

Reproducible Research in Image Processing: The Case of **IPOL**

REPPAR/Euro-Par 2016
22 – 8 – 2016

Enric Meinhardt-Llopis
CMLA, ENS-Cachan
<http://ipol.im>



Enter IPOL

(Trying to put a bit of sanity in this crazy world)

I POL: Image Processing On Line



<http://ipol.im>

*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.*

For each article, the code is:

- ▶ Reviewed by anonymous reviewers, curated and published
- ▶ Licence BSD, GPL or AGPL
- ▶ According to the journal's *software guidelines*
- ▶ Given an online demo

Anatomy of an IPOL article

Website of the article

<http://ipol.im/pub/art/2013/59>

The screenshot shows the IPOL Journal website. At the top, there's a navigation bar with links: HOME, ABOUT, ARTICLES, PREPRINTS, NEWS, and SEARCH. The main title is "LSD: a Line Segment Detector" by Rafael Grampone von Gisi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Below the title, there's a section for "article" with links to "article", "demo", and "archive". The "published" date is 2013-09-24. The "abstract" section states: "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. The method is based on Burns, Hanson, and Rosenman's method, and uses an a-posteriori validation approach according to Deschamps, Meisen, and Morel's theory. The version described here includes some further improvement over the one described in the original article." The "download" section offers a full-text manuscript (PDF, 554K) and a source code (ZIP). The "preview" section shows a thumbnail of the article's first page, which includes the IPOL logo, the title, authors, and a brief abstract.

What do you find:

1. **PDF** file with the description of the method
2. **ZIP** file with the complete and portable source code
3. **Database** of example images
4. Online **Demo**, allowing to try the method over user-uploaded images.
5. **Archive** of results of previous experiments by other readers.

IPOL: Source Code

Complete source code of a program that darkens images:

```
void darken_image_inplace(float *x, int w, int h)
{
    for (int i = 0; i < w * h; i++)
        x[i] = x[i] / 2;
}
```

```
#include "image_io.h"
int main()
{
    int w, h;
    float *x = image_load("image.png", &w, &h);

    darken_image_inplace(x, w, h);

    image_save("image_foncee.png", x, w, h);
    return 0;
}
```

IPOL Software Guidelines:

- ▶ Languages: C, C++, Matlab/Octave, Python
- ▶ Libraries: png, tiff, jpeg, fftw3, lapack, openmp (and nothing else)
- ▶ Constraints: non-interactive, 80 chars par line, Makefile, README, examples

IPOL software guidelines (checklist)

These rules are enforced by the source code reviewers

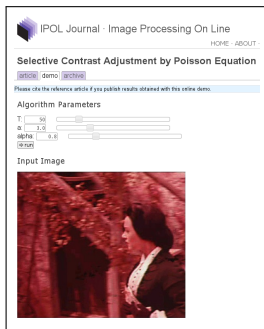
- ▶ zip or tar.gz archive `name_version.{zip,tar.gz,tgz}`, less than 2 MB
- ▶ everything into a `name_version/` folder
- ▶ file names with `a-z,A-Z,0-9,-,_,.`
- ▶ no hidden file, backup or useless file, no binary
- ▶ C89, C99 or C++98 code tested with `gcc -std=xxx -Wall -Wextra -Werror`
- ▶ portable code, 32/64-bits, nothing specific to an operating system
- ▶ only libtiff, libjpeg, libpng, zlib, fftw, libgsl, libeigen, cblas and clapack external libraries
- ▶ compilation with make or cmake, only standard options, make uses `$(CC)` or `$(CXX)`
- ▶ command-line non-interactive interface
- ▶ max 1 GB memory, max 30 s computation in the demo environment
- ▶ can read/write in PNG, TIFF, PNM, EPS, SVG, VRML or PLY format
- ▶ copyright attribution and GPL/BSD license info in every source file
- ▶ patent warning if needed
- ▶ README.txt essential information
- ▶ correct, clean code in English
- ▶ max 80 characters per line, max 1000 lines per file
- ▶ `main()`, algorithmic and auxiliary code in different files
- ▶ detailed comments for every function and every implementation step
- ▶ example input data and result

Goal: code must work in 30 years and give the same result.

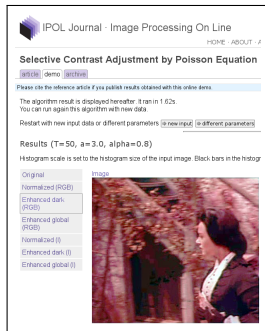
IPOL: Online demos

For each IPOL article, its online demo:

- ▶ Uses exactly the same code that is published
- ▶ Runs in real time ($< 30s$)
- ▶ Input images and parameters chosen by the reader
- ▶ Saves all the results on a public archive

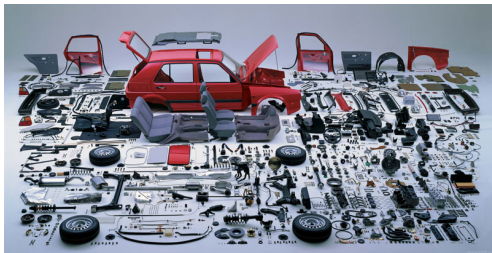


parameter selection



result

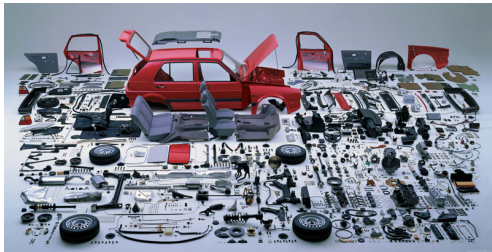
Importance of the online demo



source code

Photo: Hans Hansen, *Das Buch Von Volkswagen 1938-1988*

Importance of the online demo



source code



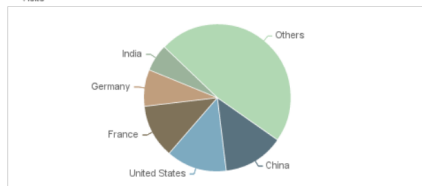
online demo

Photo: Hans Hansen, *Das Buch Von Volkswagen 1938-1988*

IPOL : statistics

- ▶ ≈ 20 articles per year since 2011
- ▶ 15 articles under review (august 2016)
- ▶ 1247 citations (Google Scholar, august 2016)
- ▶ 212.232 archived experiments (2011–2015)
- ▶ $\approx 3 \times 10^6$ demo executions (2011–2015)
- ▶ 125.000 visits, 13.000 code downloads (2012)
- ▶ 4 requests for commercial licence purchase
- ▶ 3 algorithms integrated on the OpenCV library
- ▶ authors from 11 labs in 7 countries
- ▶ main source of visibility and funding of our research group

— Visits



Universitat de les
Illes Balears



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



Real-time online demonstration

1. Overview of IPOL website
2. Take a photo with my phone
3. Process the photo with an IPOL demo
4. Look at the archive

Overview of our IPOL experience

In general:

- ✓ very attractive for readers
- ✗ extremely time-consuming for authors, reviewers and editors
- ✗ difficult to attract spontaneous submissions
- ✓ high visibility for the articles
- ✓ exhilarating sense of accomplishment when an article is published

Demos:

- ✓ learn about your own algorithms
- ✓ teaching material

Source code:

- ✓ clean, free, independent, portable and **ported**
- ✓ shall run unchanged in 20 years

✓ ✓ *make the world a better place!*

Merci!

<http://ipol.im>



Figure: IPOL demo of homographic registration and fusion