

# IPOL: Image Processing On Line

Beyond Traditional Articles

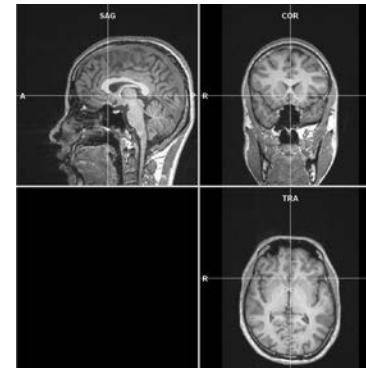
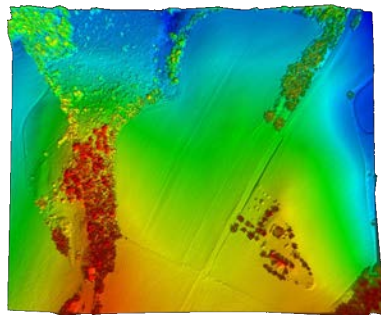
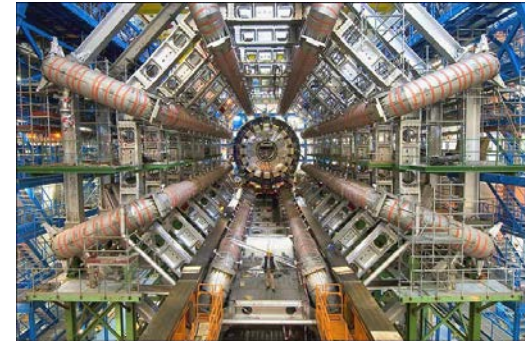
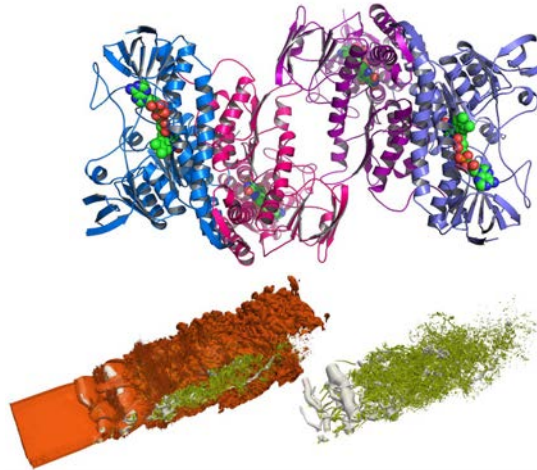
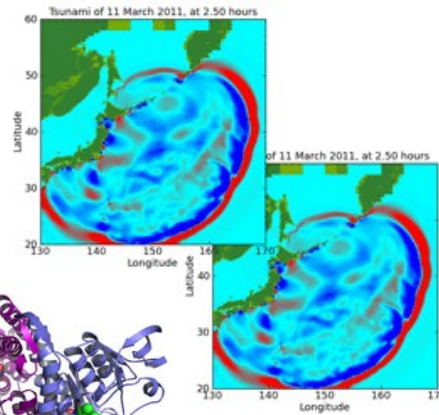
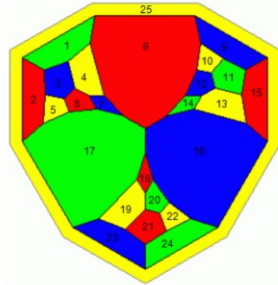
RIGOUR AND OPENNESS IN 21ST CENTURY SCIENCE

Oxford, 11th – 12th April 2013

Gabriele Facciolo  
CMLA, ENS Cachan, FR  
Image Processing On Line – IPOL  
<http://www.ipol.im/>

# Software Everywhere

- ▶ particle physics
- ▶ fluid dynamics
- ▶ econometrics
- ▶ signal processing
- ▶ quantum chemistry
- ▶ LIDAR archeology
- ▶ MRI analysis
- ▶ climate & weather
- ▶ geophysics
- ▶ ...



© CERN, rreis@flickr, rafael grompone, info-nftk@flickr, mohapj@flickr, mario stefanutti, argonne@flickr

# Software vs Science?

Research software is not like general-purpose equipment:  
telescopes, supercomputers, compilers, libraries



Software is made by scientists for scientists,  
specialized for an experimental process.

Most of the research software is:

- ▶ not released
- ▶ not published, not reviewed, not cited
- ▶ not completely specified
- ▶ ... and often buggy

# Reproducible Research

*Research is reproducible if other researchers can independently obtain the same results from the published material.*

- ▶ Theoretical sciences have proofs
- ▶ Experimental sciences have procedures
- ▶ Computational sciences have ...
  - ▶ insufficient descriptions
  - ▶ missing parameters
  - ▶ missing pre/post processing steps
  - ▶ missing data

refs: Claerbout 1992, Donoho 1995, Stodden, Vandewalle

# Reproducible Research Initiatives (some)

## **Journals:**

- ▶ *Math Programming Computation* requires the code
- ▶ *Biostatistics* stamps reproducible articles
- ▶ *JMLR* publishes software
- ▶ *Geophysics* has some software guidelines
- ▶ *Source Code for Biology and Medicine* publishes software
- ▶ *Image Processing On Line* focuses on algorithm and software
- ▶ *Computing in Science and Engineering* reviews software

## **Editors:**

- ▶ *SIAM* updated its supp. material policies to include software
- ▶ *ACM* reformed its supp. material copyright policy
- ▶ *Elsevier* experiments with “executable papers” and “post-PDF”

## **Tools and Services:**

- ▶ *RunMyCode*
- ▶ *FLOSShub, mloss/mldata*
- ▶ *Open data repositories: DataDryad, Figshare*

# Science Code Manifesto

---

- ▶ **Code:** All source code written specifically to process data for a published paper must be **available** to the reviewers and readers of the paper.
- ▶ **Copyright:** The copyright ownership and license of any released source code must be **clearly stated**.
- ▶ **Citation:** Researchers who use or adapt science source code in their research must **credit the code's creators in resulting publications**.
- ▶ **Credit:** Software contributions must be included in systems of scientific assessment, credit, and **recognition**.
- ▶ **Curation:** Source **code** must remain **available**, linked to related materials, for the useful lifetime of the publication.

<http://sciencecodemanifesto.org/>





WHAT IS

# OPEN ACCESS?



Click below to view the video:

[phdcomics.com/tv](http://phdcomics.com/tv)

**\*\*footnote:** (Can't see the video? [click here](#))

Open Access Explained!



0:05 / 8:24 HD YouTube



Watch our other videos at:  
[www.phdcomics.com/tv](http://www.phdcomics.com/tv)



Click to watch in HD



Click to watch Full screen

# Traditional Articles (comput. sciences)

- ▶ Publish description of method/algorithm and present results
- ▶ The **code** used for obtaining the results is **rarely made public**

Non reproducible results are viewed with skepticism.

## Reasons to not distribute the research code?



# Traditional Articles (comput. sciences)

- ▶ Publish description of method/algorithm and present results
- ▶ The **code** used for obtaining the results is **rarely made public**

Non reproducible results are viewed with skepticism.

## Reasons to not distribute the research code?

- ▶ Keep the advantage w.r.t. other researchers  
(better to not publish at all)

# Traditional Articles (comput. sciences)

- ▶ Publish description of method/algorithm and present results
- ▶ The **code** used for obtaining the results is **rarely made public**

Non reproducible results are viewed with skepticism.

## Reasons to not distribute the research code?

- ~~▶ Keep the advantage w.r.t. other researchers~~  
(better to not publish at all)
- ▶ Ashamed of the code  
(no time for cleanup & documentation)

# Traditional Articles (comput. sciences)

- ▶ Publish description of method/algorithm and present results
- ▶ The **code** used for obtaining the results is **rarely made public**

Non reproducible results are viewed with skepticism.

## Reasons to not distribute the research code?

- ~~▶ Keep the advantage w.r.t. other researchers~~  
(better to not publish at all)
- ▶ Ashamed of the code  
(no time for cleanup & documentation)
- ▶ Prevent incorrect use  
(by choosing wrong parameters)

# Revisit Objectives of Publishing Articles



Picture of a rare utopian community  
in the act of sharing their research code

vs

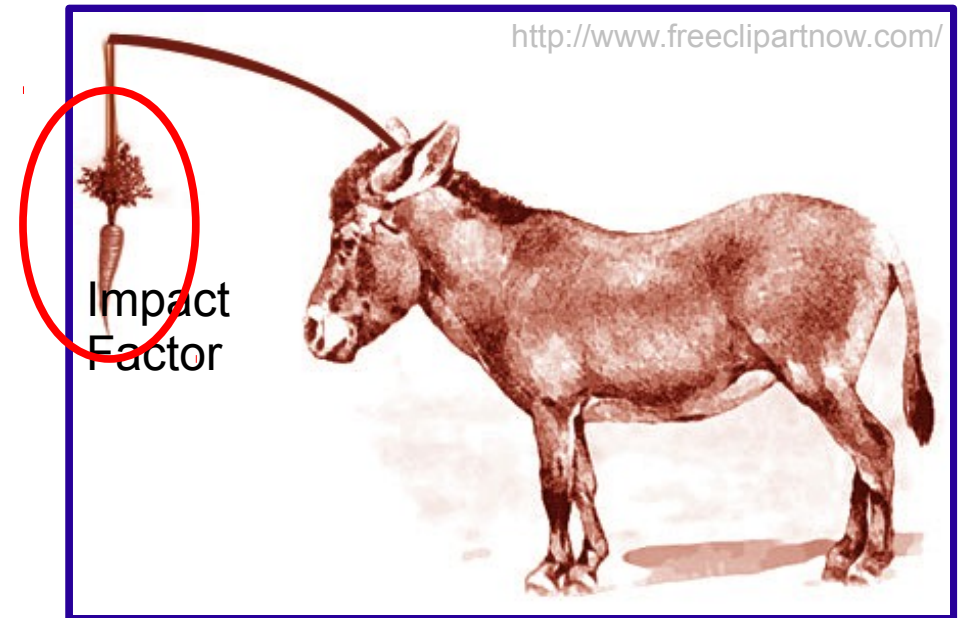


# Revisit Objectives of Publishing Articles



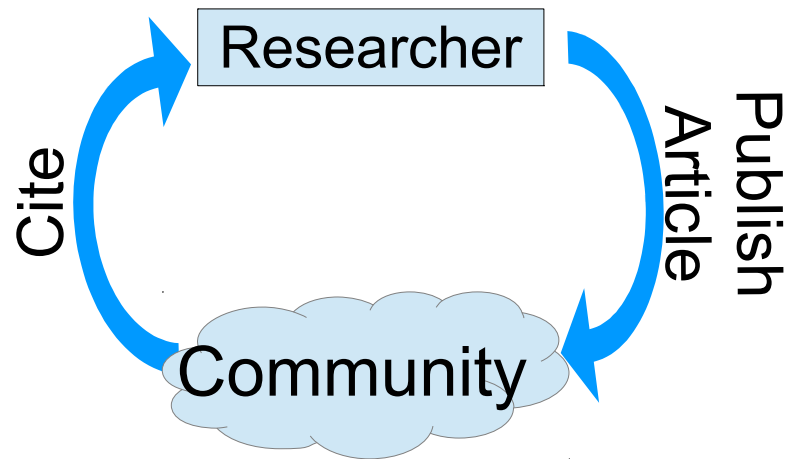
Picture of a rare utopian community in the act of sharing their research code

vs

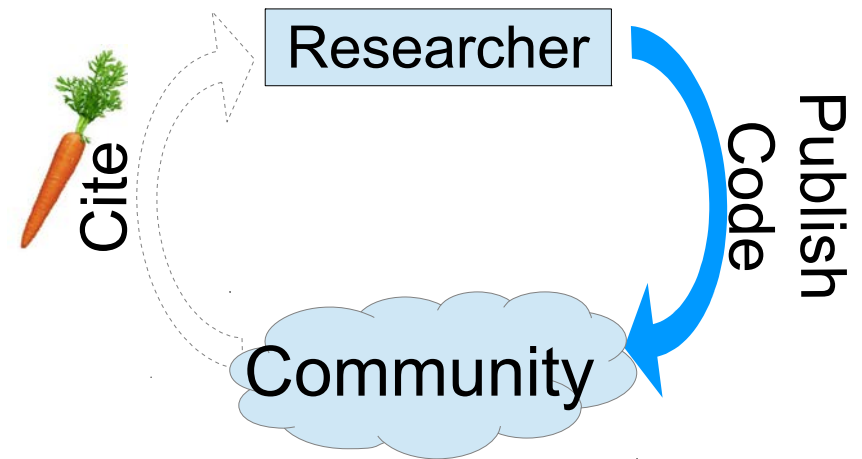


KEY to lure researchers into sharing their code

# Revisit Objectives of Publishing Articles



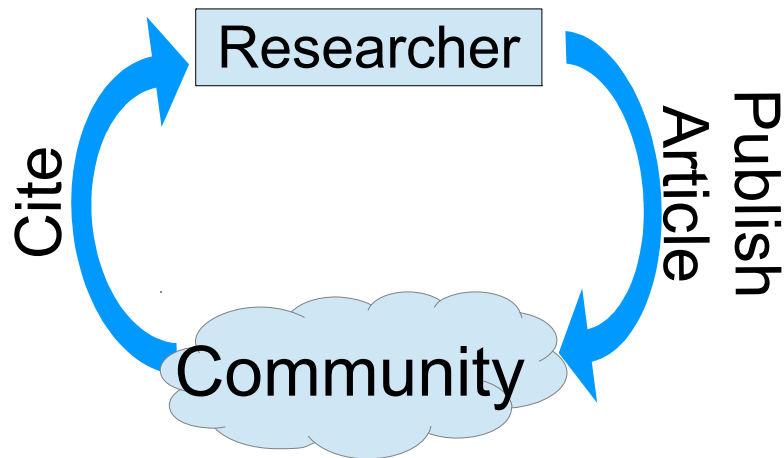
Traditional Research Articles



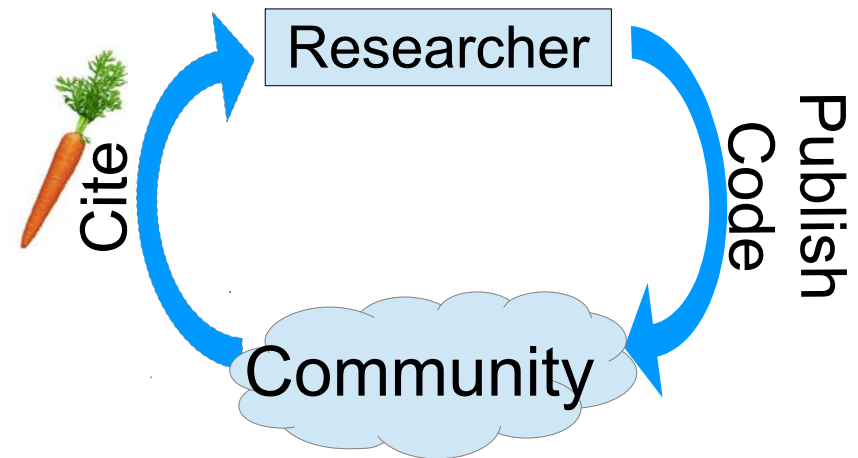
Source Code

Step 1: Make the code  
a publication by itself

# Revisit Objectives of Publishing Articles



Traditional Research Articles



Source Code

Step 2: Guide community to cite implementations



# IPOL : Image Processing On Line



*“IPOL is a **research journal of image processing and image analysis**. Each article contains a text describing an **algorithm** and **source code**, with an **online demonstration** facility and an **archive** of online experiments. The text and source code are peer-reviewed and the demonstration is controlled. IPOL follows the **Open Access** and **Reproducible Research** models.”*

<http://ipol.im/>

For every article, the implementation is:

- ▶ Reviewed and Published under GPL/BSD license
- ▶ Following Software Guidelines for correctness, portability, documentation


# IPOL : Image Processing On Line

**GOAL: provide a reference implementations of image processing algorithms**

- ▶ IPOL is not a prototype (running since 2011)
- ▶ IPOL is a journal: ISSN; DOI; Int'l editorial committee; ...
- ▶ Partnership with a SIAM journal for publication in both journals
- ▶ IPOL exists because no other journal did it
- ▶ IPOL publishes algorithms, not software. The implementations are here to provide the full details and a way to run the algorithm

# IPOL Article Components

- ▶ Algorithm **description**
- ▶ Implementation **source code**



Published in Image Processing On Line on 2012-3-24.  
ISSN 2105-1232 © 2012 IPOL & the authors CC-BY-NC-SA  
This article is available online with supplementary materials,  
software, datasets and online demo at  
<http://dx.doi.org/10.5201/ipol.2012.gjmr-lsd>

## LSD: a Line Segment Detector

Rafael Grompone von Gioi<sup>1</sup>, Jérémie Jakubowicz<sup>2</sup>,  
Jean-Michel Morel<sup>3</sup>, Gregory Randall<sup>4</sup>

<sup>1</sup> CMLA, ENS Cachan, France ([grompone@cmla.ens-cachan.fr](mailto:grompone@cmla.ens-cachan.fr))  
<sup>2</sup> TELECOM ParisTech, France ([jakubowi@telecom-paristech.fr](mailto:jakubowi@telecom-paristech.fr))  
<sup>3</sup> CMLA, ENS Cachan, France ([morel@cmla.ens-cachan.fr](mailto:morel@cmla.ens-cachan.fr))  
<sup>4</sup> IIE, UdelaR, Uruguay ([randall@iieing.edu.uy](mailto:randall@iieing.edu.uy))

### Abstract

LSD is a linear-time Line Segment Detector giving subpixel accurate results. It is designed to work on any digital image without parameter tuning. It controls its own number of false detections: on average, one false alarm is allowed per image [1]. The method is based on Burns, Hanson, and Riseman's method [2], and uses an *a contrario* validation approach according to Desolneux, Moisan, and Morel's theory [3, 4]. The version described here includes some further improvement over the one described in our original article [1].

### Source Code

The ANSI C implementation of LSD version 1.6 is the one which has been peer reviewed and accepted by IPOL. The source code, the code documentation, and the online demo are accessible at the IPOL web page of this article<sup>1</sup>.

### Supplementary Material

Also available at the IPOL web page of this article<sup>2</sup> are two older implementations of LSD, versions 1.0 and 1.5, as well as an example of applying LSD, frame by frame, to a video. The version 1.0 of LSD code corresponds better to the algorithm described in our original

## LSD: a Line Segment Detector

Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, Gregory Randall

[article](#) [demo](#) [archive](#)

published • 2012-03-24 → BibTeX  
reference • Grompone von Gioi, Rafael, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. "LSD: a Line Segment Detector." *Image Processing On Line* 2012 (2012). <http://dx.doi.org/10.5201/ipol.2012.gjmr-lsd>

*Communicated by Lionel Moisan*  
*Demo edited by Rafael Grompone*

### Abstract

LSD is a linear-time Line Segment Detector giving subpixel accurate results. It is designed to work on any digital image without parameter tuning. It controls its own number of false detections: On average, one false alarm is allowed per image [1]. The method is based on Burns, Hanson, and Riseman's method [2], and uses an *a contrario* validation approach according to Desolneux, Moisan, and Morel's theory [3, 4]. The version described here includes some further improvement over the one

## Download

- full text manuscript: PDF
- source code: ZIP

# IPOL Article Components

- ▶ Algorithm description
- ▶ Implementation source code
- ▶ **Web demo** interface



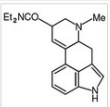
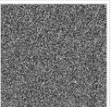
**LSD: a Line Segment Detector**

[article](#) [demo](#) [archive](#)

Please cite the reference article if you publish results obtained with this online demo.

Select Data

Click on an image to use it as the algorithm input.

Chairs image   Le Piree   LSD molecule   Noise

Upload Data

Upload your own image files to use as the algorithm input.

input image  No file chosen

Images larger than 100000000 pixels will be resized. Upload size is limited to 286MB per image file and 10MB for the whole upload set. TIFF, JPEG, PNG, GIF, PNM (and other standard formats) are supported. The uploaded files may be re-used for further analysis. Only upload suitable images. See the copyright and legal conditions for details.

feeds • sitemap • contact • privacy policy • ISSN: 2105-1232 • DOI: 10.5201/ipol  
supported by CMLA, ENS Cachan • DMI, Universitat de les Illes Balears • Fing, Universidad de la República • Gandi.net  
© 2009-2012, IPOL Image Processing On Line & the authors [Open Access](#) [Creative Commons](#) [IPOL DATA](#) [IPOL.DAT](#)

**LSD: a Line Segment Detector**

[article](#) [demo](#) [archive](#)

Please cite the reference article if you publish results obtained with this online demo.

Run again?

Result

847 Line Segments were detected. The algorithm ran in 0.27s.

You can download the result in [EPS](#) format, in [SVG](#) format, or an [ASCII](#) file (see description below).



Test the algorithm with new data and explore the parameters without compiling

# IPOL Article Components

- ▶ Algorithm description
- ▶ Implementation source code
- ▶ Web demo interface
- ▶ Public **archive** with original test data

**LSD: a Line Segment Detector**

article demo archive


Please cite the reference article if you publish results obtained with this online demo.

8954 public archives of online experiments with original images since 2009/04/27 08:11.  
This archive is not moderated. In case of copyright infringement or similar problem, please contact us to request the removal of some images. Some archived content may be deleted by the editorial board for size matters, inadequate content, user requests, or other reasons.

pages: << < > >> - 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 [448]






key 35C93AB22D7BF8CAB0314C29E286F55B  
date 2013/04/02 09:54  
LSD Version 1.6 of November 11, 2011, compiled  
version Dec 4 2012 15:49:05  
run time 0.151010036469  
(s)  
files output.txt output.eps output.svg

images







key 6ED5D2C294A071F7CEB9BD7663847DEC  
date 2013/04/02 11:07  
LSD Version 1.6 of November 11, 2011, compiled  
version Dec 4 2012 15:49:05  
run time 0.235600948334  
(s)  
files output.txt output.eps output.svg

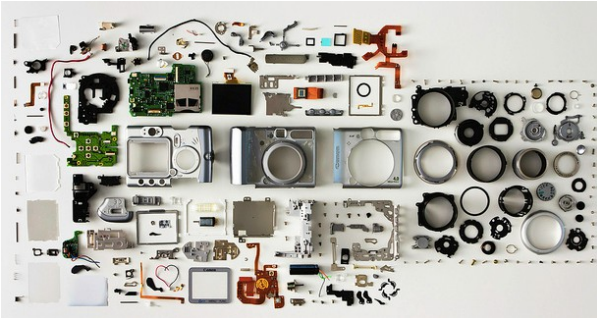
images



run time (s) 0.534783124924  
files output.txt output.eps output.svg



# Reproducible+demo → Rigour++



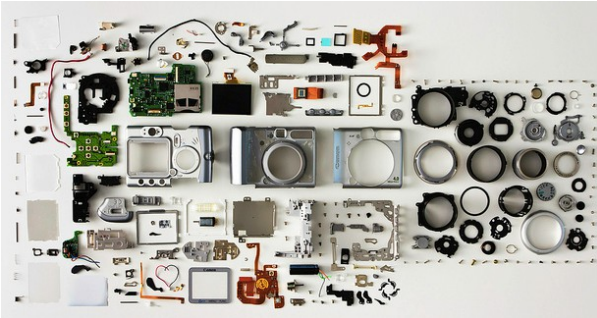
## CODE

- ▶ on-line demo facilitates experimentation
- ▶ and leads to a stricter verification of the claims

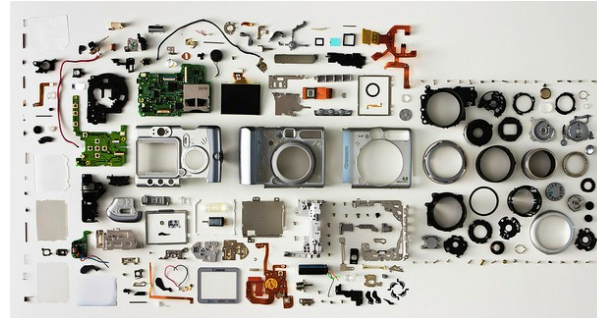
© kellyhofer@flickr



# Reproducible+demo → Rigour++



**CODE**



**CODE + DEMO**



- ▶ on-line demo facilitates experimentation
- ▶ and leads to a stricter verification of the claims

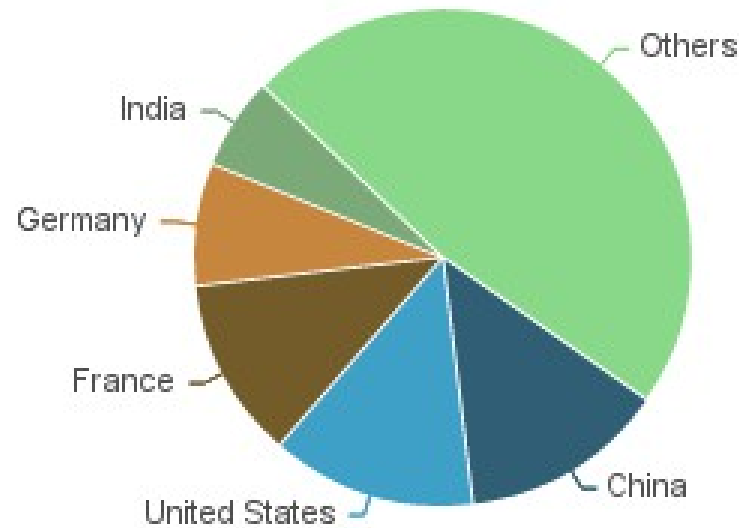
© kellyhofer@flickr



# IPOL Usage Stats

- ▶ 30 articles published with code and demo since 2011  
20 articles in preparation
- ▶ 109 citations (cf. Google Scholar)
- ▶ 2012: 125000 visits, 13000 code/data downloads
- ▶ 2012: 50000 demo runs, 30000 archived runs on original data

— Visits



# IPOL : Good Things

---

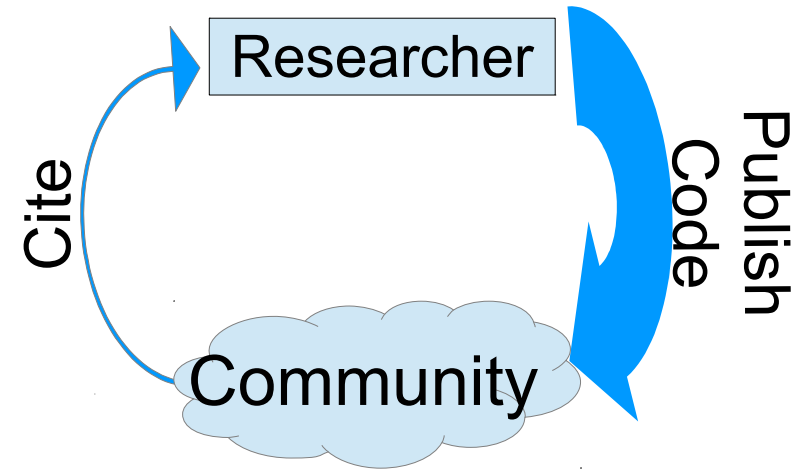
- ▶ Reference versions of algorithms
- ▶ In-depth analysis of algorithms
- ▶ Improvement of the algo. by the reviewer and user tests
- ▶ Improvement of the code by the review
- ▶ Useful to other researchers

reproducible is good, reusable is better

# Challenges

Needs momentum:

- ▶ Effort to prepare the code for publication isn't negligible
- ▶ Community must learn to cite the implementations



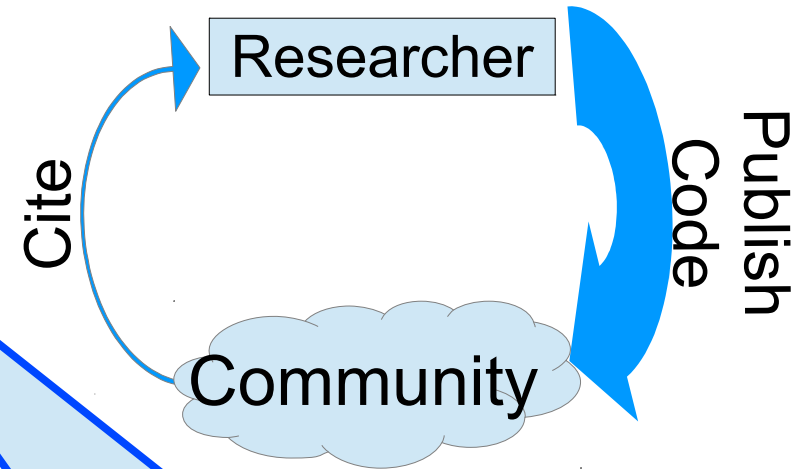
Also: Reproducible  $\neq$  Reusable

# Challenges

Needs momentum:

- ▶ Effort to prepare the code for publication isn't negligible
- ▶ Community must learn to cite the implementations

Also: Reproducible  $\neq$  Reusable



**Team-up with:**  
**WEB designers &**  
**Visualization Scientists ?**



# Follow-up to...

IPOL wouldn't be possible without the support and trust of the authors, reviewers and editors who contributed to it. Lots of help from Paris, Palma, Montevideo, Durham, ...

You can be part of it.



<http://ipol.im/>  
[edit@ipol.im](mailto:edit@ipol.im)  
[discuss@list.ipol.im](mailto:discuss@list.ipol.im)  
[@ipol\\_journal](https://twitter.com/ipol_journal) 

