LIES, DAMN LIES, AND PERFORMANCE STUDIES

Stoyan Stefanov @stoyanstefanov

NYWebPerf Meetup July 26, 2022

phpied.com/progressive

MOTIVATION

- TWEET RE: PROGRESSIVE JPEG
- REPEAT THE 2008 STUDY
  HTTPS://WEB.ARCHIVE.ORG/WEB/20081216100242/HTTP://YUIBLOG.COM/BLOG/2008/12/05/IMAGEOPT-4/
- ENCOURAGE MORE STUDIES, DONE RESPONSIBLY

STOYAN STEFANOV

- CATCHPOINT, WEBPAGETEST
- INTO WEB PERFORMANCE SINCE 200X
- PERFPLANET.COM
- BOOKS: JAVASCRIPT PATTERNS, REACT: UP AND RUNNING…

PERFORMANCE STUDIES

- 100ms earn you 1% more money?
- after 1s, XX% drop in [business metric] for every 0.5 seconds?
- UI must respond in Xms or else!?

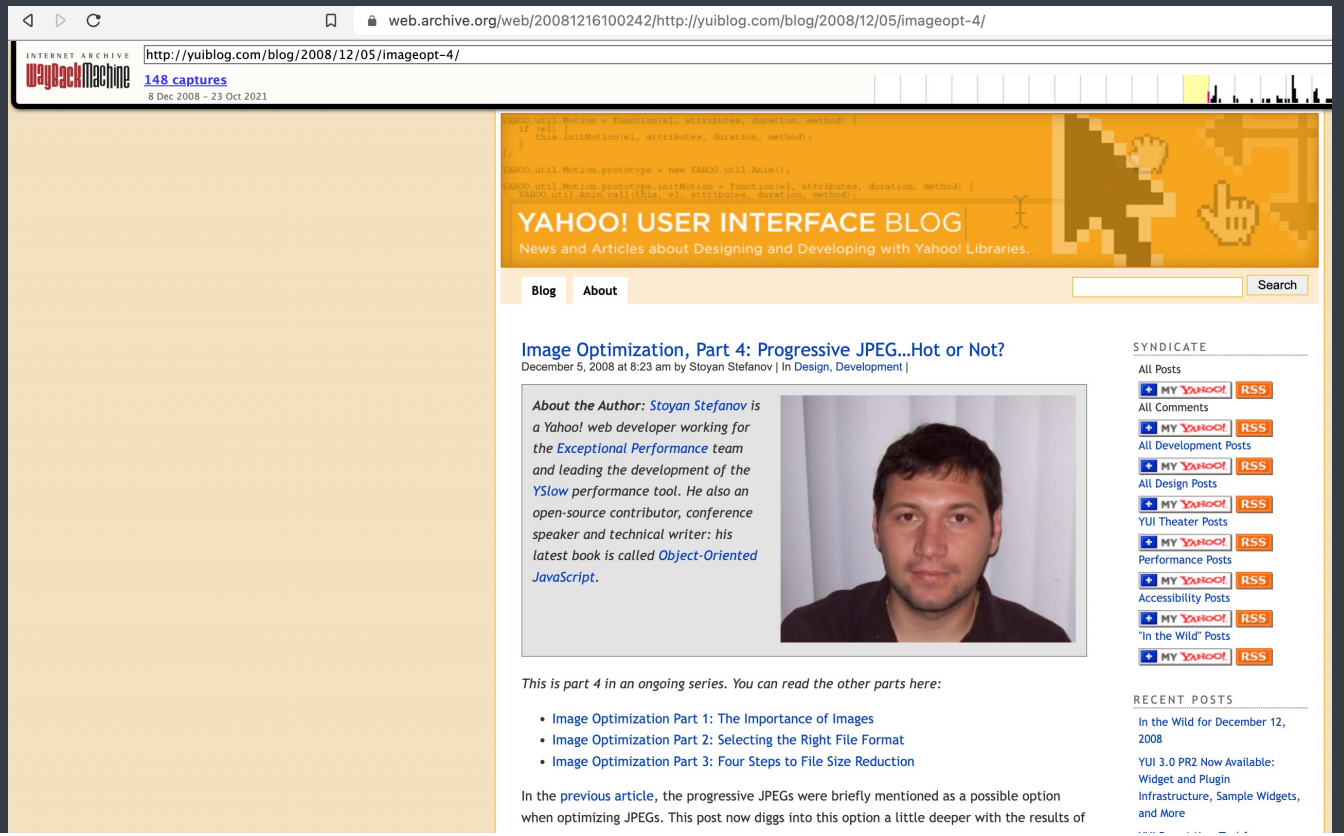
PERFORMANCE STUDIES

- Poorly done/documented studies are a disservice to the perf community
- Critical reading
- No data, no methodology, no sample size = garbage
- Interviews = rubbish

PERFORMANCE STUDIES

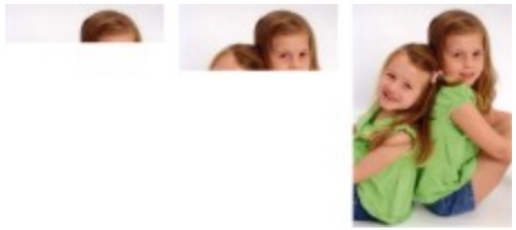
- WPOSTATS.COM
- CURATED BY TIM KADLEC AND TAMMY EVERTS
- QUALITY, BUT… USE FOR INSPIRATION
- DIY

2008 PROGRESSIVE IMAGE STUDY



Baseline vs. progressive JPEGs

Baseline are the “normal” JPEGs, the type of JPEG that all image programs write by default. The browsers load them top-to-bottom as more of the image information comes down the wire.



Loading a baseline JPEG, click to enlarge

Progressive JPEGs are another type of JPEGs, they are rendered, as the name suggests, progressively. First you see a low quality version of the whole image. Then, as more of the image information arrives over the network, the quality gradually improves.

Loading a progressive JPEG, click to enlarge


MY INTERNET IS SO SLOW THAT

Watch Video:
https://res.cloudinary.com/cloudinary/video/upload/vc_auto/non_progressive_vs_progressive_jpeg.mp4

PROGRESSIVE JPEG

- Cons
  - People may judge before the image is loaded
  - Cognitive load?

STILL WITH THE JPEG?!

- AVIF? WEBP? JPEG 2000? JPEG XR?
- YES JPEG! TILL JPEG XL, AKA JXL, ROLLS ALONG

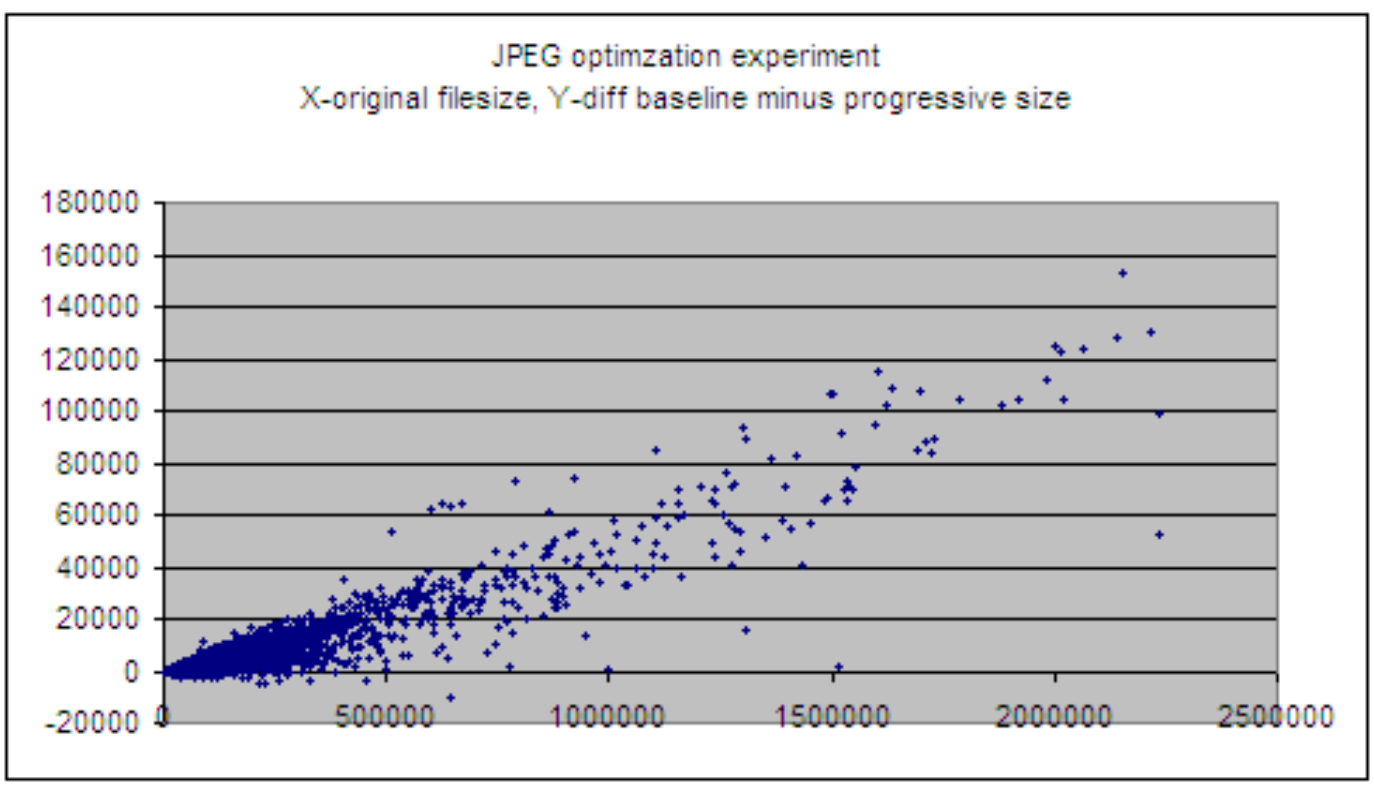
2008 PROGRESSIVE IMAGE STUDY GOALS

- Is progressive smaller
- If not, when
- Curiosities: speed of encoding
- Curiosities: what's out there compared to what it could be
- Use JPEGTran

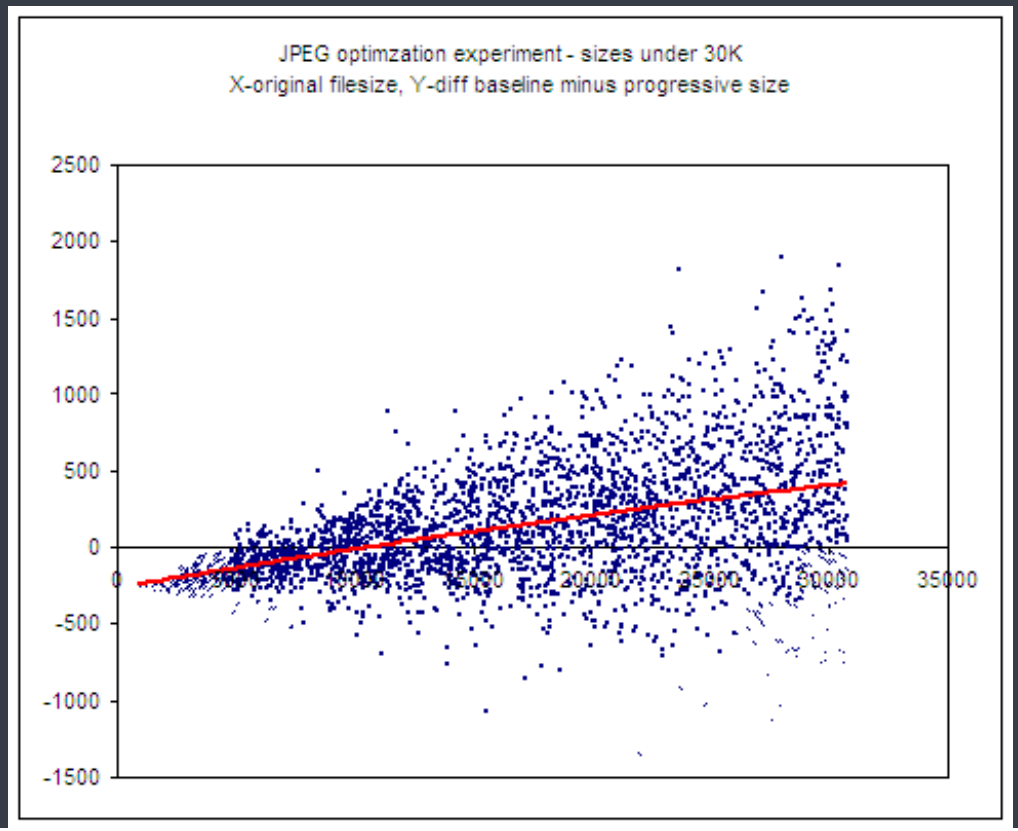
2008 PROGRESSIVE IMAGE STUDY RAW DATA

- Quote: Yahoo!... image search API. I used it to find images that match a number of queries, such as "kittens", "puppies", "monkeys", "baby", "flower", "sunset".. 12 queries in total.
- Quote: After the cleanup there were 10360 images to work with, images of all different dimensions and quality, and best of all, real life images from live web sites.

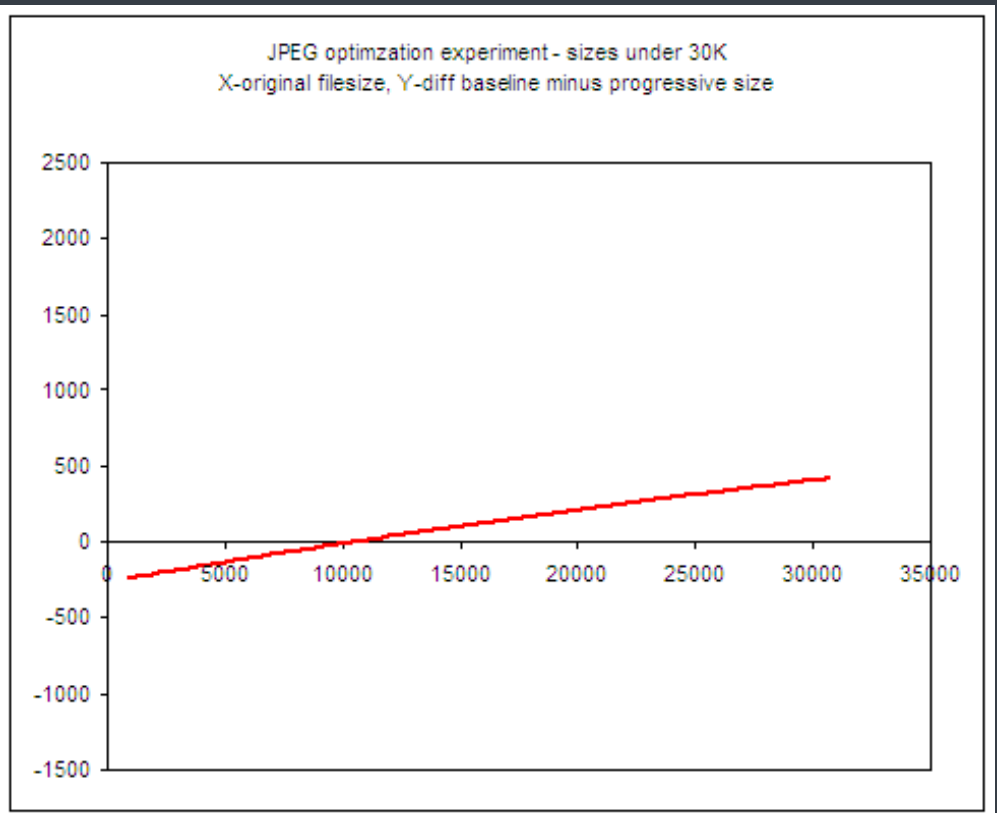
2008 IMAGE STUDY RESULTS



2008 IMAGE STUDY RESULTS



2008 IMAGE STUDY RESULTS



2008 PROGRESSIVE IMAGE STUDY RESULTS

- Files over 10K the progressive JPEG will give you a better compression (in 94% of the cases)
- baseline JPEG: 9.04% median savings
- progressive JPEG 11.45%
- baseline encoding roughly 20% faster

2022

OPTIONS FOR COLLECTING DATA TO STUDY

- Small sets of purpose-created images
- Alexa 1000 sites
- Browse the web
  - with a proxy
  - export network activity from the browser's devtools
- HTTPArchive

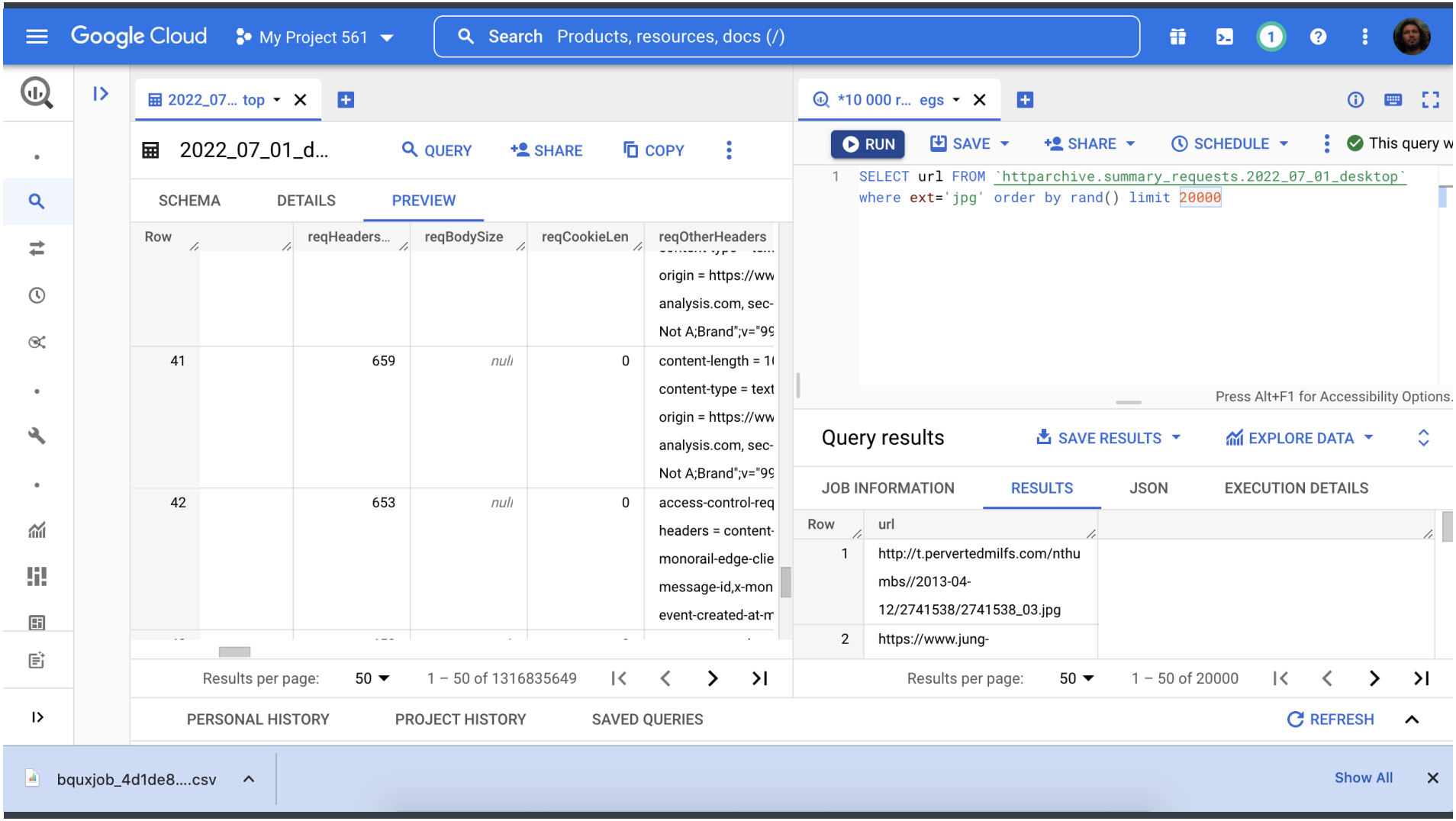
HTTPARCHIVE

- WEBPAGETEST PRIVATE INSTANCE
- RUN MONTHLY ON MILLIONS OF SITES
- YOU NEED BIG QUERY TO EXPLORE THE DATA
- BARRY POLLARD'S INTRO: HTTPS://GITHUB.COM/HTTPARCHIVE/HTTPARCHIVE.ORG/BLOB/MAIN/DOCS/GETTINGSTARTED_BIGQUERY.MD

HTTPARCHIVE

```
SELECT url FROM
`httparchive.summary_requests.2022_07_01_desktop`
WHERE ext='jpg' ORDER BY RAND() LIMIT 20000
```

- Export as CSV


Google Cloud
My Project 561
Search Products, resources, docs (/)
2022_07... top
2022_07_01_d...
QUERY
SHARE
COPY
SCHEMA
DETAILS
PREVIEW
Row
reqHeaders...
reqBodySize
reqCookieLen
reqOtherHeaders
origin = https://ww
analysis.com, sec-
Not A;Brand";v="99
41
659
null
0
content-length = 1(
content-type = text
origin = https://ww
analysis.com, sec-
Not A;Brand";v="99
42
653
null
0
access-control-req
headers = content-
monorail-edge-clie
message-id,x-mon
event-created-at-m
Results per page: 50
1 – 50 of 1316835649
*10 000 r... egs
RUN
SAVE
SHARE
SCHEDULE
This query w
1 SELECT url FROM `httparchive.summary_requests.2022_07_01_desktop`
where ext='jpg' order by rand() limit 20000
Press Alt+F1 for Accessibility Options.
Query results
SAVE RESULTS
EXPLORE DATA
JOB INFORMATION
RESULTS
JSON
EXECUTION DETAILS
Row
url
1
http://t.pervertedmilfs.com/nthu
mbs//2013-04-
12/2741538/2741538_03.jpg
2
https://www.jung-
Results per page: 50
1 – 50 of 20000
PERSONAL HISTORY
PROJECT HISTORY
SAVED QUERIES
REFRESH
bquxjob_4d1de8....csv
Show All

```
http://example.org/nthumbs//2013-04-12/2741538/2741538_03.jpg
https://example.orgfileadmin/templates/art/standard/startseite/Multimedia_ohne-Text.jpg
http://example.orgr/skin/upload/%EB%B0%B0%EC%88%98%EC%97%B0.jpg
https://example.org/s-ghux11kp0r/images/stencil/80w/products/1029/1333/1032BKE__37343.1647888080.jpg?c=1
http://example.org/images/home/type01s.jpg
https://example.org/wp-content/uploads/sicurezza-protezioni-antinfortunistiche-macchinari-industriali-milper-4.jpg
https://example.org/sites/default/files/styles/home_page_poster/public/2021-11/Elysian_Poster_Locations.jpg?itok=tvxP
https://example.org/media/wysiwyg/marki/bio-oil-logo.jpg
https://example.org/wp-content/uploads/2019/04/4.jpg
https://example.org/image/cache/catalog/vktrade/banner/proizv/jlg-logo-130x100.jpg
https://example.org/s/files/1/0313/0087/8468/products/pg_286a1613-de4d-4753-9ef3-bc6b036f509c_grande.jpg?v=1626604964
https://example.org/-_6kc1rmzI7A/YEou2XpMSkI/AAAAAAAAFCU/SYvksP1ANVAxggLl47ckyBJXj9oCuDbXgCNcBGAsYHQ/w640-h426/anupam-
https://example.org/c/11-menu_default/equipamentos-semi-novos.jpg
https://example.org/198.71.233.168/mvu.1d6.myftpupload.com/wp-content/uploads/2020/10/mini-IMG_4193-150x150.jpg
https://example.org/data/images/801003471.jpg
http://example.org/2015/11/b/0/bbedcc72d5.jpg
```

DOWNLOAD IMAGES

- 20000 URLs
- split -l 500
- Files such as urls1, urls2...
  - `wget -T 30 -t 1 -i ../urls`
  - `wget -T 30 -t 1 -i ../urls2`
- 30s timeout, no retries
- 1,465,606,852 bytes (1.58 GB on disk) for 14,511 items

CLEANUP

- Rename sequentially:

```
ls -v | cat -n | while read n f; do mv -n "$f" "$n.jpg"; done
```

- ImageMagick to figure out what we have

```
identify -regard-warnings *.jpg > ../log.txt
```

- Node script to look into log and find non-jpegs
- Delete a bunch of non-jpegs:

```
node nonjpeg.js > rm.sh
```

```
5723.jpg JPEG 127x180 127x180+0+0 8-bit sRGB 7462B 0.000u 0:00.000
5724.jpg JPEG 100x100 100x100+0+0 8-bit sRGB 3146B 0.000u 0:00.000
5725.jpg JPEG 40x40 40x40+0+0 8-bit sRGB 1240B 0.000u 0:00.000
5726.jpg JPEG 300x400 300x400+0+0 8-bit sRGB 37233B 0.010u 0:00.002
5727.jpg JPEG 307x198 307x198+0+0 8-bit sRGB 22594B 0.000u 0:00.001
5728.jpg JPEG 500x329 500x329+0+0 8-bit sRGB 90724B 0.000u 0:00.004
5729.jpg JPEG 900x500 900x500+0+0 8-bit sRGB 67179B 0.010u 0:00.008
573.jpg JPEG 571x316 571x316+0+0 8-bit sRGB 278354B 0.010u 0:00.005
5730.jpg JPEG 1080x1080 1080x1080+0+0 8-bit sRGB 68461B 0.010u 0:00.010
5731.jpg JPEG 480x463 480x463+0+0 8-bit sRGB 119999B 0.000u 0:00.007
5732.jpg JPEG 480x398 480x398+0+0 8-bit sRGB 20635B 0.010u 0:00.002
5733.jpg JPEG 400x240 400x240+0+0 8-bit sRGB 17729B 0.000u 0:00.001
5734.jpg JPEG 230x230 230x230+0+0 8-bit sRGB 6168B 0.000u 0:00.001
5735.jpg JPEG 1470x534 1470x534+0+0 8-bit sRGB 242022B 0.010u 0:00.010
```

… AND MORE CLEANUP

- JPEG BUT UNUSABLE
- MANUAL
  - ERRORS FROM `IDENTIFY -REGARD-WARNINGS`
  - ERRORS FROM RUNNING JPEGTRAN
- FROM 14,511 TO 14,126
- 2.6% TRASH

CURIOSITY

- How many are already progressive?

```
identify -format "%f,%[interlace]\n" *.jpg > ../prog-or-not.csv

node prog-or-not.js

{ prog: 4230, base: 9896 }
```

- 29.94% progressive in the source data

CURIOSITY

- Sizes: average 77K, median 26K
- Outlier: 156 bytes 1x1 ("spacer"?)
- Outlier: 12.2MB 5184x3456 with quality 99

LOSSLESS JPEG OPTIMIZATION STRATEGIES

1. JUST RUN JPEGTRAN ALREADY
    1.1. ALL DEFAULTS PLUS `-COPY NONE` AND `-OPTIMIZE`
    1.2. ALWAYS USE `-PROGRESSIVE` BECAUSE THE 2008 STUDY SAYS YOU WIN 94% OF THE TIMES
    1.3. BRUTE FORCE: TRY BASELINE AND PROGRESSIVE, PICK THE SMALLER (AS THE 2008 STUDY RECOMMENDS)
2. JUST RUN MOZJPEG ALREADY - IT DOES THE BRUTE FORCE

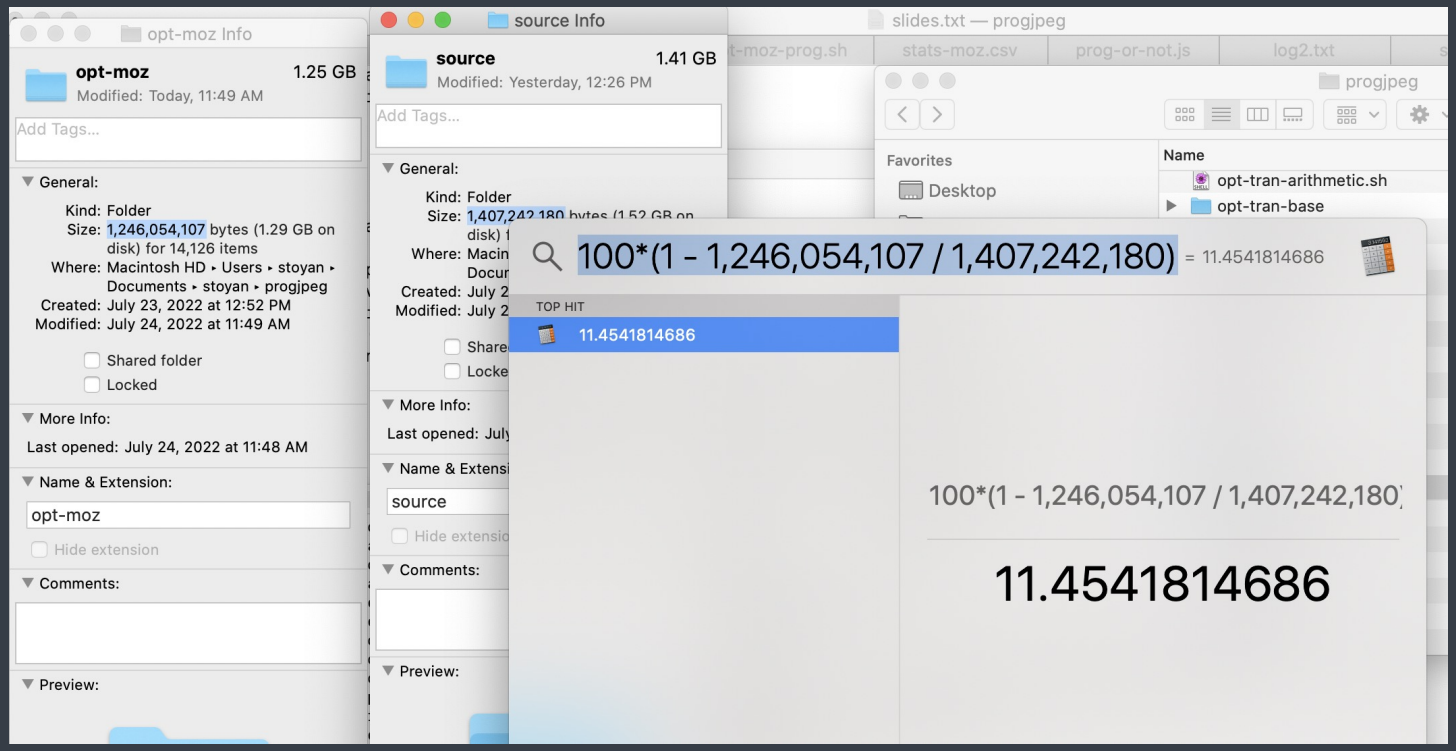
- OVERACHIEVERS ONLY: RUN JPEGMINI ON THE RESULT

START OPTIMIZING

- `node opt.sh.js > opt.sh`
- Variations:
  1. jpegtran baseline
  2. jpegtran progressive
  3. mozjpeg force baseline
  4. mozjpeg force progressive
  5. mozjpeg default

```
# BASELINE JPEGTRAN
JPEGTRAN -COPY NONE -OPTIMIZE ${SRC} > ${DEST}
# PROGRESSIVE JPEGTRAN
JPEGTRAN -COPY NONE -OPTIMIZE -PROGRESSIVE ${SRC} > ${DEST}
# BASELINE MOZJPEG
/USR/LOCAL/OPT/MOZJPEG/BIN/JPEGTRAN -REVERT -COPY NONE -OPTIMIZE
${SRC} > ${DEST}
# PROGRESSIVE MOZJPEG
/USR/LOCAL/OPT/MOZJPEG/BIN/JPEGTRAN -COPY NONE -PROGRESSIVE ${SRC} >
${DEST}
# DEFAULT MOZJPEG
/USR/LOCAL/OPT/MOZJPEG/BIN/JPEGTRAN -COPY NONE ${SRC} > ${DEST}
```

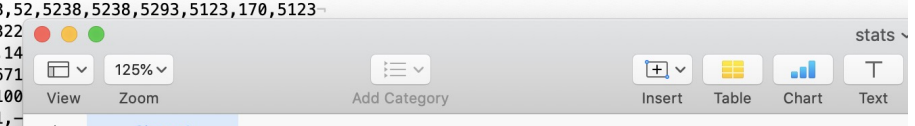
FIRST IMPRESSION: DEPRESSION



COLLECT BETTER STATS

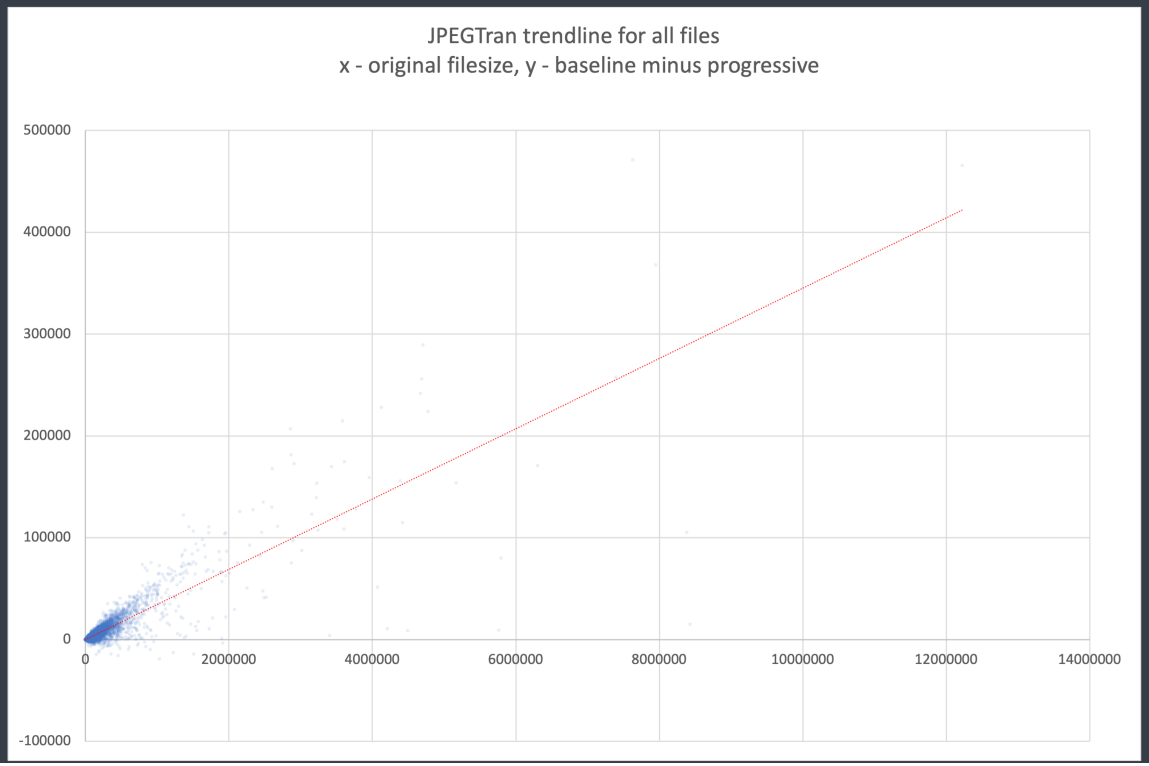
NODE STATS.JS > STATS.CSV

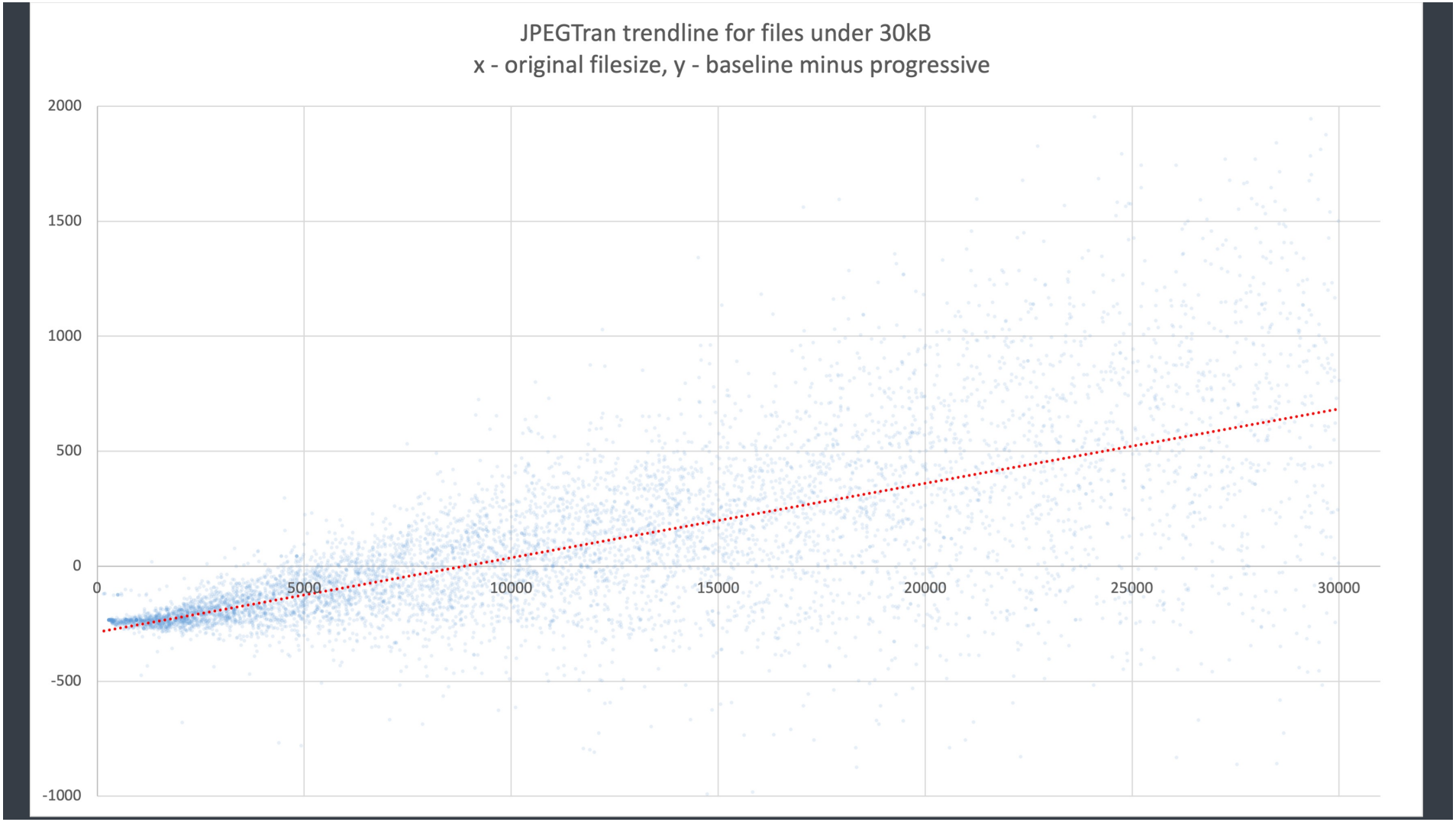
```
file,orig,tranbase,tranprog,trandiff,tran,tranopt,mozbase,mozprog,mozdiff,moz
1,2269,2236,2436,-200,2236,2236,2233,2239,-6,2239
2,2827,2138,2369,-231,2138,2138,2138,2205,-67,2205
3,51264,52092,51245,847,51245,51245,52111,50379,1732,50379
4,5293,5290,5238,52,5238,5238,5293,5123,170,5123
5,34578,34580,3322
6,140938,149157,14
7,16391,16401,1671
8,13939,11497,1100
9,3550,3165,3351,-
10,19974,19406,195
11,10280,10069,102
12,1573,1573,1845,
13,96194,103520,96
14,32895,33681,325
15,36161,34652,346
16,59769,59127,586
17,107662,111828,1
18,3722,3678,3834,
19,45783,47051,457
20,5013,4412,4616,
21,37640,7677,7572
22,26953,26824,268
23,192919,197765,1
24,25589,25540,248
25,37132,39660,382
26,17776,16928,164
27,39392,38377,382
28,12756,12333,119
```



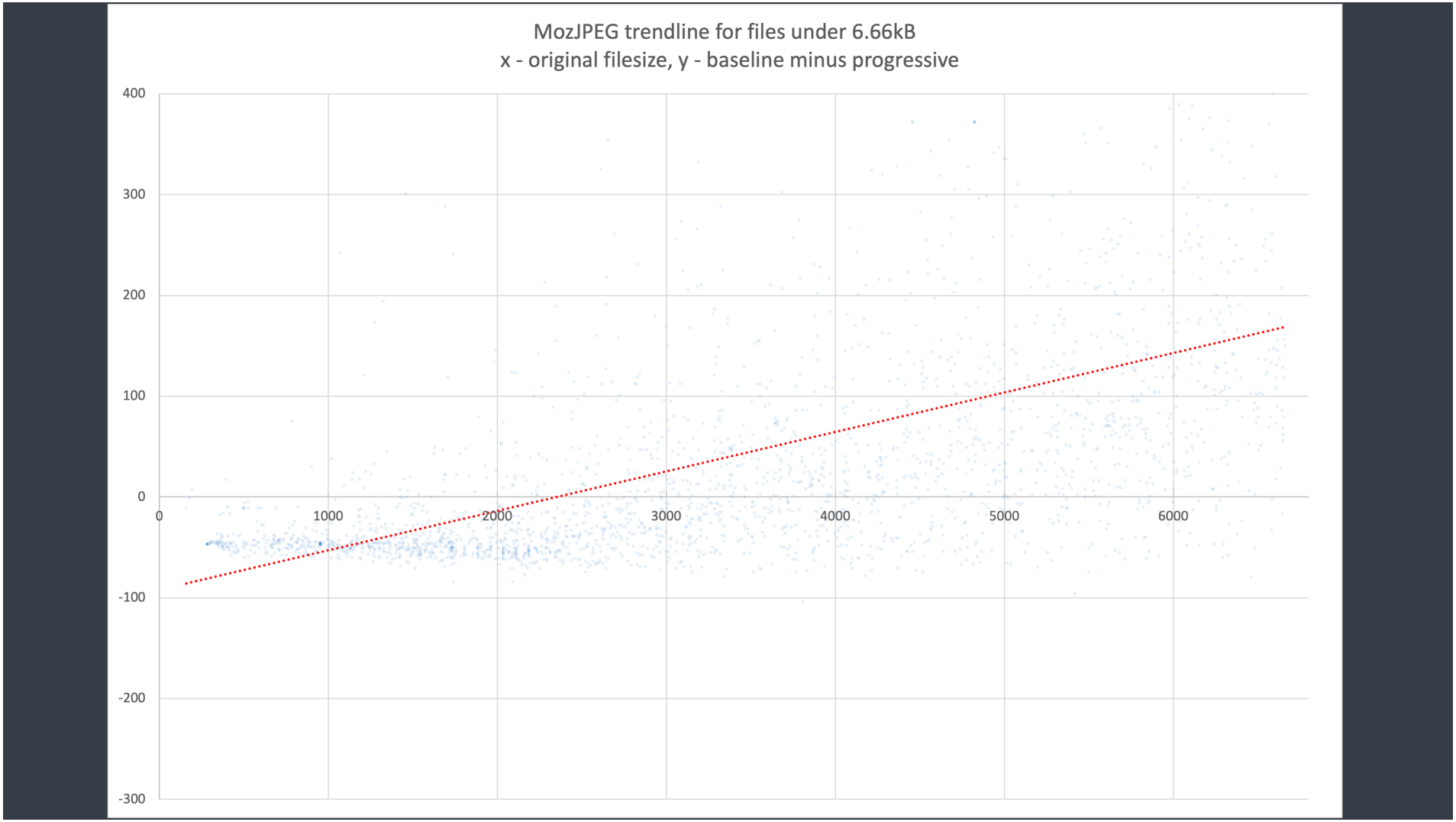
file	orig	tranbase	tranprog	trandiff	tran	tranopt	mozbase	mozprog	mozdiff	moz
1	2269	2236	2436	-200	2236	2236	2233	2239	-6	2239
2	2827	2138	2369	-231	2138	2138	2138	2205	-67	2205
3	51264	52092	51245	847	51245	51245	52111	50379	1732	50379
4	5293	5290	5238	52	5238	5238	5293	5123	170	5123
5	34578	34580	33224	1356	33224	33224	34594	32773	1821	32773
6	140938	149157	142234	6923	142234	140938	149194	140316	8878	140316
7	16391	16401	16712	-311	16401	16391	16396	16287	109	16287
8	13939	11497	11002	495	11002	11002	11504	10551	953	10551
9	3550	3165	3351	-186	3165	3165	3168	3187	-19	3187
10	19974	19406	19531	-125	19406	19406	19418	19352	66	19352

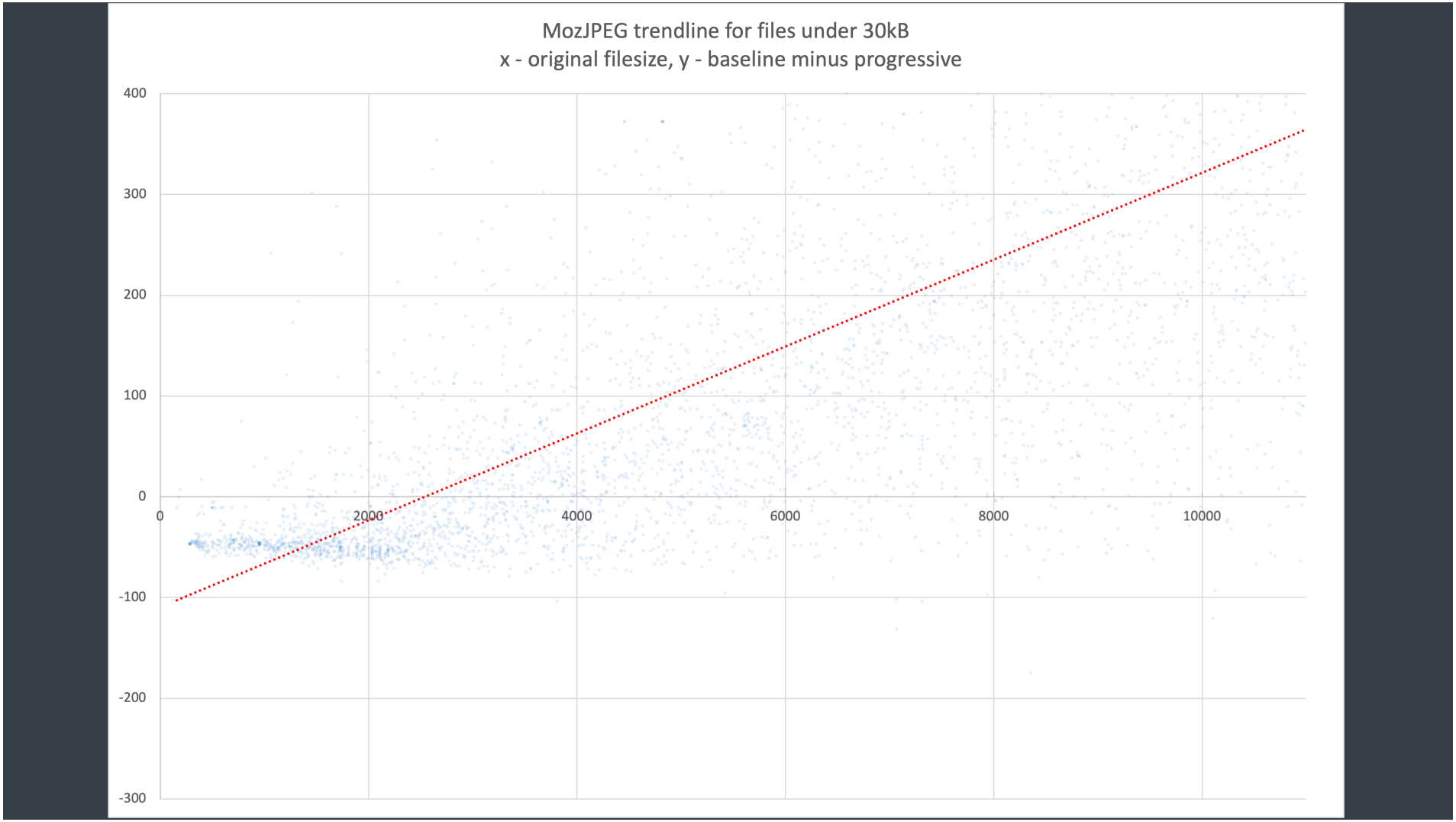
RESULTS

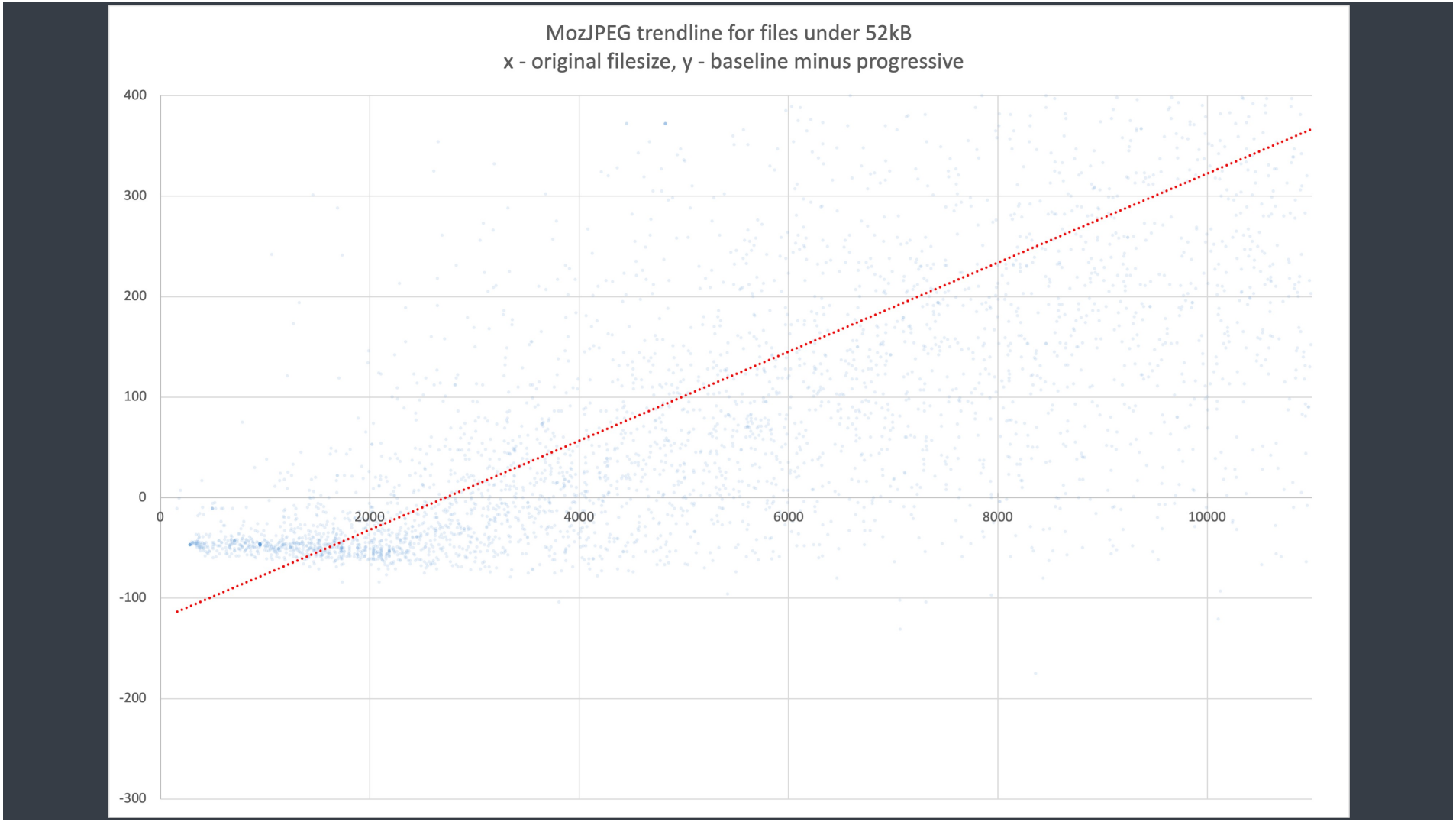

JPEGTran trendline for all files
x - original filesize, y - baseline minus progressive
500000
400000
300000
200000
100000
0
-100000
0
2000000
4000000
6000000
8000000
10000000
12000000
14000000


JPEGTran trendline for files under 30kB
x - original filesize, y - baseline minus progressive
2000
1500
1000
500
0
-500
-1000
0
5000
10000
15000
20000
25000
30000


JPEGTran trendline for files under 52kB
x - original filesize, y - baseline minus progressive
2000
1500
1000
500
0
-500
-1000
0
5000
10000
15000
20000
25000
30000


MozJPEG trendline for files under 6.66kB
x - original filesize, y - baseline minus progressive
400
300
200
100
0
-100
-200
-300
0
1000
2000
3000
4000
5000
6000


MozJPEG trendline for files under 30kB
x - original filesize, y - baseline minus progressive
400
300
200
100
0
-100
-200
-300
0
2000
4000
6000
8000
10000


MozJPEG trendline for files under 52kB
x - original filesize, y - baseline minus progressive
400
300
200
100
0
-100
-200
-300
0
2000
4000
6000
8000
10000

RESULTS

- Similar to 2008
- Progressive is almost always better, but not always
- For file sizes around under 8K, baseline encoding has a better chance (was 10K)
- But when using MozJPEG the trendline is further down, around 2.5K

ARE WE FOLLOWING THE EASIEST WEB PERF BEST PRACTICE?

LOSSLESS JPEG OPPORTUNITIES

1. Just run JPEGTran already
    1.1. baseline - 6.91% savings (9.04% in 2008)
    1.2. progressive - 10.15% (11.45% in 2008)
    1.3. Brute force - 10.40%
2. Just run MozJPEG already - 11.44%

Overachievers only: run JPEGMini - 11.58%

CURIOSITY

- 29.94% PROGRESSIVE JPEGS IN THE SOURCE DATA
- 96.91% AFTER MOZJPEG HAD ITS SAY

PARTING WORDS

- Let's just run MozJPEG already
- Let's do more studies, we need diversity
- Ask: Does study X apply to me?
- Hint: Run WebPageTest experiments

PEOPLE TO FOLLOW FOR ALL THINGS IMAGES

- **Jon Sneyers** @jonsneyers
- **Kornel Lesiński** @kornelski

THANK YOU!

Image credits and further reading:

- cloudinary.com/blog/how_jpeg_xl_compares_to_other_image_codecs
- cloudinary.com/blog/progressive_jpegs_and_green_martians
- calendar.perfplanet.com/2014/mozjpeg-3-0/
- calendar.perfplanet.com/2015/upgrading-jpegtran-to-mozjpeg/
- phpied.com/progressive