The IPOL Initiative: Publishing and Testing Algorithms on Line for Reproducible Research in Image Processing

Nicolas Limare, Jean-Michel Morel

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 00 (2011) 1–10
Procedia Computer Science

The IPOL Initiative: Publishing and Testing Algorithms on Line for Reproducible Research in Image Processing

Nicolas Limare[a], Jean-Michel Morel[b]

CMLA, ENS Cachan, CNRS, UniverSud, 61 Avenue du Président Wilson, F-94230 Cachan

[a]nicolas.limare@cmla.ens-cachan.fr
[b]jean-michel.morel@cmla.ens-cachan.fr

Abstract

Image Processing On Line (IPOL) publishes image processing and image analysis algorithms, described in accurate literary form, coupled with code. It allows scientists to check directly the published algorithms on line by providing a web execution interface on any uploaded image.

This installation acts the universality of image science. It permits to transcend the artificial segmentation of the research community in groups using this or that image software, or working on dedicated incompatible image formats. It promotes reproducible research, and the establishment of a state of the art verifiable by all, and on any image.

This paper describes the technical challenges raised by the foundation of this new kind of journal and its scientific evaluation issues. It finally analyzes the first publications, to demonstrate its potential impact on the development of image science.

Keywords: reproducible research, image processing, implementation, web interface

1. Why Image Processing on Line?

For computational sciences, detailed algorithms and implementations are an integral part of the research results and should therefore be published extensively. This has been continuously pointed out in the literature about reproducible research, from the seminal papers by Claerbout [2] and Buckeit and Donoho [3] to recent columns about software quality and computational sciences in Nature [4, 5]. This goal is far from attained in the image and signal processing community, where only 12% of the published articles include the implementation details and 9% provide a source code [6].

Indeed, image processing and computer vision are young sciences which emerged at the end of the seventies. Their publication system is far from mature. Algorithmic exchange in this community faces serious obstacles: multiple software developing environments have grown in each research group without interoperability. Source code is subject to portability issues and depends on local tools to process or exploit the result. And software maintenance is a costly issue. A tested, reliable implementation is estimated to require three times more resources than a working prototype [7], and integrating different implementations together requires again three times more. Most labs have not the permanent workforce to finalize and maintain a software and to adapt it to the changing computing environment. Next to missing software engineering skills of scientists, software maintenance and consolidation is therefore one of the main problems in this field.

In the ensuing Babel Tower, research groups are unable to communicate efficiently by software and are therefore limited to paper journals and conferences. Image processing libraries and development environments are not a suffi-

cient response, because in the absence of a common base they reinforce the software fragmentation. Existing software journals, when they only publish codes, lack the precise scientific evaluation and specification of the implemented algorithms and a discussion about their properties, qualities and limits. When an implementation is provided by the authors of a research article, it is usually available *as-is* on their personal research web pages. Out of the peer-reviewed scientific publishing process, *there is no control that the provided source code exactly implements the described algorithm, there is no guarantee on the correctness and usability of this implementation, or on its long-term availability*. We have systematically observed that the code proposed differs significantly from the paper publication, that the paper publication is not enough to characterize an algorithm, and that conversely the disclosed code contains parts that are not documented or explained in the paper.

The goal of the "Image Processing On Line" (IPOL) [1] initiative is to publish complete and certified implementations together with the precise algorithm description, submitted to a peer-review. This should enable performance and quality evaluations and comparisons between algorithms, a task difficult to achieve today because implementations are often missing or unreliable. Making implementations into a the reviewed and published material will reward the software quality, which otherwise is neglected by paper publications.

1.1. On-line Testing and Experiment Sharing

Despite the availability of a reliable source code, an algorithm may not be immediately usable because its compilation, installation and use is not straightforward. Many researchers are reluctant to get into a compilation procedure to check an algorithm. Most would prefer a quick test before they consider spending some more to study the article in depth. Our proposed solution is to provide an experimentat environment directly accessible over the network.

Despite various individual initiatives for web-based image processing, some as early as 1998[1], dozens of on-line photo editing and sharing services demonstrating the availability of the tools, industrial solutions being developed by large organizations for remote data analysis[2], and recent projects ensuring a reliable experimental environment with cloud infrastructure [8], the image processing research community still barely uses experimental resources accessed over the network. This is far behind other research fields such as the large on-line databases and research tools used in genetics[3].

We intend to remedy to this and the consequent lack of experiment sharing in computational sciences by providing a web-based test interface for all the algorithms published on IPOL, allowing any researcher to freely test any implementation on any data. The experimental data is archived and publicly accessible. This archive can be used to share experiments between researchers, and it is an efficient mean to assess the performances of an algorithm over a large collection of input images.

1.2. The IPOL Publishing Model

We devised a new model to evaluate, preserve and disseminate research. IPOL is now a solution for reproducible research, executable algorithms and experiment sharing, publicly available on `http://www.ipol.im/`:

documentation
: Articles are published as web pages with embedded formulas and figures, attached data and a portable minimal implementation. Their main content is the precise description of an algorithm.

implementation
: The implementation is used to evaluate the algorithm by careful examination of the source code and careful testing.

demonstration
: An on-line demonstration system is proposed to run the algorithm on a server over a web interface, with freely uploaded input data.

[1] Rolf Henkel's "Online Imageprocessing Pages" has been an academic on-line image processing facility created in 1998 or before and available until 2002 (see `http://axon.physik.uni-bremen.de/online_calc/`).

[2] The European Space Agency (ESA) develops KAOS as part of an infrastructure used for the analysis of satellite data using remote servers (see `http://keo-karisma.esrin.esa.int/keo-home/KAOS.html`).

[3] The National Center for Biotechnology Information (NCBI) maintains various databases and tools (see `http://www.ncbi.nlm.nih.gov/`). The Science Commons project develops knowledge management systems for biomedical research (see `http://sciencecommons.org/`).

archive

The experiments conducted with the on-line demonstration system are collected into a publicly available archive.

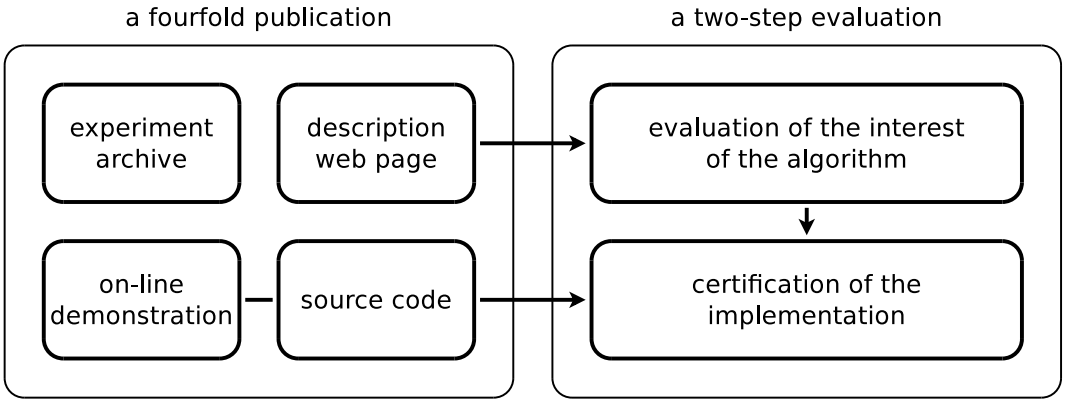

Figure 1: IPOL content and evaluation process.

Similar in its form to a classic research article but focused on the implementation, the precise description of the algorithm at IPOL is the main part of the publication. It must be detailed enough to allow any specialist to implement it in their own programming language and environment and validate it by comparison with the IPOL code and demo. IPOL aims at publishing the most accurate descriptions of existing or new algorithms, with implementations as generic and as reliable as possible. The implementation is guaranteed by the peer-reviewed evaluation to match accurately the mathematical description of the algorithm. It runs on line to grant researchers immediate testing.

The role of referees is different from (and complementary to) to a classic journal. An IPOL article evaluation runs in two steps. At the first stage, referees only have to read the submitted web page, which must contain an accurate algorithm description and the most significant experiments. If the referees are positive the article is "conditionally accepted" and the authors are invited to submit a code, carefully commented and linked to the algorithmic description.

The "conditional acceptation" criterion is not the novelty of the algorithm: in many cases the submission develops an already published algorithm, or an algorithm simultaneously submitted to a classic paper journal. The acceptation criterion is merely the interest for the community to have the proposed algorithm specified, implemented and available on line. The second phase of the evaluation is more technical and concludes with a certification by the referees. The algorithm must be completely described, in such a way that it is reproducible. The code and online demo must implement exactly the algorithm.

The on-line demonstration system and its archive are maintained by the IPOL editorial board, and each new demonstration is created and installed by the board according to the specifications given by the algorithm author, using the exact same implementation available on the algorithm web page.

1.3. The Potential Impact on the Field

There are some twenty image processing and image analysis journals and an equivalent number of international conferences with more than 4000 articles every year in the community. There is therefore a desperate need for a state of the art permitting to compare and evaluate the methods. The goal of IPOL is not only to favor a fair evaluation of new methods, but also to establish a comparative state of the art by publishing in this new form all relevant algorithms of the field. Thus, all relevant algorithms will be *republished* in IPOL.

The potential impact of IPOL on future image processing and computer vision research can be summarized in the following points:

1. establish a state of the art with benchmarks comparing the major algorithms;
2. certify a correspondence between paper, algorithm and code;
3. publish benchmark data with certified properties;
4. fix the form of classic algorithms, diminishing the time wasted by researchers on reprogramming them;
5. reward the authors for this invaluable service by a peer reviewed publication.

2. How IPOL Works

2.1. Native Web Content

Most scientific publishers produce electronic versions of printed articles (generally PDF files), distributed *via* the network. This still limits the authors to the model of printed articles and implies that any non-printable content such as data and source code is handled aside, sometimes in an on-line repository provided by the publisher, often simply on the authors' personal web space.

We prefer to publish articles *in* the network as native web pages including all the material, source code, data, files and figures required for the full evaluation of an algorithm. The Word Wide Web as a publishing and documentation model [9] provides means to distribute a document with structured text, tables, figures and any attached file, without the artificial size limits imposed in printed journals. Document layout and interaction support is reliable across major browsers for all the base technologies (content, presentation, interaction, identification) and new standards are actively developed and adopted (video, math formula, vector graphics)[4].

IPOL currently uses the Ikiwiki[5] software as a document publishing system. Other solutions are possible, from generic web content management systems to industry-grade publishing chains. The document publishing system for IPOL was chosen do be a simple and immediately available solution without in-house development, but may change in the future as new functions are added.

2.2. Standard-Compliant Portable Compiled Code

Algorithms implemented for IPOL must be usable on any major computing system. Cross-platform compatibility is required at least for the Windows, Mac OS X and Linux/Unix families of operating systems, and with 32 and 64 bits variants of the x386 processor line. This is a bare minimum, and in fact we don't expect the implementations to be tied to any operating system or hardware architecture.

Process virtual machines (Java, .NET) or interpreted languages (Perl, Python) could be a mean to achieve this portability, but this solution is not realistic because it would imply sub-efficient implementations, as shown by programming language benchmarks on computationally intensive algorithms[6]. Moreover, such programming languages would be an obstacle to practical exchange and re-use of the implementations because they would not be easily merged with other code without sacrificing performance. The same compatibility issues would be faced with proprietary scientific computing environments, which in addition don't provide any assurance of the future usability of the implementations. For these reasons, we consider that the only way to ensure efficient portability is to require the algorithms to be implemented in a low-level compiled language. C, C++, Fortran are some examples of programming languages widely used in the research community, adapted to intensive numerical computations, and supported on all common computing architectures and operating systems.

Algorithms exposed in IPOL articles must must still be usable many years later to ensure long-term verifiability of the scientific results. For this reason, implementations are required to follow an established standardized language specification (ANSI C89, C99, ISO C++98, Fortran 90, ...). We use static code analysis and strict compilation as a "best effort" procedure to test source code for future compatibility. The portability applies to all the code: no OS specific feature can be expected from the standard library and the programs will only interact via the only portable interface, the command line interface. For the same reasons, a perfect implementation would require no external library and include all the source code needed to produce the executable program. For practical reasons, exceptions are granted for well-known reliable and portable software libraries: IPOL currently allows the libpng and libtiff libraries

[4] Internet Explorer 7 (released in October 2006), Mozilla Firefox 2 (released in October 2006), Opera 9 (released in June 2006) and Safari 2 (released in April 2005) all had at least a reasonable support for the content description (HTML4), presentation (Cascaded Style Sheets — CSS) and user-side interaction (JavaScript) standards. This is expanding since with native video (HTML5 `<video>` tag), math formula (MathML), vector graphics (Scalable Vector Graphics — SVG, HTML5 `<canvas>` tag) and 3D rendering (webGL) standardized or expected soon. All this content lives in the same document space (Document Object Model — DOM) and can be interconnected for a rich interactive experience. The article and its components can be identified by unique identifiers (Digital Object Identifiers — DOI) for reliable references, citations and interlinks.

[5] Ikiwiki is a wiki compiler (see `http://ikiwiki.info/`).

[6] The "Computer Language Benchmarks Game" is an exhaustive and constantly updated benchmark over many languages and typical algorithms (see `http://shootout.alioth.debian.org/`).

to handle the image file input/output, the FFTW library for Fourier transforms, and the BLAS API and LAPACK library for linear algebra[7].

This may seem very restrictive but so far all the algorithms published or in the publishing process have been able to follow these rules or to find workarounds. Many interesting and powerful image processing algorithms can be implemented without requiring the support of external libraries, but this may not be the case in other research fields where this policy would need to be revised.

2.3. Web Execution Interface

IPOL also provides a demonstration system for every algorithm. Client-side models were excluded because, as discussed before, the solutions for a portable executable program are currently not efficient and would not show the best performances of the algorithms[8]. We preferred server-side hosted demonstrations to avoid compatibility problems and ensure consistent optimal performance in a controlled experimental environment with high-performance hardware resources. We chose to make this system accessible over a web interface because of the ubiquity of web browsers and their use as a flexible interface to any resource over the networks, with web-based applications existing since 1993 [10].

This interface is not optimal for our needs. Among other problems, we can cite the lack of a built-in persistence mechanism in the HTTP protocol [11] to keep track of the history of the user interactions and provide an "undo" option, and the very limited feature set of XHTML controls [12] (buttons, text fields, ...) compared to what is offered from any desktop graphical user interface. On the other side, a web interface has the immense advantage over other client-server models to be immediately usable for anyone with a computer connected to the network and a browser. It also is a well tested model with a 20 years history and a large collection of tools, servers, frameworks and libraries.

The typical IPOL experiment work-flow is:

1. Upload an input image or select one from a default list. The uploaded images are converted from any common format into the file format and image type expected by the algorithm implementation.
2. Set some parameters and/or perform some pre-processing. The pre-processing tools currently available include some image editing, cropping/zooming and adding noise.
3. Execute the algorithm on the server and visualize the results. Experiments conducted with original uploaded images are archived and publicly available if agreed by the uploader.

IPOL uses native HTTP server mechanisms for data transfer, user authentication, logging and usage statistics tracking. An "experiment key" is used to identify the data set and provide continuity across the successive pages required to run the experiment. Simple XHTML form controls are sufficient for most of the demonstration interfaces and some JavaScript tools have been developed when usability improvements are needed.

We chose to use the Python language and CherryPy framework[9] to implement the server-side back-end system whose tasks are to receive and store the input images and then execute the algorithm using the implementation with some optional pre-processing dialogs. The communication between users and the server is made of HTTP requests sent by the web browser as a result of their local actions and the answers received from the server, formatted as XHTML pages. Our current system can be extended for new algorithms by reusing common building blocks from an application library. Once this young software stabilizes, it will be released to help the creation of other similar initiatives.

So far, network capacity and file size are not an issue, mainly because image files are relatively light and because a limit is imposed on the execution time. A web user is usually not ready to wait for more than 10 seconds until a page is available [13, chapter 5]. In our context, we estimate that researchers can wait for 30 seconds during the execution of an algorithm if they receive a correct feedback. In order to keep the execution time under 30 seconds, the size of

[7] libpng is the reference implementation of the PNG image file format (see `http://www.libpng.org/pub/png/libpng.html`). libtiff is the *de facto* reference implementation of the TIFF image file format (see `http://www.remotesensing.org/libtiff/`). FFTW is a high-performance and portable library for discrete Fourier transforms (see `http://www.fftw.org/`). BLAS (Basic Linear Algebra Subsystem) is the specification of elementary linear algebra libraries (see `http://www.netlib.org/blas/`). LAPACK (Linear Algebra PACKage) is a library for linear algebra built on BLAS (see `http://www.netlib.org/lapack/`).

[8] An increasing part of the network traffic comes now from mobile and handheld devices with a very limited processing power.

[9] CherryPy is a light framework for web applications written in Python (see `http://www.cherrypy.org/`).

the input images is limited by cropped or resized large images, depending on the kind of algorithm. This effectively limits the volume of data exchanged and the bandwidth requirements. We expect the situation to keep balanced in the future with network capacity to grow together with hardware performances and typical file sizes.

Larger data or more computationally demanding algorithms could also be processed on the server in a non-interactive processing queue, as it is currently done by some other demonstration tools[10]:

1. upload some data or provide a way to retrieve it from the server and set the execution parameters;
2. the data is enqueued with instructions about how to process it;
3. the algorithm is run when some processing resources are available;
4. a notification email is sent to the user with the instructions needed to access the result of the algorithm.

2.4. Open Access and Free Licenses

IPOL subscribes to the Open Access principles and all its content is freely accessible. The copyright is kept by the authors on all their contributed content, allowing them to re-use their works at will. In accordance with Stodden's *Principle of Scientific Licensing* [14], dissemination, sharing, use, and re-use of the scientific research published by IPOL is facilitated by the use of Creative Commons licenses for the articles and Free Software licenses for the implementations when possible[11].

2.5. Security and Legal Context

IPOL publishes articles in the form of web pages and as such, the published content must be moderated to avoid an abuse of the service. This is achieved by identifying the authors by their name, institution and a verified contact address. The creation of any new article on IPOL requires a manual confirmation by the editorial board, and new articles are not publicly accessible until they have been reviewed and "conditionally accepted". IPOL authors are required to credit the authors republished algorithms and the origin of all the software components they re-use. Potential plagiarism issues are the same as for any scientific journal. IPOL as a web site can also be the target of external attacks, but this can be handled by generic system and network administration measures.

The demonstration system is more exposed because it provides server resources to the public by running software provided by the authors with data coming from the users. Different attacks are possible: the algorithm implementations could use all the computing resources, they could be malicious or have remotely exploitable bugs[12] and damage the server or use it to attack other servers. This exposure is limited by the careful examination of the source code during the review process, the restriction of the rights of the demonstration system on the server and of the total execution time a program can use, and the network isolation of this server. Virtual machines and operating system containers are considered to execute the demonstrations in an isolated environment without noticeable performance penalties.

Experiment archives could also be abused and unlawful images inserted. So far this has not been an issue, and the use of the demonstrations could be rate-limited or require a preliminary identification. The French law [15] (so far, IPOL operates from France) exempts IPOL as the on-line editor from liability for the archive content if it is explicitly not moderated and if unlawful content is promptly removed after notification.

3. Current Activity

The first IPOL prototypes were put on-line in December 2008 and the current version dates from December 2009 for the web publishing part and September 2010 for the demonstration server. The first article was reviewed, approved and published in July 2010. IPOL had in March 2011 a dozen algorithms published or in the final review stages and twice more in the pipeline. More than 50 international academic experts compose the scientific committee responsible

[10] In the non-interactive web demonstration interface proposed by Tomas Pajdla for his "structure from motion" algorithm, identified users upload a set of several images and receive later an email with the result of the algorithm (see `http://ptak.felk.cvut.cz/sfmservice/`).

[11] Creative Commons licenses allow the licensed content to be re-used as long as the original author is credited (see `http://creativecommons.org`). Free Software licenses allow unrestricted use, study and modification of a software with minimal restrictions. IPOL encourages the use of the GPL or BSD licenses.

[12] For example, a bug in the image reading library used by some demonstrations could be exploited by uploading specially crafted images to execute arbitrary code on the server.

for the review process, and the editorial board handling the new algorithms. A recent agreement between IPOL and the SIAM Journal of Imaging Sciences (SIIMS) [16] encourages authors to submit *simultaneously* algorithm and code to IPOL and the corresponding theory to SIIMS. Every month, the algorithm demonstrations are used for about 1500 experiments. 20000 original experimental data have been collected so far in the archives.

At this point IPOL is still experimental, and faces new challenges with almost every new algorithm publication. The delicate question of how an image processing algorithm can be explained, specified and illustrated on line is the object of many trials before some satisfactory solution is reached. Once a satisfactory on line demo solution is found for a particular image processing problem, it is copy-pasted for all further submissions dealing with the same problem. Thus, the editing workload seems to be controllable. It has led to the constitution of a fifteen members editorial board to assist authors. This board is distinct from the scientific board. Exposing an algorithm to the public and allowing it to be applied on any data is not a small challenge. It often requires from the authors a serious rethinking, rewriting and many trials. This explains why the publication of even classic algorithms ends up in authentic and innovative research.

3.1. Published Algorithms

We briefly present hereafter some of the algorithms available in IPOL. These are only seeds of what IPOL can become if the research community subscribes to this publishing model.

Figure 2: Algebraic Lens Distortion Model Estimation

Figure 3: Non-local Means Denoising

Algebraic Lens Distortion Model Estimation [17]

This algorithm presents a new method for correcting optical camera distortion (fig. 2). It is the first article evaluated and published by IPOL. The corresponding paper article was previously published in a 2009 article [18] and counts so far three citations in Google Scholar. In contrast, its IPOL archive shows 550 on line experiments with original data between May 2010 and January 2011. Its source code was downloaded 230 times in the same period of time.

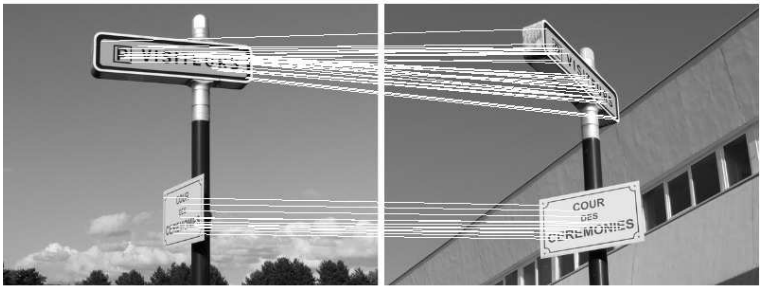

Figure 4: ASIFT: An Algorithm for Fully Affine Invariant Comparison

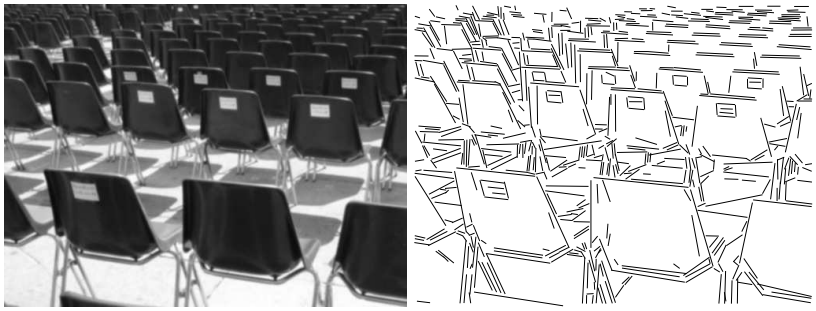

Figure 5: LSD: a Line Segment Detector

Non-local Means Denoising [19]

Any digital image taken by a camera needs a sequence of restoration operations before it is delivered in visible form. Denoising is the key operation which conditions the quality of the other ones. Non-local means (fig 3) was proposed in a 2005 article [20], referenced 477 times. Since its appearance in IPOL in November 2009, the algorithm has been tested on 1800 uploaded images and the code has been downloaded more than 450 times. Thus the observable number of readers in only one year is comparable to five years of a classic publication.

ASIFT: An Algorithm for Fully Affine Invariant Comparison [21]

This on-line article is the flagship of IPOL, with 8300 on line experiments on original image pairs (fig. 4). The reason for the interest is obvious: ASIFT is an improvement of the classic SIFT method published in 2004 [22] used to detect whether two digital images have objects in common or not. This problem is the core of the image retrieval problem, it is the first stage of many robotic and scene reconstruction projects, and it is systematically used for photo stitching. Hence the need for a standard and rigorous reference implementation. Although ASIFT only is a first approximation to that, the code has been downloaded more that 1500 times in 15 months.

LSD: a Line Segment Detector [23]

Probably the most puzzling publication is this one (fig. 5), because of its obvious impact demonstrated by 3900 on-line experiments in 18 months and more than 200 code downloads. In contrast, the 2009 journal paper [24] is so far cited 10 times! One explanation is that, while detecting segments or edges in an image has been a hot topic in the eighties and nineties, it is not currently considered a research subject in the computer vision community, and no new paper appears on it. Nevertheless, most scientific and technical imaging techniques require reliable segment detectors. The immense variety of the on-line experiments proves it. This also demonstrates that an on-line demo can characterize much better the impact on technology and society of a research than citation indexes do.

4. Conclusion: the Scientific Program

The hypothesis underlying IPOL is the *universality of image science*. Images currently created by mankind are incredibly diverse. They can be snapshots of current life and artworks for the wide public. Astronomers and geographers produce images of the universe, the Sun, the Earth and other planets in many wavelengths and resolutions. Material science and physics have an immense diversity of scientific samples. Last but not least medicine and biology create images of all living organisms in an almost infinite scale from organs to proteins. Human perception is prepared to this diversity, and seems to be able to adapt quickly to any new kind of image. The existence of universal visual primitives, of "perception atoms" is hypothesized in neurophysiology (where it leads to a cartography of visual areas by their geometric retinotopic primitives), in psychophysics, where it leads to experiments with abstract images exciting these visual primitives, and finally in computer vision where the goal is to emulate them numerically.

In this universal context of imaging science the existence of incompatible image formats and software is highly counterproductive. While some very specific imaging tasks might indeed require peculiar dedicated algorithms, computer vision and image processing are highly successful in designing algorithms valid for *any kind of images*. This is the case for two of the first published algorithms in IPOL, LSD (which extracts segments in any image) and SIFT (which can recognize any solid shape, from galaxies to logos.)

This is why IPOL will encourage researchers to publish and confront systematically existing generic image processing algorithms to new ones. In 2011–2012 at least four benchmark publications should appear in IPOL. One will deal with image denoising and will confront the state of the art methods. Other benchmarks will deal with operations as universal and basic as zooms in and out, color demosaicking, contrast adjustment, camera blur estimation. If everything goes according to plan, all researchers using images (not necessarily image processing specialists) should soon be able to test most state of the art algorithms on their own images.

[1] Image Processing On Line (IPOL).
ISSN:2105-1232 DOI:10.5201/ipol URL http://www.ipol.im/
[2] J. Claerbout, Electronic documents give reproducible research a new meaning, in: Proceedings of the 62nd Annual International Meeting of the Society of Exploration Geophysics, 1992.
[3] J. B. Buckheit and D. L. Donoho, Wavelab and reproducible research, Tech. Rep. 474, Department of Statistics, Stanford University (1995).
URL http://www-stat.stanford.edu/ wavelab/Wavelab_850/wavelab.pdf
[4] N. Barnes, Publish your computer code: it is good enough, Nature, 2010.
DOI:10.1038/467753a
[5] Z. Merali, Computational science: ...error, Nature, 2010.
DOI:10.1038/467775a
[6] P. Vandewalle, J. Kovacevic and M. Vetterli, Reproducible research in signal processing, IEEE Signal Processing Magazine.
DOI:10.1109/MSP.2009.932122
[7] F. P. Brooks, The Mythical Man-Month: Essays on Software Engineering (2nd Edition), Addison-Wesley Professional, 1995.
ISBN:0201835959
[8] P. Van Gorp and P. Grefen, Supporting the internet-based evaluation of research software with cloud infrastructure, Software and Systems Modeling, 2010.
DOI:10.1007/s10270-010-0163-y
[9] T. Berners-Lee and R. Cailliau, Worldwideweb: Proposal for a hypertext project (1990).
URL http://www.w3.org/Proposal.html
[10] D. Robinson and K. Coar, The Common Gateway Interface (CGI) version 1.1, IETF Request for Comments (RFC3875), 2004.
URL http://tools.ietf.org/html/rfc3875
[11] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee, Hypertext Transfer Protocol – HTTP/1.1, IETF Request for Comments (RFC2616), 1999.
URL http://tools.ietf.org/html/rfc2616
[12] W3C HTML Working Group, Xhtml 1.0 the extensible hypertext markup language (second edition), 2002.
URL http://www.w3.org/TR/xhtml1/
[13] J. Nielsen, Usability Engineering, Morgan Kaufmann, 1993.
[14] V. Stodden, Enabling reproducible research: licensing for scientific innovation, International Journal of Communicatons Law and Policy, 2009.
[15] French law 82-652 on audiovisual communications, 29 july 1982, article 93-3, modified by the law 2009-669, 12 June 2009, article 27.
[16] SIAM Journal on Imaging Sciences (SIIMS).
URL http://www.siam.org/journals/siims.php
[17] L. Alvarez, L. Gomez and J. R. Sendra, Algebraic lens distortion model estimation, Image Processing On Line, 2010.
DOI:10.5201/ipol.2010.ags-alde
[18] L. Alvarez, L. Gomez and J. R. Sendra, An algebraic approach to lens distortion by line rectification, Journal of Mathematical Imaging and Vision, 2009.
DOI:10.1007/s10851-009-0153-2.
[19] A. Buades, B. Coll and J.-M. Morel, Non-local means denoising.
URL http://www.ipol.im/pub/algo/bcm_non_local_means_denoising/
[20] A. Buades, B. Coll and J.-M. Morel, A review of image denoising algorithms, with a new one, Multiscale Modeling and Simulation, 2006.
DOI:10.1137/040616024
[21] G. Yu and J.-M. Morel, ASIFT: An algorithm for fully affine invariant comparison, Image Processing On Line, 2011.
DOI:10.5201/ipol.2011.my-asift
[22] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, 2004.
DOI:10.1023/B:VISI.0000029664.99615.94
[23] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, LSD: a line segment detector.
URL http://www.ipol.im/pub/algo/gjmr_line_segment_detector/
[24] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel and G. Randall, Lsd: A fast line segment detector with a false detection control, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010.
DOI:10.1109/TPAMI.2008.300