из-за потери цвета применение ограничивается анимациями в чатах.

9.9 MB

PNG

PNG is Not GIF

Формат сжатия **без потерь** и с **глубокой цветовой палитрой.**

Поддержка прозрачности.

Практическое применение — хранение изображений без потерь, элементы дизайна.

19 MB

HEIF

Сложный алгоритм сжатия.

Ни один браузер не поддерживает.

Практическое применение — хранение изображений с потерями на iOS устройствах.

9.7 MB

WEBP

Благодаря алгоритмам предсказания содержимого блоков сжимает очень хорошо (но теряет информацию о цвете)

Поддерживается многими, **кроме Apple**

Практическое применение затруднено

12.4 MB (1MB если с потерями)

JPEG

Самый распространенный формат изображения.

Сжатие с потерями.

11.8 MB

Не все так просто

21 KB в PNG

69 KB в JPG

9 KB в GIF

16 KB в WEBP

Какой же формат выбрать?

Выводы по форматам

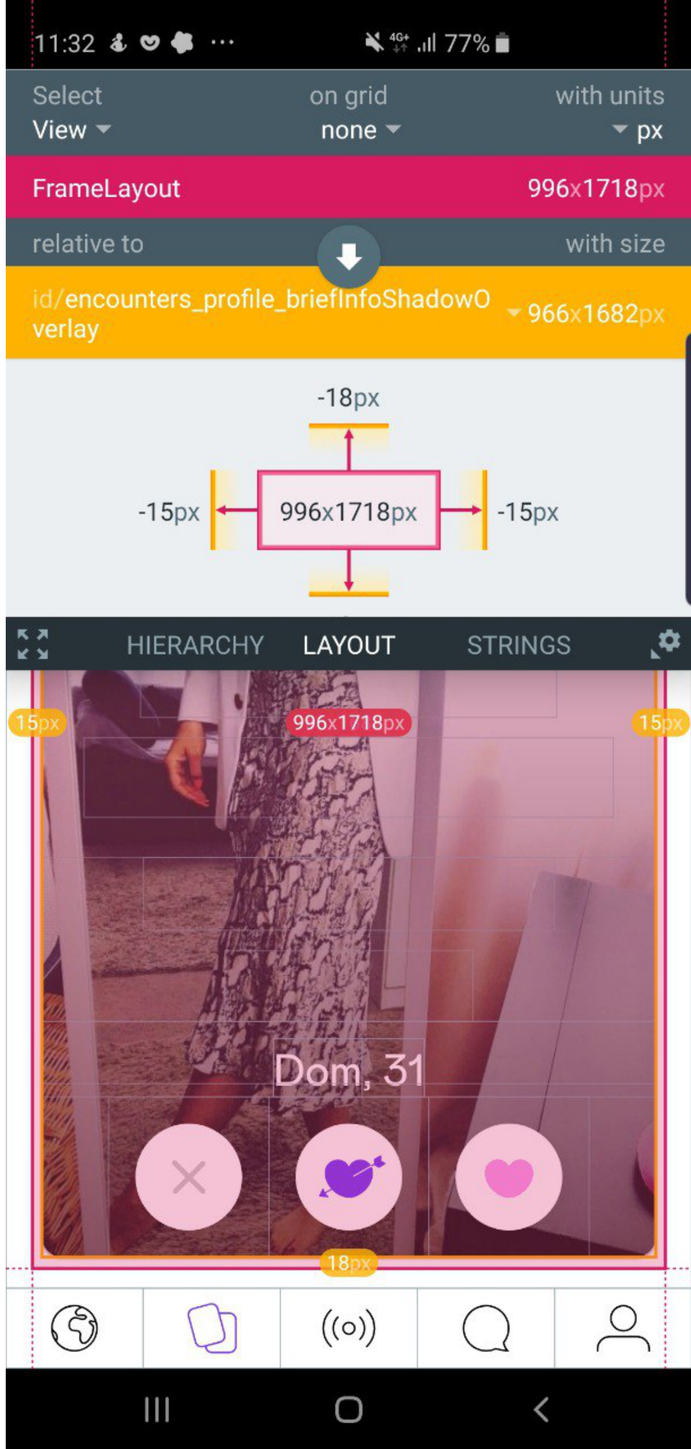
- Выбор формата зависит от решаемой задачи
- Идеального формата не существует
- Все рано или поздно скатывается к использованию jpeg (пока еще)

Режем разрешение

Режем разрешение

View Port и реальный размер

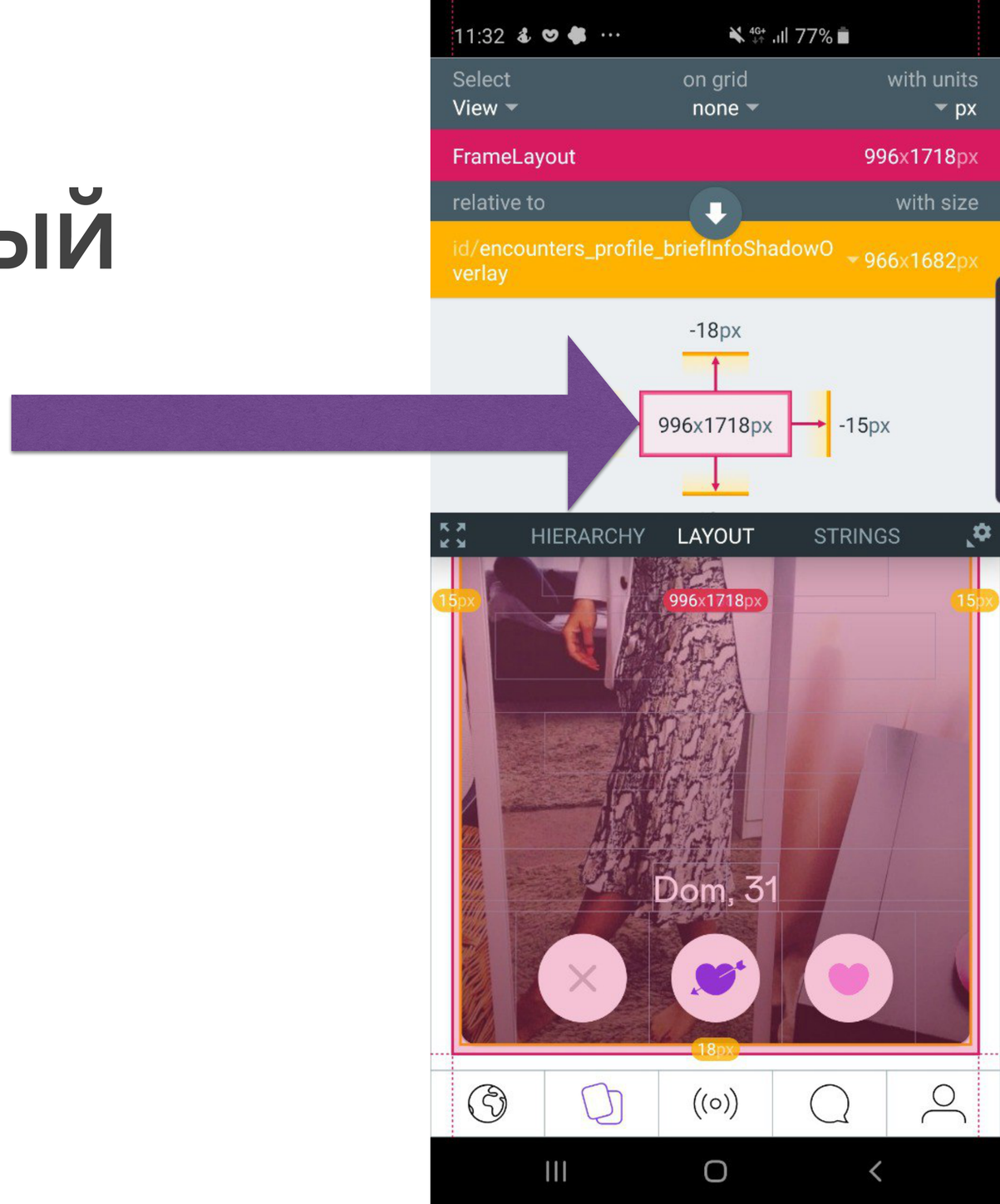
Зачем грузить лишнее?



Режем разрешение

View Port и реальный размер

Зачем грузить лишнее?



Режем разрешение

Битмапы в памяти

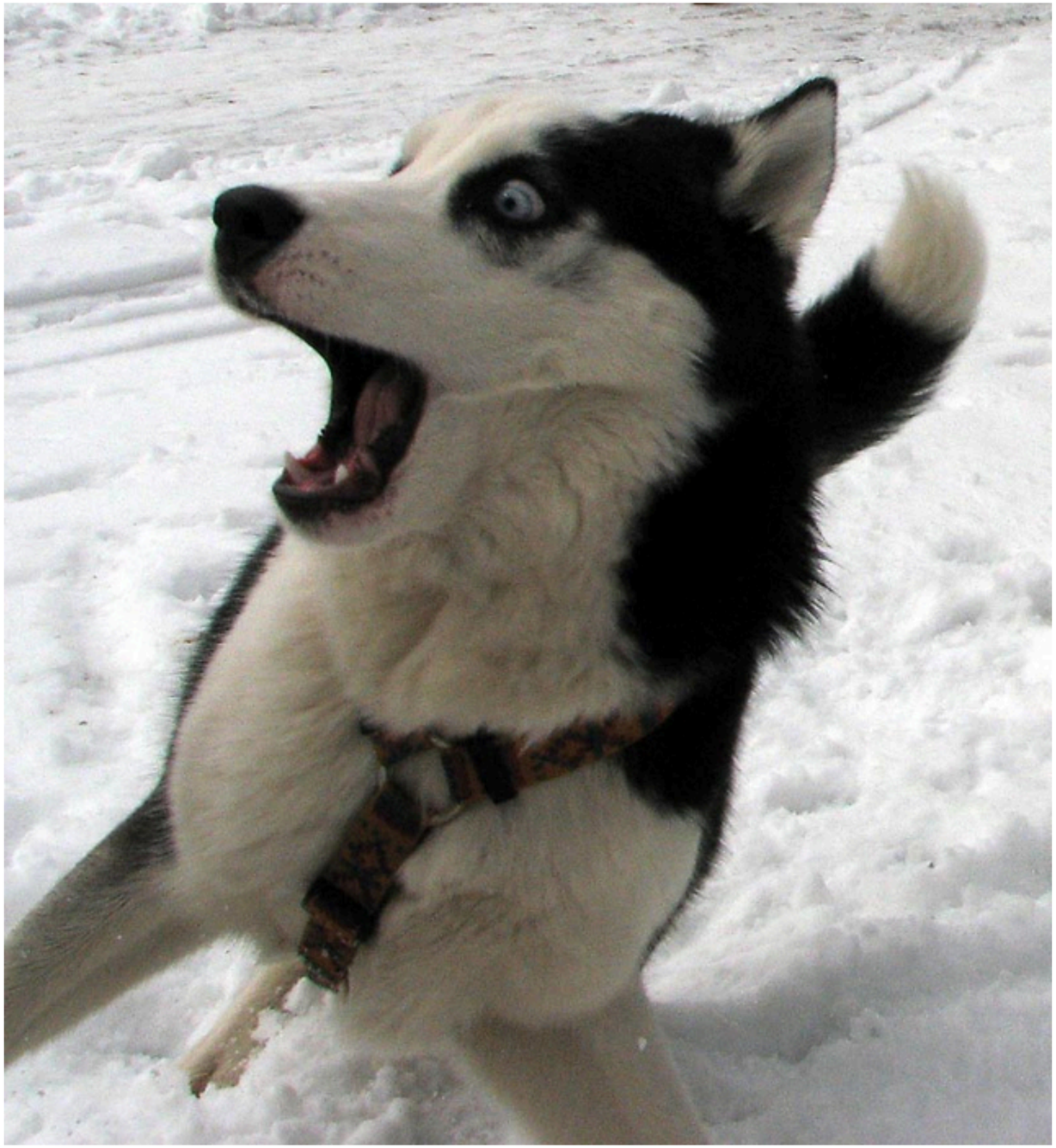
3.9MB на диске

6000x2908

Режем разрешение

Битмапы в памяти

66.7 MB в памяти!



Решение

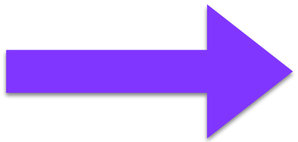
- Клиент передает размер ViewPort, для которого нужно изображение.
- Сервер либо масштабирует на лету, либо имеет заранее нарезанные картинки и отдает их клиенту.

Результат

3.9MB на диске

66.7MB в памяти

6000x2908



844KB на диске

16.6MB в памяти

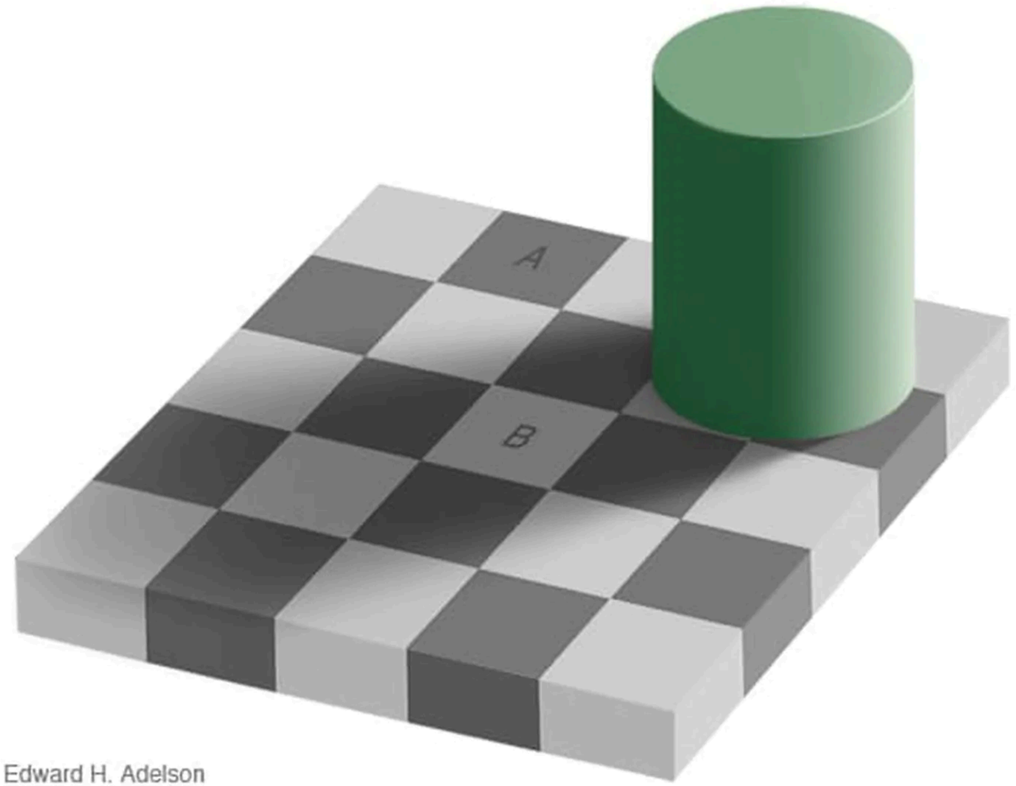
3000 × 1454

Как кодируется изображение

Зрение

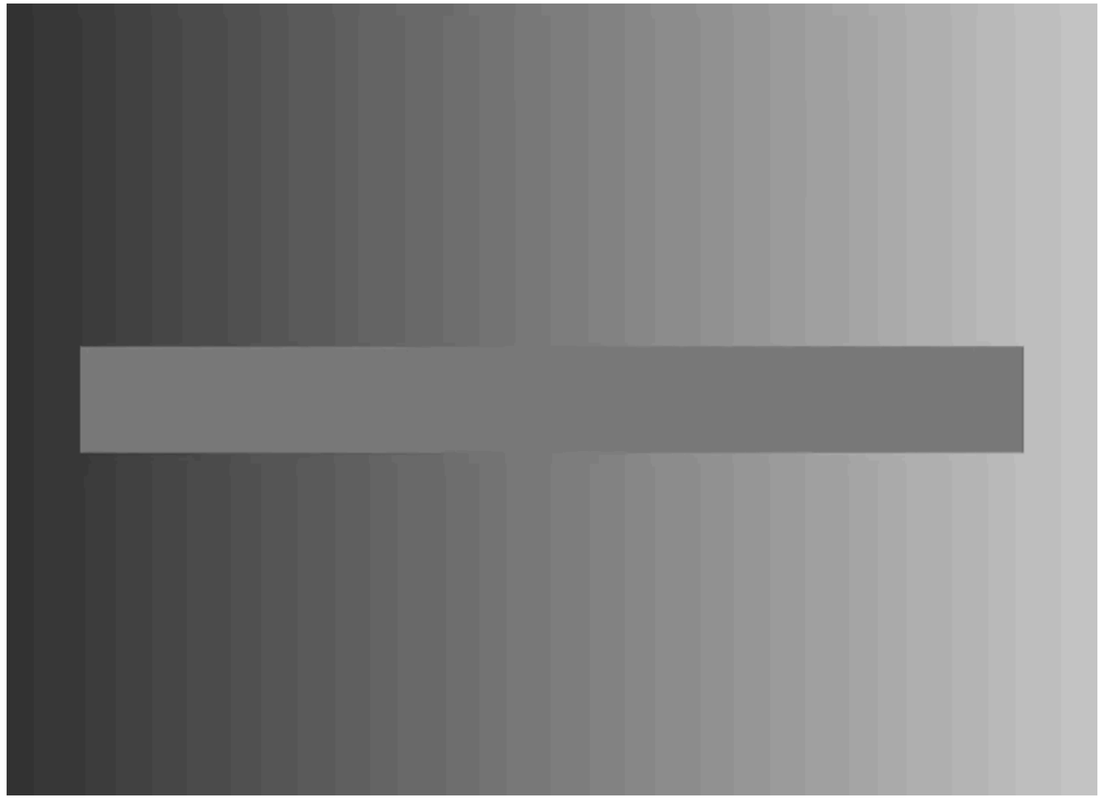
Таблица RGB — 16 млн цветов

Зрение

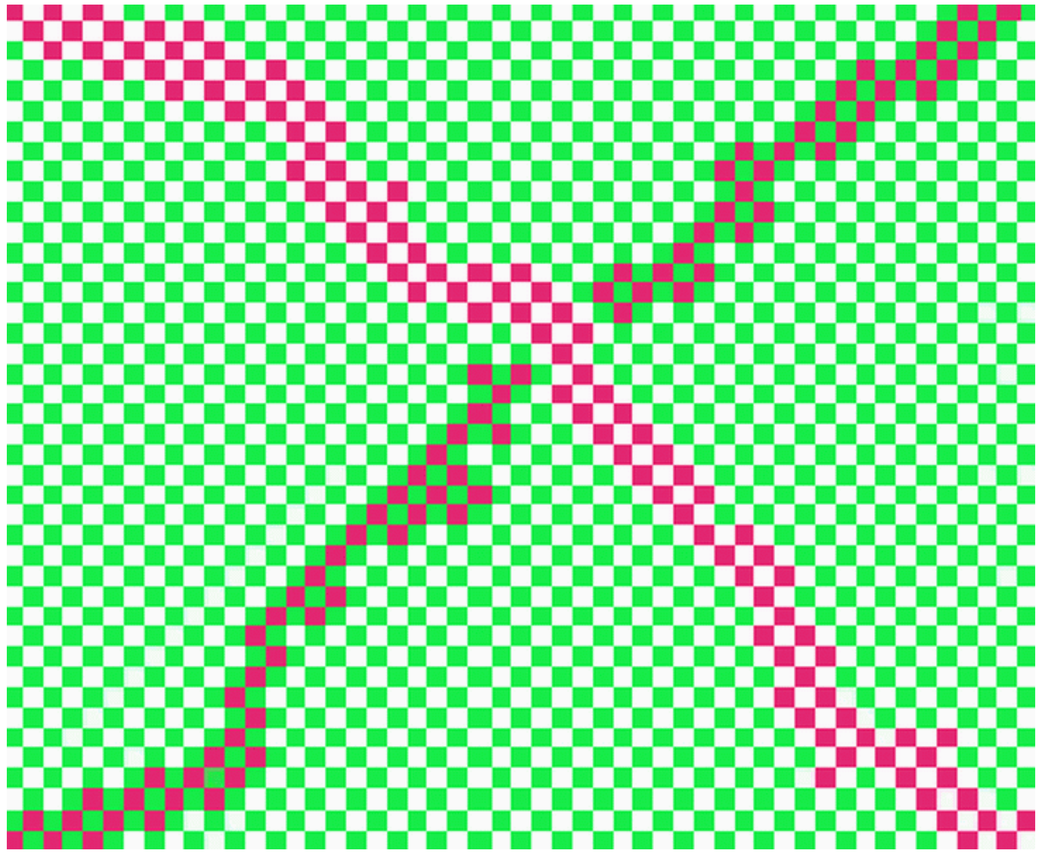


Edward H. Adelson

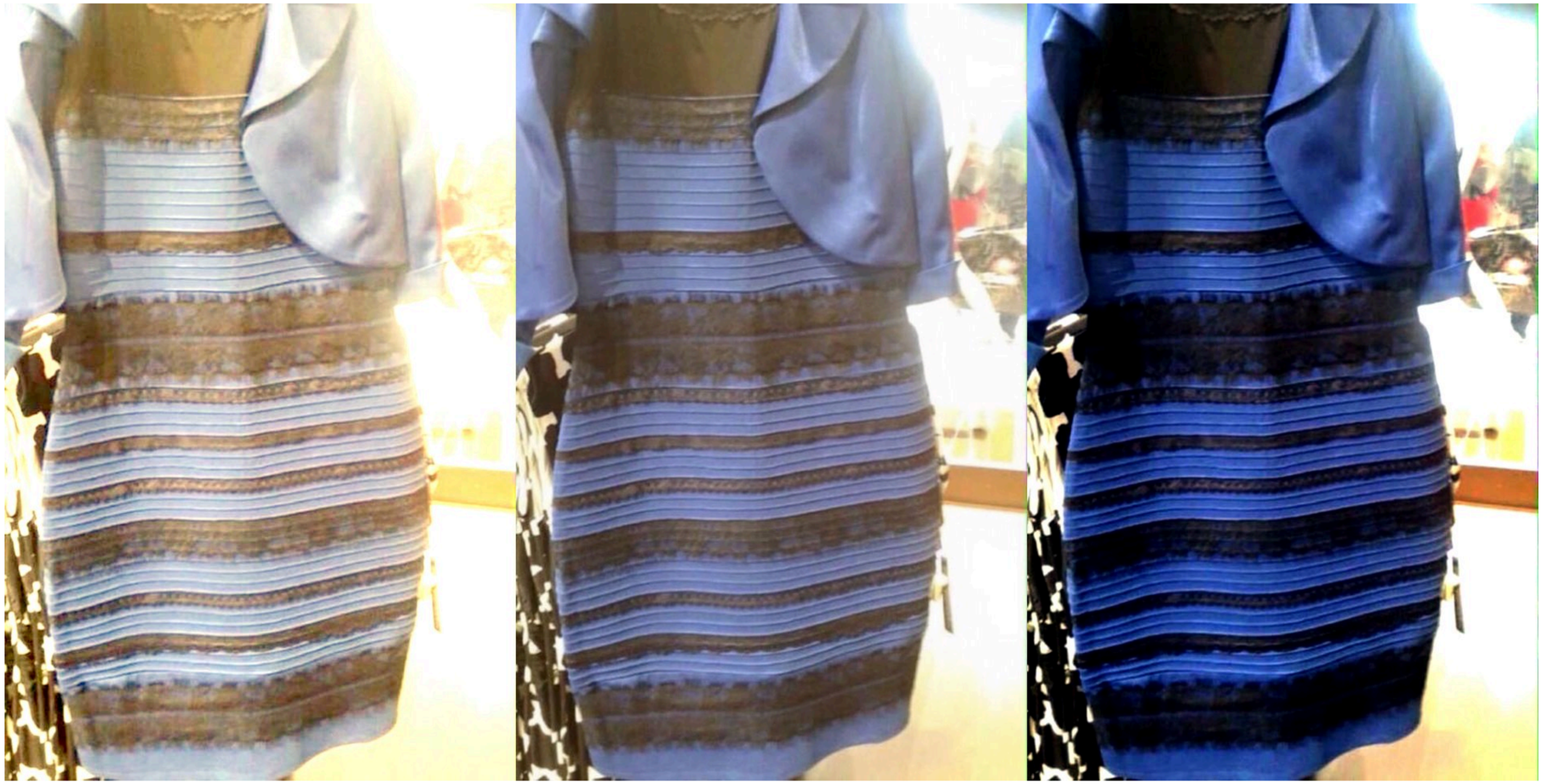
Зрение



Зрение



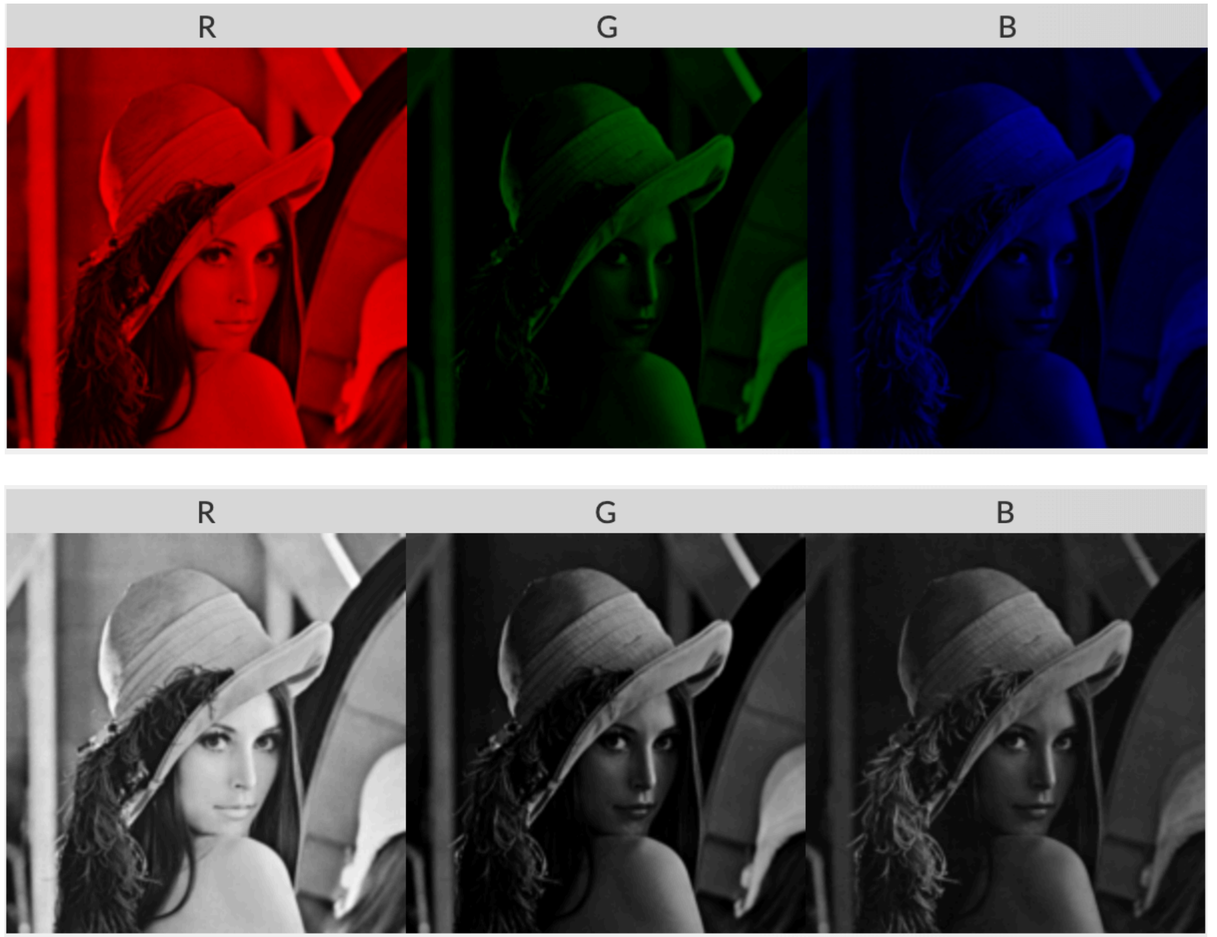
Зрение



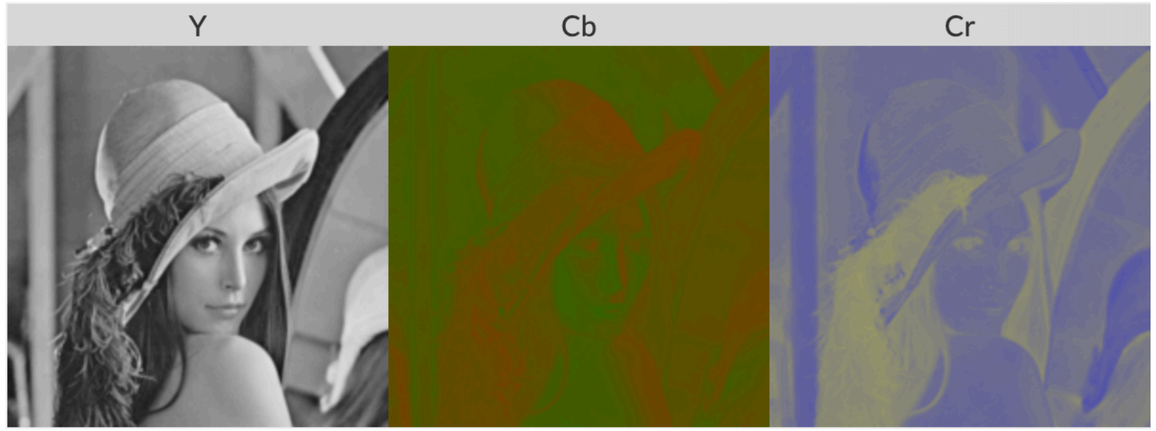
Lenna

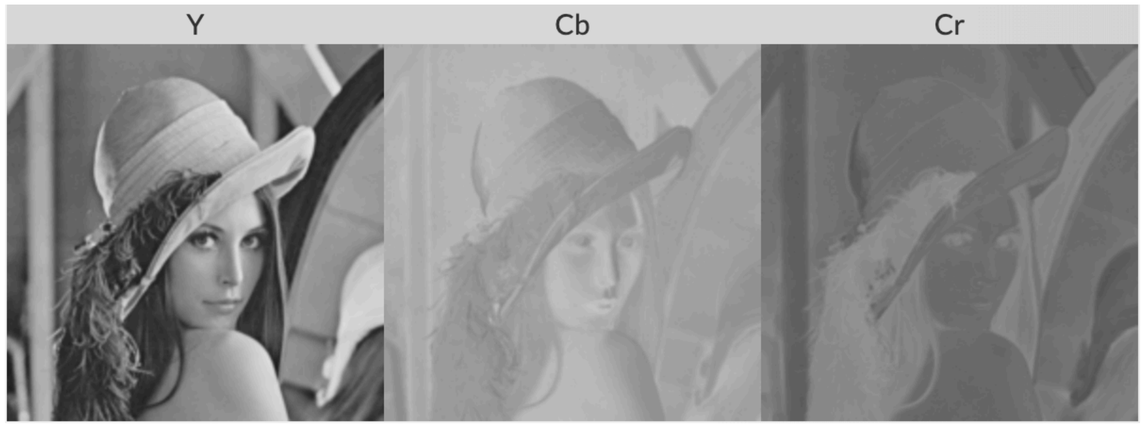
Стандартное тестовое изображение с 1973 года

Конвертирование цвета



Конвертирование цвета





Перевод из RGB в Y’CbCr

Сабсэмплинг

Перевод из RGB в Y’CbCr

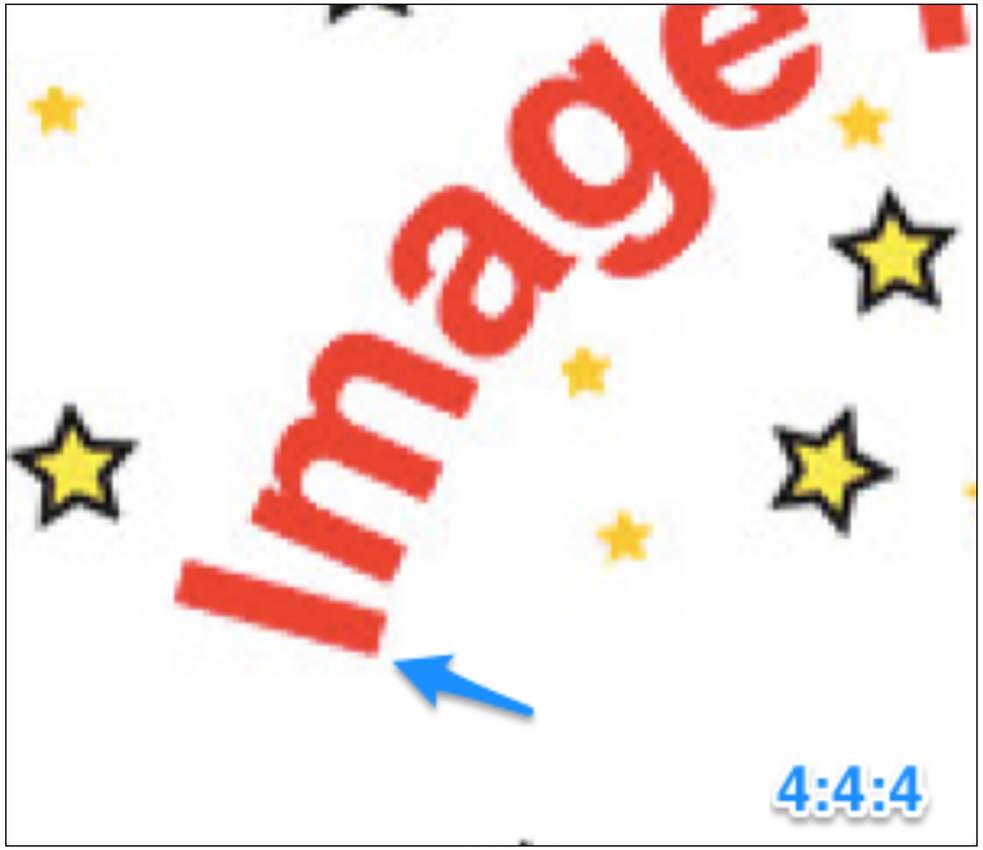
Сабсэмплинг

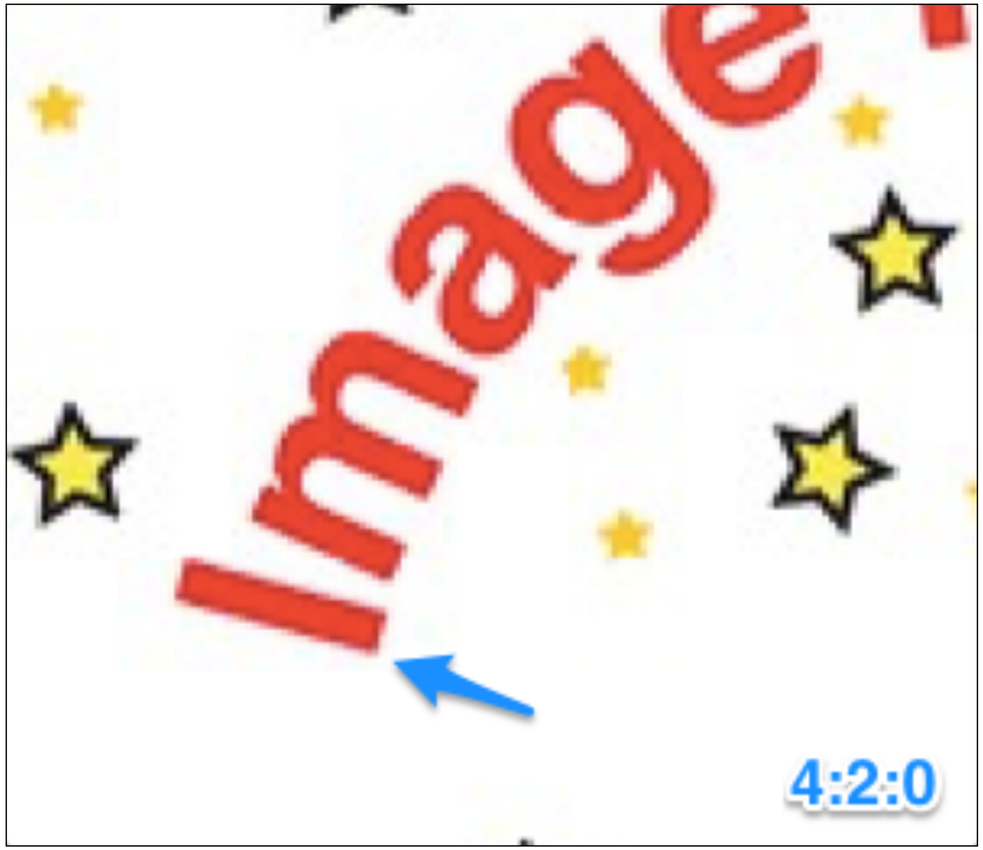
Сабсэмплинг

407KB

215KB

Сабсэмплинг





Сабсэмплинг

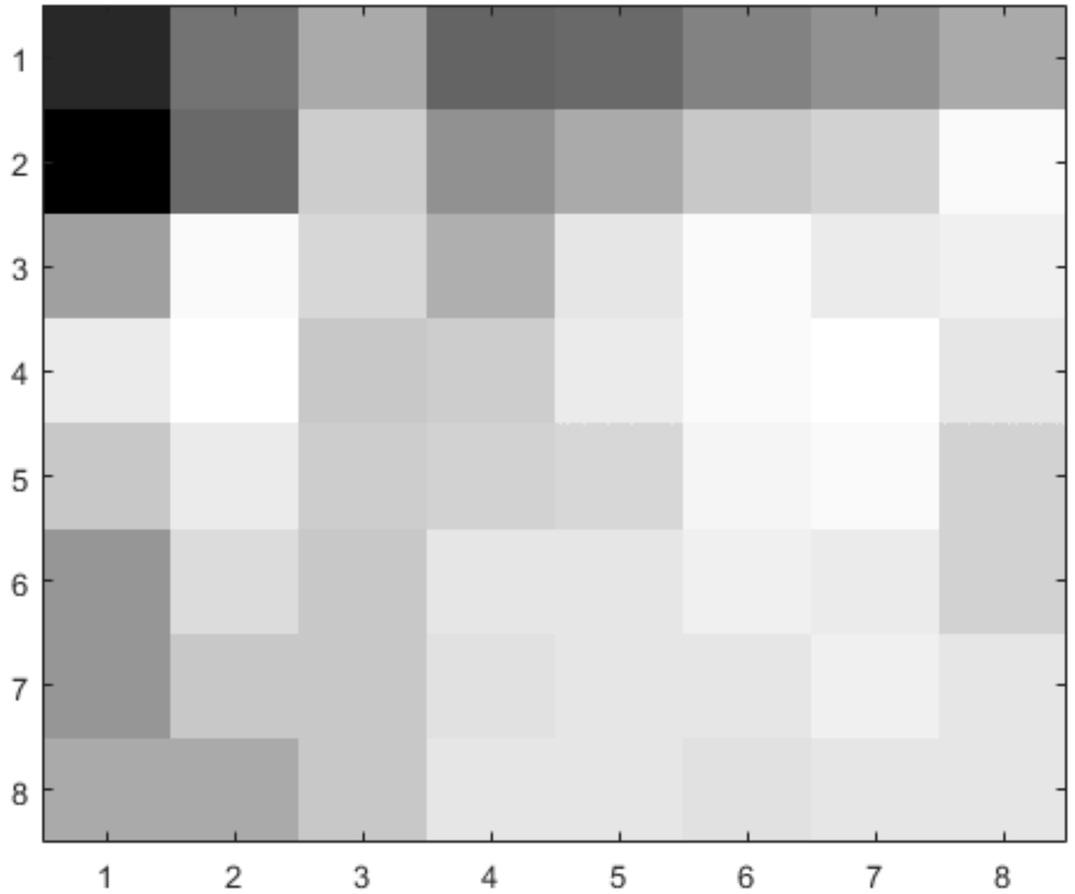
12–17% экономии

Выводы

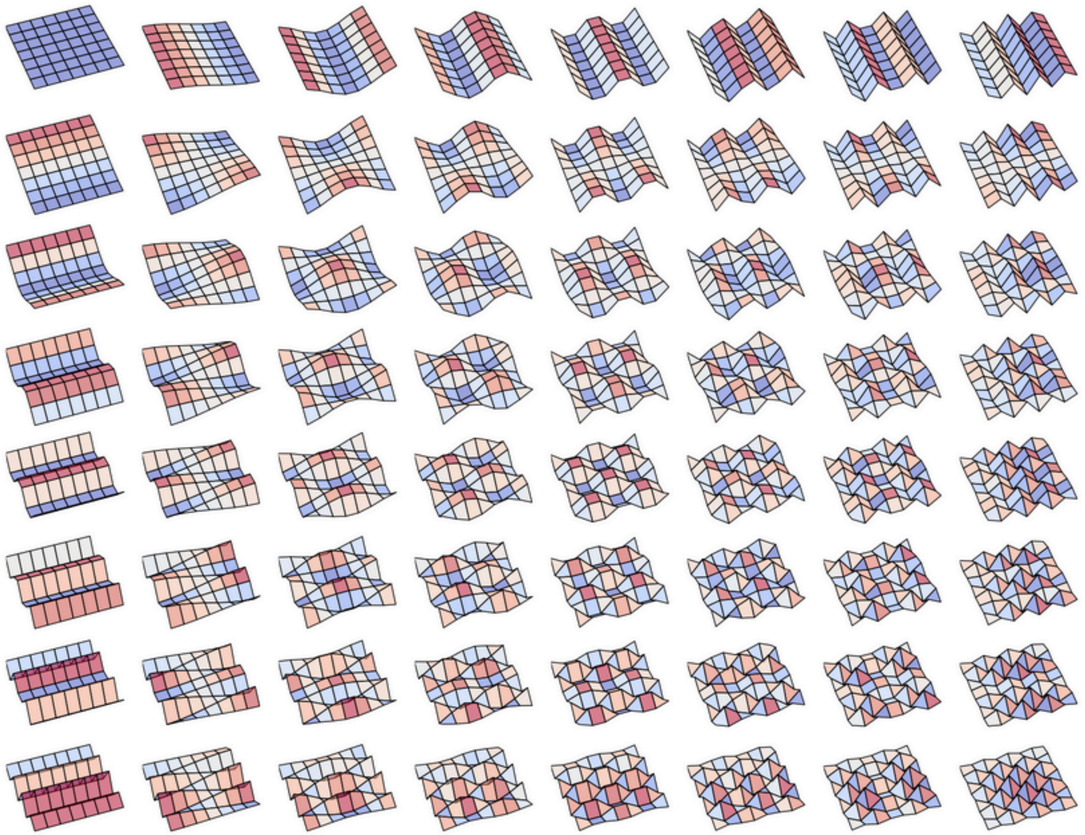
- Сабсемплинг хорошо уменьшает изображение.
- Можно применять для большинства фотографий, не боясь за цветопередачу.
- Для изображений с резкими переходами возможно изменение оттенка.

Качество сжатия

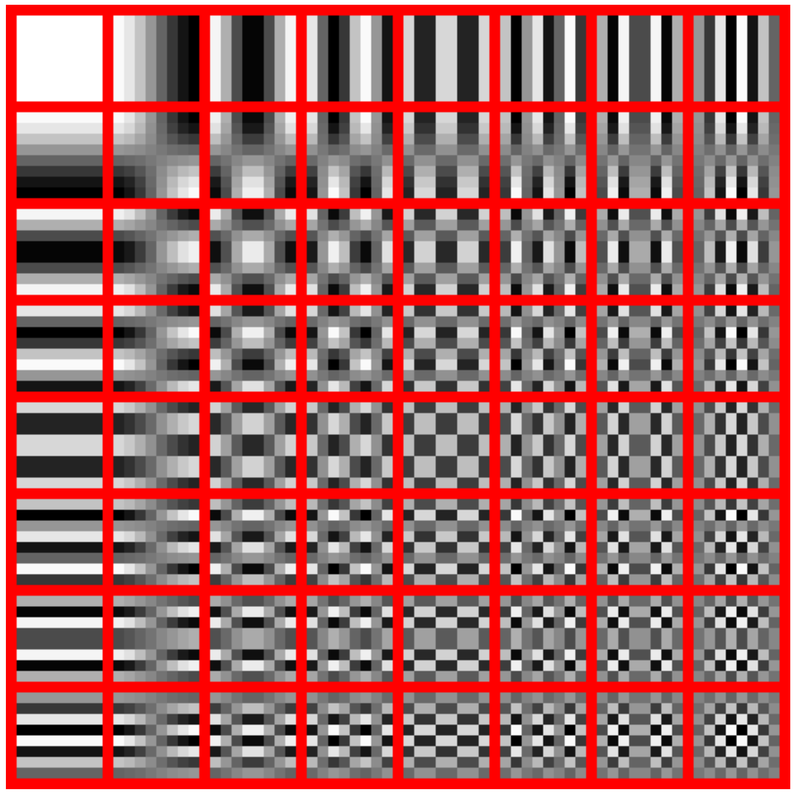
Разбиение на блоки



Дискретное косинусное преобразование



Дискретное косинусное преобразование



Дискретное косинусное преобразование

$$
G = \begin{bmatrix}
-415.38 & -30.19 & -61.20 & 27.24 & 56.12 & -20.10 & -2.39 & 0.46 \\
4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\
-46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\
-48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\
12.12 & -6.55 & -13.20 & -3.95 & -1.87 & 1.75 & -2.79 & 3.14 \\
-7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\
-1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\
-0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68
\end{bmatrix}
\begin{matrix} \\ \\ \\ \\ \\ \\ \\ \downarrow v. \end{matrix}
\qquad
\begin{matrix} u \\ \longrightarrow \end{matrix}
$$

Дискретное косинусное преобразование

Качество сжатия

Котик закодирован с переменным качеством от 1 до 100.

Чем ниже настройки качества, тем больше элементов из таблицы будут отброшены.

Качество сжатия — квантование

Таблица квантования

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Качество сжатия — квантование

$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

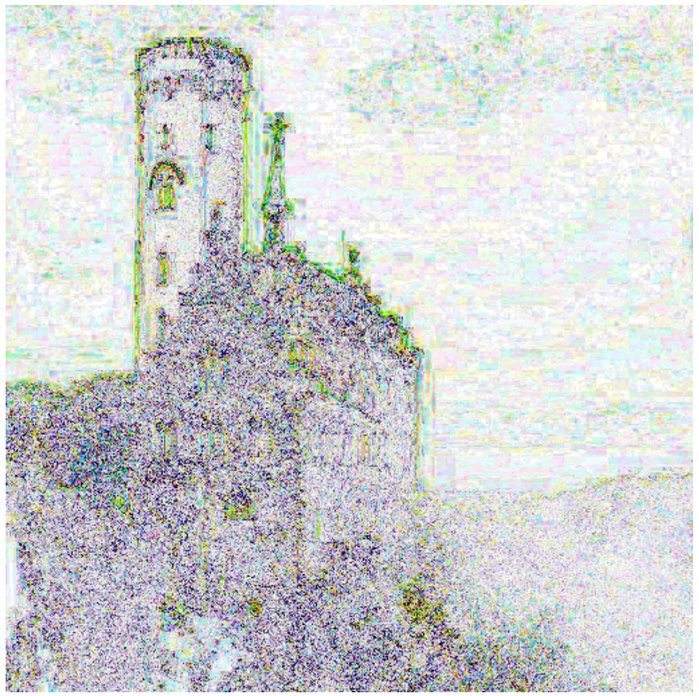
Выбор качества

5% 25% 50% 75% 100%

Выбор качества

Выбор качества

Выбор качества



Выбор качества

А как оценить качество?

1. На глаз
2. SSIM — индекс структурного сходства
3. PSNR — пиковое отношение сигнала к шуму
4. А может Butteraugli?

Выбор качества

Выбор качества

Выводы

- Человеческий глаз несовершенен при оценке различий в мелочах.
- Нет понятия оптимального качества.
- При качестве от 75 до 100 визуальные различия незначительны, а размер растет нелинейно.

Выводы

11.9MB

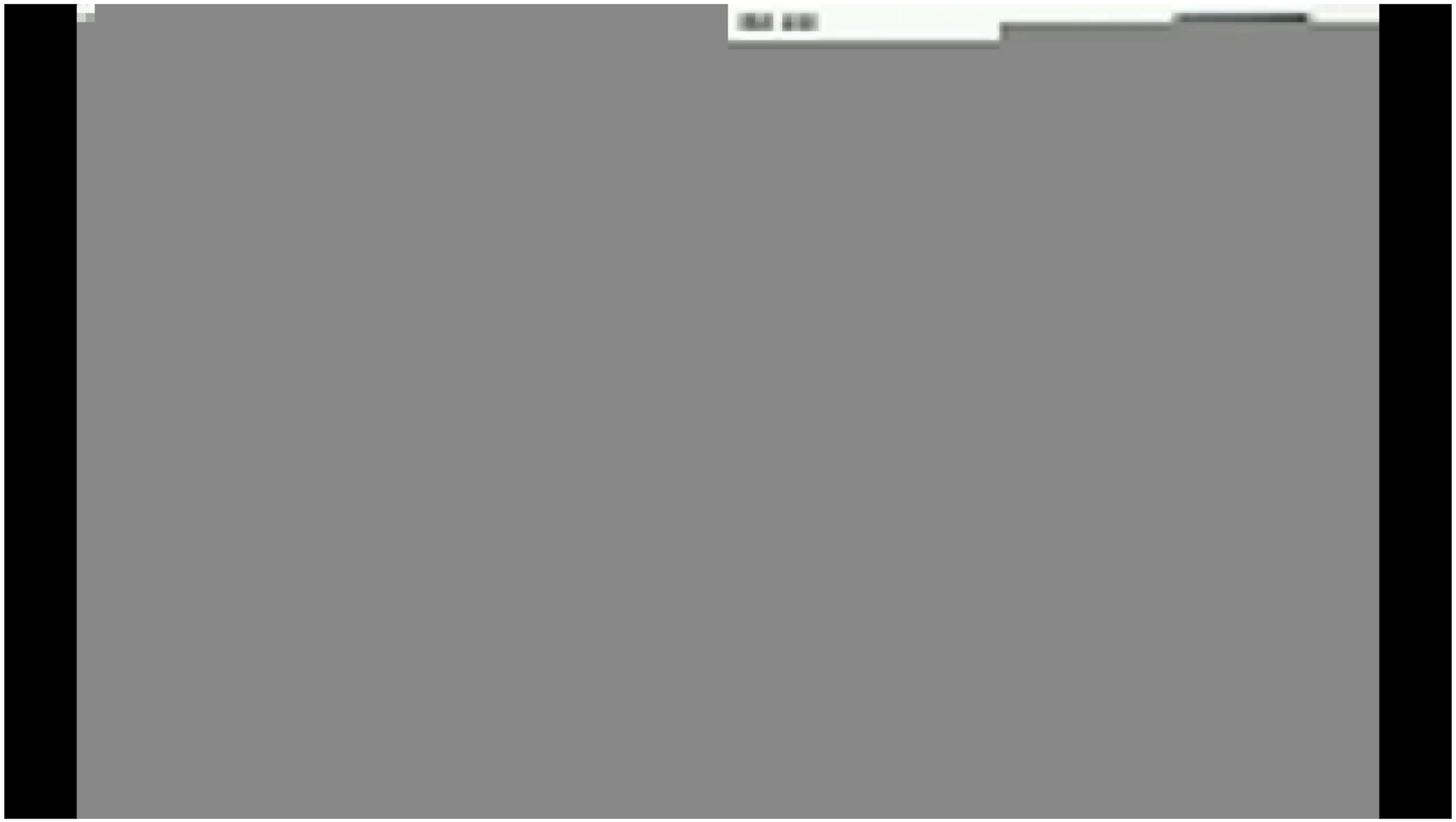
100%

2.9MB

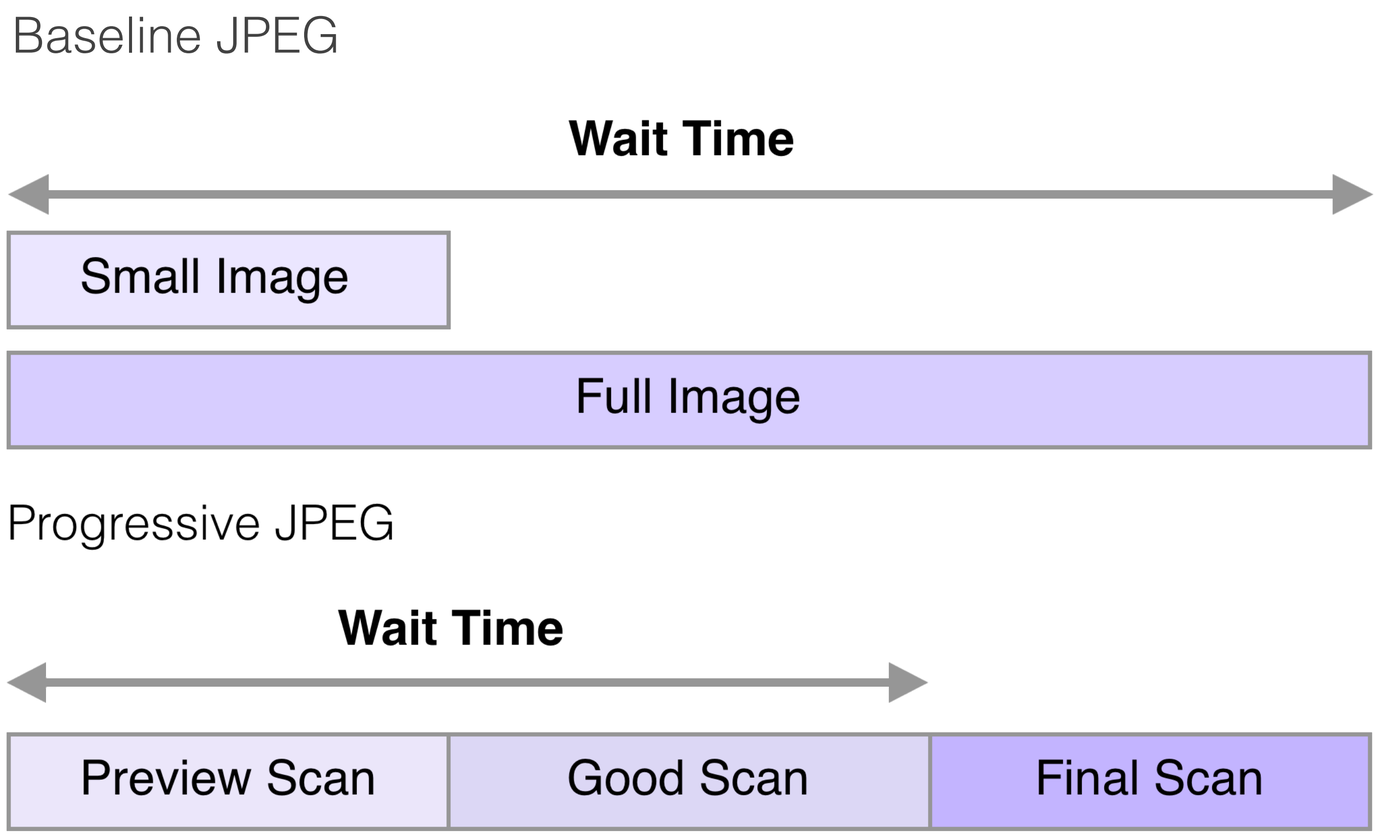
89%

Baseline vs Progressive

Baseline vs Progressive



Progressive vs Baseline



Progressive vs Baseline

- Для использования всех преимуществ Progressive придется использовать Fresco.
- Android из коробки не умеет работать с Progressive и показывает картинку как Baseline.
- Подавляющее количество изображений в сети – Baseline.

Progressive vs Baseline

2-10% меньше

Progressive vs Baseline

-5%

Выводы

- Progressive JPEG меньше, чем Baseline.
- Без разницы как запакован файл – он все равно отобразится.
- Использование особенностей Progressive (загрузка определенного скана) позволяет разгрузить канал.

Кодеки

Кодеки

Для кодирования JPEG нет требований к качеству сжатия!

Есть только требование к восстановлению файла в изображение.

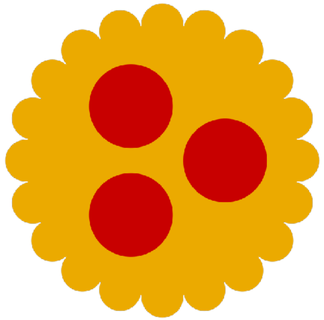
JPEG 9

STILL IMAGE CODEC

libjpeg-turbo

mozjpeg

Guetzli



Кодеки

Стандартная библиотека JPEG

1.7MB (качество 75%)

Время сжатия 2.164 сек

JPEG 9
STILL IMAGE CODEC

Кодеки

Приоритет — **скорость сжатия**

1.7MB (качество 75%)

Время сжатия 0.129 сек

libjpeg-turbo

Кодеки

Приоритет — размер файла

Декодирование быстрее

1.2 MB (качество 75%)

Время сжатия 1.830 сек

mozjpeg

Кодеки

Приоритет — лучшее качество

3.9MB (качество 96%)

Время конвертации - 16 минут!

Guetzli

Выводы

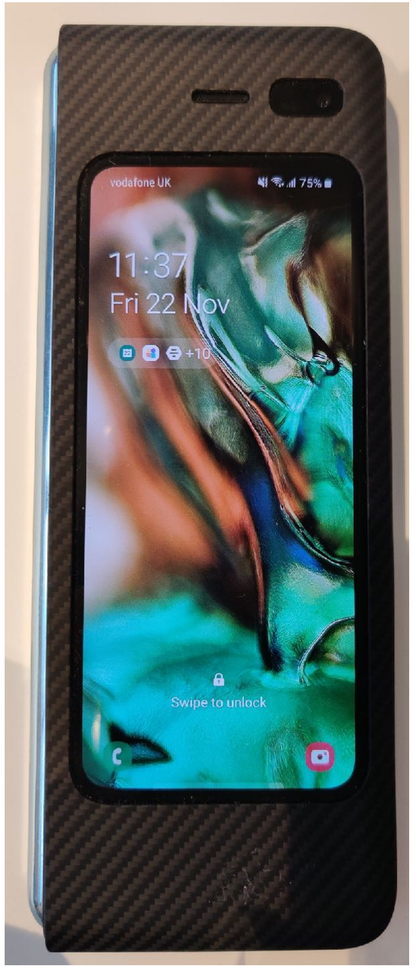
- Выбор кодека зависит от решаемой задачи
- Идеального кодека не существует
- Использовать Guetzli для динамических изображений ресурсоемко

Метаданные

Чистим файл от ненужной информации

В файле может быть много ненужной для нас информации.

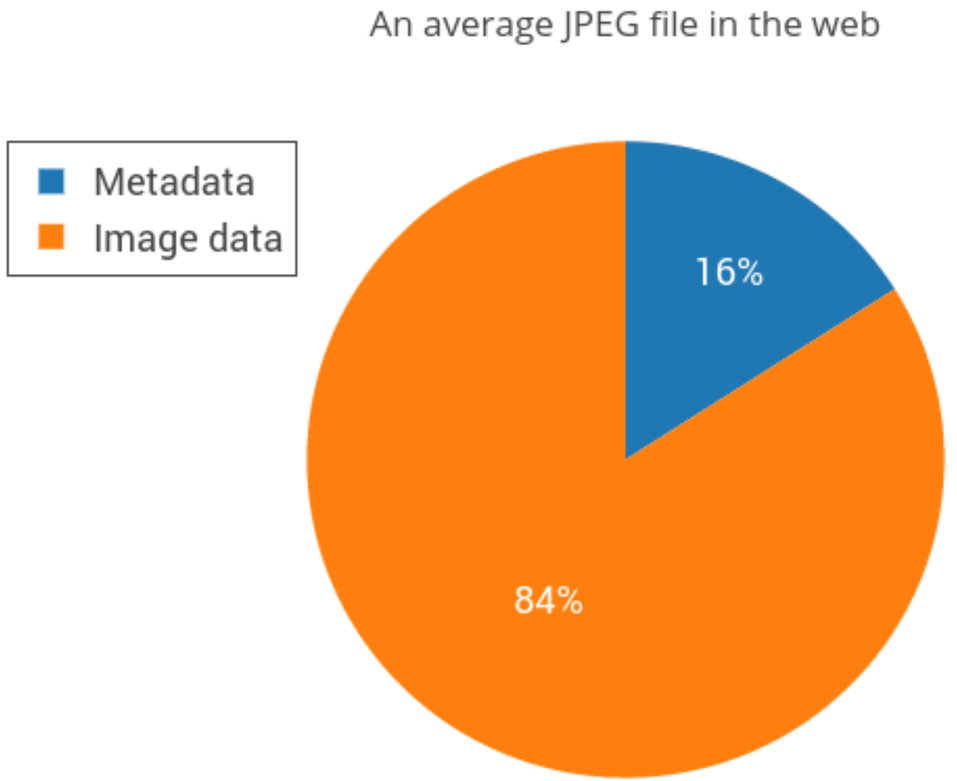
Last opened:	22 Nov 2019 at 11:46
Dimensions:	3000×4000
Device make:	OnePlus
Device model:	GM1913
Colour space:	RGB
Colour profile:	sRGB IEC61966-2.1
Focal length:	4.755 mm
Alpha channel:	No
Red-eye:	No
Metering mode:	Centre-weighted average
F number:	f/1.65
Exposure program:	Normal
Exposure time:	1/50
Latitude:	51° 30' 54.258" N
Longitude:	0° 8' 3.628" W



Чистим файл от ненужной информации

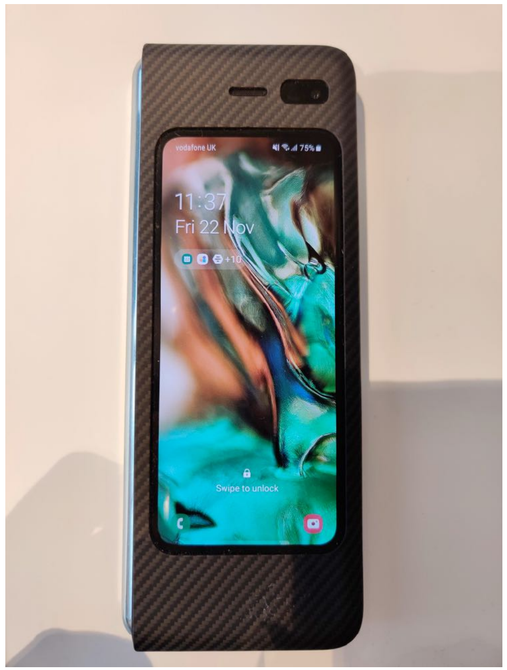
Большая часть избыточной информаци идет:

- для статических изображений — от графических редакторов
- для динамических изображений — от камеры

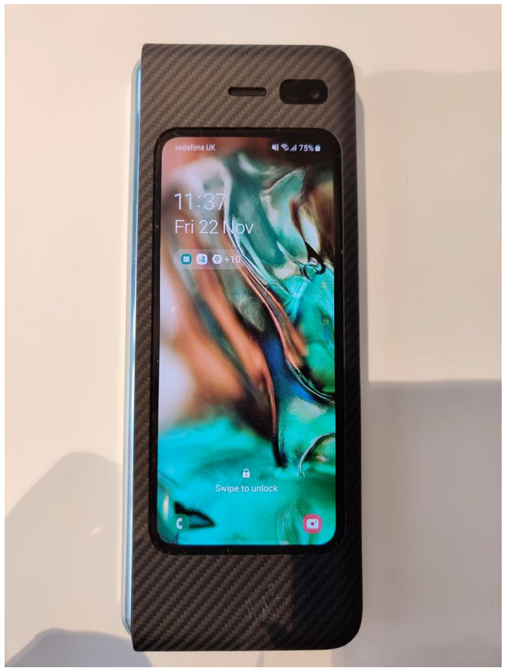


Чистим файл от ненужной информации

307KB



282KB

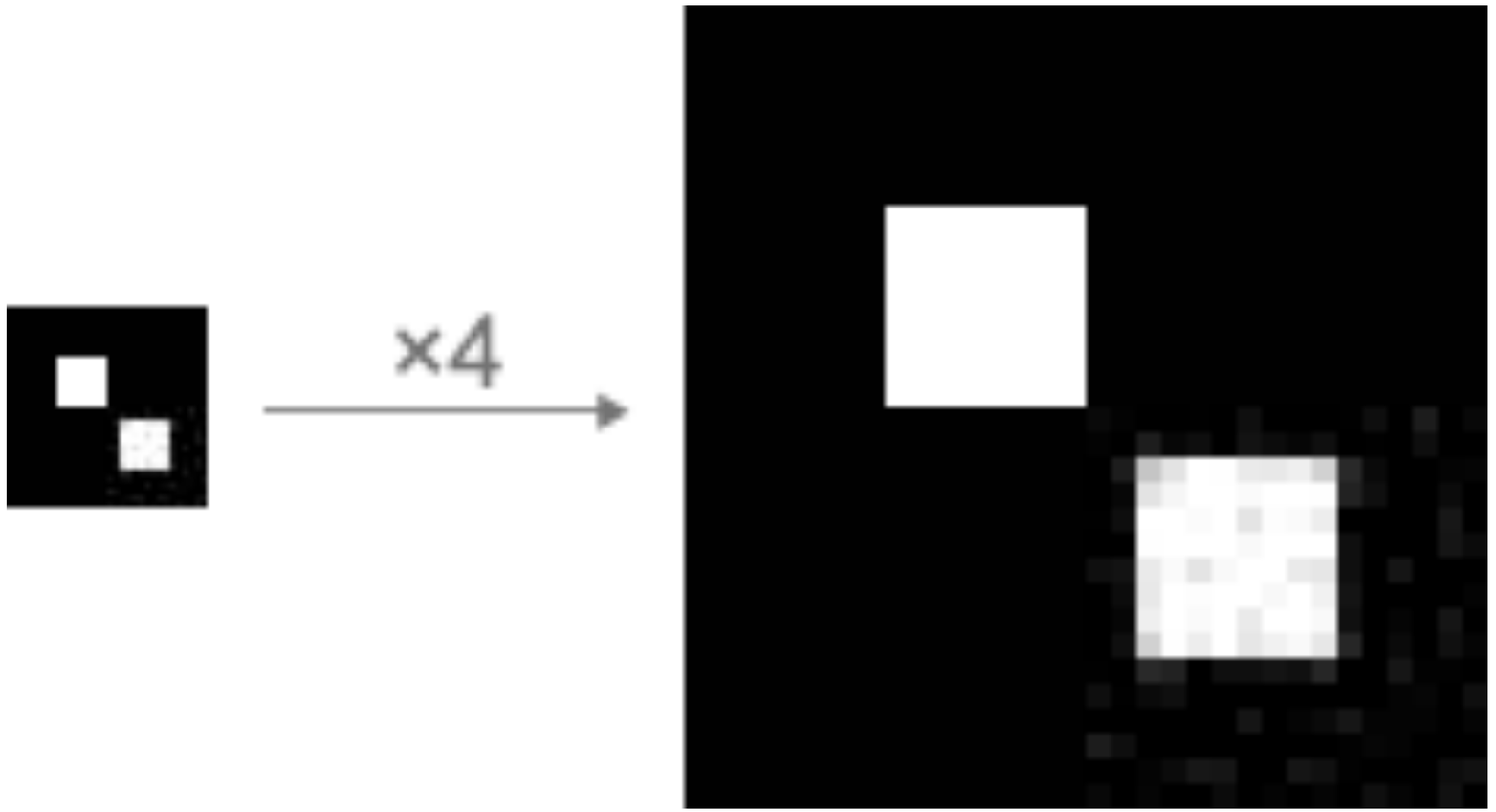


Выводы

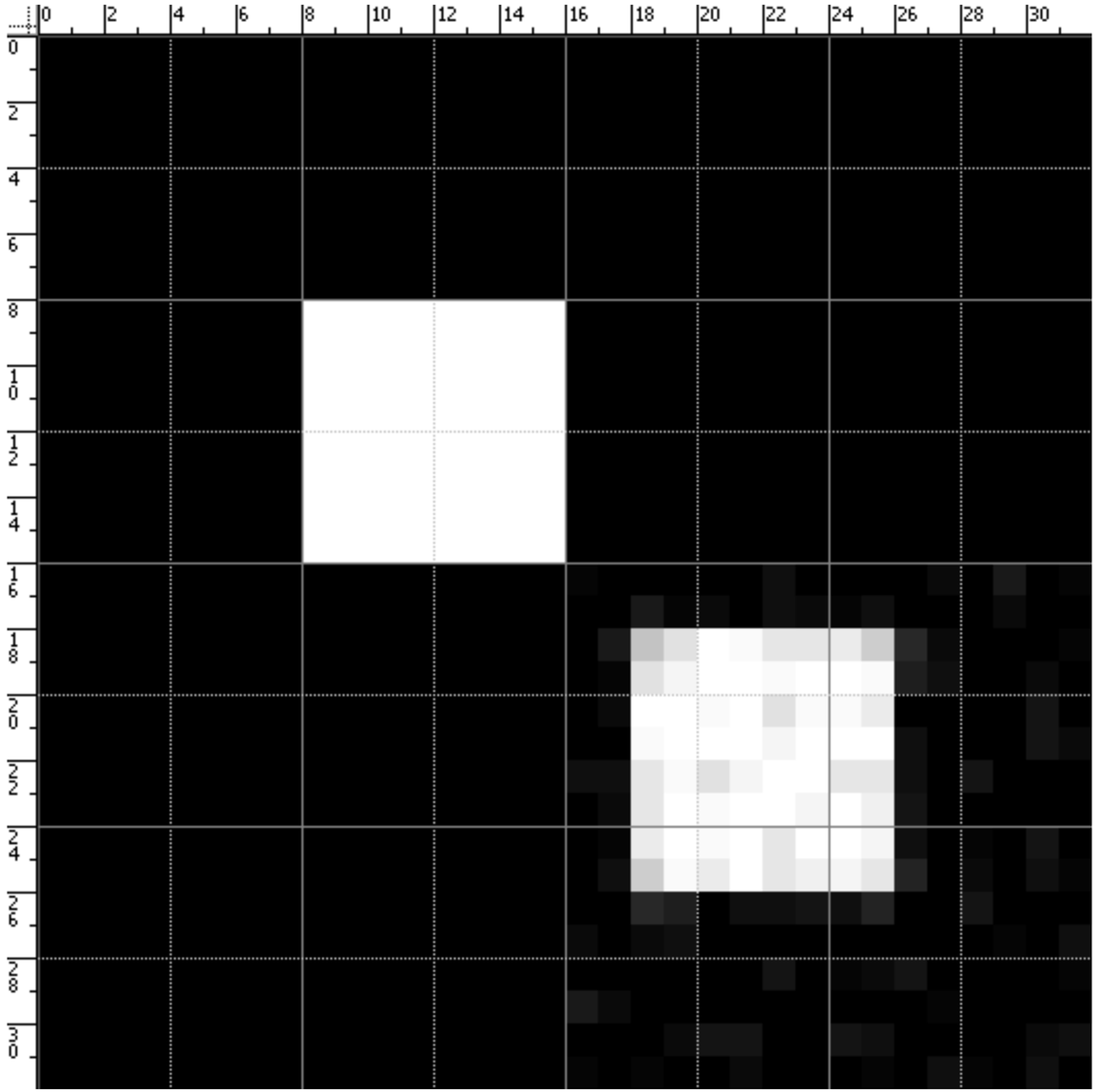
- Метаданные могут быть не нужны конечному пользователю
- Для кодеков удаление метаданных – это простая операция
- Не только динамика страдает от лишней информации

Выравнивание сетки

Выравнивание сетки



Выравнивание сетки

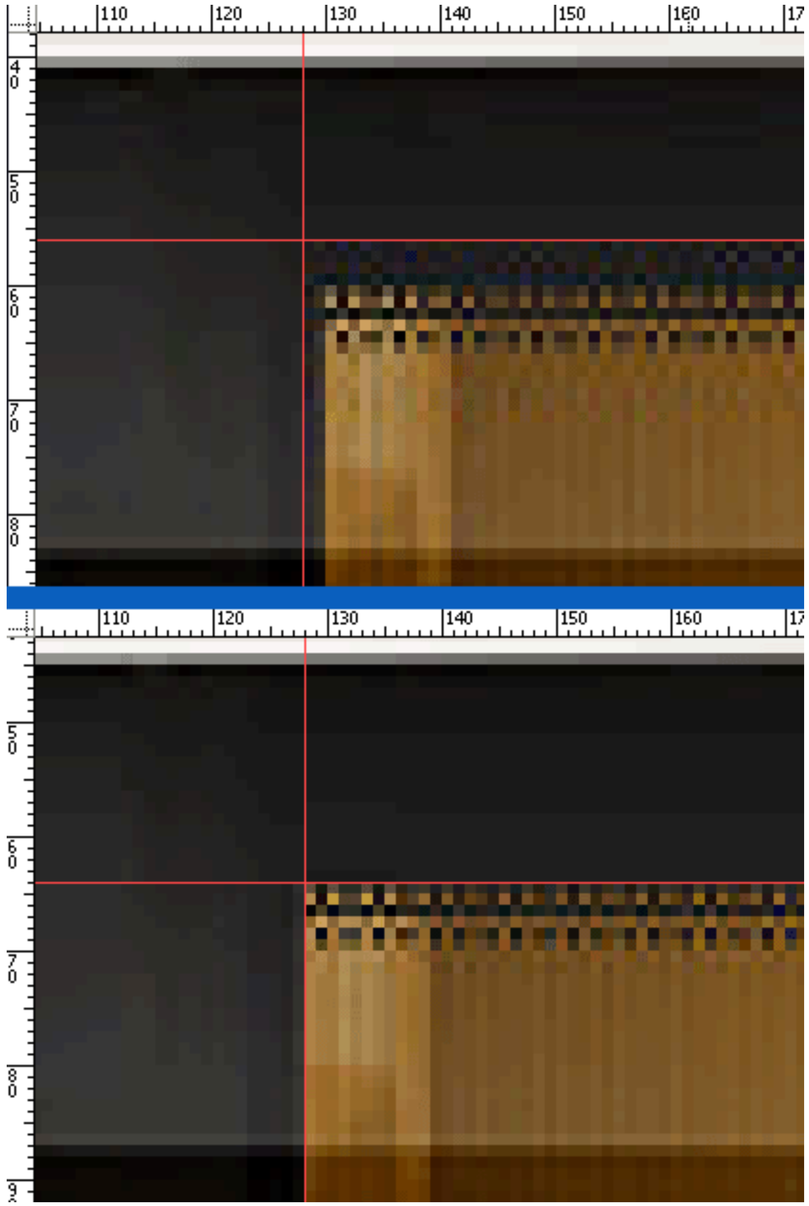


Выравнивание сетки

13.51 KB

12.65 KB

Выравнивание сетки



Выравнивание сетки

2,425B

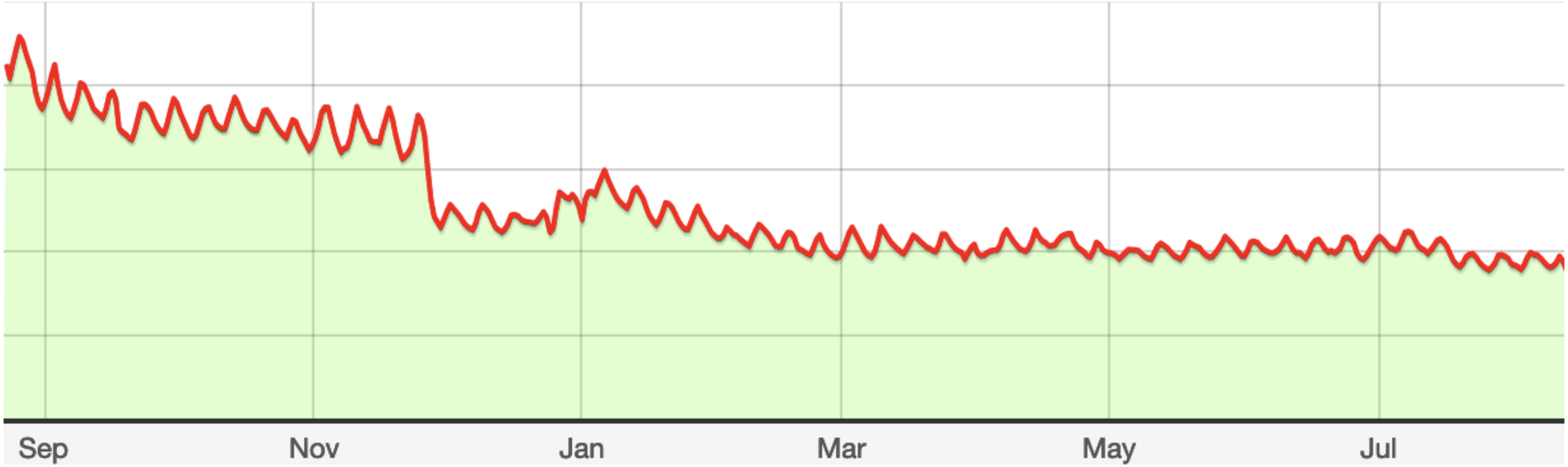
1,316B

Итоги

Что можно сделать?

1. Уменьшить разрешения изображений
2. Использовать оптимальный формат (если есть возможность)
3. Пользоваться особенностями формата
4. Выбрать оптимальные настройки кодека
5. Удалять метаданные
6. Немного подравнять статику

Ради чего?



Почитать

Ссылки

1. Как правильно грузить изображения в Android https://developer.android.com/topic/performance/graphics/load-bitmap
2. Декодирование JPEG для чайников https://habr.com/ru/post/102521/
3. Изобретаем JPEG https://habr.com/ru/post/206264/
4. Butteraugli https://github.com/google/butteraugli
5. Поддерживаемые форматы Android https://developer.android.com/guide/topics/media/media-formats
6. Оптические иллюзии http://brainden.com/color-illusions.htm
7. Способы кодирования Progressive Jpeg https://cloudinary.com/blog/progressive_jpegs_and_green_martians
8. Исследование Progressive Jpeg https://yuiblog.com/blog/2008/12/05/imageopt-4/
9. Влияние метаданных на размер файла https://dexecure.com/blog/impact-of-metadata-on-image-performance/
10. Использование Progressive Jpeg на проде https://engineering.fb.com/ios/faster-photos-in-facebook-for-ios/
11. Как работает Jpeg https://www.freecodecamp.org/news/how-jpg-works-a4dbd2316f35/
12. Уменьшение размера Jpeg https://medium.com/@duhroach/reducing-jpg-file-size-e5b27df3257c

Почитать

Ссылки

13. Libjpeg-turbo https://libjpeg-turbo.org/
14. Оптимизация изображений для WEB https://images.guide/
15. Матрицы квантования различных фотоаппаратов https://www.impulseadventure.com/photo/jpeg-quantization.html
16. Почему webp? https://insanelab.com/blog/web-development/webp-web-design-vs-jpeg-gif-png/
17. Mozjpeg и его преимущества https://libjpeg-turbo.org/About/Mozjpeg
18. Строение файла Jpeg https://parametric.press/issue-01/unraveling-the-jpeg/
19. Обзор mozjpeg 3.0 https://calendar.perfplanet.com/2014/mozjpeg-3-0/
20. Почему сабсэмплинг важен https://calendar.perfplanet.com/2015/why-arent-your-images-using-chroma-subsampling/
21. Что значат настройки качества в Photoshop https://photographylife.com/jpeg-compression-levels-in-photoshop-and-lightroom
22. Визуализация Дискретных косинусы преобразований http://weitz.de/dct/
23. Цветовая модель https://en.wikipedia.org/wiki/Color_model
24. Цветовое пространство https://en.wikipedia.org/wiki/Color_space

Почитать

Ссылки

25. Применение ДКП https://en.wikipedia.org/wiki/Discrete_cosine_transform#Common_applications
26. Guetzli https://en.wikipedia.org/wiki/Guetzli
27. Формулы и принципы кодирования JPEG https://en.wikipedia.org/wiki/JPEG#Encoding
28. YCbCr https://en.wikipedia.org/wiki/YCbCr
29. GIF https://ru.wikipedia.org/wiki/GIF
30. SSIM https://ru.wikipedia.org/wiki/SSIM
31. WebP https://ru.wikipedia.org/wiki/WebP
32. PSNR https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio
33. Типы метаданных в изображениях https://wp-rocket.me/blog/image-metadata-can-impact-web-performance-security/
34. Опыт улучшения изображений без потерь качества в YELP https://engineeringblog.yelp.com/2017/06/making-photos-smaller.html
35. Еще пару слов про Progressive https://calendar.perfplanet.com/2012/progressive-jpegs-a-new-best-practice/

Cheers!

n.kozlov@magiclab.co

techblog.badoo.com

badoo_tech