# Automatic data organization, storage, and analysis of camera trap pictures

**Jim Sanderson**[1,2] and **Grant Harris**[3]

[1]Small Wild Cat Conservation Foundation,
[2]Wildlife Conservation Network
[3]US Fish and Wildlife Service

Corresponding author: Jim Sanderson, email: *gato_andino@yahoo.com*

### Abstrak

Sistem jaringan kamera otomatis mampu menghasilkan ribuan gambar dalam waktu yang singkat. Sebagai contoh, 35 kamera di Sevilleta National Wildlife Refuge  di new Mexico, Amerika Serikat telah memproduksi lebih dari 1.9 juta gambar sejak Juni 2009. US Fish & Wildlife program monitoring menghasilkan kira-kira 30.000 gambar per minggu. Meskipun pengambilan dan penyimpanan gambar adalah hal yang dianggap sepele, entri dan analisa data, keduanya masih menyita waktu serta rentan mengalami kesalahan. Hal ini terutama pada saat entri data kedalam lembaran data dengan pengetikan tangan menggunakan keyboard. Tujuan kita dalam hal ini adalah untuk meningkatkan kecepatan entri data dengan meminimalkan kesalahannya, lebih  mudah melakukan analisa, serta memungkinkan data dari berbagai lokasi dapat terhubung dan dianalisa dalam satu rangkaian data. Tujuan tersebut dapat dicapai dengan mengeliminasi entri data dengan tangan menggunakan keyboard, dan lebih mengarahkan pengguna untuk berinteraksi dengan file data gambar melalui penggunaan perangkat open software. Pengecekan kesalahan juga secara otomatis. Disini kita memperbaharui metodologi yang dijelaskan dalam Harris et al. (2010) dengan menyediakan langkah-langkah panduan penyimpanan data perangkap kamera dan analisa, tanpa melakukan entri data dengan keyboard. Dari komunitas pengguna yang telah ada, metodologi ini sudah digunakan dan memberikan keuntungan. Program ini tersedia secara gratis di http://www.smallcats.org/CTA-executables.html.

### Abstract

Networks of automatic cameras are producing many thousands of images over modest time periods. For example, 35 cameras at Sevilleta National Wildlife Refuge in New Mexico, USA have produced more than 1.9m useful images since June, 2009. A US Fish and Wildlife monitoring program is producing about 30,000 images per week. Although image file retrieval and storage is trivial, data entry and analysis are both time consuming and error prone since data is most often entered by hand from a keyboard into a spreadsheet. Our objectives were to increase data entry speed while minimizing data entry errors, easily run data analysis, and enable data from multiple locations to be concatenated and analyzed as a single data set. These objectives are achieved by eliminating the task of hand data entry via a keyboard, and managing user interactions with image file data through the use of a suite of open software tools. Error-checking is also automatic. Here we update the methodology described in Harris et al. (2010) by providing a step-by step guide for automatic camera trap data storage and analysis without entering data by hand from a keyboard.  This methodology is already in use and is benefiting from an established user communitym. The programs are available free on http://www.smallcats.org/CTA-executables.html.

*Key words*: Automatic data organization; analysis; camera trap

## Introduction

The use of automatic cameras to monitor wildlife has increased dramatically globally. The greater community of users recognizes the need for a common, simply way to organize, store, analyze, and share data from camera traps. Most users realize that data entry by hand from a keyboard is both cumbersome and error prone (Maydanchik, 2007). Moreover, once data is entered into a spreadsheet, data manipulation is done mostly by hand. Harris et al. (2010) offered a methodology to overcome many of the issues faced by users of automatic cameras. The methodology is intuitive and does not require data entry by hand from a keyboard.

Received 22 October 2012; revision accepted 10 November 2012

Each step from data organization to data analysis and sharing is facilitated by powerful software available free on the internet. Numerous advantages occur when using a standardized methodology, not the least of which is that data from multiple sites can be analyzed as a single data set. Moreover, intermediate files that are produced automatically can be loaded error-free into spreadsheets.

The methodology described in Harris et al. (2010), having been adopted by many users, has since been improved and streamlined. What follows is a step-by-step end-to-end description of the methodology that by its very nature recognizes that camera data collection and analysis is iterative and ongoing. No attempt is made to replicate other camera analysis software tools.

## Methods

Data relabeling and storage is a two-step process. First, all images are labeled with their date and time simultaneously and automatically. Second, images are sorted into location/species/number-of-individual folders. Once this step is completed data analysis is set up by running program DataOrganize, adding the UTM and elevation information to each camera location, and then running program DataAnalyze. Since the methodology is iterative, an inexperienced user can go through the entire procedure with a few images taken from several camera locations. Before the first step can be done, preliminary groundwork is needed once and only once.

### Data retrieval  and relabeling

First, a summary of the camera trap data retrieval, relabeling, identifying and counting, and then storing the images is given. Following the summary, each step is detailed. Also described is a program that relabels jpg-files that have been incorrectly dated.

*Groundwork details*

First create a top-level folder here called AllCameraTrapData that will contain all the programs, How_to text files, PowerPoint presentation, published paper by Harris et al. (2010), and three folders: AllLocations, SpecialCases, and Unsorted. Open http://www.smallcats.org/CTA-executables.html and download and save all the files into folder AllCameraTrapData. Folder AllLocations (any name can be used so long as it is one word without embedded blanks) will contain all the camera trap images. Folder SpecialCases (again any name will suffice so long as it is one word without embedded blanks) will contain all images whose date and time are not correct. Folder Unsorted (or whatever name is used) will contain all unsorted relabeled images organized by location and date (Fig.1). Alternatively, folder "Unsorted" can be placed in each location folder.

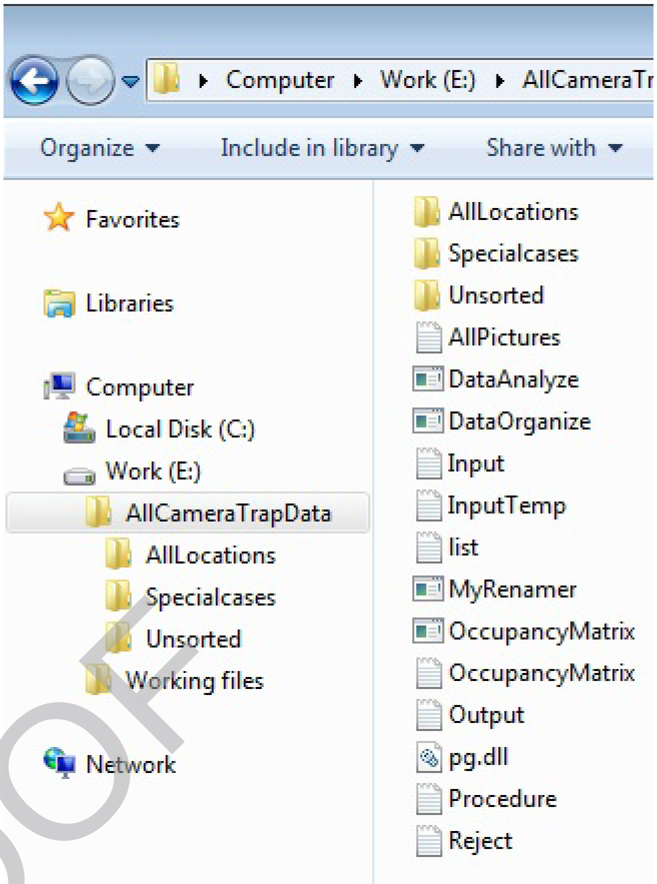


**Figure 1.** The folder here called AllCameraTrapData contains three folders: AllLocations, SpecialCases, and Unsorted. All programs and files are downloaded sand saved in folder AllCameraTrapData.

*I. Create a hierarchal folder structure to hold the images*

The following file structure is needed within the AllLocations folder (Fig. 2). Each image will ultimately be stored in AllLocations/Location/Species/# folder. Note that a template location folder containing species/# folders can be copied, pasted, and relabeled with the new location. Folders can also be created when needed such as during the drag-and-drop procedure described below.

(1) Site refers to the general area where the camera traps are running. Site is the top-level main directory where all files are stored. This might be Gunung Kerinci, Komodo, or AllCameraTrapData (used here). This folder contains all the programs, How_to text files, and all other files necessary for camera trap data analysis (Fig. 1).

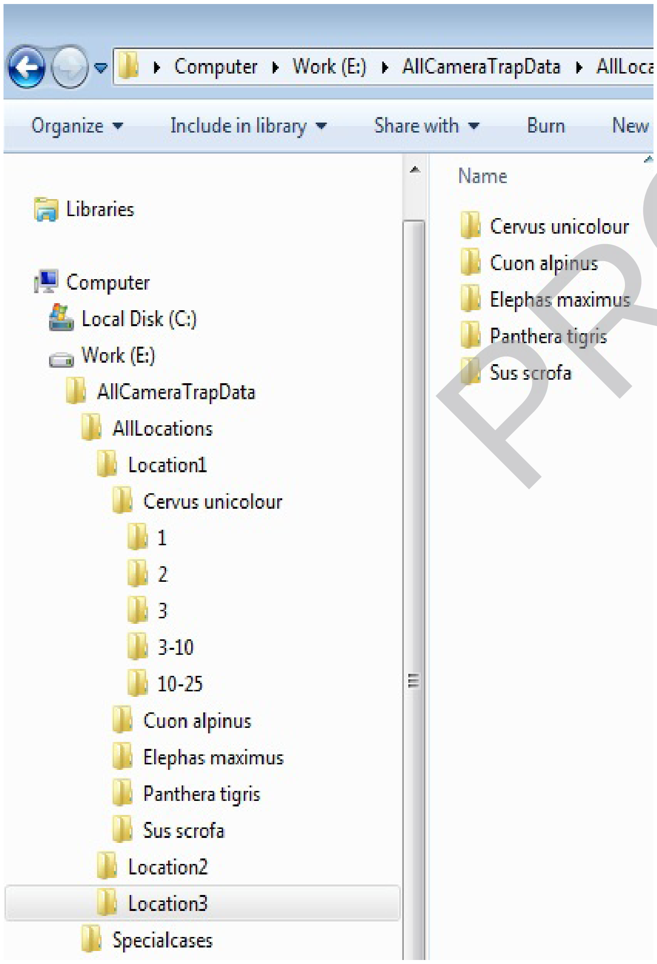


**Figure 2.** An example of a hierarchal folder camera trap data structure. Note that the folder structure within AllLocations is three and only three levels deep.

(2) Location is a place where a camera trap is set. Locations might be Loc01, Loc02, or have names like Sg. Young, Bekol, or Bukit Lawang. Make sure no two locations have the same name. Location names are limited to 25 characters (Fig. 2).

(3) Species folders are folders of species names in each location folder. These might be Tiger, elephant, and Domestic dog. Species names are limited to 25 characters, and it is recommended that Latin names are used for consistency.

(4) Number (#) are folders in each species folder labeled with the number-of-individuals in the image: 01, 02, …, 99. Note that a species folder labeled "Unknown" has a number-of-individuals folder 01. Without exception, all jpg files reside in number-of-individuals folders. Not all 01-99 are necessary. It does not matter if some are empty or are not present. It is likely that for most species there will be only a single folder (namely 01) (Fig. 2).

(5) In addition to actual species, folders can represent multiple species such as Birds, Amphibian, or Skunks that represent several skunk species that can be later identified by an expert. Other typical folders are "Ghost" where waving grass triggered an image, or "Unknown" such as an image taken at night that contains an animal that cannot be identified, Human, and Vehicle. These are typically not included in data analysis. The "AllLocations" folder can have any name so long as it is one word with no embedded blanks (Fig. 2).

*II. Visit a camera trap, remove the memory card, and store the images in a temporary folder here called Unsorted.*

Storing all the images from a memory card in a temporary location folder here called "Unsorted" and then relabeling them in step III prevents subsequent loss of image files. Image file counting resets to zero when images are removed from the memory card. Subsequent file moves can overwrite existing files. Relabeling the image files with their date and time makes the image files unique and prevents files from being over-written.

*IIIa. Digital images. Relabel all the files in the temporary folder simultaneously with the date and time the image was recorded.*

The freeware program "ReNamer" is used to simultaneously relabel all images in the temporary location folder. ReNamer is downloaded from: http://www.snapfiles.com/downloads/denrenamer/dldenrenamer.html.

Follow the download instructions and create a short-cut on your desktop. Then,

(1) Open ReNamer by double clicking on the icon.
(2) Click on Settings, then Meta Tags.
(3) Remove the punctuation in the date so that a single blank space separates each field. The date and time should look like:

yyyy mm dd hr mn sc

Then click on the save button.

(4) Drop and drag a file, group of files, or a folder into the lower frame of ReNamer. The files are listed.

(5) Click + Add to add the first rule. Select "Delete". Then in the "Until box" click on the button "Till the end". The button "Skip extension" has already been selected. This rule deletes the name of each jpg file, but gives an error that there are multiple files with the same blank name. Ignore the error. Click on + Add Rule to add the rule.

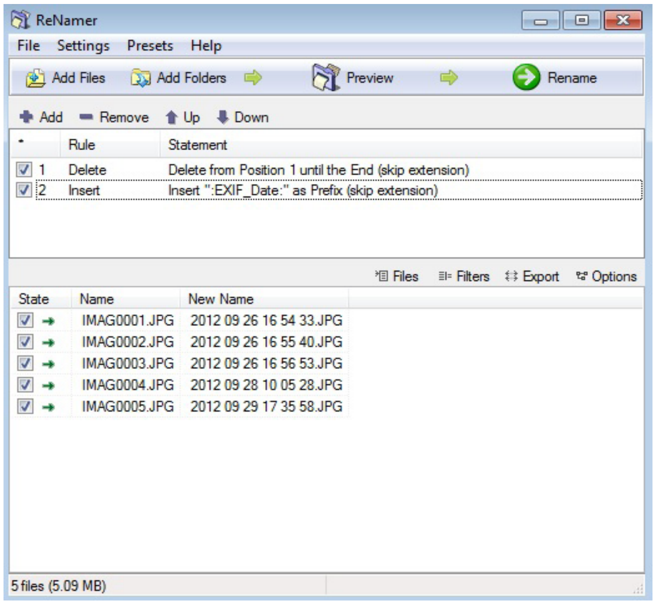
(6) Click +Add to add the second rule. Select Insert, then click the yellow lightening bolt Insert Meta Tag, and select from the drop-down list EXIF_Date. Then click the green check Insert. Click +Add Rule.

(7) Click on Preview on the upper right of the ReNamer box and the new names should appear. If the file label does not appear as yyyy mm dd hr mn sc.jpg then chances are the camera trap does not keep track of seconds so that two files have the same name but were taken in different seconds. Add five seconds to the time. To the right of "Preview" click on the green "Rename" right arrow and the files are relabeled and saved (Fig. 3).

(8) From the top menu list select "Presets", save as, and then give your rules a name like "EXIF_Date_Time". You can then use the rules again without re-entering them simply by loading from Presets.

(9) We recommend creating three Preset rules. In (6) present rule EXIF_Date_Time was created. Create two more preset rules using step (5) and then in (6) chose from MetaTags File_DateModified. Store this as Preset File_DateModified. Next create another Preset called File_DateCreated by choosing in step (5) File_DateCreated from the MetaTags menu.

(10) Each of the three Preset Rules contains two rules. The first rule deletes the current label, and the second inserts the EXIF_date & time, FileDateModified, or File_DateCreated –



|  | Rule | Statement |
|---|---|---|
| ☑ 1 | Delete | Delete from Position 1 until the End (skip extension) |
| ☑ 2 | Insert | Insert ":EXIF_Date:" as Prefix (skip extension) |

| State | Name | New Name |
|---|---|---|
| ☑ → | IMAG0001.JPG | 2012 09 26 16 54 33.JPG |
| ☑ → | IMAG0002.JPG | 2012 09 26 16 55 40.JPG |
| ☑ → | IMAG0003.JPG | 2012 09 26 16 56 53.JPG |
| ☑ → | IMAG0004.JPG | 2012 09 28 10 05 28.JPG |
| ☑ → | IMAG0005.JPG | 2012 09 29 17 35 58.JPG |

5 files (5.09 MB)

**Figure 3.** A screen-shot of ReNamer after selected pictures have been renamed according to yyyy mm dd hr mn sc.jpg.

whichever works for the camera traps. Check that the date and time label matches the date and time shown on the image. If the label does not match the date and time on the image, apply a different Preset Rule. Note that only the file label is altered. The image metadata is not altered.

*IIIb. Film images. Relabel all the files in the temporary folder by hand with the date and time the image was recorded.*
Unfortunately for film images this step must be done manually since the year and month are inferred and the day and time are imprinted on the image. The subsequent steps are the same.

*IV. For each image file, identify the species, count the number-of-individuals of each species recorded, and store the file in the proper location/species/number-of-individuals folder (Fig.2).*
View an extra-large icon of each relabeled image in the temporary folder, identify the species recorded, count the number-of-individuals, and drag-and-drop the file to the proper location/species/number-of-individuals folder. In the event that two or more species are recorded, store a copy for each species in the proper species/number-of-individuals folder.

*V. Return to step II above until all camera traps have been visited and images processed. Note that at no time was data enter data by hand.*

## Data storage

### I. ID, count, drag & drop

In Windows use Explorer to open the folder AllCameraTrapData and display the location, species, number-of-individuals folders on the left (if you prefer to drag left) or right side (if you prefer to drag right) on the screen (Fig.2). Double-click the Unsorted, location-date folder that contains the images that have been relabeled with their date and time but have not yet been sorted. If extra-large icons are used it is most often unnecessary to open the image to id and count the species and individuals. For example, suppose the first image at Location 01 contains 10 Sambar deer. This image is dragged and dropped to its Location 01 folder, species folder Sambar deer, number-of-individuals folder 10. Using large icons about 500 images per hour can be stored. This procedure is repeated for all subsequent images stored in Unsorted/Location 01 until the Location 01 is emptied. All camera locations are treated the same way each and every time an SD card is removed from the camera.

Relabeled images are dragged-and-dropped from the Unsorted/Location folder to the sorted Location/Species/# folders. This completes the data organization step. Images accumulate in the same folder hierarchy.

## Data creation and analysis

There are two steps: (1) data creation and (2) data analysis. Both are done with software.

### I. Creation of AllPictures.txt and InputTemp.txt

Data creation is necessarily an iterative step since DataOrganize terminates on each error it finds, but attempts to show where the error occurred. Two files are used to find relabeling or storage errors: AllFoldersFiles.txt and AllInfo.txt. If DataOrganize terminates normally these two files do not appear.

By this step all camera trap images have been relabeled with their date and time and stored in the Location/Species/Number-of-Individuals folders. Folder AllLocations (or whatever you called this folder) has all the location folders.

Double-click on DataOrganize, enter the name of the AllLocations folder making sure that the spelling matches the folder name exactly. Allow the program to run to completion which might take some time since the files must all be sorted chronologically within each folder.

If the program ends with the message *press enter to terminate normally* congratulations. AllPictures.txt and InputTemp.txt have been created and added automatically to your folder. Sometimes, there will have been an error in the way images were relabeled or stored causing DataOrganize to abort prematurely. If DataOrganize does not terminate normally, with

100% certainty there is an error someplace. Open AllInfo.txt and use it to track down and repair the error that can be pinpointed in AllFoldersFiles.txt. AllFoldersFiles.txt is complex but systematic list of what is in folder AllLocations. DataOrganize is infallible. After repairing the error, double-click on DataOrganize again and repeat this procedure.

Repeat this step until all errors are corrected and the message *press enter to terminate normally* appears. Only AllPictures.txt and InputTemp.txt appear. AllInfo.txt and AllFoldersFiles. txt are automatically deleted.

AllPictures.txt is a simple text file listing of all that is known of each camera trap image ordered first by location, then species, date, time, and number of individuals. Each line appears as:

```
Location 01 Sambar deer 2012 10 10 12 51 35 10
```

Text file AllPictures.txt has three roles. First, since each line is space delimited AllPictures.txt can be loaded into any spreadsheet program such as Excel. Second, AllPictures.txt is easily e-mailed. Third, AllPictures.txt from other sites can be easily concatenated and analyzed together as a single camera trap effort.

Text file InputTemp.txt is also produced. InputTemp.txt is a list of all camera locations, their start and stop dates, and has place holders to enter the UTM location and elevation. InputTemp.txt also has a list of all species including Ghost, Unknown, Human, Vehicle, and other species that might not be included in the analysis step. InputTemp.txt is edited and saved as Input.txt for use in the data analysis step. Input.txt serves three purposes. First, it contains UTM and elevation information for the camera location. Second, it is used to control what camera data is analyzed. Third, it can be easily e-mailed.

InputTemp.txt looks like this. Comments are in [*].

```
CAMERA TRAP DATA FROM XXXX   [single line information header edited by user]
2  UTME-W  UTMN-S ELEV  [2 is the number of camera locations].
Arroyo Seco        000000    000000     000  [first location and space holder for UTM and elevation].
1 [1 is the number of start-stop dates that is accepted or edited by user].
2005  8  1 2008 10 14   [sequential list of start-stop dates of camera].

Gundagai  000000    000000      000
1
2005  8  5 2008 11 17
8 [number of species taken from all location folders]
Elephant [list of species in alphabetical order]
Ghost
Human
Peacock
Sparrow
Unknown
Vehicle
Zebra
```

Editing InputTemp.txt to create Input.txt

InputTemp.txt is edited by the user. Here we have changed the header to be more informative, entered the UTM and elevation, changed the number of start-stop dates for each camera, and edited the species list to contain only the species we want to analyze. Note that the number of start-stop dates changed from 1 to 6 for the first camera and 1 to 5 for the second camera. Note also that only the first list species will be analyzed; in this case two mammals will be analyzed. The list can be switched and birds will be analyzed next. Species Ghost, Human, Unknown, and Vehicle will not be included in the analysis. InputTemp.txt is then saved as Input.txt.

```
SUMATRA CAMERA TRAP DATA
2                        UTME-W UTMN-S ELEV
Arroyo Seco               524907    511503    316
6
2005    8  1 2005  8  1
2005    9 14 2005  9 30
2006    1 16 2006  5 31
2006    7  7 2007  8 12
2007    9 23 2007 10 10
```

```
2007    11 29 2008 10 14
2005 8 1 2008 10 14

Gundagai                523778  507548  285
5
2005  7  5 2005 11  8
2006  1  2 2006  1 15
2006  2 17 2006  5  6
2006  6 22 2006  7 29
2006 12 23 2008 11 17
2
Elephant
Zebra
2
Peafowl
Sparrow
```

## Data analysis

Data analysis uses AllPictures.txt and Input.txt. Double-click on DataAnalyze. File Output.txt contains the full analysis of your data.

## Additional features: correcting corrupted jpg-files

Program MyRenamer is used to correct corrupted jpg files. Typically MyRenamer is used when a camera trap's date and/or time have not been properly initialized. Store all the jpg files that need correcting in a folder here called SpecialCases in folder AllCameraTrapData folder where all the programs are located. Double-click on MyRenamer. The program asks for the folder where the jpg files are stored. Type SpecialCases (or whatever you called it), then Enter.

The program asks for the offset. The offset is 6 integers each of which is>= 0 and each separated by a space. The offset is added to the date and time to correct them. Note that carry-forward happens. Thus, if 7 months are added to a jpg file that shows 9 months, then the result will be that the final correct month is 04 and one year will be added to the year automatically. That is, just as is the case with the camera: a certain number of years, months, days, hours, minutes, and seconds are behind, and this "constant offset" is true for every image. Note that if an error has been made in the offset then the offset can be corrected and MyRenamer run again. The second, or corrected offset, is not a correction to the first offset but rather the actual jpg file as it came from the camera trap. The date the jpg file was taken according to the camera trap can be seen in the outMyRenamer.txt file.

Sequential or burst images with the same date and time can be relabeled properly using MyRenamer. If the offset is 0 0 0 0 0 0 then MyRenamer will read the date and time from the image metadata, and add 5 seconds until each image is uniquely labeled with its date and time + some multiple of 5 seconds.

### *Preserving changes in Input.txt when more data is added*

Program UpdateInput is used to create a new Input.txt file from an old Input.txt file so that you do not have to retype the UTMs, elevations, and alter start times each time more data is stored. First rename Input.txt as InputOld.txt. Double click on DataOrganize to produce new AllPitcures.txt and InputTemp.txt files.

Double-click on UpdateInput. This produces InputNew.txt that preserves the first line, all UTM and elevation information, and start dates you may have changed. Note that the species list is not preserved since it is replaced when new images are included. New stop times and new locations are added by DataOrganize. You must now edit InputTemp.txt to produce Input.txt that will be used in the analysis.

## Creating multiple start-stop times

Program CorrectInput uses AllPictures.txt and Input.txt to create a new list of start-stop times for each camera trap location. At each location, consecutive images separated by more than 29 days are assumed to have resulted from a dead camera trap. This dead-time gives rise to multiple start-stop times. File DateList.txt contains the list of start-stop dates that can be substituted into InputTemp.txt.

### Spreadsheet camera trap data

Much camera trap data is presently stored in spreadsheets. In this case, the spreadsheet functions can be used to produce the AllPictures.txt that contains a list of images in the form (*) above. Since no protocols for storing camera trap data exist, it is not possible to give a typical example of how to produce (*) above from existing spreadsheet data. However, we have not yet found an example of spreadsheet data that cannot be converted and stored in text form identical to (*) above that is required by the analysis program software. Presuming AllPictures.txt has been created InputTemp.txt is easily created by double-clicking program CreateInput.

### Analyzing multiple sites

Multiple sites can be analyzed by concatenating AllPictures.txt from each site. Double-click CreateInput to create InputTemp.txt. Edit InputTemp.txt and save as Input.txt. Double-click DataAnalyze to produce a full analysis of all sites.

### Creating occupancy matrices for programme presence

Every user knows that creating matrices for program PRESENCE to estimate occupancy is difficult and fraught with errors. Moreover experimenting with the number of so-called camera trap occasion days is difficult. Program MatrixOccupancy removes these difficulties and is error-free. MatrixOccupancy uses AllPictures.txt to produce matrices for program PRESENCE using the number of camera trap days specified by the user. Periods where a camera is not operated appear as a "-", an absence is "0", and a presence is "1" as is required by program PRESENCE. The text matrix is simply copied and pasted into program PRESENCE. Simply double-click MatrixOccupancy and the enter the number of days in a camera trap occasion.

### Discussion

Many millions of camera trap pictures have been taken from diverse locations around the world. Unfortunately far fewer have been analyzed. The vast majority of camera trap pictures remain simply pictures, disconnected and possibly endangered records existing on a computer hard drive somewhere. These camera trap pictures, some dating back 30 years, would be extremely valuable if they were all relabeled with their date and time, sorted, and stored. Mining this veritable mountain of camera trap data would become possible, and results collected today could be compared with results collected in the past.

We believe that camera trap pictures are the equivalent of museum specimens; both require curating. The methodology we described aids curating camera trap pictures, and we believe that whether our methodology for relabeling and sorting pictures is used or not, using a standardize method is better than using no method. Migrating from one relabeling and sorting procedure to another can be facilitated with software but waiting to adopt the latest methodology is like waiting to purchase a new more powerful computer – it is easier to upgrade than to start from the beginning.

Users enjoy the power of community. We have had discussion with camera trap manufacturers about labeling pictures with the date and time the picture is recorded on the memory card. When enough users of our methodology make the same demand perhaps the manufacturers will answer our requests. Users of our methodology agree that entering data by hand at a keyboard is now a thing of the past and recently we added the option of "drag-and-drop sorting step" sorting by voice commands. The user's voice is transmitted to the computer program that interprets the voice command, and executes it. For example, if a picture of two elephants appears on the screen the user speaks "elephant". If no elephant folder exists, one is created in the proper location folder. The question "How many?" appears on the screen. "Two" the user answers, and the picture is automatically moved to the Elephant/02 folder and the next picture appears automatically without touching the keyboard or mouse.

However, one step still requires a human: identifying the species and counting the number of individuals in a picture. No computer program exists to do this. Even when the list of species at a location is small, say on the order of 10, no program exists to identify the species and count the number of individuals. Pattern recognition remains a difficult problem that goes far beyond identifying animals in pictures. Perhaps the growing user community will encourage others to solve this outstanding problem.

The growing user base means that more useful programs will be written. Now available on http://www.smallcats.org/CTA-executables is a new program called GroupLocations. Program GroupLocations uses AllPictures.txt, Input.txt (both generated by program DataOrganize) and a user provided text file GroupMap.txt. For instance, imagine a camera trap study of 5 mountain ranges, each with some number of cameras. We wish to analyze and present the analysis of all the data by mountain range rather than by camera trap location as program DataAnalyze does. GroupMap.txt lists all the camera locations and the "group" that each is mapped into. Double-clicking GroupLocations creates a full analysis of each mountain range, treating each camera in each location as independent (just as program DataAnalyze does). Of course, such an analysis could be created by hand from output produced by DataAnalyze, however, GroupLocations does this much faster and flawlessly.

Another user has suggested the need for program Corridor that can identify when, where, and how long it takes a species to traverse a corridor between a pair of cameras. As with all the analysis programs Corridor starts with AllPictures.txt and InputTemp.txt (generated by DataOrganize). The user provides a list of paired cameras that act as possible corridors.

We have no doubt other users' ideas will benefit a growing global user community. Our hope is that camera trap surveying and monitoring programs will continue to expand, and at the same time begin mining the mountains of accumulated camera trap data that resides on countless hard drives across the world before it is lost.

## References

Harris, G., Thompson, R., Childs, J.L. and J.G. Sanderson (2010). Automatic Storage and Analysis of Camera Trap Data. *Bulletin of the Ecological Society of America* **7**:352-360.

Maydanchik. A. (2007). Data quality assessment. Technics Publications, Bradley Beach, New Jersey, USA.