imager: an R package for image processing based on Clmg

Simon Barthelmé¹ and David Tschumperlé²

1 CNRS, Gipsa-lab, Grenoble INP, France 2 CNRS, Greyc, ENSICAen, France

Summary

R is a language designed for the processing of (numerical) data, and images are just numerical data. The original motivation for developing imager was that existing image processing facilities in R were at the time rather limited, and that R had the potential to be a great environment for processing image data. Many scientists now use R as their primary computing environment, and some amount of image processing is a requirement in fields like biology (when microscopy is used), or astronomy.

Because R already provides a lot of generic data processing tools, imager was designed to be "idiomatic", so that native R tools could be used easily on image data. On the other hand, because R is also slow, and ill-adapted to many image processing algorithms that involve looping over each pixel in an image, imager was designed to provide easy access to a C++ image processing library as a fall-back option (using Rcpp (Edelbuettel et al., 2011) as glue).

There are many C++ libraries for image processing, and among them OpenCV is arguably dominant. However, its API is quite complex and targets computer vision rather than image processing per se. A popular alternative is the Clmg library developed by the second author (Tschumperlé, 2012). Clmg provides an elegant API, and is easy to learn. In addition, Clmg is fast, parallel (using OpenMP), and supports image data in up to four dimensions, meaning that it can process hyperspectral or video data. It is already used in many OSS applications partial list here. Notably, Clmg is the back-end for G'MIC, a popular command-line tool and Gimp/Krita plugin for image processing.

imager is based on Clmg but is not a simple wrapper. It is organised around two main types of objects: raster images (with cimg type), and pixel sets (with pixset type). The first are meant to represent images, while the second represent regions of interest, such as foreground pixels, cells, etc. Both come with standard R generics, like print, plot, or as.data.frame. Internally, they are nothing more than 4D arrays, of numeric or logical type, that Clmg can directly work with.

imager provides all classical image processing operations (filtering, morphology, etc.), along with more advanced functionality. It integrates well with packages from the tidyverse such as purrr. It is extensively documented, with two vignettes (getting started, and pixsets), and many worked-out examples on the website.

Current status, and plans for the future

The package is now relatively mature (over 3 years old). As a testament to its usefulness, we note that imager is used by several other OSS projects: as of Aug. 2018, 18 dependencies are listed on CRAN. The package contains at this stage far more functions than its author

https://doi.org/10.21105/joss.01012
can remember, so the current focus for development is on enhancing documentation and “discoverability” of imager’s content, and consistency of the API. Another interesting area is support for missing values: this is generally a strong point of R, and a weak point of image processing libraries. *imager* provides limited support for missing values, but future versions should increase the level of support.

**Acknowledgements**

Simon Barthelmé is supported by a grant from the French National Agency for Research (ANR GenGP, ANR-16-CE23-0008).

**References**
