PORTFOLIO

Benjamin Gallois

INTRODUCING

ABOUT ME

Hi, I’m Benjamin Gallois, Ph.D. in physics and computer vision developer. For four years, I have had the opportunity to work with a vast spectrum of technologies, including data analysis, computer vision, and software design.

I’m working remotely as a freelancer, providing my computer vision and software conception expertise using C++, Python, and OpenCV from prototyping to deployment.

BENJAMIN GALLOIS
Computer Vision Expert

EDUCATION

Trained as a scientist, I had the chance to work on multidisciplinary projects that taught me much about computer vision, software design, and rigorous work methods.

2017 - 2021
PH.D. IN PHYSICS

SORBONNE UNIVERSITY, PARIS

2015 - 2017
M.SC. IN PHYSICS

PARIS DIDEROT UNIVERSITY, PARIS

2012 - 2015
B.SC. IN PHYSICS

PARIS DIDEROT UNIVERSITY, PARIS

1000+

DOWNLOAD TOTAL

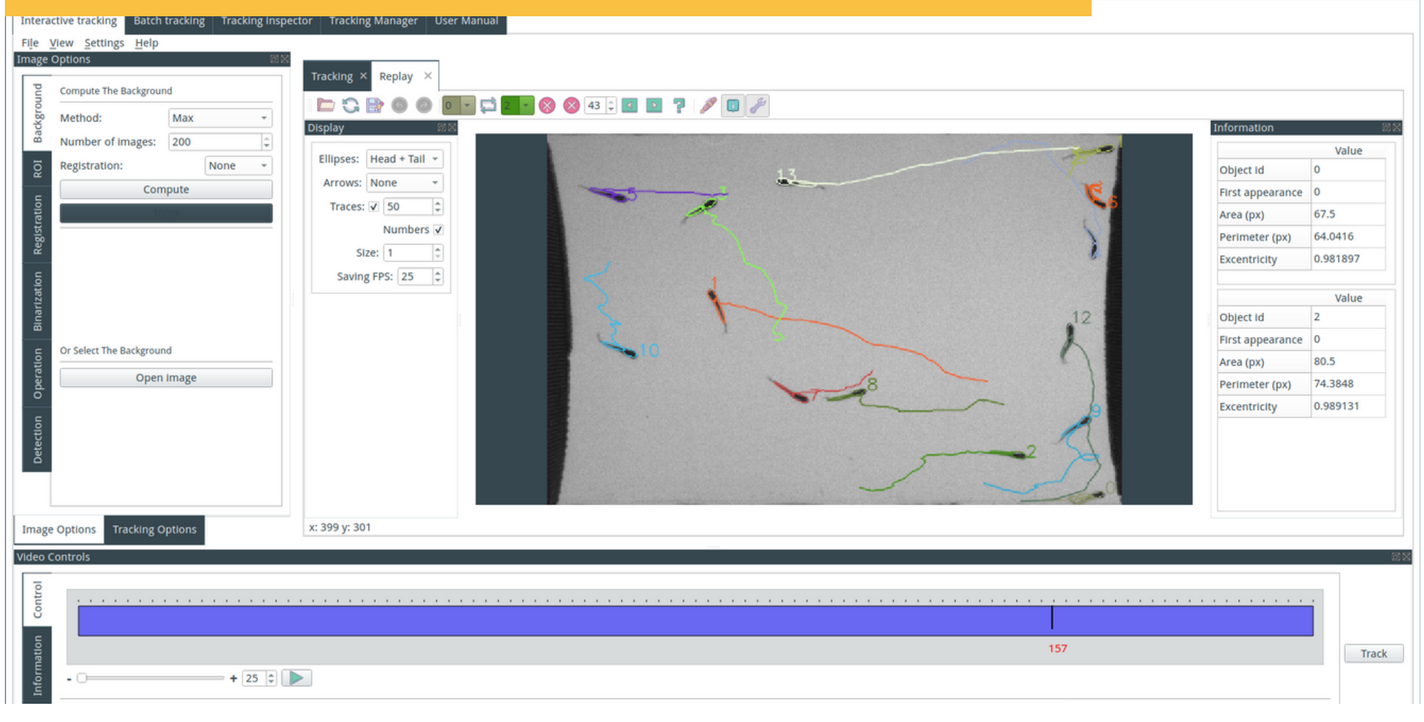
55

STARS ON GITHUB

FASTTRACK

Tracking objects in video recording is a common but difficult task for many scientists. Free and open-source tracking software are notoriously tricky to install and requires extensive knowledge. Their closed-source counterparts are expensive and restricted.
To solve this problem, I created FastTrack, a cross-platform tracking software with an intuitive but complete user interface that is not a chore to install.

I designed FastTrack independently, from low-level algorithms to complete CI/CD producing end-users installers. It is built in C++ using Qt for the user interface, OpenCV for image processing, SQLite to store the tracking data, and Google Test for unit testing. A complete CI/CD provides the end-user with an installer for Windows, a dmg for MacOs, and an AppImage for Linux, allowing easy installation and integration on any system. The developer manual is generated using Doxygen, and the user manual using Docusaurus.



5/5

WHAT CLIENT SAY

Outstanding experience! Benjamin has delivered what many other developers couldn't, very competent developer, will be working again with him very soon!

SCANNED IMAGES WARPING

Scanning image from prints comes with several specific problems. Correcting them is essential to get the image close as possible to the original material.

A common problem when scanning files is small deformations that can occur if the scanned material is not flat during the scanning. These deformations can be of two types, affine (rotation) or perspective (shearing).

Correcting these deformations can be done by finding four points of reference in the image and mapping them to their original coordinates. The tricky part is finding these four points with pixel accuracy. In our case, we detect four corners based on color landmarks consistent between scanned documents from a binarized version of the image using a specific color range selection.

WHAT USERS SAY

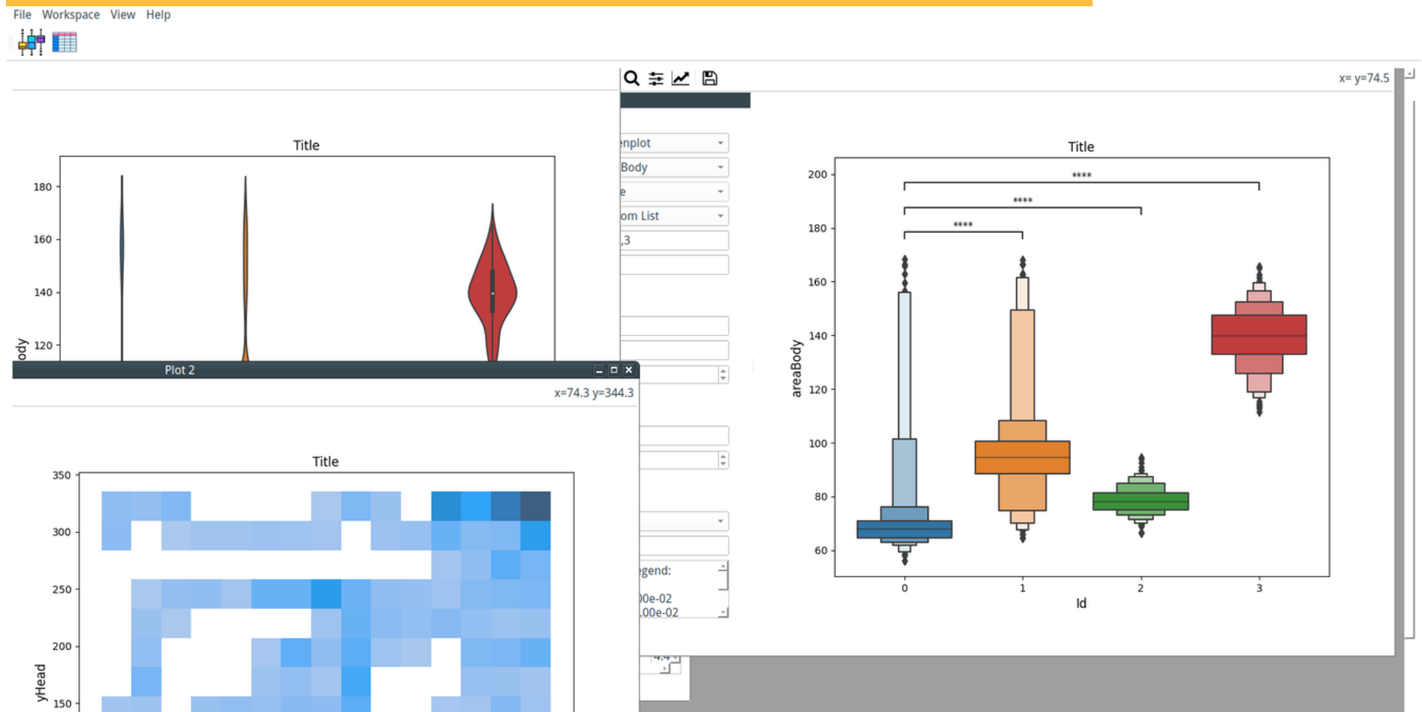
Will simplify my analysis pipeline. Possibility for low-level API call is a must!

FASTANALYZER

FastAnalyzer is a software for analyzing tracking trajectories extracted using FastTrack. A complete and intuitive user interface allows complex statistical analysis without technical knowledge in programming.

Tracking objects from video recording is generally only the first step of any scientific analysis. The second task, trajectory analysis, can be pretty daunting. Numerous tools are available, from scripting languages to complete user interface environments that require various learning curves.

FastTrack allows an easy and a fast tracking from any video recording. To make the trajectory analysis as quick and easy as possible, we started the development of FastAnalyzer. FastAnalyzer is built on top of the Python scientific ecosystem (SciPy, NumPy, Matplotlib, and Seaborn) and the existing FastAnalysis library.



WHAT CLIENT SAY

Fast and accurate. Delivered on time. Excellent developer to work with.

DETECTION FROM RTSP STREAM

An automatic front doorbell camera with an alert when someone or something is at the door can be a helpful tool. A low-tech option with a Raspberry camera and low-footprint desktop software running in the background can be an ideal solution.

The task was to provide an automatic detector of humans on low-quality video feeds that can run with low computational power.

We provided an optimized algorithm that can run on a single-board computer taking advantage of the Yolov5n6 detector and OpenCV for decoding the RTSP video feed. The detected object is then saved, and we use Plyer to display notifications on the user's desktop.

5/5

WHAT CLIENT SAY

It was great to work with Benjamin. He’s very knowledgeable about computer vision, with a straightforward and optimized implementation.

REFLECTION REMOVAL

Taking a picture from a glossy medium can be quite a challenge. It is almost impossible to avoid any reflection.

The task was to remove all the light reflections in the picture using several photos from the same objects with a slight change in perspective.

The complete process is detailed in the blog post https://gallois.cc/blog/glare/. To summarize, a rigid registration is performed to align all the images. Then a non-rigid registration is used to correct local deformations. Finally, we select the pixels with the minimum intensity to construct the final image. This process can be applied to colored images using the HSV space.

WHAT USERS SAY

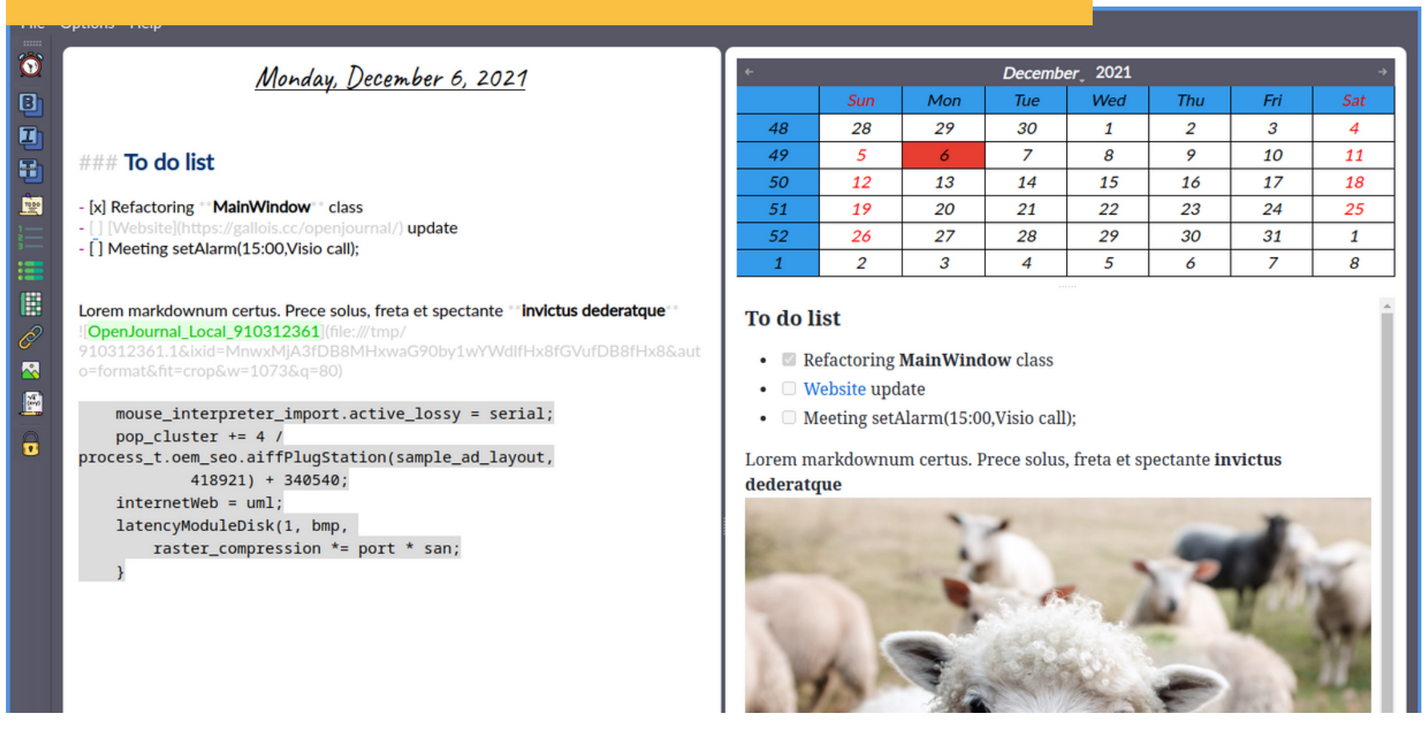
I don't have the laziness to keep my journal every night anymore

OPENJOURNAL

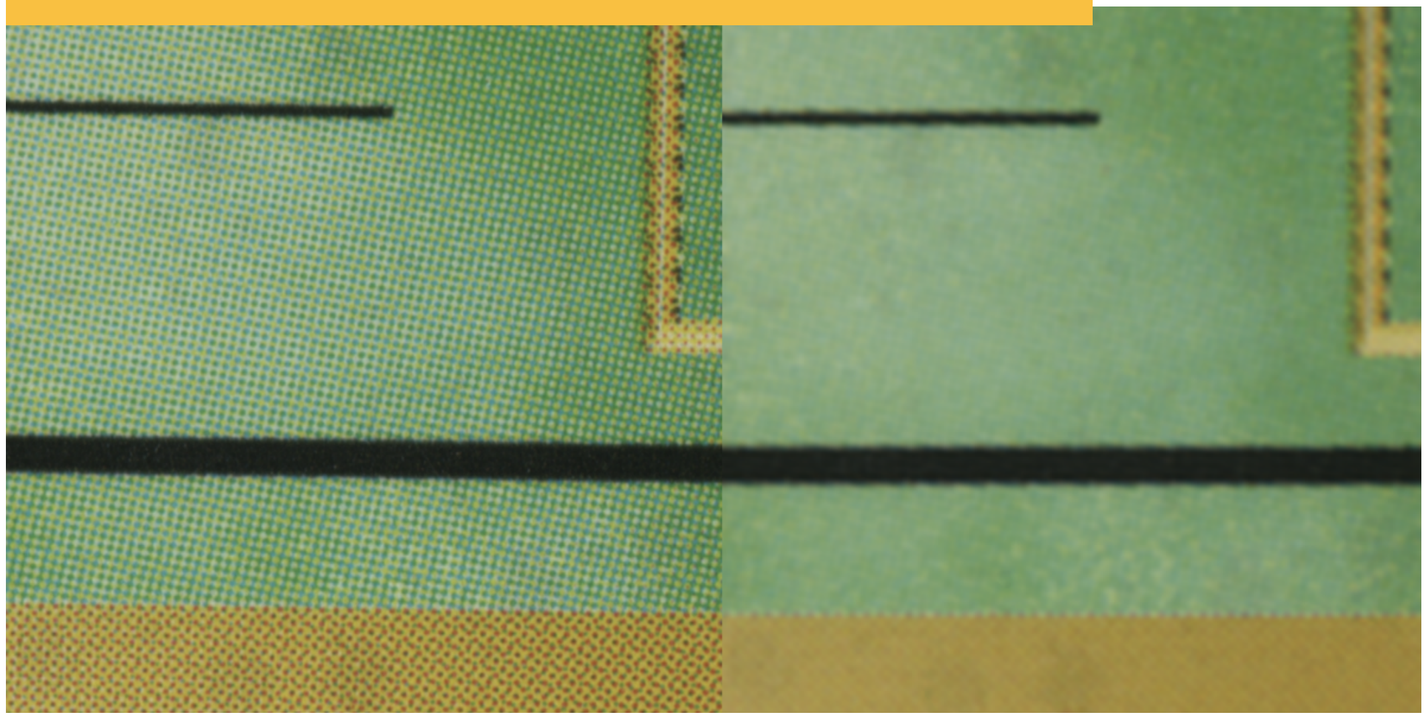
OpenJournal is a simple journal, note-taking, and assistant application supporting Markdown syntax and rendering thanks to qmarkdowntextedit. You can write your thoughts and to-do list and never forget a meeting by setting visual and audible alerts.

Scheduling a day, taking notes, write small calculations and equations all in one place can be very helpful. I created OpenJournal, a simple cross-platform journal, note-taking, and assistant application supporting Markdown syntax and rendering with referenced images stored locally. It supports LaTeX equations, unit conversion, symbolic and numeric calculation, and alarms never to forget a meeting.

OpenJournal is built in C++ with Qt for the user interface and an SQLite database to store the data. A cloud storage option built with Flask is provided to keep journals remotely. A complete CI/CD provides the end-user with an installer for Windows, a dmg for MacOs, and an AppImage for Linux.



SCANNED IMAGES ENHANCEMENT



Scanning image from prints comes with several specific problems. Correcting them is essential to get the image close as possible to the original material.

The first and most visible problem is the addition of a moiré pattern to the image. It is caused during the scanning process by the interference between two sets of fine grids. The second is to have a coherent result between scans (contrast, sharpness, etc.) Finally, the compatibility issue with OpenCV and high-resolution 16-bit images is to consider.

Removing the moiré pattern, also called descreening, can be performed using Fourier space and removing frequency peaks. In addition, it is possible to use a low-pass filter for a smoother result. Then we develop an algorithm to automatically detect the blur level in an image and correct it to a given level using an unsharpening mask. We also use Scikit-image to compensate for the inability of OpenCV to work with more than 8-bit images for specific algorithms.