E-Foto and MicMac: synergetic benefits of integrating open-source digital photogrammetry software

João Araujo Ribeiro
jaraujo@uerj.br

Rio de Janeiro State University

Open Source Geospatial Research and Education Symposium
Espoo, Finland, June 2014
Why are we doing this?
First reason:

Because we can!

But, you could ask me: Why do you can? We can not because we are smart people, not because we have superpowers. The main reason we can is because both teams work with Open Source. Not only in Linux, of course. Both programs work on windows too.

http://tux.crystalxp.net/en.id.18041-brunocb-tux-obama-linux.html
Second reason

Because we need!

- ![Clock] We don’t have time. We have a lot of other tasks. E-foto is not the only thing we do.

- ![Dollar] We don’t have money to hire programmers.

- ![Person] We don’t have people enough to do our goal. Well, this is a consequence from “we don’t have money”
Our goal

A complete educational, digital photogrammetric workstation under the General Public License (GNU/GPL).
What is E-foto?

We have started E-foto ten years ago from a white page, and we have developed every single line of code, based on classical photogrammetric algorithms. And, from the beginning, everything was open source.
E-foto is a pedagogical tool for the self-teaching/learning of photogrammetry. So, each step is designed to make easier to understand the photogrammetric concepts.
The E-foto work flow

Project Manager

Interior Orientation

Phototriangulation

Exterior Orientation by
Spatial Resection

Orthorectification

Stereo Plotter

Digital Elevation Model Extraction

The first five steps are in a good stage of development, the two last ones we need to polish the code to get better results.
What is E-foto?

Phototriangulation Module
What is MicMac?

- A powerful photogrammetric software;
- each task is accomplished by a different tool;
- it has automatic tie points generation;
- MicMac uses regular expressions to select images.
What is MicMac?

Photo and Digital Elevation Model generated by MicMac

On the left we have the airborne image and on the right the model generated by MicMac.
Typical MicMac session

But MicMac is not a simple tool. As you can see here, it works with a command line approach, with a lot of regular expressions. This can be a little intimidating to new users.

Tapioca All "IMGP416[0-5].JPG" 1000
Tapas RadialExtended "IMGP416[0-5].JPG" Out=Calib
Tapioca Line "IMGP41((6[7-9])|([7-8][0-9])).JPG" 1000 4
Tapas AutoCal "IMGP41((6[7-9])|([7-8][0-9])).JPG" InCal=Calib
AperiCloud "/IMGP41((6[7-9])|([7-8][0-9])).JPG" Mur
SBGlobBascule "IMGP41((6[7-9])|([7-8][0-9])).JPG" Mur MesureBasc.xml LocBasc PostPlan=_MasqPlan DistFS=0.6
Tarama "IMGP41((6[7-9])|([7-8][0-9])).JPG" LocBasc
Malt Ortho "IMGP41((6[7-9])|([7-8][0-9])).JPG" LocBasc
Tawny Ortho-MEC-Malt/
Tawny Ortho-MEC-Malt/ DEq=0
Tawny Ortho-MEC-Malt/ DegRapXY=[2,0]
MicMac and E-foto have a lot of common points. It’s logical to join efforts to have a better program.
Both projects are developed with C++, XML and, the most important, both are open source, so we can read all code and learn from each other.
Both programs run on Linux and Windows. Also, both projects run better on Linux than on Windows. All development of E-foto is made first on Linux, after we will compile the windows version. So, it is logical to have a program more stable on Linux.
Differences

We have also some differences. The most important one is connected with its interface. MicMac uses command line without graphical user interface. E-foto, in order to be easier to learn, uses Qt in its graphical user interface. Almost all problems we have with E-foto stability are related with its interface.
What borrow from MicMac

What can we borrow from MicMac in order to make E-foto a better photogrammetric software?

E-foto only works with airborne imagery

MicMac works with generic images
What borrow from MicMac

E-foto needs a calibrated photogrammetric camera. We do not have autocalibration.

MicMac can use consumer cameras. Even a simple cellphone is good enough to get images, but with less precise results. Thanks to its autocalibration.
What borrow from MicMac

E-foto lacks an automatic tie point generation tool. Here we have two images taken with a little different point of view. With E-foto we are not capable to find the corresponding points between the two images.
What borrow from MicMac

MicMac has automatic tie point generation. Each red points has a corresponding point on the other image. That is made as the first step of processing.
Is it easy? What kind of problems we have with this task?
Kind of problems

To accomplish our task we have three kinds of problems:
Technical, Human and Conceptual
**Technical**

MicMac works in image space

E-foto works in object space

MicMac do all calculations in image space. Only when needed it will be done in object space. This make a very good 3D model with all calculations in pixels, not in meters. Efoto has a different approach. It calculates in meters since the beginning. That is not a huge problem. We can do conversions between the two systems very quickly, but we have to know that we can not use all the results of MicMac directly.
Difficulties

Technical

Lack of documentation

In the open source world we work almost all the time with small teams, sometimes only programmers. We like writing code, but we do not like to write documents. So, sometimes, the code is always a little different from the description we have for it. Mainly, we have students writing code to E-foto and they are good programmers but not always good on documentation.
Human

E-Foto has a small team of programmers and almost 44,000 lines of code. This is a consequence from “we do not have money”. We have a lot of code to write. A photogrammetric workstation is not a simple program. We have thousands of lines of code, (44,000 lines of code to keep) and we have an unstable team (we work with students).

Difficulties

Conceptual

The conceptual problems are harder to deal. We have to be very careful with them. For example: E-foto has a pedagogical approach. MicMac doesn’t have. We must use the good things from MicMac, as the powerful algorithms, without lose our approach and without compromise backward compatibility.
What are the benefits of integration?
What kind of lessons we have learned from this integration?
Is this good for us?
What kind of benefits we have?
To join efforts, keeping different approaches and different teams

Even if we can modify the MicMac code, we decided not to touch its code, so we can benefit from all the ongoing developments from MicMac.
To share experiences
To improve distinct project
This is a work in project

We did not finish it. Before to bring news features from MicMac, we need to correct some bugs from E-foto we have discovered. So, there is a lot to do.
Thanks

This work was conducted during my sabbatical year, thanks to the support of Capes and of Rio de Janeiro State University (UERJ) from Brazil. I also wish to thank the École Nationale de Sciences Géographiques (ENSG, France) for receiving me during this sabbatical period.
Contact and Questions

http://www.efoto.eng.uerj.br

jaraujo@uerj.br