IPOL: a new journal for fully reproducible research; analysis of four years development

M. Colom$^1$, B. Keraultret$^2$, N. Limare$^1$, P. Monasse$^3$, and J.M. Morel$^1$

$^1$CMLA, ENS Cachan $^2$LORIA, Univ. de Lorraine $^3$LIGM, Univ. Paris-Est

Worskshop RCBR 2015, NTMS 2015 July 29, Paris
Reproducible research in sciences:

- *Theoretical scientists* share demonstrations;
1. The IPOL Journal: context of reproducible research (1)

Reproducible research in sciences:

- *Theoretical scientists* share demonstrations;
- *Experimental scientist* share procedures;
Reproducible research in sciences:

- *Theoretical scientists* share demonstrations;
- *Experimental scientists* share procedures;
- *Computational scientists*... ?
Reproducible research in sciences:

- *Theoretical scientists* share demonstrations;
- *Experimental scientist* share procedures;
- *Computational scientists*... ?

Computer Science:

- Description of methods/algorithms;
- description often limited (constraints on page limits);
- parameters not given or not well described;
- steps of pre/post processing missing.
1. The IPOL Journal: context of reproducible research (2)

Research in Computer Science:

1. New idea;
2. Demonstration, implementation;
3. Article publication.
1. The IPOL Journal: context of reproducible research (2)

Research in Computer Science:
1. New idea;
2. Demonstration, implementation;
3. Article publication.

Reusable Research:
1. Article which seems interesting;
2. Re-implement the algorithm;
3. Conformity of the results with the original.
Frequent difficulties in computer science (image processing):

- **Source code often not available** (or not reviewed);
- **quality/stability** of the results not easy to analyze;
- testing with **different input data** not possible.
1. The IPOL Journal: context of reproducible research (3)

Frequent difficulties in computer science (image processing):

- Source code often not available (or not reviewed);
- quality/stability of the results not easy to analyze;
- testing with different input data not possible.

Consequences:

- Comparisons and experiments difficult;
- potential time loss for the reader;
- limits the diffusion of research.
Providing source code/data

- ✨ A real added value for the publication;
- ✨ increases the impact/comparisons;
- ✗ software is not really acknowledged;
- ✗ important effort (documentation, tests, user maintenance).
1. The IPOL Journal: context of reproducible research (4)

Providing source code/data

- A real added value for the publication;
- increases the impact/comparisons;
- software is not really acknowledged;
- important effort (documentation, tests, user maintenance).

Software Diffusion

- Specialized journals in software:
  
  Source Code for Biology and Medicine, Journal of Open Research Software, Computing in Science and Engineering, ...

- Diffusion platform:
  
  RunMyCode / Run&Share, Figshare, DataDryad, Harvard Dataverse, Ubiquity Metajournals, Zenodo (with doi) ...

⇒ no validation, no scientific review (reliability and durability problem).
Origin:

- Journal started in October 2009;
- under the initiative of Nicolas Limare, Jean-Michel Morel and the Image Processing team at the CMLA lab (ENS-Cachan);
- first article published in 2010.
1. The IPOL Journal: origin and motivation

Origin:

- Journal started in October 2009;
- under the initiative of Nicolas Limare, Jean-Michel Morel and the Image Processing team at the CMLA lab (ENS-Cachan);
- first article published in 2010.

Motivation [Limare & Morel 2009]:

- Reproducible research;
- new way to publish research results;
- allows everybody to test the algorithms;
  - with their own images
- independent of the platform (the demos execute on the server side and the results are shown to the user using a web interface).
1. IPOL Journal: principle and current form (1)

Characteristics:

- Research journal in image processing;
- each article contains a description of one algorithm and its source code;
- association of each article with its online demonstration, with archived experiments;
- the peer-review process includes the article, demo, and source code;
- Open Science journal and Reproductible Research.
Philosophy of the journal:

- Follows the guideline on reproducible research topics;
- reproducible research standard [Stodden 09a] [Stodden 09b];
- answer to credibility crisis in scientific computation (as pointed out by Donoho [Donoho et al. 09]).
1. IPOL Journal: principle and current form (2)

Philosophy of the journal:
- Follows the guideline on reproducible research topics;
- reproducible research standard [Stodden 09a] [Stodden 09b];
- answer to credibility crisis in scientific computation (as pointed out by Donoho [Donoho et al. 09]).

What IPOL is not:
- IPOL publishes algorithms along with their implementation, but not compiled software;
- IPOL is not a software library (each code has minimal dependencies);
- IPOL is not a software or code diffusion platform.
1. IPOL Journal: principle and current form (3)

- Current form: “classic” (with online PDF).
Current form: “classic” (with online PDF).

Associated demos.
1. IPOL Journal: principle and current form (3)

- Current form: “classic” (with online PDF).
- Associated demos.
- Archive containing experiments with data uploaded by users.
1. IPOL Journal: editorial structure

Same aspects as a classical journal:

- Editorial project, editorial committee;
- articles, authors, editors;
- reviewing process and validation;
- ISSN, DOI;
- special issues;
- currently indexed by:
  
  Scirus, Google Scholar, DBLP, DOAJ, SHERPA/RoMEO, Héloïse, WorldCat, CrossRef, Ulrich, Index Copernicus, PBN, JGate, VisionBib, CVonline, JournalSeek, and NewJour.
1. IPOL Journal: editorial structure

Same aspects as a classical journal:

- Editorial project, editorial committee;
- articles, authors, editors;
- reviewing process and validation;
- ISSN, DOI;
- special issues;
- currently indexed by: Scirus, Google Scholar, DBLP, DOAJ, SHERPA/RoMEO, Héloïse, WorldCat, CrossRef, Ulrich, Index Copernicus, PBN, JGate, VisionBib, CVonline, JournalSeek, and NewJour.

Software point of view:

- Each article should propose an implementation;
- reviewing step, verification, validation, and publication;
- reviewer: check the correspondence between the algorithm description in the article and code (+ code readability and code documentation).
2. Scientific & technical achievements to establish a state of the art (1)

Image Denoising

- Papers on image denoising cover most of the state of the art in image denoising.
  ⇒ analyze and finalize the often incomplete algorithms.
- Online implementation allows the first objective comparison.
- *complete state of the art of denoising*

⇒ See the noise Clinic [Lebrun et al. 15]
2. Some scientific and technical achievements

2. Scientific & technical achievements to establish a state of the art (1)

**Image Denoising**
- Papers on image denoising cover most of the state of the art in image denoising.
  ⇒ analyze and finalize the often incomplete algorithms.
- Online implementation allows the first objective comparison.
- complete state of the art of denoising

⇒ See the noise Clinic [Lebrun et al. 15]

**Stereovision**

The stereovision category at IPOL contains five fundamental algorithms:
- Quasi-Euclidean Epipolar Rectification [Monasse 11].
- Kolmogorov & Zabih’s graph cuts stereo matching algorithm [Kolmogorov et al. 14].
- Stereo disparity through cost aggregation with guided filter [Tan & Monasse 14].
- Integral images for block matching [Facciolo et al. 14].
- Bilaterally weighted patches for disparity map computation [Fernandez & Monasse 15].

⇒ Others in preparation.
2. Some scientific and technical achievements

2. Scientific & technical achievements to establish a state of the art (2)

Stereovision

- First workshop demo delivering a 3D digital elevation model of the ground from satellite images.
- Available here: http://dev.ipol.im/~carlo/ipol_demo/workshop_s2p
First problem: reference programming languages

The chosen languages must:

- have a **stable API**;
- be **used intensively** by researchers and the industry;
- have stable **standard libraries**.
3. Technical Issues Overcome Through the Development of IPOL

First problem: reference programming languages

The chosen languages must:
- have a **stable API**;
- be **used intensively** by researchers and the industry;
- have stable **standard libraries**.

Accepted languages and libraries

- **First accepted languages**: ANSI C/C++;
- **currently accepted libraries**: libtiff, libjpeg, libpng, zlib, FFTW, GSL, Eigen, cblas, and clapack;
- **currently accepted MATLAB toolboxes**: Image Processing, Optimization, Wavelet;
- **new accepted frameworks**: MATLAB, Python (**with** NumPy and SciPy);
- other languages might be considered and accepted.
Second problem: design an online demo system

A **demo system** was created for this purpose, from scratch. It has to **manage the execution** of each online experiment:

- The **parameters and result pages** are different in each demo;
- the **test images** are different in each demo;
- users must be able to **upload their own images**;
- it has to be **fast** enough to allow online execution → **Multiple CPU** system;
- it has to **archive** user experiments (input data, input parameters, and results).
Second problem: design an online demo system

A demo system was created for this purpose, from scratch. It has to manage the execution of each online experiment:

- The parameters and result pages are different in each demo;
- the test images are different in each demo;
- users must be able to upload their own images;
- it has to be fast enough to allow online execution → Multiple CPU system;
- it has to archive user experiments (input data, input parameters, and results).

Problems detected in the current system

The current system is usable and functional, but we detected several problems:

- Creating a new demo implies coding in Python and designing HTML templates;
- non-scalable, too complex internal structure, and with lack of modularization.

⇒ Now moving towards a modular system with automatic demo generation.
Problems in the current demo system

- Most of the **problems** related to the **architecture** of the system;
- system designed as an object-oriented **monolithic** kernel;
- **too complex**, with **tightly interface-coupled** components;
- **non-scalable**;
- **not easy to distribute** the system over different machines.
Problems in the current demo system

- Most of the **problems** related to the **architecture** of the system;
- system designed as an object-oriented **monolithic** kernel;
- **too complex**, with **tightly interface-coupled** components;
- **non-scalable**;
- **not easy to distribute** the system over different machines.

Improvements

We’re currently working on **improving the system**

- **Modular architecture:**
  - Specialized standalone modules;
  - the monolithic kernel becomes a simple controller;
  - the core controller and the modules communicate via webservies.

- **Automatic demo generation:**
  - Each demo is specified using a simple textual description (name, type, and default value of the parameters and format of result page);
  - no need to code or design any page to build a new demo.
Quality of articles and a “natural selection”

Most **failed IPOL projects** aborted when:

- the described **algorithm** was **incomplete**;
- **did not give all the results** described in the paper;
- **run time** not reasonable;
- worked **only on a certain type of data**.
Progress towards the establishment of a full state of the art in each of the main sections of the journal

Identify algorithms representing a very substantial portion of the state of the art:

- Most efficient algorithms should be published;
- proposed theories and methods, even though they are not (or no longer) considered the best.
Progress towards the establishment of a full state of the art in each of the main sections of the journal

Identify algorithms representing a very substantial portion of the state of the art:

- Most **efficient algorithms** should be published;
- proposed theories and methods, **even though they are not** (or no longer) considered the best.

Some sections currently in IPOL

- Color and Contrast (10 articles)
- Denoising (15 articles)
- Demosaicking (6 articles)
- Interpolation (4 articles)
- Optical Flow (6 articles)
- Vision Through Turbulence (3 articles)
- ...
Goal: cover must of their respective state of the art (performance and descriptions of main mathematical techniques)

Others sections

- 3D
- Blur
- Computational Photography
- Geometry
- Infrared
- Learning and Detection
- Inpainting
- Image Comparison
- PDE
- Stereovision
- Texture
- ...

→ However, still incomplete!
Example of advantages of a reproducible algorithms with Stereovision: Middlebury stereo evaluation

- One method [Fernandez & Monasse 15] was implemented from an algorithm on adaptive neighborhoods [Yoon & Kweon 06];
- the obtained results were different!
- it was discovered that a post-processing must be applied to the main algorithm;
- there is no mention of a post-processing in the original article [Yoon & Kweon 06]!
Criticism to IPOL

- **Excessive effort** (rigor, run on any data, code description);
- **Excessive length** of the peer review report (including code review: slow and demanding);
- **Large number of objects** to be published (article, source code, demo);
- **No official impact factor**;
- **Frustrating to work on algorithms** designed by others;
- **Restricted number of authorized libraries or toolboxes** (as libjpeg, libtiff, FFTW, MATLAB Image Processing toolbox,...)
Authors and publishers praised IPOL for:

- Immediate impact of their publication;
- Impact due to the very existence of the demo;
- Gain tangible industrial and academic credibility;
- Facilitate obtaining research funding (ERC, ONR, ANR, DGA, CNES, FUI, ... )
Conclusions (I)

- **Reproducible Research** redefines the *output of the research*: not only the article, but also the *source* code and the *data*.
- **IPOL** is a complete and fully functional Reproducible Research journal: *articles, demos, data*. Everything is *free* or *open source*.
- For the **authors** it takes more effort to write Reproducible Research articles, but:
  - ⇒ *benefit is immediate* (credibility and *number of citations*);
  - ⇒ for both the IPOL article and the one published in a different journal with a demo available in IPOL).
- It is important to **adapt the journal to the community** needs and usages:
  - ⇒ *accept* commonly used languages, libraries, and frameworks.
Conclusions (II)

Pseudo-code is the main production of IPOL over the source code itself:

- The pseudo-code describes the significant parts of the algorithm;
- it does not contain all the details needed to encode it using an actual programming language;
- the pseudo-code is aimed to be readable, and reusable; ⇒ in general, it is between one and two orders of magnitude shorter than the actual source code.
- the pseudo-code is unambiguous to the mathematician reader.
What is Next? (I)

Facts:

- More than 5000 articles/year on Computer Vision and Image processing describing algorithms;
- however, about 200 of these articles would we enough to cover the CV and IP state of the art. Less than 400 if exhaustive.

IPOL is producing 40 articles/year → It can exhaust the state of the art (old and new) in about 7 years.
Moving towards a new way to do research

- What is next, then? → Focus on incremental research.
- Antecedents: analysis of the genome.
- Cycle:
  1. review established and published algorithms;
  2. combine them;
  3. improve them to achieve new and better results and applications.

In definitive, a new methodology to do research in CV and IP
Acknowledgement

Work partly founded by the **European Research Council** (advanced grant Twelve Labours), the **Office of Naval Research** (ONR grant N00014-14-1-0023), and **ANR-DGA** project ANR-12-ASTR-0035.
References I

[Stodden 09a] V. Stodden
The legal framework for reproducible scientific research: Licensing and copyright

[Stodden 09b] V. Stodden
Enabling reproducible research: Open licensing for scientific innovation

[Buckheit & Donoho 95] J. B. Buckheit and D. L. Donoho
Wavelab and reproducible research

[Gentleman 05] R. Gentleman
Reproducible research: A bioinformatics case study

Reproducible research in computational harmonic analysis

[Donoho 10] D. L. Donoho
An invitation to reproducible computational research

The Noise Clinic: a Blind Image Denoising Algorithm,
References II

[Monasse 11] P. Monasse
Quasi-Euclidean Epipolar Rectification

Kolmogorov and Zabih’s Graph Cuts Stereo Matching Algorithm

Stereo Disparity through Cost Aggregation with Guided Filter


Bilaterally Weighted Patches for Disparity Map Computation

The ipol initiative: Publishing and testing algorithms on line for reproducible research in image processing

[Gamma et al. 94] E. Gamma, R. Helm, R. Johnson and J. Vlissides
Design patterns: elements of reusable object-oriented software
REFERENCES III

[Yoon & Kweon 06] K.-J. Yoon and I. Kweon
Adaptive support-weight approach for correspondence search

[CGAL] Computational Geometry Algorithms Library,
http://www.cgal.org

[DGtal] DGtal: Digital Geometry tools and algorithms library,
http://libdgtal.org

IPOL Project Presented at the CMLA Seminar
CMLA ENS Cachan
Acknowledgement

IPOL citations (07/28/2015)

<table>
<thead>
<tr>
<th>Title</th>
<th>Cited by</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSD: a line segment detector</td>
<td>300</td>
<td>2012</td>
</tr>
<tr>
<td>Asift: An algorithm for fully affine invariant comparison</td>
<td>60</td>
<td>2011</td>
</tr>
<tr>
<td>Non-Local Means Denoising</td>
<td>82</td>
<td>2011</td>
</tr>
<tr>
<td>Rudin-Osher-Fatemi total variation denoising using split Bregman</td>
<td>48</td>
<td>2012</td>
</tr>
<tr>
<td>An Analysis and Implementation of the BM3D Image Denoising Method</td>
<td>32</td>
<td>2012</td>
</tr>
<tr>
<td>DCT image denoising: a simple and effective image denoising algorithm</td>
<td>30</td>
<td>2011</td>
</tr>
<tr>
<td>Implementation of the &quot;Non-Local Dayes&quot; (NL-Dayes) Image Denoising Algorithm</td>
<td>26*</td>
<td>2013</td>
</tr>
<tr>
<td>TV-L1 Optical Flow Estimation</td>
<td>24*</td>
<td>2013</td>
</tr>
<tr>
<td>Horn-chunick optical flow with a multi-scale strategy</td>
<td>19</td>
<td>2013</td>
</tr>
<tr>
<td>Automatic homographic registration of a pair of images, with a contrario elimination of outliers</td>
<td>19</td>
<td>2017</td>
</tr>
</tbody>
</table>

M. Colom et al. (CMLA,LORIA,LIGM)  IPOL: a new journal for fully reproducible research  Workshop RCBR 2015
### Acknowledgement

**IPOL citations (07/28/2015)**

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic homographic registration of a pair of images, with a contrario elimination of outliers</td>
<td>2012</td>
</tr>
<tr>
<td>Total variation deconvolution using split Bregman</td>
<td>2014</td>
</tr>
<tr>
<td>Automatic color enhancement (ACE) and its fast implementation</td>
<td>2012</td>
</tr>
<tr>
<td>Algebraic lens distortion model estimation</td>
<td>2011</td>
</tr>
<tr>
<td>Chan-Vese segmentation</td>
<td>2012</td>
</tr>
<tr>
<td>Simplest color balance</td>
<td>2011</td>
</tr>
<tr>
<td>An Analysis of the Viola-Jones Face Detection Algorithm</td>
<td>2014</td>
</tr>
<tr>
<td>Exemplar-based Texture Synthesis: the Efros-Leung Algorithm</td>
<td>2013</td>
</tr>
<tr>
<td>Micro-texture synthesis by phase randomization</td>
<td>2011</td>
</tr>
</tbody>
</table>