

LIES, DAMN LIES, AND PERFORMANCE STUDIES

STOYAN STEFANOV @STOYANSTEFANOV

NYWEBPERF MEETUP JULY 26, 2022

PHPIED.COM/PROGRESSIVE

MOTIVATION

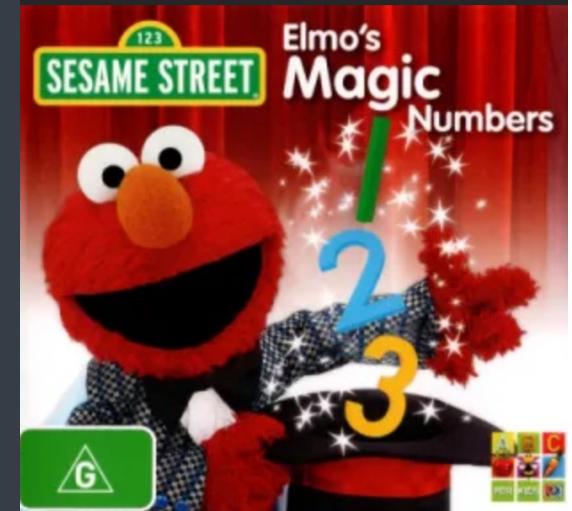
- TWEET RE: PROGRESSIVE JPEG
- REPEAT THE 2008 STUDY
[HTTPS://WEB.ARCHIVE.ORG/WEB/20081216100242/HTTP://YUIBLOG.COM/BLOG/2008/12/05/IMAGEOPT-4/](https://web.archive.org/web/20081216100242/http://yuiblog.com/blog/2008/12/05/imageopt-4/)
- ENCOURAGE MORE STUDIES, DONE RESPONSIBLY

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

INTERNET ARCHIVE Wayback Machine <http://yuiblog.com/blog/2008/12/05/imageopt-4/> 148 captures 8 Dec 2008 – 23 Oct 2021

The screenshot shows a web browser window displaying a blog post from the YAHOO! User Interface Blog. The page has a yellow header with the title "YAHOO! USER INTERFACE BLOG" and a subtitle "News and Articles about Designing and Developing with Yahoo! Libraries.". Below the header, there are two tabs: "Blog" (which is selected) and "About". A search bar is also present. The main content area features a large, partially loaded progressive JPEG image of a man's face. To the left of the image, there is a sidebar with author information and a list of related articles. On the right side, there are syndication links for various categories like All Posts, All Comments, and YUI Theater Posts, each with an RSS link. At the bottom, there is a section for recent posts.

YAHOO! USER INTERFACE BLOG
News and Articles about Designing and Developing with Yahoo! Libraries.

Blog **About** Search

Image Optimization, Part 4: Progressive JPEG...Hot or Not?
December 5, 2008 at 8:23 am by Stoyan Stefanov | In Design, Development |

About the Author: Stoyan Stefanov is a Yahoo! web developer working for the [Exceptional Performance](#) team and leading the development of the [YSlow](#) performance tool. He also an open-source contributor, conference speaker and technical writer: his latest book is called [Object-Oriented JavaScript](#).



This is part 4 in an ongoing series. You can read the other parts here:

- [Image Optimization Part 1: The Importance of Images](#)
- [Image Optimization Part 2: Selecting the Right File Format](#)
- [Image Optimization Part 3: Four Steps to File Size Reduction](#)

In the [previous article](#), the progressive JPEGs were briefly mentioned as a possible option when optimizing JPEGs. This post now digs into this option a little deeper with the results of

SYNDICATE

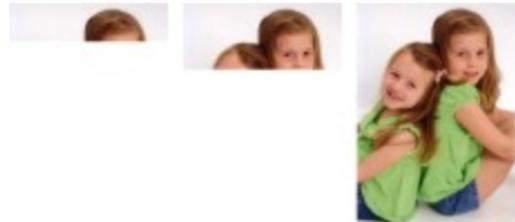
All Posts [RSS](#)
All Comments [RSS](#)
All Development Posts [RSS](#)
All Design Posts [RSS](#)
YUI Theater Posts [RSS](#)
Performance Posts [RSS](#)
Accessibility Posts [RSS](#)
"In the Wild" Posts [RSS](#)
My YAHOO! [RSS](#)

RECENT POSTS

In the Wild for December 12, 2008
YUI 3.0 PR2 Now Available: Widget and Plugin Infrastructure, Sample Widgets, and More

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](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?!



XL

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

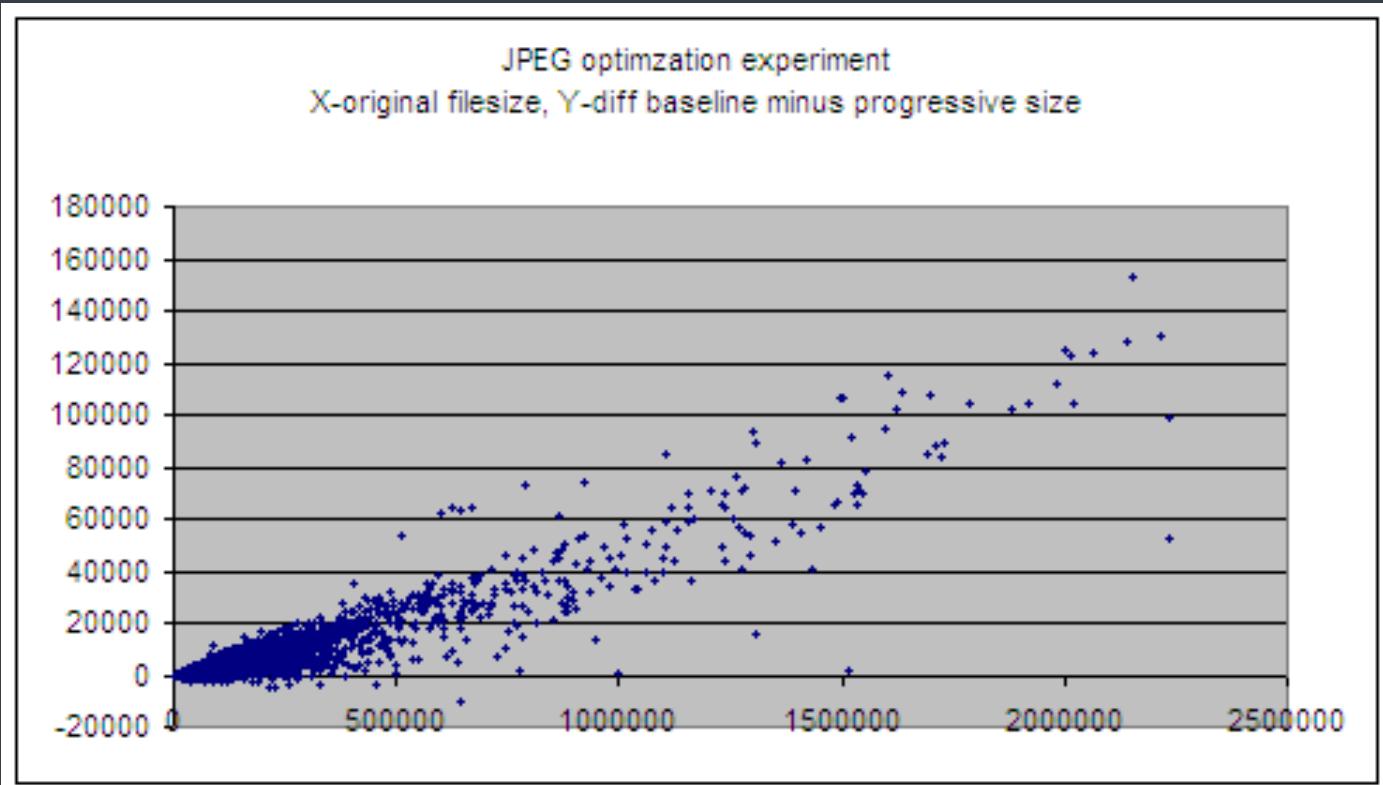
2008 PROGRESSIVE IMAGE STUDY GOALS

- IS PROGRESSIVE SMALLER
- IF NOT, WHEN
- CURIOSITIES: SPEED OF ENCODING
- CURIOSITIES: WHAT'S OUT THERE COMPARED TO WHAT IT COULD BE
- USE JPEGTRAN

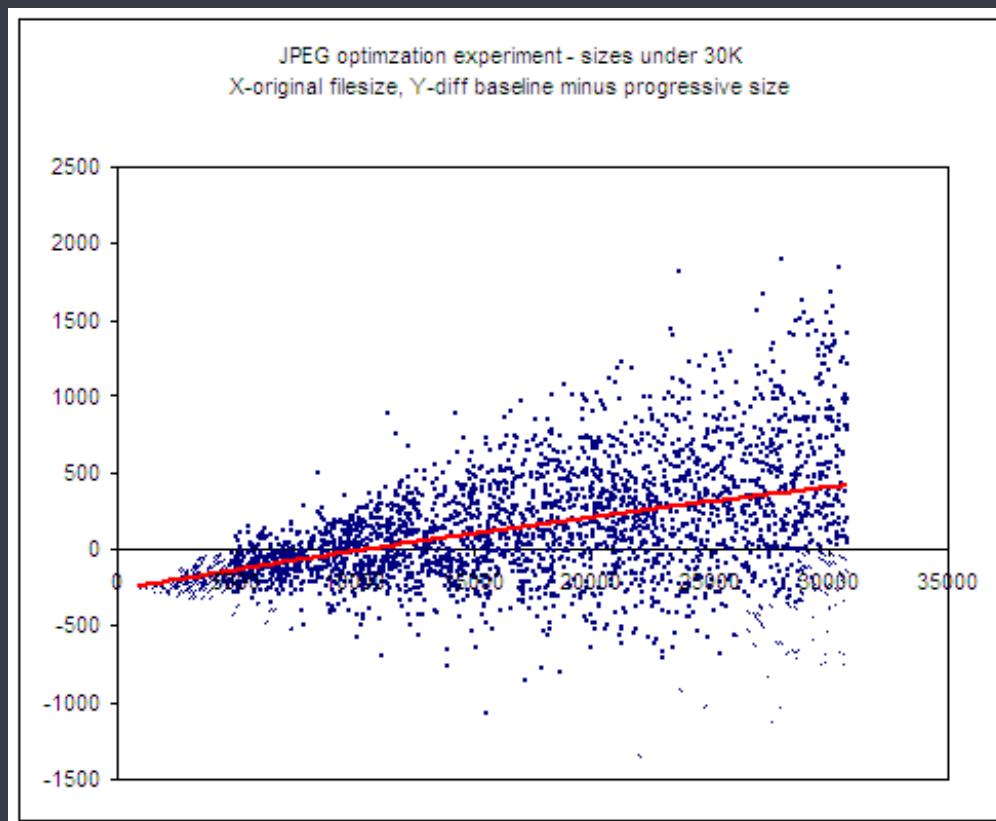
2008 PROGRESSIVE IMAGE STUDY RAW DATA

- 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.

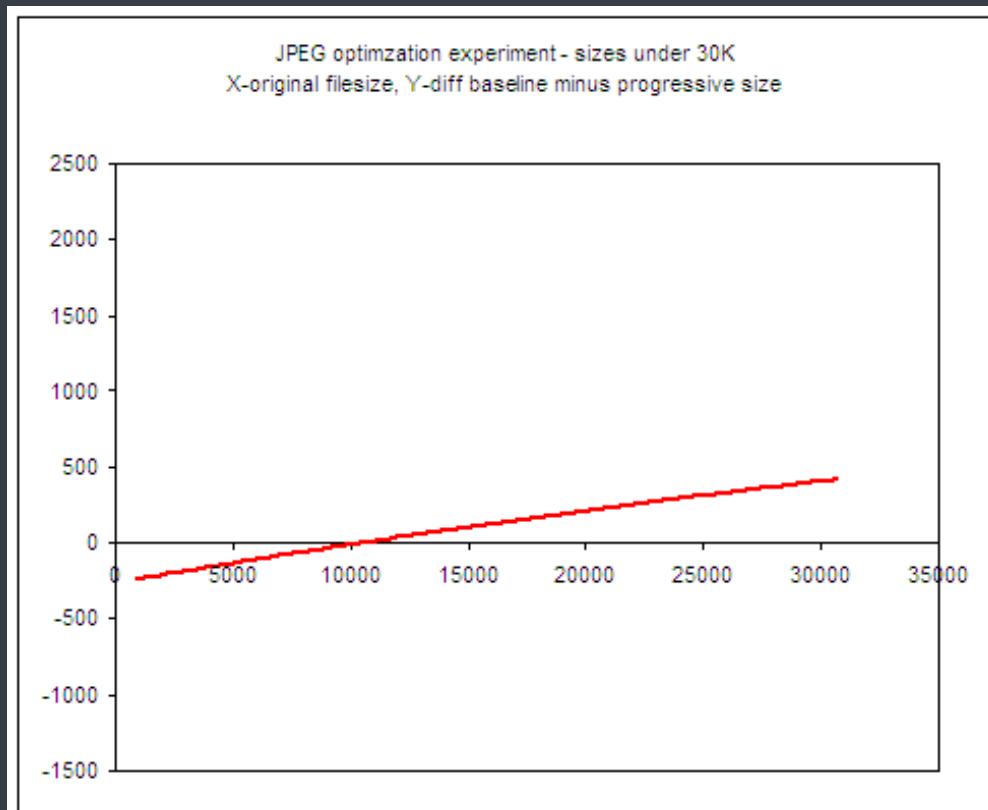
2008 IMAGE STUDY RESULTS



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](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 (/) 1 ? :

2022_07... top X +

2022_07_01_d... QUERY SHARE COPY ...

SCHEMA DETAILS PREVIEW

Row	reqHeaders	reqBodySize	reqCookieLen	reqOtherHeaders
41		659	null	origin = https://www.analysis.com, sec-Not A;Brand";v="99
42		653	null	content-length = 11 content-type = text origin = https://www.analysis.com, sec-Not A;Brand";v="99

Results per page: 50 ▾ 1 – 50 of 1316835649 |< < > >|

*10 000 r... egs X +

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 X

```
1 http://example.org/nthumbs//2013-04-12/2741538/2741538_03.jpg
2 https://example.org/fileadmin/templates/art/standard/startseite/Multimedia_ohne-Text.jpg
3 http://example.orgr/skin/upload/%EB%B0%EC%88%98%EC%97%B0.jpg
4 https://example.org/s-ghux11kp0r/images/stencil/80w/products/1029/1333/1032BKE_37343.1647888080.jpg?c=1
5 http://example.org/images/home/type01s.jpg
6 https://example.org/wp-content/uploads/sicurezza-protezioni-antinfortunistiche-macchinari-industriali-milper-4.jpg
7 https://example.org/sites/default/files/styles/home_page_poster/public/2021-11/Elysian_Poster_Locations.jpg?itok=tvxP
8 https://example.org/media/wysiwyg/marki/bio-oil-logo.jpg
9 https://example.org/wp-content/uploads/2019/04/4.jpg
10 https://example.org/image/cache/catalog/vktrade/banner/proizv/jlg-logo-130x100.jpg
11 https://example.org/s/files/1/0313/0087/8468/products/pg_286a1613-de4d-4753-9ef3-bc6b036f509c_grande.jpg?v=1626604964
12 https://example.org/-_6kc1rmzI7A/YEou2XpMSkI/AAAAAAAFCU/SYvksP1ANVAXggLl47ckyBJXj9oCuDbXgCNcBGAsYH0/w640-h426/anupam-
13 https://example.org/c/11-menu_default/equipamentos-semi-novos.jpg
14 https://example.org/198.71.233.168/mvu.1d6.myftpupload.com/wp-content/uploads/2020/10/mini-IMG_4193-150x150.jpg
15 https://example.org/data/images/801003471.jpg
16 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-JPEGGS

- DELETE A BUNCH OF NON-JPEGGS:

```
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

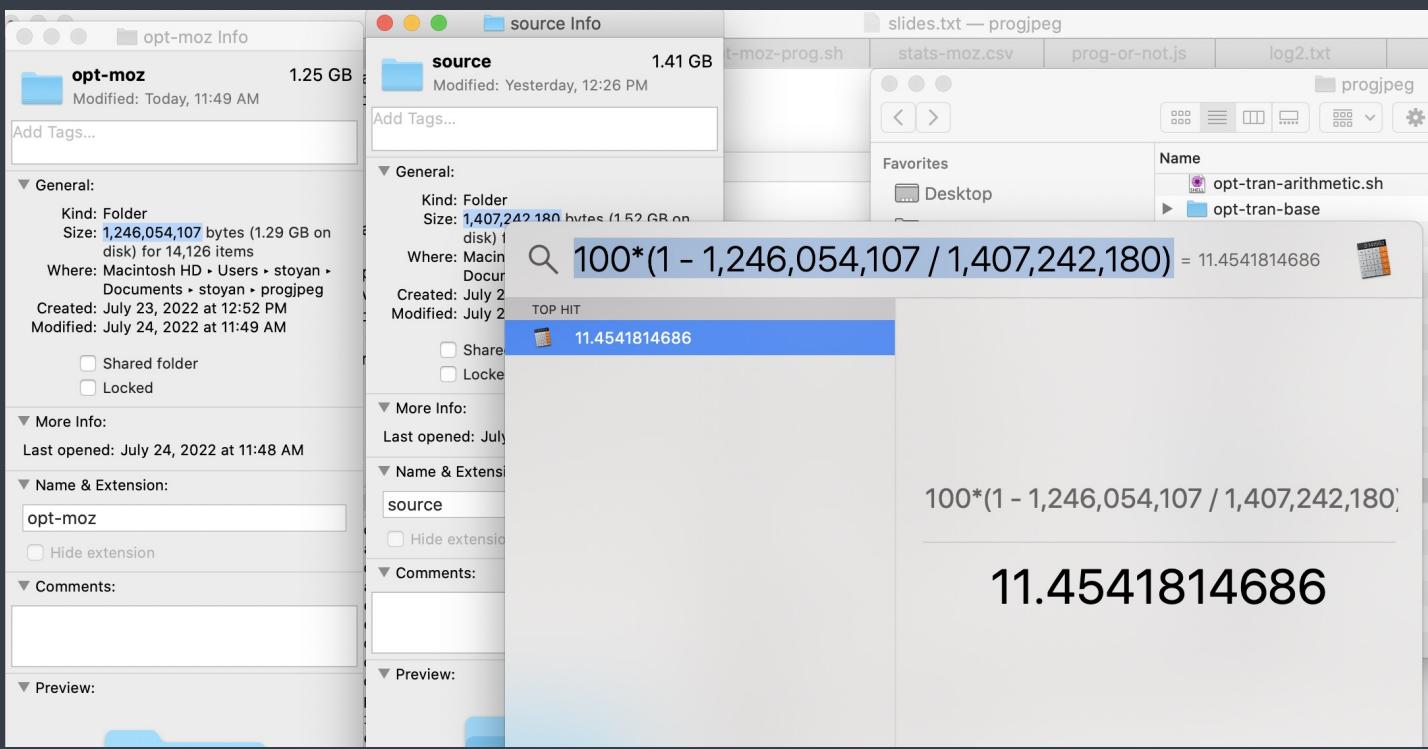
1. JUST RUN `JPEGTRAN` ALREADY
 - 1.1. ALL DEFAULTS PLUS `-COPY NONE` AND `-OPTIMIZE`
 - 1.2. ALWAYS USE `-PROGRESSIVE` BECAUSE THE 2008 STUDY SAYS YOU WIN 94% OF THE TIMES
 - 1.3. BRUTE FORCE: TRY BASELINE AND PROGRESSIVE, PICK THE SMALLER (AS THE 2008 STUDY RECOMMENDS)
2. JUST RUN `MozJPEG` ALREADY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

The screenshot shows a spreadsheet interface with two main sections. On the left, there is a code editor or terminal window displaying the raw CSV data. On the right, there is a table view with various filters and tools.

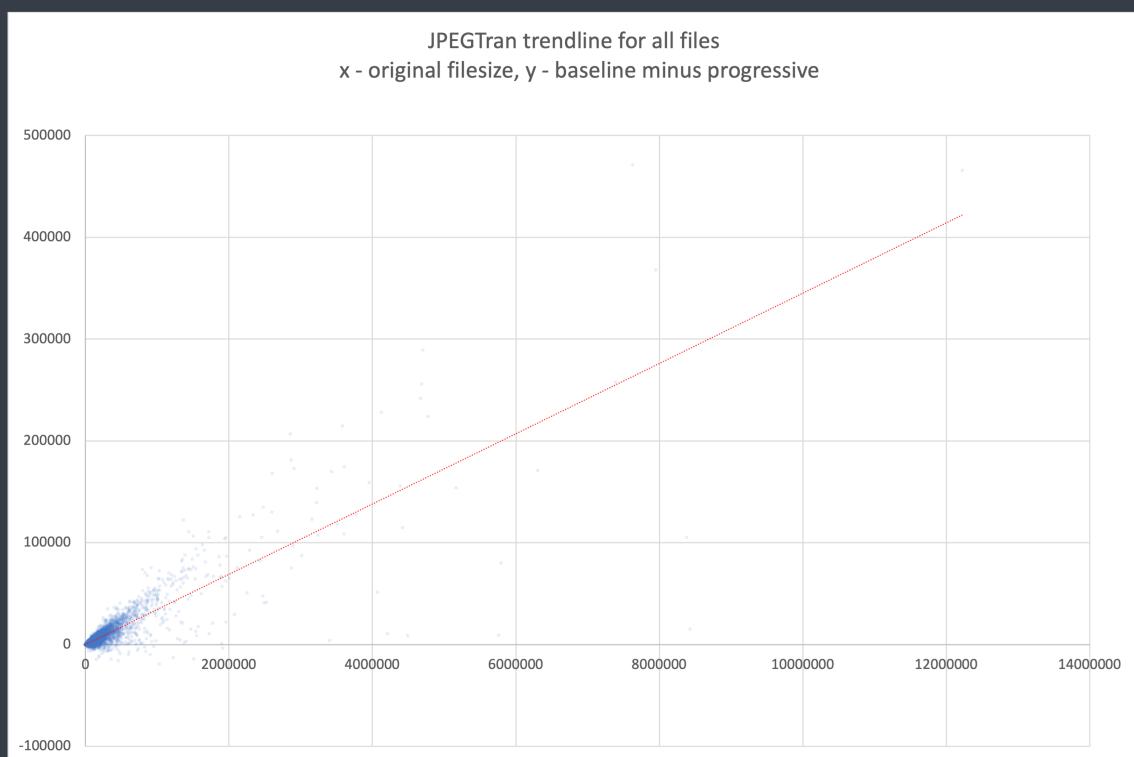
Raw CSV Data (Left):

file	orig	tranbase	tranprog	trandiff	tran	tranopt	mozbase	mozprog	mozdiff	moz	
1	1,2269	2236	2436	-200	2236	2236	2239	-6	2239	-6	2239
2	2,2827	2138	2369	-231	2138	2138	2205	-67	2205	-67	2205
3	3,51264	52092	51245	847	51245	51245	52111	50379	1732	50379	1732
4	4,5293	5290	5238	52	5238	5238	5293	5123	170	5123	170
5	5,34578	34580	3322	3322	3322	3322	34594	32773	1821	32773	1821
6	6,140938	149157	14	14	14	14	149194	140316	8878	140316	8878
7	7,16391	16401	1671	1671	1671	1671	16391	16287	109	16287	109
8	8,13939	11497	1100	1100	1100	1100	13939	10551	953	10551	953
9	9,3550	3165	3351	3351	3351	3351	3168	3187	-19	3187	-19
10	10,19974	19406	19531	19531	19531	19531	19418	19352	66	19352	66
11	11,10280	10069	102	102	102	102	102	102	102	102	102
12	12,1573	1573	1845	1845	1845	1845	1845	1845	1845	1845	1845
13	13,96194	103520	96	96	96	96	96	96	96	96	96
14	14,32895	33681	325	325	325	325	325	325	325	325	325
15	15,36161	34652	346	346	346	346	346	346	346	346	346
16	16,59769	59127	586	586	586	586	586	586	586	586	586
17	17,107662	111828	1	1	1	1	1	1	1	1	1
18	18,3722	3678	3834	3834	3834	3834	3834	3834	3834	3834	3834
19	19,45783	47051	457	457	457	457	457	457	457	457	457
20	20,5013	4412	4616	4616	4616	4616	4616	4616	4616	4616	4616
21	21,37640	7677	7572	7572	7572	7572	7572	7572	7572	7572	7572
22	22,26953	26824	268	268	268	268	268	268	268	268	268
23	23,192919	197765	1	1	1	1	1	1	1	1	1
24	24,25589	25540	248	248	248	248	248	248	248	248	248
25	25,37132	39660	382	382	382	382	382	382	382	382	382
26	26,17776	16928	164	164	164	164	164	164	164	164	164
27	27,39392	38377	382	382	382	382	382	382	382	382	382
28	28,12756	12333	119	119	119	119	119	119	119	119	119
29											

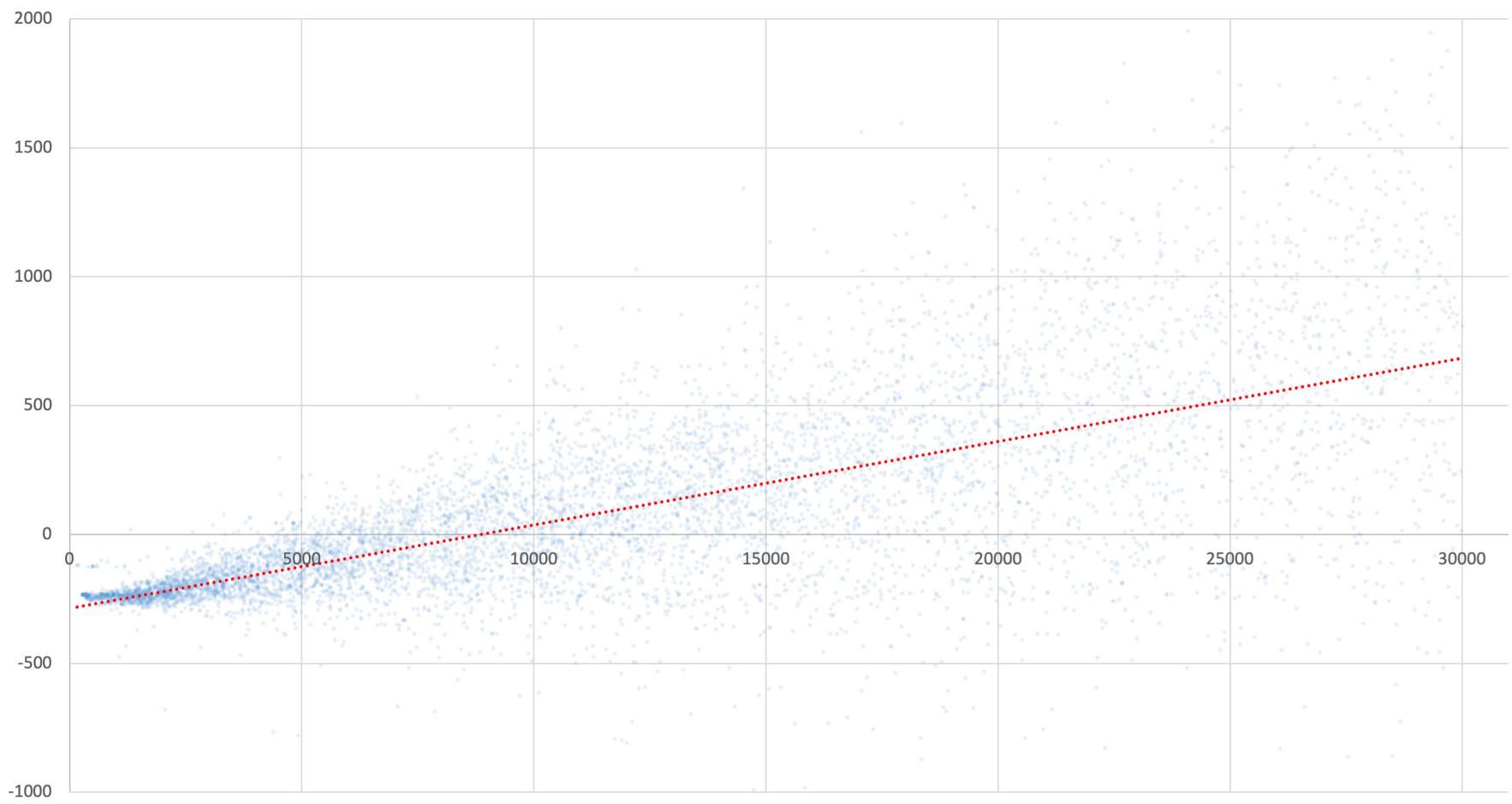
Table View (Right):

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	2138	-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

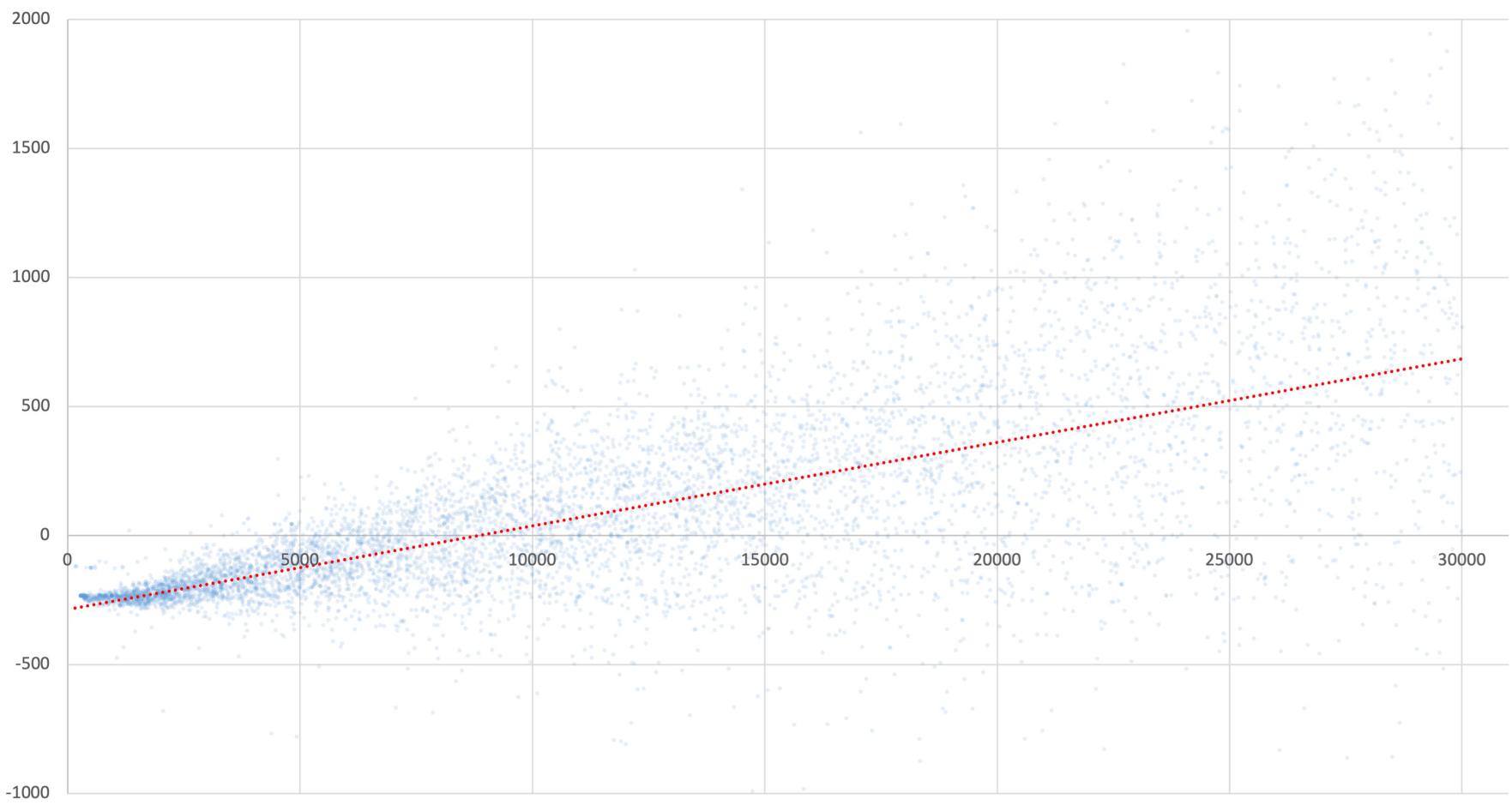
RESULTS



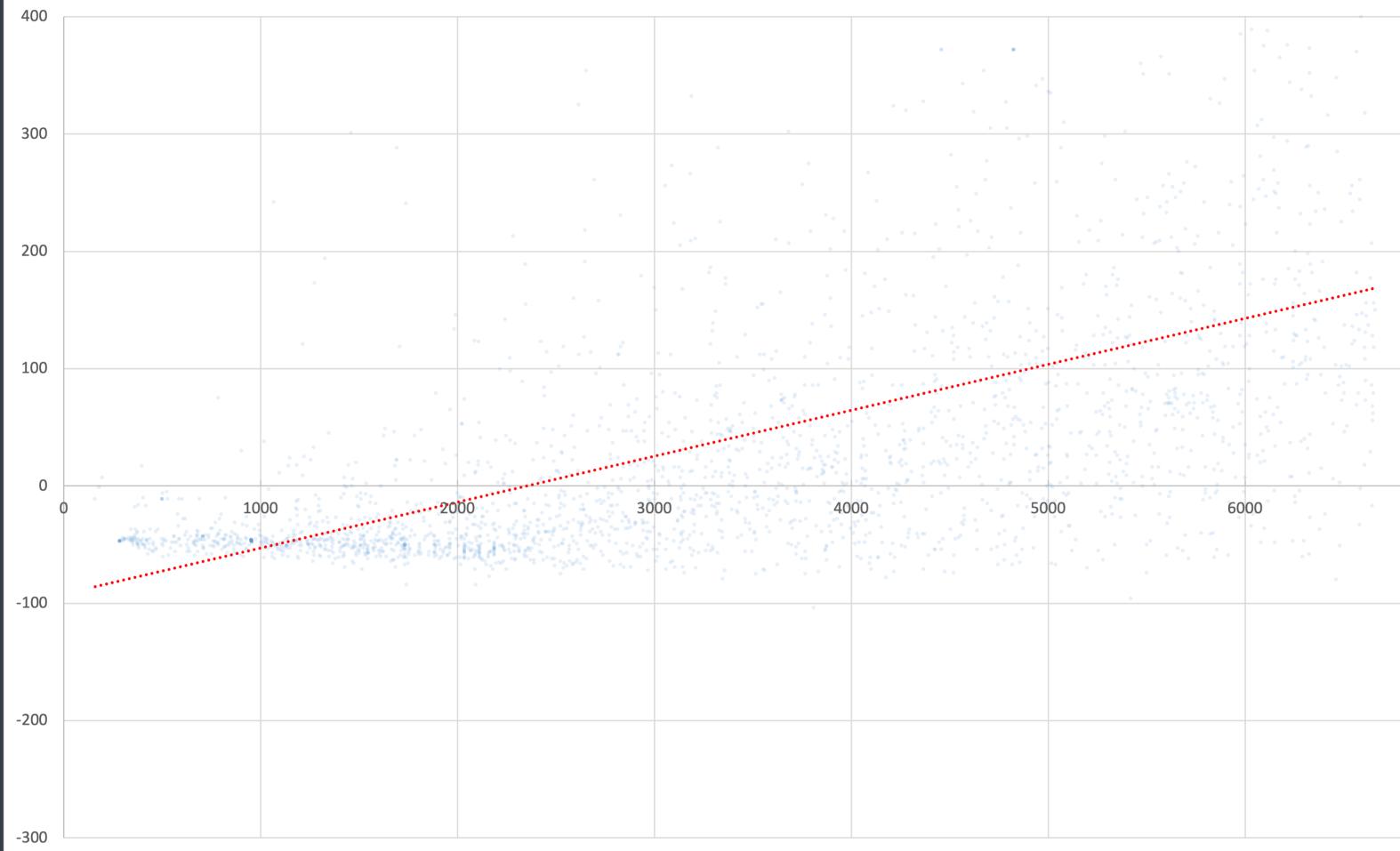
JPEGTran trendline for files under 30kB
x - original filesize, y - baseline minus progressive



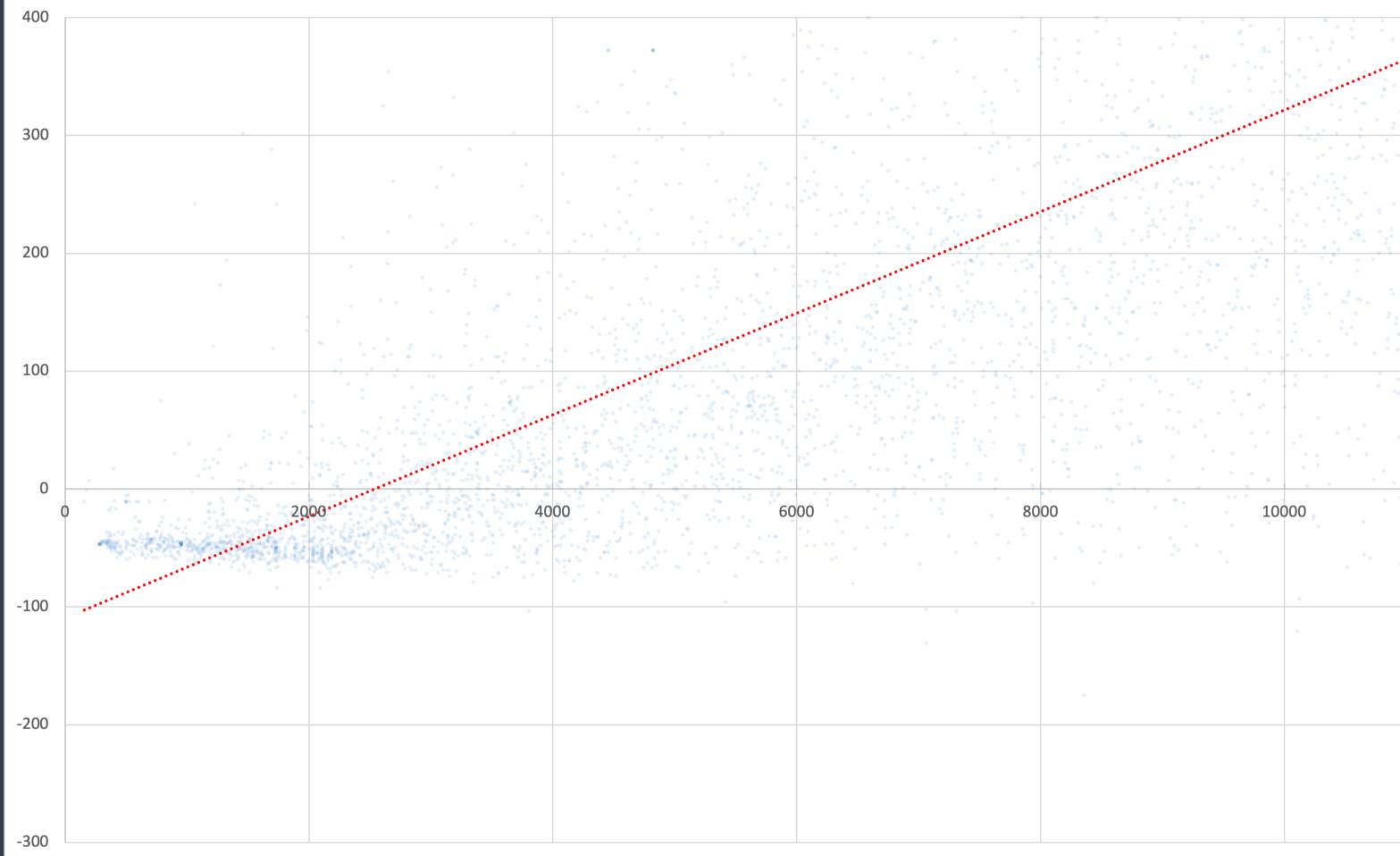
JPEGTran trendline for files under 30kB
x - original filesize, y - baseline minus progressive



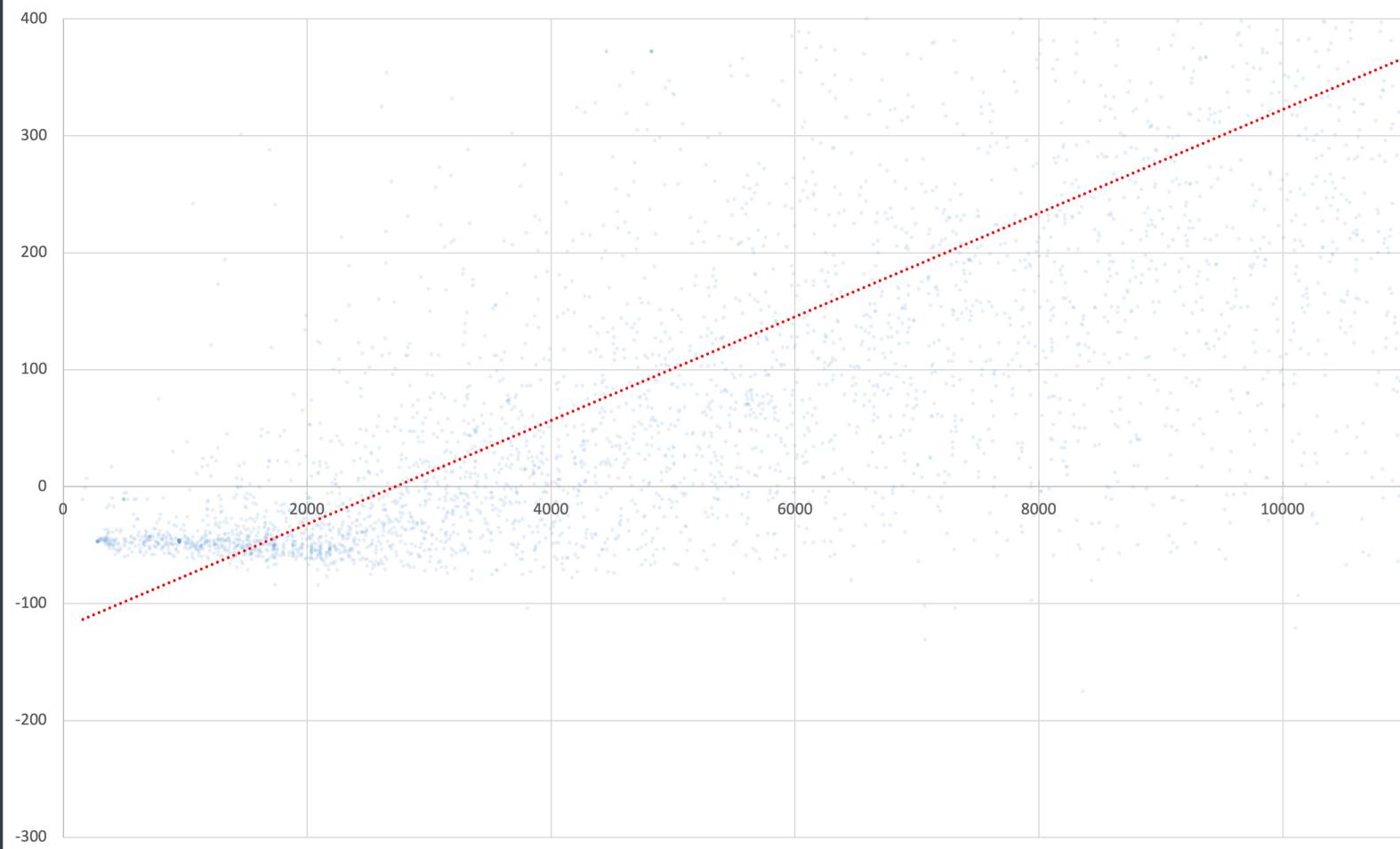
MozJPEG trendline for files under 6.66kB
x - original filesize, y - baseline minus progressive



MozJPEG trendline for files under 30kB
x - original filesize, y - baseline minus progressive



MozJPEG trendline for files under 52kB
x - original filesize, y - baseline minus progressive



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