rePhoto vignette

Matthew Gilbert

1/5/2022

A demonstration of the functions for the rePhoto R package

Developed by Matthew Gilbert (megilbert at ucdavis dot edu)

Introduction

You have a historical photograph and want to re-photograph it from the same location as the original. But how do you find the original camera location? The rePhoto package has functions that apply a geometric method to calculate probable search areas for the original camera location.

This code accompanies a currently unpublished manuscript: METHODS FOR FINDING THE LOCATION OF HISTORICAL PHOTOGRAPHS FOR REPEAT PHOTOGRAPHY

Functions in the package take inputs of user specified geographic points in an image and the latitude and longitude of those points. The functions output a predicted search area; see examples below. It is unnecessary to understand the calculations involved, but the general principle is that a value called a Theta difference is close to zero at the original camera location of a historical photograph. Package functions output a plot of contours of similar Theta differences, the smaller the difference the more likely that the location corresponds with the historical photograph location. Theta differences are a value that represents the difference between distances measured on a photograph and those on a map. Theta for the photograph is calculated from distances between X pixel coordinates measured using Microsoft Paint, ImageJ or other image analysis software (see Fig. 1).
Theta for the map is calculated from geographic coordinates (latitude and longitude) measured with GIS software such as Google Maps, Google Earth, CalTopo etc. The difficulty is finding at least three points of reference in a photograph that can also be found on a map/satellite photograph.

Step 1: Set up R environment

If you haven't already, download and install R: here (https://cran.r-project.org/)

Download and install the rePhoto package from GitHub here (https://github.com/MatthewGilbertUCD/rePhoto.git). You can do this in R with the following commands, these require the devtools package (instructions can be found here (https://cran.r-project.org/web/packages/devtools/readme/README.html)):

```
require(devtools)
install_github("MatthewGilbertUCD/rePhoto",force=TRUE)
require(rePhoto)
```

The *rgdal* R package and dependency package *sp* are needed to output KML files, but not to run the other functions. If you want to output KML files for viewing in Google Earth, you'll need to install these packages from the CRAN website: here (https://cran.r-project.org/)

The Rtools package may be needed for these installations: here (https://cran.rstudio.com/bin/windows/Rtools/).

Having installed these you should now be all set to go.

Step 2: Data input

Input consists of a data frame with three or four rows of data, each row representing a point of reference in a photograph (Fig. 1). The data frame needs three columns representing: the X pixel coordinate of a point of reference in a historical photograph and the corresponding latitude (Lat) and longitude (Long) of the same point of reference.

Fig. 1 An example of a photograph at Butte Lake, Lassen National Park, with unknown camera position. Each circle indicates a point of reference with known horizontal distances between points (derived from the X pixel values found in Microsoft Paint) and known geographical location (latitude and longitude from Google Maps)

For the example in Fig. 1 the X, Lat and Long columns would be:

```
X=c(1817.5,2976,2981.5,4133.5) # corresponding to the X pixel value in the original photograph
Lat=c(40.5626732036708,40.5572747839369,40.5521645296348,40.5538526778492)
Long=c(-121.2881588081880,-121.2863588017280,-121.2840814300800,-121.2866216468020)
```

Combine them into a data frame and view the result:

```
data1=data.frame(X,Lat,Long)
data1
```

```
##          X      Lat      Long
## 1 1817.5 40.56267 -121.2882
## 2 2976.0 40.55727 -121.2864
## 3 2981.5 40.55216 -121.2841
## 4 4133.5 40.55385 -121.2866
```

Note: dput(data1) gives the full resolution of the values. The data can also be entered via a file or clipboard:

```
data1=read.csv(file="yourfilename.csv") # file in the format above
# or
data1=read.table('clipboard') # clipboard in the format above
```

Step 3: Run the core rePhotograph function

The core rePhotograph function evaluates a large grid of possible locations from which the historical photograph could have been taken. Depending upon settings this function can take some time (1 to 10 seconds or more).

```
out <- rePhotograph(data1,PtDensity=1000)
```

'out' is the grid evaluated with values for latitude and longitude, and a series of values that relate to how likely the photograph was taken at that each of those grid locations, for instance here are the first five locations:

```
out[1:5,]
```

```
##          Lat      Long diff1 diff2 diffcomb include
## 1 40.54598 -121.2968     1     1        1       0
## 2 40.54600 -121.2968     1     1        1       0
## 3 40.54602 -121.2968     1     1        1       0
## 4 40.54605 -121.2968     1     1        1       0
## 5 40.54607 -121.2968     1     1        1       0
```

'diff1' corresponds to the Theta difference using the first three points, 'diff2' corresponds to the Theta difference using the first, third and fourth points (if four points were supplied). 'diffcomb' is the geometric mean of 'diff1' and 'diff2' (if you only supply three points 'diff2' is not used). For advanced use (explained below): if the location is excluded due to camera angle or order of points, then 'diffcomb' is assigned a value of 1 i.e. it won't be included in the final search area.

Step 4: View the contours in R plot

```
plot.search.area(data1,out)
```

Search areas are represented by lower values
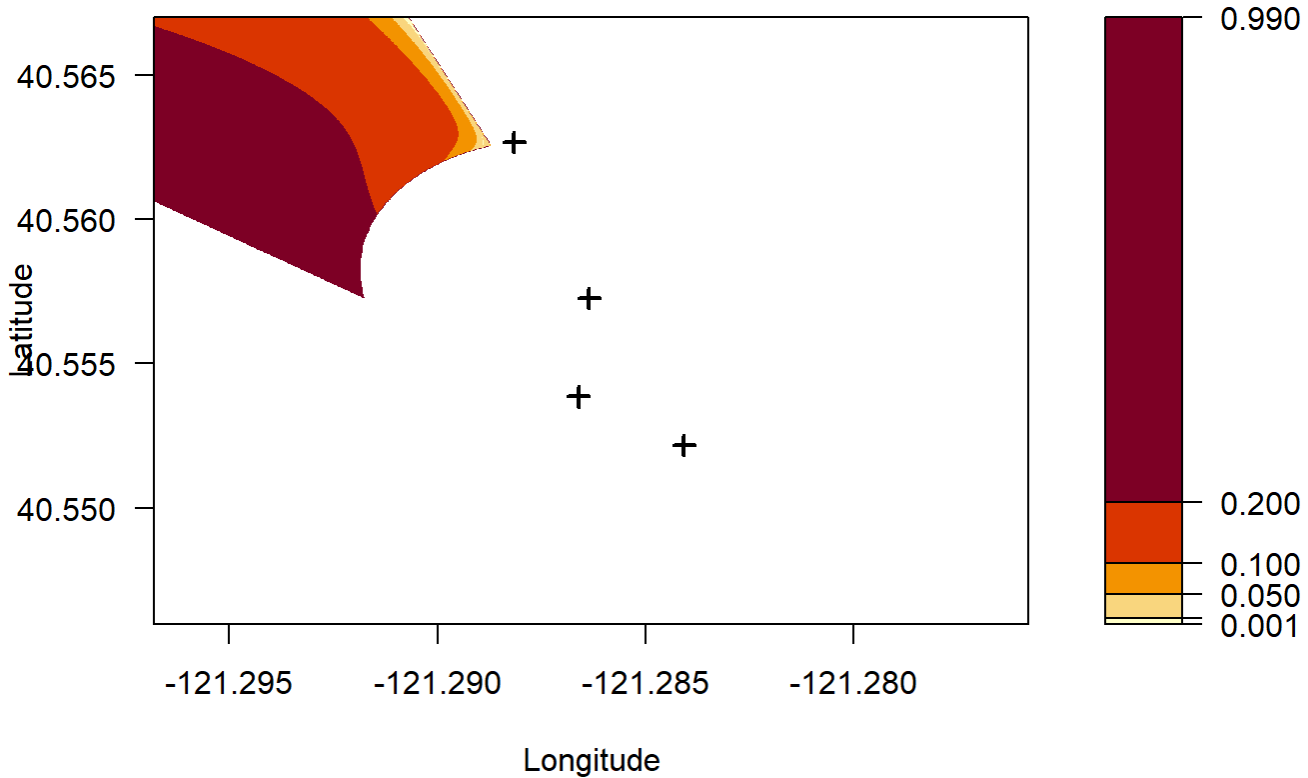
The contour plot provides a visual output, indicating the location of the points of reference (+'s) supplied in 'data1', and contours of areas that vary in Theta difference. A low Theta difference (light colors) represent more likely locations for the original camera position. The white areas are areas excluded based upon advanced criteria such as the order of points of reference in the original photograph and camera lens maximum angle.

Step 5: Output probable search area as KML file

You can create a KML file with a contour showing likely search areas for the camera location of the original photograph:

```
outputKML(out,filename="MyFileName1")
```

```
## Warning in outputKML(out, filename = "MyFileName1"): Warning: Filename in use,
## saved as: MyFileName11234567891011121314
```

The KML file can be opened in Google Earth/Projects/Open or CalTopo/Import and compared to satellite photographs or topographic maps.

Step 6: Make adjustments to settings

The functions run well with the default settings, but not always. In the above example, it is clear that the photograph original location is likely on the north western side of the grid. The area searched was set by the default *scaleToSearch*=2 setting, signifying that a range of latitudes and longitudes should be evaluated at *twice*

the range of the points of reference. We could change this value to change the region searched i.e. a larger region *scaleToSearch*=3.

```
out2=rePhotograph(data1,scaleToSearch=3,PtDensity=1000)
```

Or we could choose to evaluate a specific range of latitudes and longitudes for possible locations by using the *grid* setting:

```
out2=rePhotograph(data1,grid=c(40.550,40.570,-121.300,-121.280),PtDensity=1000)
```

Note:setting *grid* values masks *scaleToSearch*. We can see the new range by replotting the output:

```
plot.search.area(data1,out2)
```


Search areas are represented by lower values
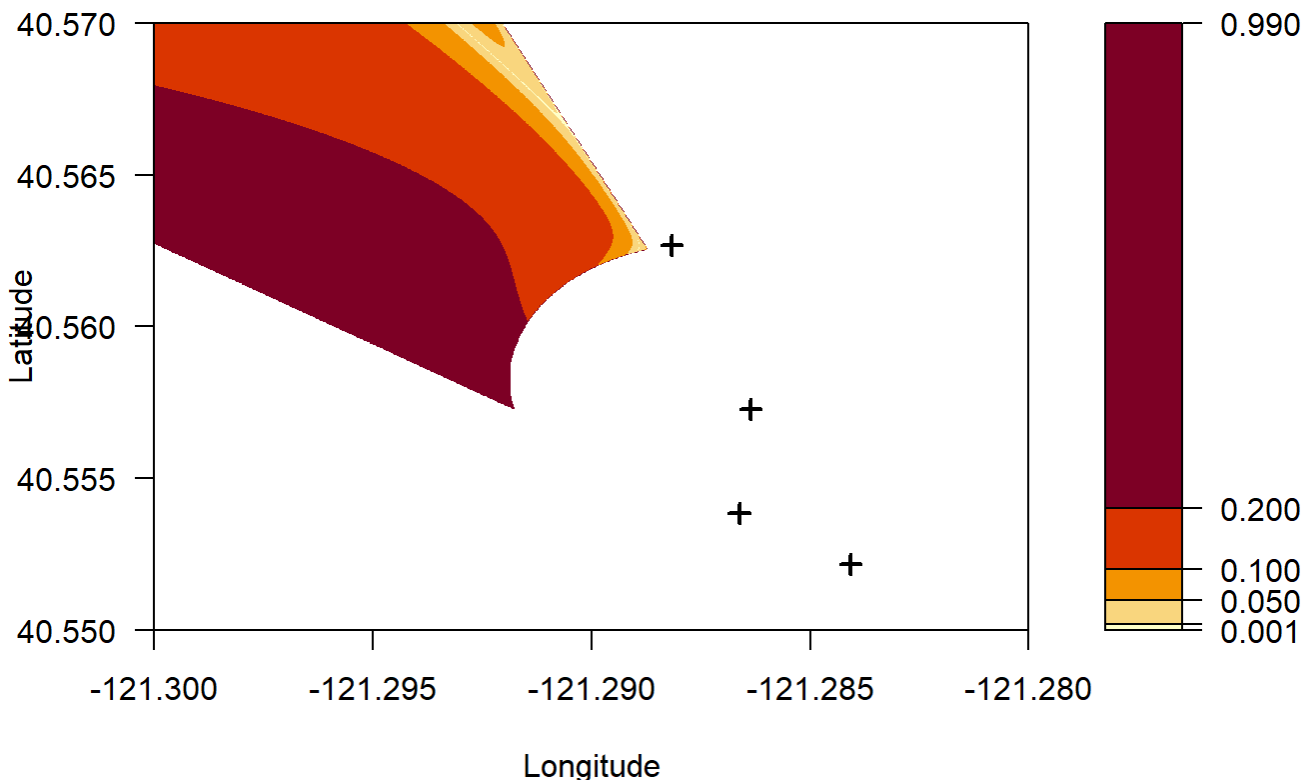
The other default values can be adjusted:
PtDensity changes the resolution of the grid evaluated, but values >1000 can lead to long processing times >10 seconds (default value 1000: a grid of 1000x1000 locations).
UseOrder = FALSE disables the use of the order of the points of reference as a criteria for the search area (default TRUE). In Fig. 1 the order of points a,b,c,d from left to right is distinctive. If a location had a different order then it would be excluded if UseOrder=TRUE.
CameraLens = 45 readjusts the function to only include locations where the points of reference are less than 45 degrees i.e. to be in the frame of a camera lens with 45 degree field of view (default 90 degrees).
Putting these together:

```
out3=rePhotograph(data1,grid=c(40.550,40.570,-121.300,-121.280),
                         PtDensity=1500,
                         UseOrder=TRUE,
                         CameraLens=45)
```

```
## [1] "At PtDensity > 1000 the grid evaluation can take some time"
```

We can replot the analysis, this time using the *extrapoints* input to add reference points, in this case a distinctive feature of the lake shore (will be shown as a purple circle).

```
plot.search.area(data1,out3,extrapoints=data.frame(Lat=40.5661999192325,Long= -121.2909089047998
5))
```

Search areas are represented by lower values
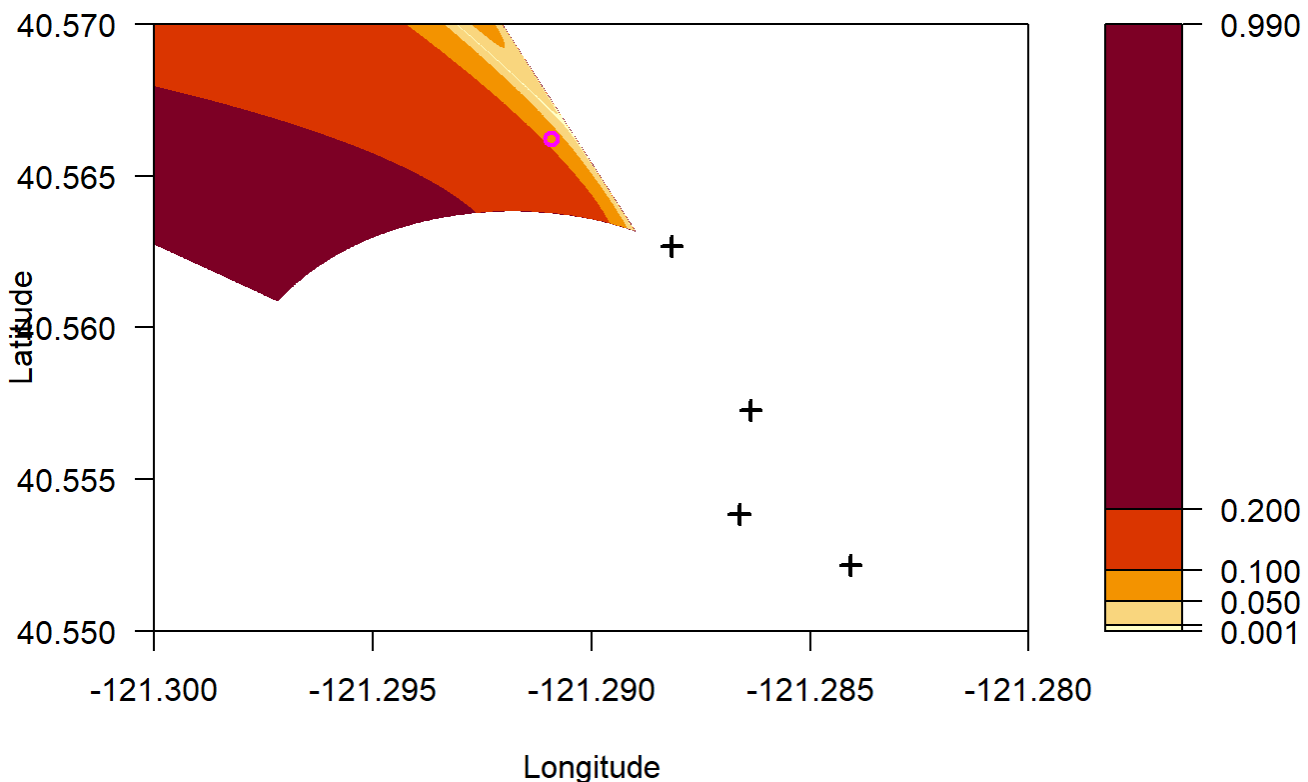
In the plot, it seems that some locations are a very close match (Thetadiff<0.01), thus the threshold could be updated to narrow the region output in the KML contour:

```
outputKML(out3,filename="MyFileName2",Thres = 0.01)
```

```
## Warning in outputKML(out3, filename = "MyFileName2", Thres = 0.01): Warning:
## Filename in use, saved as: MyFileName21234567
```

I encourage you to go see how closely the region in the KML file corresponds with the actual photograph.

Opening KML files

KML files can be opened in a number of software, here are instructions for CalTopo and Google Earth.

CalTopo

1. Go to https://caltopo.com/map.html (https://caltopo.com/map.html)
2. On the left tab select *Import*
3. Choose your KML file in its folder
4. Select *Import*
5. CalTopo doesn't automatically pan to the KML contour lines, so you'll have to navigate to the location the KML file is plotting. e.g. enter some coordinates
6. The right tab on CalTopo allows you to change your base layer to historical maps or satellite photographs, and create printable maps detailing the search area for the camera location

Google Earth

1. Go to https://earth.google.com/ (https://earth.google.com/)
2. *Launch* Google Earth
3. Click on the three horizontal bars in the upper left corner
4. Select *Projects*
5. Select *New Project*
6. Select *Import KML file from computer*
7. Browse the folders to fine your KML file
8. Select *Open*
9. The view will pan to the location of the KML file.

Conclusions

To read more about this technique see: Link to paper will be added when published

The following websites have vast databases of landscape photographs to which this technique can be applied:
https://calisphere.org/ (https://calisphere.org/)
https://www.loc.gov/collections/ (https://www.loc.gov/collections/)
http://vtm.berkeley.edu/#/data/photos (http://vtm.berkeley.edu/#/data/photos)
https://www.nps.gov/media/multimedia-search.htm (https://www.nps.gov/media/multimedia-search.htm)