For this UI run uicaller.py

Set the data folder. *

Do you want the results as screenshots?
If not, they are shwon on a window

Set the dimensions to visualize **

Choose a stat to show
(mean/ sum)

Screenshots resolution

Sampling resolution of
the 2D KDE function

Gamma correction factor for HDR

Colors for color coding:
--Color1 low heat
--Color2 high heat

Size of KDE gaussian kernels

Show the heatmap as
areas with contours

Filter to smooth the heatmap (if no contours)

Filp the heatmap on the X/Y axis

Choose a transformation
on data

Save the KDE densities as CSV files

Run the visualizer
with these parameters

Generate command line
for the visualizer
with these parameters
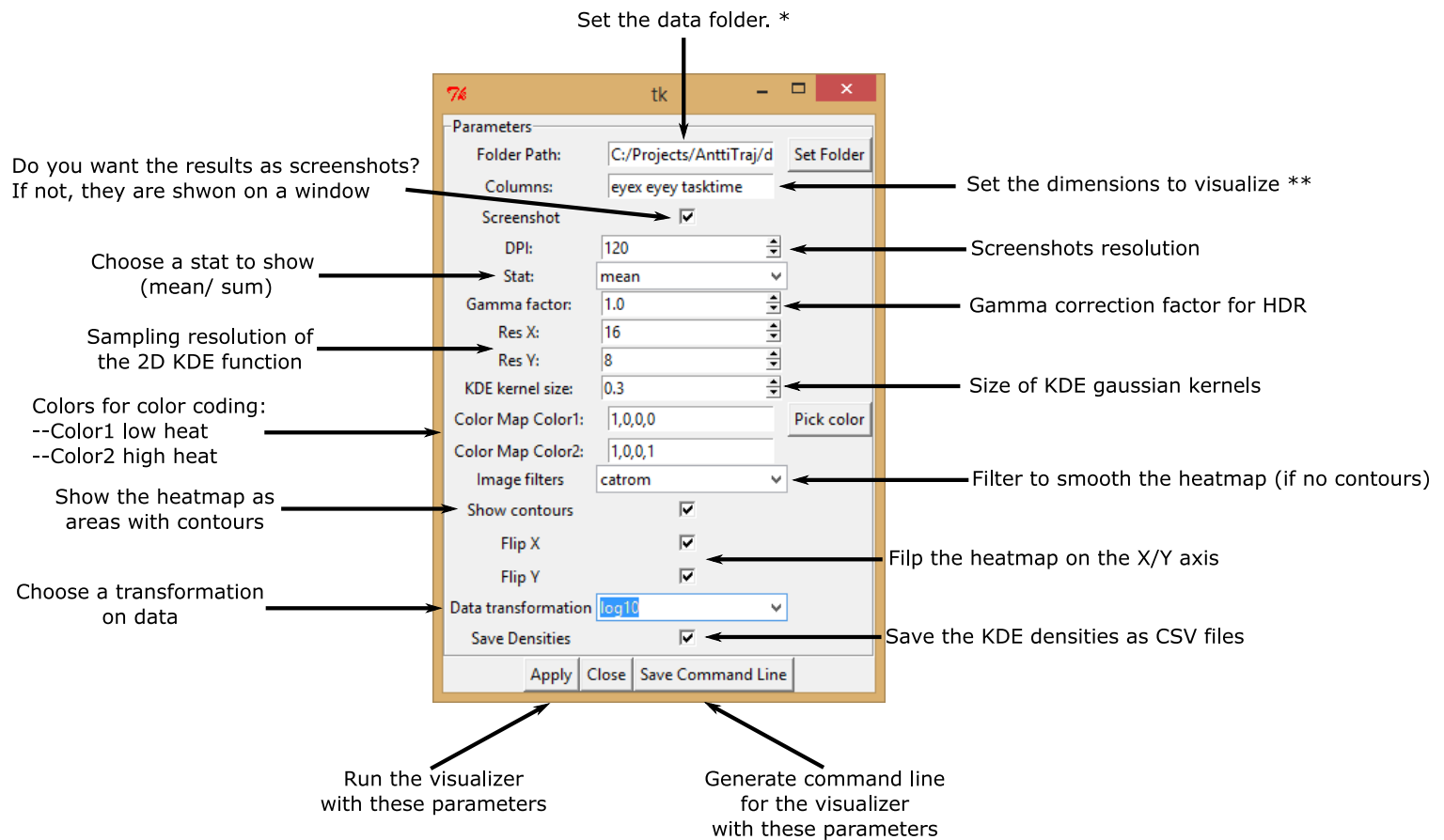
* UI image files should have the same name as the data file but with *.png extension.
The tool reads only CSV files.

** at least 2 column names separated by spaces. No dots or other special characters: eye.x -> eyex

Example of command line generated with the UI in the image above:

python heatmap_gen.py -open C:/Projects/AnttiTraj/data/jussi -vis eyex eyey tasktime -screenshot -res 16 8 -ksize 0.3
-stat mean -gamma 1.0 -dpi 120 -color1 1 0 0 0 -color2 1 0 0 1 iimgfilter catrom -showcontours -flipx -flipy
-transform log10 -saveKDE