PySAP: Python Sparse Data Analysis Package for Multidisciplinary Image Processing

S. Farrens\textsuperscript{a,b}, A. Grigis\textsuperscript{b}, L. El Gueddari\textsuperscript{b}, Z. Ramzi\textsuperscript{a,b}, Chaithya G. R.\textsuperscript{b}, S. Starck\textsuperscript{c}, B. Sarthou\textsuperscript{d}, H. Cherkaoui\textsuperscript{b}, P. Ciuciu\textsuperscript{b}, J.-L. Starck\textsuperscript{a}

\textsuperscript{a}AIM, CEA, CNRS, Université Paris-Saclay, Université Paris Diderot, Sorbonne Paris Cité, F-91191 Gif-sur-Yvette, France
\textsuperscript{b}CEA/DRF/Joliot NeuroSpin, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France
\textsuperscript{c}EPITA, 14-16 rue Voltaire, 94270 Le Kremlin-Bicêtre, France
\textsuperscript{d}ENSTA ParisTech, 828 Boulevard des Maréchaux, 91120 Palaiseau, France

Abstract

We present the open-source image processing software package PySAP (Python Sparse data Analysis Package) developed for the COmpressed Sensing for Magnetic resonance Imaging and Cosmology (COSMIC) project. This package provides a set of flexible tools that can be applied to a variety of compressed sensing and image reconstruction problems in various research domains. In particular, PySAP offers fast wavelet transforms and a range of integrated optimisation algorithms. In this paper we present the features available in PySAP and provide practical demonstrations on astrophysical and magnetic resonance imaging data.

Keywords: Image processing, convex optimization, reconstruction, open-source software

1. Introduction

The ability to obtain high quality data in a short amount of time or indeed to recover high resolution images from undersampled blurred and noisy data can significantly improve the results of experiments potentially leading to new and exciting scientific discoveries. While the benefits of the mathematical methods that make this possible are relatively well known, robust and easy-to-use software tools that implement these techniques are extremely rare. The Compressed Sensing for Magnetic Resonance Imaging and Cosmology (COSMIC) project (\url{http://cosmic.cosmostat.org/}) was funded by the Fundamental Research Division (DRF) at the French Alternative Energies and Atomic Energy Commission (CEA) to provide precisely these tools.

COSMIC is a collaboration between two CEA groups with signal processing expertise: NeuroSpin, specialists in Magnetic Resonance Imaging (MRI), and CosmoStat, specialists in astrophysical image analysis. There is significant overlap in these fields, especially for astrophysical radio imaging that, like MRI, collects data in Fourier space. The primary output of this collaboration has been the development of the Python Sparse data Analysis Package (PySAP).

PySAP is an open-source software package written in Python that provides highly optimised sparse image transforms and a library of modular optimisation tools for solving linear inverse problems. While PySAP has been designed with specific applications to the MRI and astrophysics domains in mind, the versatility of the software and the universality of the mathematical techniques mean that it can also be applied to a variety of other imaging domains such as microscopy, tomography and echography.

This paper is organised as follows. Section 2 provides a detailed description of the structure and features of PySAP with particular focus on the image transforms and optimisation tools. Section 3 demonstrates practical applications of PySAP on MRI and astrophysical data. Finally, conclusions and plans for the future development of the package are presented.

2. PySAP Features

In essence, the base PySAP package serves as a front-end that comprises several specialised modules. PySAP provides a simplified framework in which to combine these modules as well as managing file IO, visualisation and exception handling.

The core modules that provide the PySAP features are:

- Sparse2D: Sparse Image Transforms
- ModOpt: Modular Optimisation Tools
- Plug-ins

Fig. 1 illustrates the core structure of the PySAP package. Each of these modules is described in detail in the following subsections.

2.1. Sparse Image Transforms

The most essential tools for implementing compressed sensing or sparsity in signal processing problems are effi-
Figure 1: Illustration of the structure of the PySAP package. The SPARSE2D and ModOpt core libraries are represented in orange and red, respectively. The various plug-in applications appear in blue.

In practice, these dictionaries correspond to a series of data transforms ranging from wavelets to curvelets that can convert the data into a domain where the majority of the information is concentrated in very few non-zero coefficients, a concept called sparsity or compressibility.

Sparse2D is a C++ package, which forms part of iSAP (Interactive Sparse Astronomical data Analysis Packages), that provides a wide range of robust and efficient sparse transforms for 1D, 2D and 3D signals. In particular, Sparse2D includes a collection of undecimated wavelet transforms (UWT) that provide shift invariant properties for image reconstruction, such as the starlet transform (Starck et al., 2007) or the 7/9 UWT. These transforms are well documented in Starck et al. (2015). These software tools have been extensively tested on astrophysical data producing high quality results on a range of different topics (Bobin et al., 2014; Leonard et al., 2014; Ngole Mboula et al., 2015; Lanusse et al., 2016). The fact that this package relies on a set of fixed multiscale dictionaries means that the transforms are very computationally efficient and are therefore ideally suited to on-line MR image reconstruction (Ramzi et al., 2019).

PySAP provides Python bindings for the Sparse2D C++ libraries, thus enabling fast and efficient implementation of the sparse transforms inside of a Python environment. This allows these tools to be more easily integrated into optimisation problems without any loss of performance (see section 2.2). Additionally, through the PySAP interface, Sparse2D transforms can be applied to MRI data separately on the real and imaginary parts.

Table 1: List of proximity operators currently available in ModOpt.

| Proximity Operator | Application         |
|--------------------|---------------------|
| Positivity         | Image analysis      |
| $\ell_1$ Minimisation | Sparse Regularisation |
| $\ell_*$ Minimisation | Low-rank Regularisation |

2.2. Modular Optimisation Tools

Linear inverse problems, such as compressed sensing, are ill-posed because they are under-determined, i.e. the number of measurements is far below the number of image pixels. To cope with this issue and make the inverse problem well posed, one usually resorts to regularisation. The image solution is then obtained as the minimiser of an optimisation problem. One of the main features of PySAP is a series of modular optimisation tools designed for solving linear inverse problems that comprise a subpackage called ModOpt.

This package is particularly well suited for solving linear inverse imaging problems of the following form

$$\mathbf{y} = \mathbf{H} \mathbf{x} + \mathbf{n}$$  \hspace{1cm} (1)

where $\mathbf{y}$ is the observed image obtained from the detector in question, $\mathbf{H}$ is a degradation matrix that could constitute blurring, sub-sampling, distortion, etc., $\mathbf{x}$ is the true image that one aims to recover and $\mathbf{n}$ is noise.

ModOpt provides robust and extremely flexible implementations of cutting-edge optimisation algorithms such as Forward-Backward, FISTA (Beck and Teboulle, 2009), Generalized Forward-Backward (Raguet et al., 2011), Condat-Vu (Condat, 2013; Vu, 2013) and POGM (Kim and Fessler, 2017). The flexibility of these implementations is provided via means of Python class composition. All of the proximity and linear operators as well as the gradient utilised by a given algorithm can be provided as class instances that inherit a parent structure to ensure smooth cohesion. The modularity of this approach means that any potential bug can be easily identified and fixed, thus ensuring a well maintained and robust framework. Additionally, this structure facilitates the future implementation of virtually any optimisation algorithm.

Predefined proximity operators are provided for implementing sparse and low-rank regularisation (i.e., $\ell_1$ and nuclear norms, respectively) as well as a positivity constraint, which is commonly required in image analysis problems. Tools are included that allow the automatic setting of the regularisation parameters using the noise properties of the observed data. New proximity operator instances can easily be generated using the parent class. A list of the proximity operators currently available in ModOpt is provided in table 1. This structure includes a method that automatically calculates a given operator’s contribution to the overall cost of the optimisation problem at hand.

The linear operator parent class enables the use of any of the sparse image transforms described in section 2.1.
The structure of the this class also requires the definition of the adjoint process for a given transformation.

A standard gradient implementation of the form

\[ \nabla F(x) = H^T (Hx - y) \]

is included, where \( F(x) \) is a convex function of the form \( F(x) = \|Hx - y\|_2^2 \). The parent class structure ensures that the gradient required for a given inverse problem can be easily implemented. As with the proximity operators, the gradient’s contribution to the total cost is built into the class structure.

A cost function class is also provided that automatically sums up the contributions from the proximity and gradient operators. This class has a built-in framework to test for convergence up to a given tolerance.

Finally, a reweighting class is provided to counteract the bias introduced into a given solution owing to the use of soft-thresholding in sparse regularisation. At present, the method of \( \text{Candes et al.} \) (2008) is included.

The combination of these tools enables the user to very quickly prototype robust codes for tackling a variety of inverse imaging problems.

2.3. Plug-ins

PySAP also provides application specific plug-ins. In this module algorithms and operators from ModOpt can be combined with with Sparse2D transforms to develop tools for a given application. The objective being to produce user-friendly functions, designed to solve well defined problems, that can be applied directly to data.

At present, this module contains plug-ins that demonstrate the applicability of PySAP to astrophysical and MRI data. For example, the MRI plug-in adds the ability to deal with non-Cartesian data using non-uniform or non-equispaced FFT tools, while the astrophysics plug-in provides easy-to-use tools for denoising or deconvolving survey images.

The plug-in framework has been designed to promote collaboration by providing a template for creating new plug-ins for virtually any imaging domain. Future plug-ins have already been planned for microscopy and electron tomography.

3. Practical Applications

3.1. Astrophysical Images

One straightforward application of PySAP on astrophysical data is to the problem of galaxy image deconvolution. Astrophysical images obtained with optical telescopes are subject to a blurring caused by internal factors, such as imperfections in the optical system, and external factors, such as the atmosphere for ground based instruments. The sum of these aberrations is commonly referred to as the Point Spread Function (PSF).

Removing the effects of the PSF from noisy observations amounts to solving a non-trivial inverse problem that requires the use of regularisation owing to the ill-conditioned nature of the degradation matrix, which corresponds to convolution with the PSF in this case. This problem can be solved using sparse regularisation following the same prescription described in \( \text{Farrens et al.} \) (2017) using PySAP. A deconvolution example is provided in PySAP that demonstrates this process in a few lines of code. This example performs deconvolution using the Condat-Vu algorithm. An isotropic undecimated wavelet transform from Sparse2D is used for the linear operator, and a positivity constraint and soft-thresholding of the sparse coefficients are used as the proximity operators. The results of this example are shown in figure 2.

Another application is simply removing noise from observations. This is a particularly challenging problem when the object in question contains important high-frequency spatial features that need to be preserved. Figure 3 presents the results of denoising an image of the galaxy NGC2997 using PySAP. For this example the same isotropic undecimated wavelet transform from Sparse2D is used to decompose the image, which is in turn thresholded by weights learned from the image itself.

Note that the data used for the examples presented in this paper are provided in PySAP. Therefore, all of the example outputs can be reproduced exactly by users.

3.2. MRI

MRI is probably one of the most successful applications of compressed sensing. The ability to reconstruct high-fidelity MR images from massively undersampled data in a short amount of time is of paramount importance. This is achievable as signals obtained in the Fourier domain are no longer acquired on a rectangular grid, but rather on a polar grid or more recently using custom trajectories \( \text{Boyer et al.} \) (2010). The images can then be reconstructed using state of the art optimisation algorithms. The idea is to take advantage of these time-saving tactics, not only to increase spatial resolution in anatomical imaging, but also to improve spatio-temporal resolution in functional MRI.

Take, for example, the problem of compressed sensing parallel imaging reconstruction. For this example let \( L \) be the number of coils used to acquire the NMR signal, \( N \) be the number of pixels of the complex-valued image \( x \) to be reconstructed and \( M \) the number of samples collected per channel during acquisition. We denote by \( y_\ell \in \mathbb{C}^M \) the complex-valued data recorded by the \( \ell \)-th channel, \( S_\ell \in \mathbb{C}^{N \times N} \) the corresponding diagonal sensitivity matrix. Let \( F \) be the NFFT and \( \Omega \in \{1, \ldots, N\} \) the sampling pattern in the \( k \)-space, with \( |\Omega| = M \ll N \). The CS-PI acquisition model thus reads:

\[ y_\ell = F_\Omega S_\ell x + n_\ell, \quad \forall \ell = 1 : L, \]

where \( n_\ell \) is additive zero-mean Gaussian noise of variance \( \sigma_n^2 \), which can be characterized by a separate scan (without RF pulse) considering the same bandwidth as the prospective CS acquisition. This problem can easily be solved using sparse regularisation following the same prescription.
Figure 2: Example of galaxy image deconvolution using PySAP. Top left: True galaxy image, Top right: observed galaxy image, Bottom left: deconvolved galaxy image, Bottom right: deconvolution residual.
Figure 3: Example of galaxy image denoising using PySAP. Top left: True galaxy image, Top right: observed galaxy image, Bottom left: denoised galaxy image, Bottom right: denoising residual.
Figure 4: Cartesian MRI decimated wavelet-based reconstruction. *Top left:* Cartesian reference, *Top right:* K-space mask, *Bottom left:* Zero-filled reconstruction ($SSIM = 0.82$), *Bottom right:* Decimated wavelet based reconstruction ($SSIM = 0.91$).
Figure 5: Non-cartesian MRI undecimated wavelet-based reconstruction. *Top left:* Cartesian reference, *Top right:* K-space mask, *Bottom left:* Zero-filled reconstruction (SSIM = 0.67), *Bottom right:* Undecimated wavelet based reconstruction (SSIM = 0.92).
described in [Chaari et al. (2011); Guerquin-Kern et al. (2011)] using PySAP in just a few lines of code.

Several example applications to MR data are provided in the PySAP package. A first example is shown in figure 1. It shows the reconstruction of a single-coil MR image undersampled (with an acceleration factor of 2.28) in the k-space using the mask shown using the FISTA (Beck and Teboulle (2009)) algorithm with sparsity promoted with the decimated symmlet 8 transform. A second example is depicted in figure 5 where the undersampling mask is this time non-cartesian (radial, with an acceleration factor of 8) and the sparsity is promoted using an anisotropic undecimated wavelet transform form Sparse2D.

Some codes spinets can be found in the PySAP documentation gallery: https://cea-cosmic.github.io/pysap/auto_gallery/gallery

3.3. Gadgetron

Speeding up the acquisition time is one of the most challenging issues in high magnetic field MRI. This goal is achieved by using fancy sampling schemes in the measurement space (Fourier space). One of these strategies, the so-called “SPARKLING” trajectories, has been implemented as a Gadgetron reconstruction following the same prescription described in [Lazarus et al. (2017)] using PySAP. Gadgetron is a streaming data processing framework for medical image reconstruction that can replace the scanner’s built-in reconstruction algorithm (Hansen and Sørensen (2013)). Such reconstruction strategies can be shared across research institutes even if the scanner manufacturer or software versions differ.

Gadgetron natively supports Python packages and therefore PySAP can easily be installed on any MRI scanner where the Gadgetron framework is in place.

4. Conclusions

In this paper we have presented the image processing package PySAP, its principal features and example applications to MR and astrophysical images. In particular, examples demonstrate how PySAP can be applied to image processing problems such as denoising, deconvolution and compressed sensing employing state-of-the-art reconstruction algorithms and wavelet transforms. In each case the plugin framework provides easy-to-use tools for solving these problems for specific applications.

The flexibility and modularity of this package permit a wide range of possible future developments. In particular, we aim to continue to add new and cutting-edge optimisation algorithms, reweighting methods, etc. We additionally aim to add further features for handling 4D data and optimising the computation time by exploring GPU implementations. Another important aspect to which we plan to dedicate effort is to integrating machine learning techniques into the existing architecture.

One of the most exciting uses of PySAP comes from the Gadgetron implementation. The universality of this system and the growing community mean that PySAP can readily be used at MRI scanners around the world, potentially leading to some fascinating developments in the biomedical imaging domain.

Finally, we intend to seek out new applications of this software in a variety of different fields. In fact, work has already begun on developing a PySAP plugin for electron tomography.

Reproducible research. In the spirit of reproducible research PySAP is made publicly available and fully open source. Documentation and installation instructions are available on the PySAP website (https://cea-cosmic.github.io/pysap/). The authors kindly request that any academic publications that make use of PySAP cite this paper.

Acknowledgments

The authors wish to acknowledge the COSMIC project funded by the CEA DRF-Impulsion call in 2016, the UniEarthS Labex program of Sorbonne Paris Cite (ANR-10-LABX-0023 and ANR-11-IDEX-0005-02), the European Community through the grant DEDALE (contract no. 665044) within the II2020 Framework Program of the European Commission, the Euclid Collaboration, the European Space Agency and the support of the Centre National d’Etudes Spatiales. The authors additionally acknowledge Hannae Carrie, Sylvain Lannuzel and Fabrice Poupin for contributing to the development of this package.

References

Beck, A., Teboulle, M., 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Img. Sci. 2, 183–202.
Bobin, J., Sureau, F., Starck, J.L., Rassat, A., Paykari, P., 2014. Joint Planck and WMAP CMB map reconstruction. Astronomy & Astrophysics 563, A105. doi:10.1051/0004-6361/201323272
Boyer, C., Chauffert, N., Ciuciu, P., Kahn, J., Weiss, P., 2016. On the Generation of Sampling Schemes for Magnetic Resonance Imaging. SIAM Journal on Imaging Sciences 9, 2039–2072.
Candès, E.J., Wakin, M.B., Boyd, S.P., 2008. Enhancing sparsity by reweighted $l_1$ minimization. Journal of Fourier Analysis and Applications 14, 877–905. URL: http://dx.doi.org/10.1007/s00041-008-9045-x
Chaari, L., Pesquet, J.C., Benazza-Benyahia, A., Ciuciu, P., 2011. A wavelet-based regularized reconstruction algorithm for SENSE parallelMRI with applications to neuroimaging. Medical Image Analysis 15, 185–201.
Condat, L., 2013. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. Journal of Optimization Theory and Applications 158, 460–479. URL: https://hal.archives-ouvertes.fr/hal-00609728
doi:10.1007/s10957-012-0245-x
Farrens, S., Ngol`e Mboula, F.M., Starck, J.L., 2017. Space variant deconvolution of galaxy survey images. Astronomy & Astrophysics 601, A66. doi:10.1051/0004-6361/201629709 arXiv:1703.02305
Guerquin-Kern, M., Haberlin, M., Pruessmann, K., Unser, M., 2011. A fast wavelet-based reconstruction method for magnetic resonance imaging. IEEE Transactions on Medical Imaging 30, 1649–1660.

Hansen, M.S., Sørensen, T.S., 2013. Gadgetron: An open source framework for medical image reconstruction. Magn Reson Med 69, 1768–1776. URL: http://dx.doi.org/10.1002/mrm.24389
doi:10.1002/mrm.24389

Kim, D., Fessler, J.A., 2017. Adaptive Restart of the Optimized Gradient Method for Convex Optimization. arXiv e-prints, arXiv:1703.04641[arXiv:1703.04641].

Lanusse, F., Starck, J.L., Leonard, A., Pires, S., 2016. High resolution weak lensing mass mapping combining shear and flexion. Astronomy & Astrophysics 591, A2. doi:10.1051/0004-6361/201628278[arXiv:1603.01599].

Lazarus, C., Weiss, P., Chauffert, N., Vignaud, A., Ciuciu, P., 2017. SPARKLING: nouveaux schémas d’échantillonnage compressif prospectif pour l’IRM haute résolution, in: GRETSI, Juan les Pins, France. URL: https://hal.inria.fr/hal-01577207.

Leonard, A., Lanusse, F., Starck, J.L., 2014. OLIIMPSE: accurate 3D weak lensing reconstructions using sparsity. Monthly Notices of the Royal Astronomical Society 440, 1281–1294. doi:10.1093/mnras/stu273[arXiv:1408.1353].

Ngol’ Mboula, F.M., Starck, J.L., Ronayette, S., Okumura, K., Amiaux, J., 2015. Super-resolution method using sparse regularization for point-spread function recovery. Astronomy & Astrophysics 575, A86. doi:10.1051/0004-6361/201424167[arXiv:1410.7679].

Raguet, H., Fadili, J., Peyré, G., 2011. Generalized Forward-Backward Splitting. ArXiv e-prints, arXiv:1108.4404[arXiv:1108.4404].

Ramzi, Z., Ciuciu, P., Starck, J.L., 2019. Benchmarking proximal methods acceleration enhancements for CS-acquired MR image analysis reconstruction, in: SPARS 2019 - Signal Processing with Adaptive Sparse Structured Representations Workshop, Toulouse, France. URL: https://hal.inria.fr/hal-02298569.

Starck, J.L., Fadili, J., Murtagh, F., 2007. The undecimated wavelet decomposition and its reconstruction. IEEE Transactions on Image Processing 16, 297–309. doi:10.1109/TIP.2006.887733[arXiv:1603.01599].

Starck, J.L., Murtagh, F., Fadili, J., 2015. Sparse Image and Signal Processing: Wavelets and Related Geometric Multiscale Analysis. Cambridge University Press.

Vũ, B.C., 2013. A splitting algorithm for dual monotone inclusions involving cocoercive operators. Advances in Computational Mathematics 38, 667–681. URL: https://doi.org/10.1007/s10444-011-9254-8[arXiv:1108.4404].