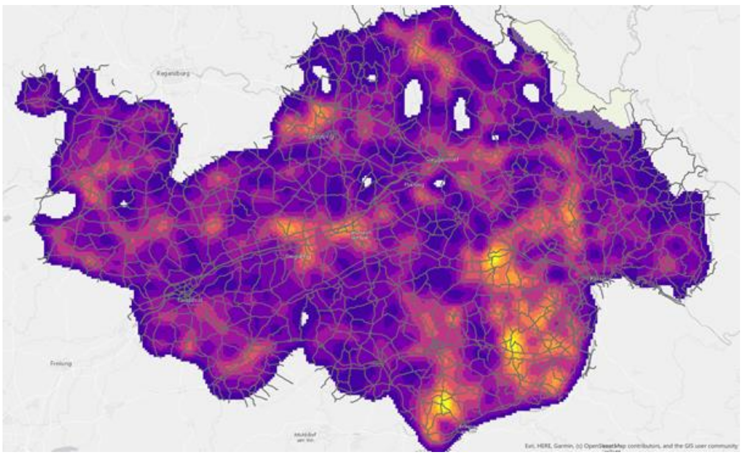
Automated Kernel Density Estimation for extracting Wildlife Vehicle Accidents (WVA)

The idea of this project was to automate the process of filtering out hotpots of wildlife-vehicle accidents alongside roads. Filtering out actual hotspots can help to mitigate wildlife vehicle accidents and may lead to a reduction in accidents. The data was collected by the Bavarian Police for damage compensation reasons and since 2009 also GPS-data is collected alongside animal-species, time, date and some other not so important information. This data set includes 550.000 accidents over a period of ten years all over Bavaria. And the number rises daily.

This automatization process was supposed to be done with a python script in ArcGIS Pro. The analyzation process was supposed to be done with a Kernel Density Estimation. Further information on Kernel Density Estimation and Wildlife Vehicle Accidents can be found in this story-maps: https://storymaps.arcgis.com/stories/wildlife-vehicle-accidents

https://pro.arcgis.com/de/pro-app/latest/tool-reference/spatial-analyst/how-kernel-density-works.htm

But for calculating the Kernel Density one can not only include all the data and make a Kernel Density Estimation. Because this is what an output may look like:



But this is not the product that filters out small-scale accident hotspots, but regional differences which may be strongly dependent on animal and traffic density. So there have to be other solutions for having a better small- scale outcome for realizing projects to mitigate future accidents. A Corine Landcover can be added as a basemap for regarding the preferred landscape of WVAs.

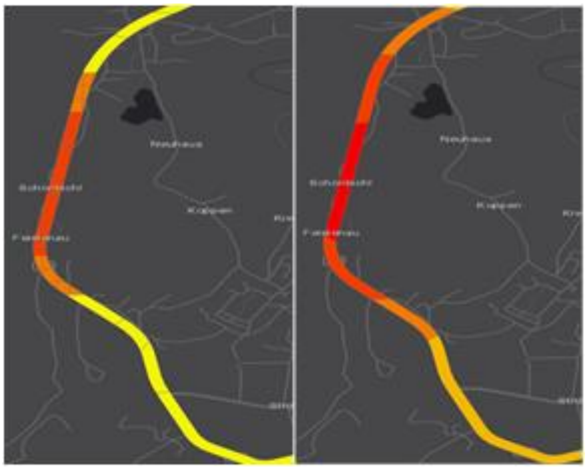
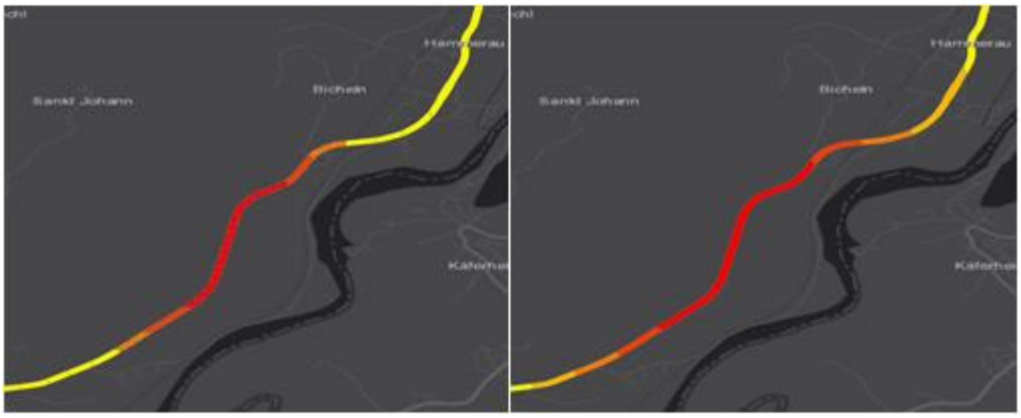


Figure 2 and 3 show on the left hand side the completely automated KDE-result and on the right hand side the wanted result calculated with KDE_Part1 and KDE_Part2 and the manually done Symbology and Reclassification. Both outputs vary in classification values as you can see clearly.



Automatization/ Script

The end result is three scripts, the KDE is split into two parts as the reclassification is better down manually to achieve more accurate results. The third part is the PDF output, which prints out the hotspots.

Steps

- Get input parameters from the user: wildlife accidents, streets, and input name
- Create a 40m buffer of the streets to get the width of the streets (20m taken as a middle value)

```
# Buffered inputStreet Dataset in Diameter D_40 + D_400
# Create 40m buffer (diameter of most streets)
# Input variables
output_buffer = wildlifeGDB + "\\" + "bufferD40"
buffer_distance = "20 meters"
# Buffer Function
arcpy.analysis.Buffer(inputStreet, output_buffer, buffer_distance, dissolve_option = "ALL")

# Create 400m Buffer
output_buffer = wildlifeGDB + "\\" + "bufferD400"
buffer_distance = "200 meters"
arcpy.analysis.Buffer(inputStreet, output_buffer, buffer_distance, dissolve_option = "ALL")
```

- Create a 400m buffer of the streets
- Intersect the 40m buffer of the street with all accidents

```
# Intersect inputStreet with all accidents (inputAccidents) to get the accidents lying on the inputStreet
in_features = ["bufferD40", inputAccidents]
out_feature_class = wildlifeGDB + "\\" + "intersectedAccidents"
join_attributes = "ONLY_FID"
output_type = "point"
arcpy.analysis.Intersect(in_features, out_feature_class, join_attributes)
```

-
- Kernelfunction of the intersection

```
# KERNELFUNCTION
# use the intersected accidents for calculation
in_features = "intersectedAccidents"
# KDE Function (search radius could have to be adjusted depending on street length etc.)
KDE = arcpy.sa.KernelDensity(in_features, population_field="NONE", search_radius=0.01, area_unit_scale_factor="SQUARE_MAP_UNITS",
 out_cell_values="DENSITIES", method="PLANAR", in_barriers="")
```

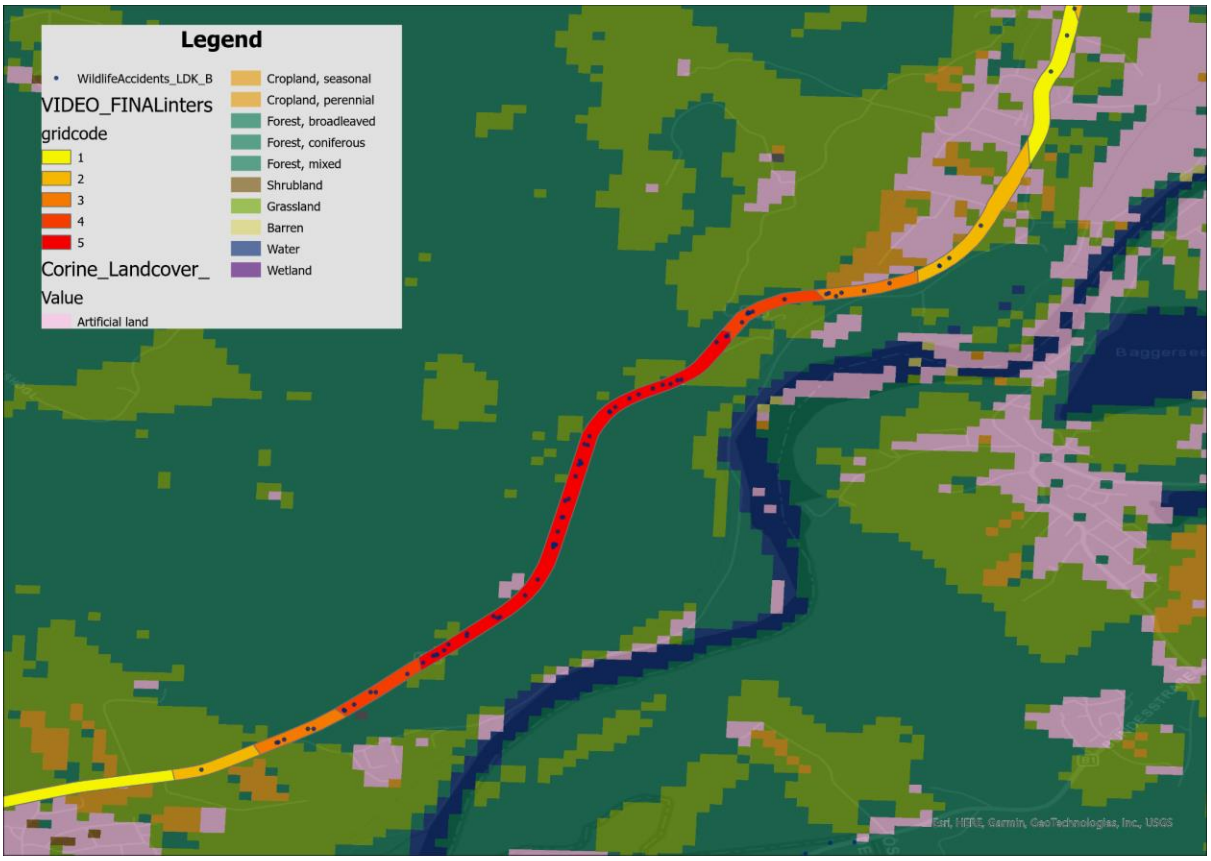
-
- Mask over the calculation with a 400m buffer to limit the hotspots to the streets

```
# set a mask over the calculation
KDE = ExtractByMask(KDE, "bufferD400")
# saves the kernel in the database
KDE.save(wildlifeGDB + "\\" + inputName)
```

Problems/ Further possible development

One of the remaining problems is that the best-suited search radius can differ regarding the study area, the density of streets, and the length of the street. The used search radius of 0.01 fits for the used study area well but when transferring it for different regions with other parameters it could result in unsatisfactory results.

This is what a final Layout may look like:


Legend
WildlifeAccidents_LDK_B
VIDEO_FINALinters
gridcode
1
2
3
4
5
Corine_Landcover_
Value
Artificial land
Cropland, seasonal
Cropland, perennial
Forest, broadleaved
Forest, coniferous
Forest, mixed
Shrubland
Grassland
Barren
Water
Wetland
Baggersee
Esri, HERE, Garmin, GeoTechnologies, Inc., USGS