

Accepted Manuscript

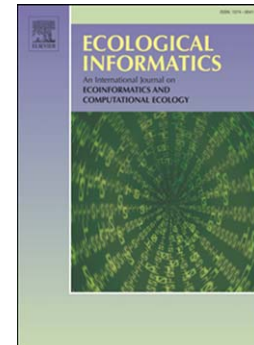
Advances in camera-trap data management tools: Towards collaborative development and integration with GIS

B. Zaragozaí, A. Belda, P. Giménez, J.T. Navarro, A. Bonet

PII: S1574-9541(15)00117-X
DOI: doi: [10.1016/j.ecoinf.2015.08.001](https://doi.org/10.1016/j.ecoinf.2015.08.001)
Reference: ECOINF 594

To appear in: *Ecological Informatics*

Received date: 4 February 2015
Revised date: 30 July 2015
Accepted date: 3 August 2015



Please cite this article as: Zaragozaí, B., Belda, A., Giménez, P., Navarro, J.T., Bonet, A., Advances in camera-trap data management tools: Towards collaborative development and integration with GIS, *Ecological Informatics* (2015), doi: [10.1016/j.ecoinf.2015.08.001](https://doi.org/10.1016/j.ecoinf.2015.08.001)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



Advances in camera-trap data management tools: towards collaborative development and integration with GIS

B. Zaragoza^{a,b}, A. Belda^c, P. Giménez^a, J.T. Navarro^a, A. Bonet^b

^a*Instituto Interuniversitario de Geografía, Universidad de Alicante.*

^b*Instituto Multidisciplinar para el Estudio del Medio “Ramón Margalef”, Universidad de Alicante*

^c*Dpt. Ciencias de la Tierra y del Medio Ambiente. Universidad de Alicante.*

Abstract

Camera traps have become a widely used technique for conducting biological inventories, generating a large number of database records of great interest. The main aim of this paper is to describe a new Free and Open Source Software (FOSS), developed to facilitate the management of camera trapped data originated in a protected Mediterranean area (SE Spain). In the last decade, some other useful alternatives have been proposed, but ours focuses especially on a collaborative undertaking and on the importance of spatial information underpinning common camera trap studies.

This FOSS application, namely “Camera Trap Manager” (CTM), has been designed to expedite the processing of the pictures on the .NET platform. CTM has a very intuitive user interface, automatic extraction of some image metadata (date, time, moon phase, location, temperature, atmospheric pressure, among others), analytical (Geographical Information Systems, statistics, charts, among others) and reporting capabilities (ESRI Shapefiles, Microsoft Excel Spreadsheets, PDF reports, among others). Using this application we have achieved a very simple management, fast analysis, and a significant reduction of costs. While we were able to classify an average of 55 pictures per hour manually, CTM has made it possible to process over 1,000 photographs per hour, consequently retrieving a greater amount of data.

© 2015 Published by Elsevier Ltd.

Keywords: camera traps, data management and analyses, free and open source software, geographical information systems, wildlife monitoring

1. Introduction

Camera trapping has become a common non-invasive method for wildlife monitoring in almost any soil or weather condition in recent years. It has proved successful in not only accurately identifying a variety of species but also, given an appropriate setup [1, 2], correctly estimating features such as age, sex, population structure, and density over large areas.

Most methods for finding out the number of wildlife species are fundamentally concerned with three major ecological characteristics of the species: (1) the scarcity of individuals, (2) the wide range of the species, and (3) the secrecy

of their habits [3]. Traditional surveys are efficient and inexpensive, but they closely depend on adequate environmental conditions and trained personnel [1]. Even though camera trapping might appear to be more expensive at first, it is not as dependent on the environment neither does it require intensive field work nor is it greatly affected by the inexperience of the researchers [1]. Furthermore, using these methods makes the study of rare, endemic, threatened or endangered species feasible, particularly if authorities have restricted or even prohibited their monitoring [4]. The use of camera trapping techniques has positively interesting advantages but caution should be taken when managing the data. The following problems may appear:

1. Lack of systematization. Studies using camera trapping techniques tend to usually be poorly standardized and systematized [5]. A higher systematization of camera trap databases would facilitate the integration of these pictures into different scientific collections, significantly contributing to its use in global biodiversity studies [4].
2. Large volumes of data. Depending on the requirements, an average project could produce tens of thousands to hundreds of thousands of photos over a one-year campaign[6]. In order to analyse the geographical distribution of a target species, all these pictures have to be processed and classified.
3. Lack of regular quality controls. Obviously, these processes increase the final costs [6]. Inconsistencies and errors in data capture could be prevented by systematization and use of specific tools.

Every biodiversity project will have devised a more or less practical solution for these problems, but nowadays there is a general consensus in that the higher use of camera traps has not been coupled with powerful software solutions to manage and analyse the huge data volume required [7, 8].

The main objective of this work is to provide a Free and Open Source Software (FOSS) for providing appropriate management of camera trapped pictures. For practical purposes, finding a FOSS solution, modifiable to satisfy any particular project needs is of utmost importance. By becoming involved in the search for a particular solution, the base code keeps evolving through collaboration across the developers community. However, two important requirements need be added: 1) the tool must ensure Geographical Information System (GIS) software compatibility to handle cartography for spatial analysis, and 2) the tool should be extensible and usable for a wide users community. These project needs have been met in three main steps. First, we have evaluated the available options and then developed a desktop prototype application for the .NET platform, which accounts for our research aims. Finally, this application has been tested in a real case: tracking mammals in the Natural Park of the Sierra de Mariola (SE Spain) since 2009.

2. Camera trap software for extracting and managing data

The problems mentioned are rather common in camera trap projects. It is for this reason that several useful alternatives for specific or generic software have been proposed to handle photos taken by camera traps [7, 9, 8, 6, 10]. We clearly recommend weighed considerations to be made before choosing or developing a new one.

In his initial proposal, [7] suggests a management system executed in four stages. First, organize photos into folders in relation to the area surveyed by each camera and rename them automatically so that each photo is identified by date and time of capture. Second, review all the pictures taken at each site and identify the species and number of animals captured in each photo. The user must manually move the picture to a manually created subfolder which is kept within the species folder –one subfolder for each number of individuals of that species. For example, a photo of two wild boars should be moved to a folder named “C:\Location \Boar \2”. And thirdly, produce a text file that lists all files and directories of the project. Then, analyse and summarize the directory structure in a spreadsheet, for later analysis.

[9] explains that the proposal made by [7] involves several problems that would be easily fixed if the classification were done using agile tools for general purpose image classification (Photospread or Picasa). However, neither of the two proposed alternatives has been designed for camera trap studies, which will probably cause errors in handling and certainly will lack of both integration with GIS tools and specific reporting capabilities.

Camera Base (Tobler, 2013; <http://www.atrium-biodiversity.org/tools/camerabase/>) is a well known approach which has powerful functionalities for data management and analysis. It runs on Microsoft Access, with custom forms for different types of projects, and has several types of predefined reports. In spite of Camera Base code being open source, it still requires a license and forces the user to work on proprietary file-based non spatial databases (*.mdb) with limitations in storage and management capabilities.

Possibly the most complete and coherent approach of those described in the scientific literature is presented by [8] within the Tropical Ecology Assessment and Monitoring Network (TEAM network). [8] points out some requirements this type of software must portray. DeskTEAM’s –their desktop application– meets virtually all those requirements. This research community has managed over 1,500,000 photographs worldwide by 2011. We may safely assume that the TEAM network has processed many millions of frames using this application. DeskTEAM source code is partially shared in a Google Code repository. However, even though the project deserves to stand out for its technical achievements, its authors maintain that DeskTEAM is not intended for general purpose projects or devices.

More recently, [6] concludes that the best alternative for projects with reduced budgets may well be to prepare custom forms on Microsoft Access (similar to Camera Base). Through custom forms biodiversity researchers have successfully managed over 300,000 pictures from natural parks in Canada. However, this option is limited by the fact

67 that they are based on a common relational database. The pictures are not storing geospatial data, thus jeopardizing
68 actions like writing some important reports (ESRI Shapefiles and generating other formats, cartography, statistics by
69 location, etc).

70 Finally, a new development has been proposed by [10]. This software has already been tested in an important
71 three-year camera trap project, which generated over one million photographs [11]. The software system, namely
72 Aardwolf, has the ability to automatically extract metadata from images and add customized metadata tags to the
73 images. Aardwolf is a true FOSS project written in C++ and developed to run on the most popular desktop operating
74 systems. Moreover, there are plans to make it more scalable and extendable through a web interface. Its minimalistic
75 approach superbly handles most of the metadata from the camera trapped pictures. However, GIS integration issues
76 are not yet contemplated.

77 From all previous solutions we may say that for a software package to be complete certain features need to be
78 present:

- 79 • An intuitive classification/querying desktop application.
- 80 • Multi-platform development (running on Windows, Mac-OS or Linux platforms)
- 81 • Exif (Exchangeable image file format) standard metadata extracting capabilities and custom manufacturer in-
82 formation.
- 83 • FOSS for ensuring collaborative development and affordable camera trap prices.
- 84 • Collaborative work-flow in the Cloud (with Internet connection or sync capabilities).

85 At this point, we would like to introduce further considerations for present and future developments. New projects
86 should look for:

- 87 • properly integrating GIS capabilities. Many camera trap projects need to manage captures in the form of spatial
88 data, species distribution or abundances required for completing species distribution models and compiling
89 reports on biodiversity management [12].
- 90 • efficiently building on sound projects and platforms. Underlying well known projects will presumably attract
91 the community of developers to work collaboratively, eventually upgrading the project.
- 92 • effectively translating this type of software into mobile platform environs. Developing these tools for smart
93 devices, such as smartphones or tablets, would have some practical advantages, but may also present some
94 hardware hurdles to overcome [13].

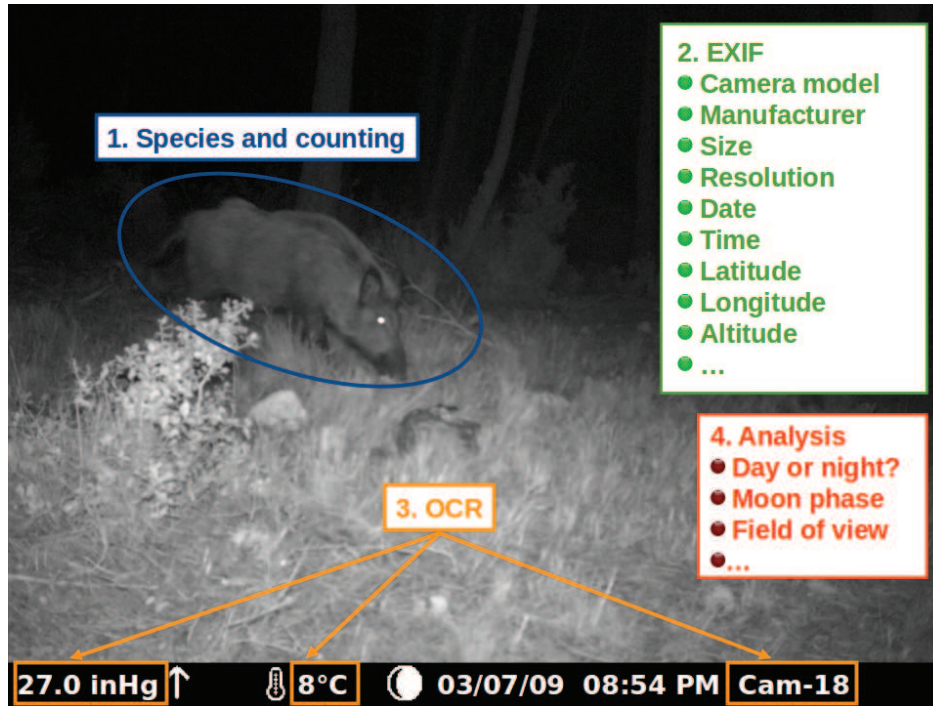


Figure 1. Camera Trap Manager extracts more than 30 different useful variables from images generated by the Moultrie Game Spy I-60 camera traps

3. Development of Camera Trap Manager

From the alternatives discussed earlier, we consider that no one meets all the desirable requirements highlighted in this research project. For this reason, we want to describe our effort in developing a new solution to expedite data management and geographical modelling based on camera trapped data. The design of our application, namely Camera Trap Manager (CTM), and its work-flow are a consequence of the data of interest that can be extracted from the images obtained with camera trap cameras (see Figure 1). The data extracted for our particular case study may well serve in a wide variety of standard formatted image files.

3.1. Case Study

Since 2009 we have been analysing the composition and distribution of the vertebrate community in the Natural Park of Sierra de Mariola (SE Spain) with 17,500 hectares of protected area, within seven urban communities. The study area has a very mountainous and rugged relief, with the exception of some river valleys. Due to its geographical location, this park boasts a typical Mediterranean climate with mild temperatures. Its landscape is mostly natural (67%); there are some areas of non-irrigated crops (24%), residential areas (5%), abandoned lands (3%) and irrigated farmlands (1%). This natural area also features high biodiversity rates in plants and animals. There is a wide variety

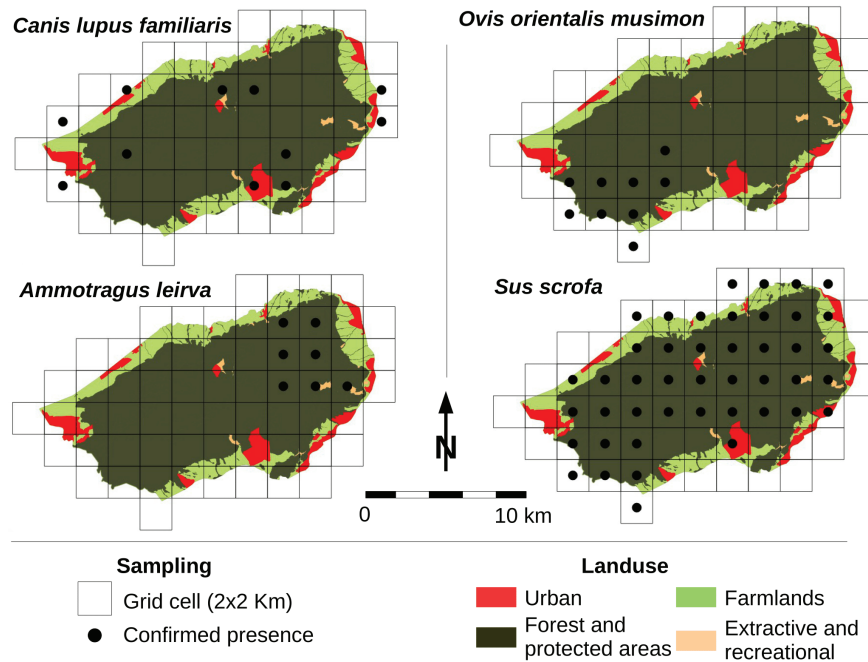


Figure 2. Examples of some species studied using Camera Trap Manager (2009-2010). The grid layer with counting statistics was generated with CTM

of carnivorous mammals, ungulates and several game species. The objective of this project is to watch and analyse the areas biodiversity in terms of sustainability [14].

We used 25 infra-red cameras with motion sensor (Moultrie Game Spy I-60 c), obtaining over 20,000 valid photographs per year. We define valid photographs as those pictures enabling both the species identification and tally keeping. Blurred images captured in movement, for example, would be considered non-valid. The cameras were deployed to determine the abundance of the species sampled in 63 grid cells (2 * 2 Km), covering the whole study area.

Considering this workload and certain project limitations (budget, personnel, time, etc), some level of automation was needed for data processing. Therefore, we developed the application in C#, on the .NET platform, making the most from other well known FOSS projects, such as NetTopologySuite or AForge.NET (typically under General Public License or alike). All of them are based on open standards and analogous projects in many platforms (Java, C++, R, among others). In the following sections, we will briefly explain the main features of Camera Trap Manager and how we achieve a significant increase in productivity (>1000 photos/hour) along with time savings when performing some project outputs. In Figure 2, some examples of spatial outputs are presented. These outputs are spatially aggregated reports, which is our key contribution: converting raw data into geospatial information.

124 3.2. Automatic metadata extraction

125 The main aim of most of CTMs code is to extract data automatically to save time. The application is able to extract
126 Exif metadata, perform Optical Character Recognition (OCR), and calculate several equations based on the available
127 metadata. Modern digital cameras can store some useful information in the form of image metadata, using Exif,
128 International Press Telecommunications Council (IPTC), Extensible Metadata Platform (XMP), and other formats.
129 CTM takes advantage of Exif metadata only.

130 Exif metadata is used for registering data related to capturing conditions (including GPS coordinates), light/temperature
131 conditions, among others, provided that the sensors had been connected to the camera when the picture was taken. It
132 might be the case that images taken by modern or old devices may contain some data directly stamped on the picture
133 itself. This data is not necessarily stored as metadata but manufacturers may decide to include it if appropriate.

134 CTM allows for reading this data using Tesseract 3.0 (<https://code.google.com/p/tesseract-ocr/>), a powerful OCR
135 library able to parse alphanumeric characters stamped on a camera trap picture. With OCR, data such as pressure,
136 temperature and camera name can be directly retrieved from many camera trap pictures where this data is not available
137 as metadata (see Figure 1). Finally, new information can be derived by further processing such metadata. For exam-
138 ple, knowing the date the photo was taken, we may calculate the moon phase, which is certainly illustrating in different
139 mammal studies (see code in https://github.com/benizar/cameratrapmanager/blob/master/CameratrapManager_lib/Analysis/MoonPhas
140 Besides, we need only a GPS position and a date/time to decide whether a picture was taken during the day or at night
141 (see code in https://github.com/benizar/cameratrapmanager/blob/master/CameratrapManager_lib/Analysis/SunriseSunsetCalculations.

142 3.3. GIS-based work-flow

143 In our project, the operator needs to perform two actions by hand: (1) introduce data to identify the species and
144 (2) keep a tally of the species entered, relying on CTMs user interface tools. Other projects will also need to identify
145 animals individually by name, gender, and other variables. For our case study, the current user interface simplifies
146 this process through a Windows Forms environment whose options depend on the project configuration file.

147 As already stated, ensuring GIS integration by following Open Geospatial Consortium (OGC) standards is a key
148 issue for our project. In this way, we can perform some spatial queries (e.g. retrieve camera traps located close
149 to a specific infrastructure, or compute grid density estimations). Geospatial data Integration is achieved through
150 NetTopologySuite (NTS; <https://github.com/NetTopologySuite/NetTopologySuite>), an Application Programming In-
151 terface (API) for GIS operations which implements these standards. NTS was originally ported from the Java Topol-
152 ogy Suite framework (JTS). NTS is compliant with the OGC Simple Feature Access specification, so it provides
153 developers with robust 2D spatial algorithms, a standard geometry model implementation, read/write capabilities for

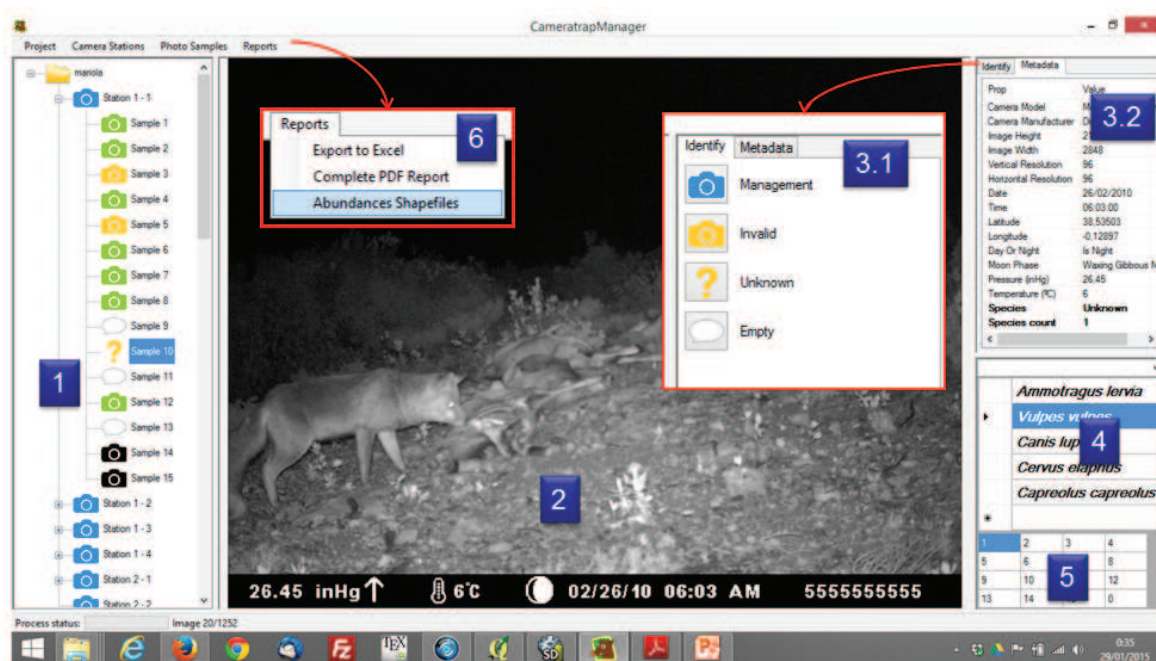


Figure 3. Camera Trap Manager. 1) project tree-view which shows the current processing status of each picture. The user interface focus is always on it, so it is simple to pick options with a mouse and simultaneously navigate through the project structure with the keyboard, 2) image space, 3.1) image status tags with meaningful colours and shapes (e. g. black camera icon is for those pictures processed automatically, but not yet classified by any user), 3.2) most important metadata for each picture, 4) self completing and configurable species list, 5) numeric picker for animals count, and 6) reporting menu

standard vector GIS formats, some overlay functions (intersection, difference, union, symmetric difference), buffer, convex hull, area, perimeter and many more.

Each new CTM project relies on a tree database structure which is an image of the internal Object-Oriented model. Current CTM releases are using SQLite as local binary project repositories, but almost any embedded database system could be used for the purpose. A projects database is structured in “stations” (grid cells) which store “samples” (pictures). A CTM wizard organizes the database following spatial criteria. The user can specify an ESRI Shapefile delimiting some boundaries of the study area (polygon shape). Once the study area is set, a guided process helps the user to create a spatial grid that can be used to organize the camera distribution and calculate grid based statistics and reports. Although camera locations may be manually typed, a more convenient way can be proposed: loading a GPX file (GPS eXchange Format) or a geotagged picture so that the coordinates will directly be loaded into the database.

When pictures are imported, they are linked to their corresponding station (camera). During the loading process, CTM checks if there are any geotagged pictures and all pictures are saved into the project database. Then, all available metadata is extracted, and the necessary calculations performed. Typical elapsed times take approximately 30 seconds for loading a set of 100 medium sized pictures (1.5 to 2 Mb) on an ordinary laptop running on Windows 8 (Intel Core

i5-160 3337U 1.8GHz, 4GB RAM and HHD-500GB). Depending on the project requirements, OCR processing can be omitted, speeding up the loading process by reducing computational requirements by about five times.

Once the photo loading process ends, the user can scroll through the list of images using the keyboard arrow keys, and select the species and its number with simple mouse click actions (see Figure 3). The users can also benefit from several tools to increase productivity (e.g. self completing and configurable taxonomic lists) and a visual tagging system.

With regards to output capabilities, CTM can create different types of reports (see Figure 4), in different formats (e.g. GIS, Excel, PDF), by exploiting the power of Lambda Expressions for writing the required statistics (.NET's LINQ). Currently, these statistics include aggregated countings by area or by species, which can be output as GIS file-based exports or tables and charts in a PDF report.

4. Results and discussion

The main result of this work is the source code of the application and its libraries, which are licensed under the General Public License (GPL), though admitting some degree of compatibility with more permissive software licenses, such as the original MIT/X license, BSD licenses, MPL 2.0, and LGPL. As mentioned in several studies, any development of software published under FOSS license will not be constrained or limited in the future [15, 16]. The source code is available for cloning on the GitHub platform(<https://github.com/benizar/cameratrapmanager>). Not only is it possible to modify and improve the original software project but also some code examples can be adapted for building any specific development.

Our productivity has been greatly enhanced by automating many different tasks. Among the most valuable ones, the ability to readily perform spatial aggregation queries and obtain results in a variety of commonly used formats, such as Excel spreadsheets, GIS exchange files and PDF reports. In the first place, CTM provides a spreadsheet which returns a file very similar to those that had to be compiled by hand a few years ago. As of current release, GIS files consist of two ESRI Shapefiles, a points file for presenting statistics aggregated by cameras and a polygons file for the grid aggregated statistics. These capabilities may be extended in the future to also support open inter-operable GIS formats, such as GML or GeoJSON. Finally, there is an option for creating a PDF report with some projects metadata, data tables and time series statistics for understanding animals behaviour, all grouped by species and by grid cells.

This project research we believe is a modest but useful contribution to address some of the questions mentioned in the introduction. We summarize these contributions in order of magnitude and also present some of the drawbacks we have detected in CTM:

- FOSS project. CTM is a project-specific development tested on three different camera models. Because of our

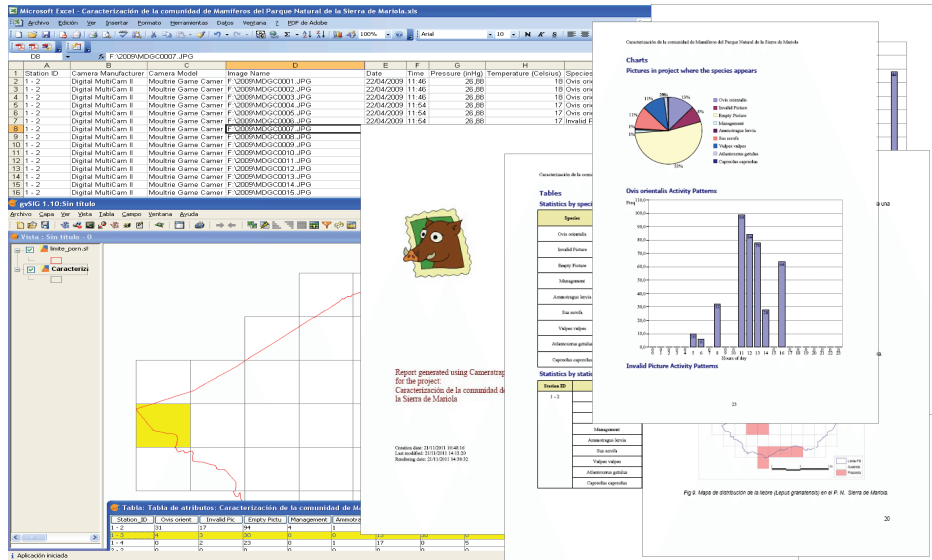


Figure 4. Camera Trap Manager reports examples

limited resources, the software could not match other project requirements. However, this FOSS code is now made available to developers, who can take care of these potential conflicts to meet their specific needs.

- GIS integration. This software is not GIS proper, but its work-flow is based on spatial information, and collected data can be easily explored at different spatial aggregation levels.
- Automatic image metadata extraction. CTM extracts many Exif metadata tags and uses the data for analytical purposes. However, there remain many issues to be solved, such as exploring specific manufacturers metadata or alternatively using some other metadata standards, such as Adobe XMP, for example.
- Multi-platform installation. Provided that this software can be considered a desktop multi-platform development, and assuming that certain capabilities may be readily implemented, we find it completely feasible to develop CTM for smart devices as well, improving its usability in many contexts. Since the CTM code is written in C#, it is more immediate for its migration to the Android platform.
- Collaborative development. One of our biggest concerns is to develop a software package for a larger user community, calling forth greater participation of developers, thus guaranteeing our projects viability. Ours is a work in progress that we intend to accomplish by migrating CTM to smart device platforms.

Considerations we have addressed in this paper point to further developments to be targeted at rewriting routines of the data storage routines in parallel to a migration to the Android platform. With regard to data access, our plans envis-

age the evaluation of different alternatives to avoid the mismatch between CTMs Object-Oriented model and SQLites relational storage. Two approaches are presented: the first, the use of an Object-Relational Mapping (ORM) framework such as greenDAO (<http://greendao-orm.com>) or switching to a NoSQL storage solution. The second, an assessment of Object-Oriented storage engines, such as Realm (<http://realm.io/>) and Perst (<http://www.mcobject.com/perst>). We expect the migration to Android to be immediate, given the language similarities, free software availability and abundant documentation. The Android app is currently being developed, and it will be soon published as an alpha release. Prototype source code can already be checked at its GitHub repository (<https://github.com/benizar/ctmdroid>). We acknowledge that being FOSS is not enough if this software does not take into account its target users. Developing a project in a high-level programming language may help boosting a developers community while making it available for mobile devices, such as tablets or smartphones. The installation process will presumably become much easier and productivity greatly enhanced if we are capable of providing a much simpler and more intuitive user interface. Similarly, programming for battery-efficient devices can be an advantage in several contexts, as for example, when fieldwork is performed in remote areas.

Our experience with CTM has shown that maximizing productivity in processing photographs taken by camera traps speeds up the analysis of geographical distribution and the setting-up of predictive models within a reasonable time. Unfortunately, there are not enough studies discussing productivity achieved by other software alternatives. According to [6], using a custom form on a Microsoft Access database, the least regular contributors to the project were able to classify about 500 photos/hour while a technician regularly performing this task is able to classify, about 700-1000 photos/hour on average.

In our case study, we have achieved very interesting results in terms of productivity. Whereas in manual processing an average of 55 photos/hour was reached (extracting only six variables), CTM processes more than 1,000 photographs/hour and gets a larger amount of information (more than 30 variables are extracted from metadata, read with OCR or derived by calculations). This figures are achieved by well-trained users, which are familiar with protocols and species.

Acknowledgements

This project has been carried out thanks to the UNCROACH project (CGL-2011-30581-CC02-01) from the Spanish Ministry of Economy and Competitiveness; and has been partially funded by the public call for research projects of the Conselleria of Education (T6217-2010); and the Institute of Culture Juan Gil-Albert-Alicante.

References

- [1] L. Silveira, Camera trap, line transect census and track surveys: a comparative evaluation, *Biological Conservation* 114 (3) (2003) 351–355. doi:10.1016/S0006-3207(03)00063-6.
URL <http://linkinghub.elsevier.com/retrieve/pii/S0006320703000636>
- [2] A. F. OConnell, J. D. Nichols, K. Ullas Karanth, *Camera traps in animal Ecology*, 1st Edition, Springer, New York, 2011.
- [3] K. U. Karanth, J. D. Nichols, Estimation of Tiger Densities in India Using Photographic Captures and Recaptures, *Ecology* 79 (8) (1998) 2852–2862. doi:10.1890/0012-9658(1998)079[2852:EOTDII]2.0.CO;2.
URL <http://www.esajournals.org/doi/abs/10.1890/0012-9658%281998%29079%5B2852%3AEOTDII%5D2.0.CO%3B2>
- [4] F. Botello, G. Monroy, P. Illoldi-rangel, I. Trujillo-bolio, V. Sánchez-Cordero, Sistematización de imágenes obtenidas por fototrampeo: una propuesta de ficha, *Revista Mexicana De Biodiversidad* 78 (2007) 207–210.
- [5] M. J. Kelly, Design, evaluate, refine: camera trap studies for elusive species, *Animal Conservation* 11 (3) (2008) 182–184. doi:10.1111/j.1469-1795.2008.00179.x.
URL <http://www.blackwell-synergy.com/doi/abs/10.1111/j.1469-1795.2008.00179.x>
- [6] M. Barrueto, A. P. Clevenger, B. Dorsey, A. T. Ford, A better solution for photo-classification, automatic storage and data input of camera data from wildlife crossing structures, in: *International Conference on Ecology and Transportation (ICOET 2013)*, no. Icoet, Scottsdale, Arizona, USA, 2013, pp. 1–11.
URL http://www.icoet.net/ICOET_2013/documents/papers/ICOET2013_Paper402B_Barrueto_et_al.pdf
- [7] G. Harris, R. Thompson, J. L. Childs, J. G. Sanderson, Automatic Storage and Analysis of Camera Trap Data, *Bulletin of the Ecological Society of America* 91 (2010) 352–360. doi:10.1890/0012-9623-91.3.352.
- [8] E. H. Fegraus, K. Lin, J. a. Ahumada, C. Baru, S. Chandra, C. Youn, Data acquisition and management software for camera trap data: A case study from the TEAM Network, *Ecological Informatics* doi:10.1016/j.ecoinf.2011.06.003.
URL <http://linkinghub.elsevier.com/retrieve/pii/S1574954111000549>
- [9] S. R. Sundaresan, C. Riginos, E. S. Abelson, Management and Analysis of Camera Trap Data: Alternative Approaches, *Bulletin of the Ecological Society of America* 92 (2011) 188–195. doi:10.1890/0012-9623-92.2.188.
- [10] Y. S. Krishnappa, W. C. Turner, Software for minimalistic data management in large camera trap studies, *Ecological Informatics* 24 (2014) 11–16. doi:10.1016/j.ecoinf.2014.06.004.
URL <http://dx.doi.org/10.1016/j.ecoinf.2014.06.004>
- [11] W. C. Turner, K. L. Kausrud, Y. S. Krishnappa, J. P. G. M. Cromsigt, H. H. Ganz, I. Mapaure, C. C. Cloete, Z. Havarua, M. Kusters, W. M. Getz, N. C. Stenseth, Fatal attraction: vegetation responses to nutrient inputs attract herbivores to infectious anthrax carcass sites, *Proceedings of the Royal Society B: Biological Sciences* 281 (1795) (2014) 20141785–20141785. doi:10.1098/rspb.2014.1785.
URL <http://rspb.royalsocietypublishing.org/cgi/doi/10.1098/rspb.2014.1785>
- [12] F. Rovero, F. Zimmermann, D. Berzi, P. Meek, "Which camera trap type and how many do I need?" A review of camera features and study designs for a range of wildlife research applications, *Hystrix* 24 (February) (2013) 148–156. doi:10.4404/hystrix-24.2-6316.
- [13] A. G. F. Teacher, D. J. Griffiths, D. J. Hodgson, R. Inger, Smartphones in ecology and evolution: A guide for the app-rehensive, *Ecology and Evolution* 3 (16) (2013) 5268–5278. doi:10.1002/ece3.888.
- [14] A. Belda, B. Zaragozí, I. Belda, J. Martínez, E. Seva, Traditional knowledge of medicinal plants in the Serra de Mariola natural park, south-eastern Spain, *African Journal of Traditional Complementary and Alternative Medicines* 10 (2) (2013) 299 – 309. doi:http://dx.doi.org/10.4314/ajtcam.v10i2.15.
- [15] S. Steiniger, G. J. Hay, Free and open source geographic information tools for landscape ecology, *Ecological Informatics* 4 (4) (2009) 183–

1
2
3
4 281 195. doi:10.1016/j.ecoinf.2009.07.004.

5 282 URL <http://linkinghub.elsevier.com/retrieve/pii/S1574954109000363>

6 283 [16] B. Zaragozí, A. Belda, J. Linares, J. Martínez-Pérez, J. Navarro, J. Esparza, A free and open source programming library for landscape
7 metrics calculations, *Environmental Modelling & Software* 31 (2012) 131–140. doi:10.1016/j.envsoft.2011.10.009.

8 284
9 285 URL <http://linkinghub.elsevier.com/retrieve/pii/S1364815211002209>

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

ACCEPTED MANUSCRIPT

Highlights

- We present a new software for managing and analysing camera trapped data
- Users can spatially organise projects and generate aggregated reports
- The software may facilitate biodiversity studies by reducing processing time
- We illustrate the software performance in a real case study in a Mediterranean region (SE Spain)
- The software is open source and freely available (FOSS).