Annex 1: Alternative method for processing LandScan data into LAEA 1km grid cells

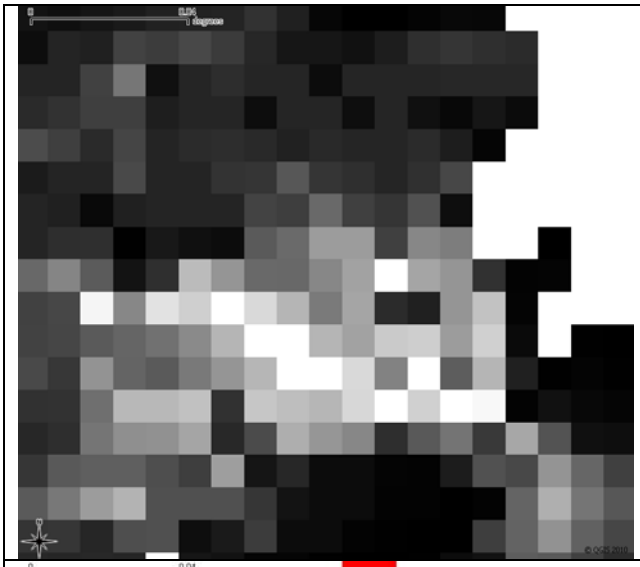
The methodology presented here:

- has a more complete coverage of Europe, as no nask has been applied to the source dataset
- is a better numerical representation of the source dataset (nearly identical population counts before and after processing. Only 4304 people are "lost" in the processing. According to the source dataset, the defined area of interest (UL = -35,75 ; LR = 45,25) has a total population of 967 187 142 people, the relative loss through data processing is hence ~0.000444%)
- only takes ~1 day for the actual data processing, given a working GRASS installation (incl. preparatory work and consistency checks). Part of the script runs could be automated to save some time so that the processing could be reduced to about 0.5 person-days.

Comparison of converted LandScan population data by LAEA 1km grid cells:
ETC-LUSI results (left) and Hermann's results (right)

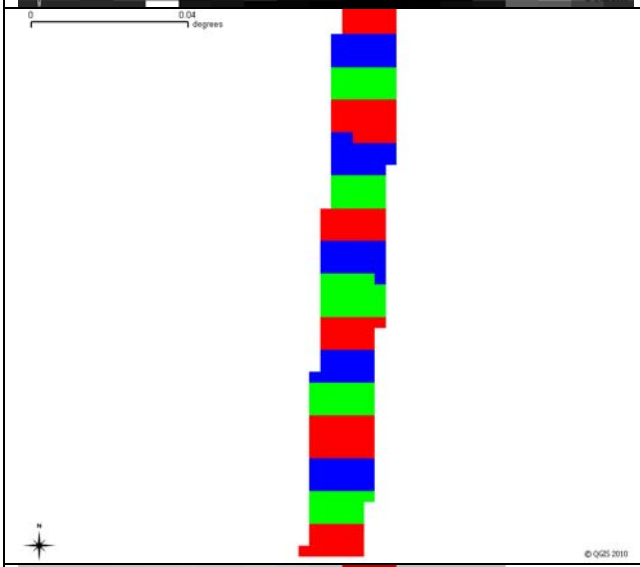
The individual steps of the data processing in a Debian Linux working environment including a GRASS 6.4.0 installation is described on the following pages.

Methodology details: snapshots from raster layers in the Copenhagen area



a) Input dataset: lspop2008_10sec:

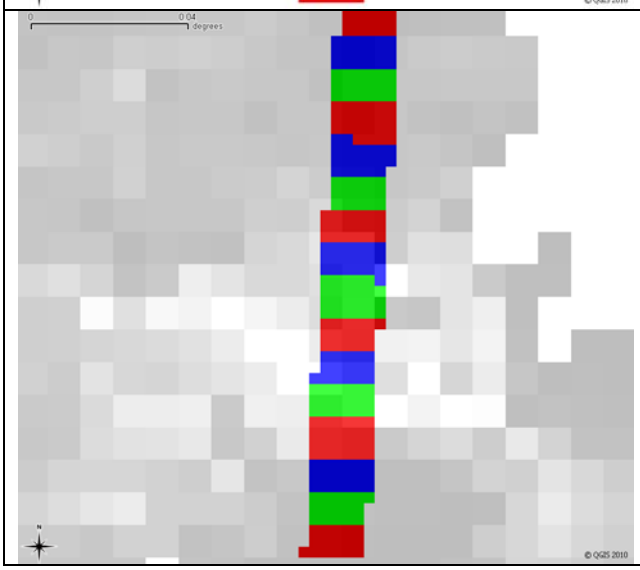
What looks like single cells in the picture on the left are actually groups of 3x3 cells, each individual cell has a size of 10x10 sec. The population value for each of these 10x10 sec cells is set to 1/9 of the original LandScan value which represents the population count for a 30x30 sec cell (30x30 sec is the original resolution of LandScan population data)).



b) Input dataset: laea_grid:

The picture on the left shows selected LAEA grid cells reprojected as raster layer to WGS84, in 10x10 sec resolution.

Originally, the all cells cover 1x1 km in LAEA projection. After nearest neighbor reprojection, the new areas for 1 LAEA cell cover a varying number of 10x10 sec cells, depending on the latitude value. (Typically there are 10..20 target cells, the numbers increase when going south.



Overlay of a) and b) in GRASS

The applied raster processing was:

```
r.statistics   base=laea_grid  \
               cover=lspop2008_10sec \
               method=distribution
```

See above for more details on the output of GRASS' r.statistics module and how it was further processed and converted back to LAEA projection

LAEA grid preparation

 # 0. Create a sufficiently large LAEA grid

```
$ cat laea_grid.awk

# Create LAEA grid in AAIGrid format
# Hermann,  November 2010
BEGIN {

  # Add these header lines
  print "ncols         10000"
  print "nrows         10000"
  print "xllcorner     0"
  print "yllcorner     0"
  print "cellsize      1000"

  # Print Int32 CELLCODE values in AAIGrid format
  # Ex.: CELLCODE = 1kmE2668N1925, value = 126681925
  for (row = 9999; row >= 0; row--)
    for (col = 0; col <= 9999; col++)
      printf "1%04d%04d\n", col, row
}

$ awk -f laea_grid.awk > laea_grid.asc

# Assign CRS EPSG:3035 to the .asc file

$ gdal_translate -of vrt laea_grid.asc laea_grid.vrt -a_srs epsg:3035
```

 # 1. gdal_warp laea_grid.vrt to WGS84 with target resolution 10 x 10 sec

```
# target extent -35 25 75 45, which covers EEA32+Balkan

$ gdalwarp --debug on --config GDAL_CACHEMAX 2000 -wm 500 \
  laea_grid.vrt laea_grid.tif \
  -te -35 25 75 45 -t_srs epsg:4326 \
  -tr .002777777777778 .002777777777778 \
  -co compress=lzw -co tiled=yes -dstnodata 0
```

 # 2. Import laea_grid.tif into GRASS mapset wgs84 (EPSG:4326)

```
GRASS 6.4.0 (wgs84): r.in.gdal input=laea_grid.tif out=laea_grid
```

End LAEA grid preparation

```
### LandScan data preparation ###


# 0. Source data from ORNL (received from Oscar)

$ ls -l lspop2008.???
-rw-r--r-- 1 peifer mtr 3608064000 Jun 26  2009 lspop2008.flt
-rw-r--r-- 1 peifer mtr        160 Jun 26  2009 lspop2008.hdr

# Assign the missing CRS
$ gdal_translate lspop2008.flt lspop2008.vrt -a_srs epsg:4326 -of vrt

# Translate a sufficiently large area into GeoTIFF format
$ gdal_translate lspop2008.vrt lspop2008_laea_extent.tif -ot int32 \
    -projwin -50 75 75 25 -co compress=lzw -co tiled=yes


# 1. Import processed source data into GRASS mapset wgs84

GRASS 6.4.0 (wgs84): r.in.gdal lspop2008_laea_extent.tif
out=lspop2008_laea_extent

# Change resolution to 10 sec

GRASS 6.4.0 (wgs84): g.region rast=lspop2008_laea_extent res=0:00:10


# 2. Calculate a population map in 10 sec resolution

# Generated Int32 values represent original POPCOUNT * 10000

GRASS 6.4.0 (wgs84): r.mapcalc "lspop2008_10sec = \
                     round(10000 * lspop2008_laea_extent / 9 )"


# 3. Check population before and after the calculation

GRASS 6.4.0 (wgs84): r.sum lspop2008_10sec
SUM = 9671828380524  # Population count after calculation (POPCOUNT * 10000)

GRASS 6.4.0 (wgs84): g.region res=0:0:30 # change resolution

GRASS 6.4.0 (wgs84): r.sum lspop2008_laea_extent
SUM = 967187142      # Original lspop2008 population count in laea_extent

# Result: only 4304 people have been lost in a total of 967+ million (error =
~0.000444%)


### End LandScan data preparation ###
```

```
### The actual data processing ###

# 0. Prepare for population counts in 10 sec resolution

GRASS 6.4.0 (wgs84): g.region rast=laea_grid

GRASS 6.4.0 (wgs84): r.statistics base=laea_grid cover=lspop2008_10sec \
                     method=distribution > lspop2008_10sec_by_laea_grid

# The above calculation results into a plain text table with 31+ million records.

# Some sample lines:
109452768       12      1111    9
109452768       12      5555    2
109452768       12      48888   1

# Explanation
# Basemap cat value 109452768 (CELLCODE 1km945E2768N)
# has an area of 12 pixels in WGS84 / 10 sec resolution
# - 9 of these basemap pixels are covered by a population value of 1111
# - another 2 basemap pixels are covered with a population value of 5555,
# - the remaining basemap pixel is covered with a population value of 48888

# The total population for CELLCODE 1km2990E4147N is:
# ( 9 * 1111 + 2 * 5555 + 1 * 48888 ) / 10000 = 6.9997 # rounded to int32 = 7

# 1. Sum up population counts by LAEA CELLCODE

# The calculation is done with sumup.awk; results are printed in AAIGrid format.

$ cat sumup.awk
# Sum up r.statistics method=distribution output
# Hermann, November 2010

BEGIN { FS = SUBSEP = "\t" }

{
  col = substr($1, 2, 4) + 0
  row = substr($1, 6, 4) + 0

  row_min = row_min == "" || row < row_min ? row : row_min
  row_max = row_max == "" || row > row_max ? row : row_max

  col_min = col_min == "" || col < col_min ? col : col_min
  col_max = col_max == "" || col > col_max ? col : col_max

  a[row,col] += $3 * $4
}

# Print Int32 CELLCODE values in AAIGrid format
END {

  # print header lines
  print "ncols         " col_max - col_min + 1
  print "nrows         " row_max - row_min + 1
  print "xllcorner     " col_min "000"
  print "yllcorner     " row_min "000"
  print "cellsize      1000"
  print "nodata_value  -99"

  for (row = row_max; row >= row_min; row--)
    for (col = col_min; col <= col_max; col++)
      print (row,col) in a ? a[row,col] : -99
}
```

```
$ awk -f sumup.awk lspop2008_10sec_by_laea_grid_250m_wgs84 > lspop2008_laea.asc

# Assign CRS EPSG:3035

$ gdal_translate -of vrt -a_srs epsg:3035 lspop2008_grid.asc lspop2008_grid.vrt

# 2. Import results into GRASS, for double checks and country stats

GRASS 6.4.0 (laea): r.in.gdal input="lspop2008_grid.vrt" out="popcount_grid"

# Check total in the area of interest, result has to be divided by 10000

GRASS 6.4.0 (laea): r.sum popcount_grid | awk '{printf"%.0f\n", $NF/10000}'
967182838

# Result: 967 182 838 original population in lspop2008_10sec map, versus
#         967 182 838 population after processing. So nobody lost. Hurray.

# 3. Additional check: population counts in LAEA grid cells, by country

# based on rough country borders from the EEA map template (rasterized to 1km)

GRASS 6.4.0 (laea): cat country_stats.sh

# GRASS shell script for checking values by country
# Hermann, November 2010

# Check if GRASS is running
if [ -z "$GISBASE" ] ; then
  echo "You must be in GRASS GIS to run this program." >&2
  exit 1
fi

# GRASS map to process
if [ -z "$1" ]; then
  map=popcount_grid
else
  map=$1
fi

# Set the region
g.region rast=eeatemplate_countries

printf "%s\t%s\t%s\t%s\n" "ID" "ISO_2DIGIT" "MIN_CNTRY_" "$map"

# List of countries in eeatemplate shapefile
awk 'NR > 1' ../clc/eeatemplate_countries.txt |

# IFS is a tab character
while IFS=" " read -r id code junk name1 name2

do
  printf "%s\t%s\t%s\t" "$id" "$code" "$name1"

  r.mask --quiet -o eeatemplate_countries maskcats=$id
  r.sum --quiet $map | awk -vmap="$map" '
       {printf "%.0f\n", $NF / (map == "popcount_grid" ? 10000 : 1 )}'

done

r.mask --quiet -r

exit 0
```

Results of the consistency checks by country

For all 3 country comparisons (original LandScan data, ETC-LUSI results and Hermann’s results), the same (rough) country borders have been used.

The results show that original Landscan data (column [3]) and Hermann's raster processing match well. ETC-LUSI results (column 4) are OK, numbers are somewhat lower, perhaps because of LEAC masking.

Attributes from EEA map template		[3]Original data	[4]ETC-LUSI	[5]Hermann	[3]-[4]	[3]-[5]
ISO_2DIGIT	MIN_CNTRY_	LandScan.WGS84	LandScan	popcount_grid	Diff1_%	Diff2_%
KS	Kosovo*	1792303	1775809	1792576	0.92	0.02
1	West Bank	2718473	0	2736962	NA	0.68
2	Vatican City	6596	0	0	NA	NA
3	Jersey	70476	0	71711	NA	1.75
4	Guernsey	61979	0	61857	NA	0.20
5	Isle of Man	72835	71571	71781	1.74	1.45
AD	Andorra	82642	81612	82833	1.25	0.23
AL	Albania	3539464	3573744	3538757	0.97	0.02
AM	Armenia	2552081	10903	2555470	99.57	0.13
AT	Austria	8195813	8250786	8198788	0.67	0.04
AZ	Azerbaijan	58979	27	57230	99.95	2.97
BA	Bosnia & Herzegovina	4565962	4558654	4558229	0.16	0.17
BE	Belgium	10440492	10441535	10439628	0.01	0.01
BG	Bulgaria	7222584	7199645	7212686	0.32	0.14
BY	Belarus	9684526	66096	9685993	99.32	0.02
CH	Switzerland	7682877	7660599	7687633	0.29	0.06
CY	Cyprus	762819	752450	760843	1.36	0.26
CZ	Czech Republic	10237824	10266875	10237816	0.28	0.00
DE	Germany	82239677	82053839	82228161	0.23	0.01
DK	Denmark	5164900	5081242	5155840	1.62	0.18
DZ	Algeria	33201350	0	32595284	NA	1.83
EE	Estonia	1258368	1249111	1249004	0.74	0.74
EG	Egypt	80139693	0	63248380	NA	21.08
EH	Western Sahara	296306	0	0	NA	NA
ES	Spain	39489699	39123897	39410024	0.93	0.20
FI	Finland	5093084	5061224	5090752	0.63	0.05
FO	Faroe Is.	38887	0	40026	NA	2.93
FR	France	60885206	60915245	60895243	0.05	0.02
UK	United Kingdom	59628495	59667804	59491930	0.07	0.23
GE	Georgia	4053072	3789	4039993	99.91	0.32
GI	Gibraltar	29017	32252	22853	11.15	21.24
GL	Greenland	551	0	0	NA	NA
GR	Greece	10203050	10056861	10156953	1.43	0.45
HR	Croatia	4203406	4141808	4198573	1.47	0.11
HU	Hungary	9936750	10002130	9934635	0.66	0.02
IE	Ireland	4015034	3981780	4002474	0.83	0.31
IL	Israel	6735625	0	6699932	NA	0.53
IQ	Iraq	18678694	8372	17195474	99.96	7.94
IR	Iran	684725	5999	684946	99.12	0.03
IS	Iceland	261085	264430	259442	1.28	0.63
IT	Italy	56565609	55959576	56327302	1.07	0.42
JO	Jordan	7002639	0	7003109	NA	0.01
LB	Lebanon	3806049	0	3765665	NA	1.06
LI	Liechtenstein	36410	39742	36879	9.15	1.29
LT	Lithuania	3569629	3567678	3578549	0.05	0.25
LU	Luxembourg	495510	485304	496322	2.06	0.16
LV	Latvia	2190046	2182184	2183548	0.36	0.30
LY	Libya	6033736	0	5540638	NA	8.17
MA	Morocco	33725737	9525	33522694	99.97	0.60
MC	Monaco	9539	18735	22176	96.40	132.48
MD	Moldova	4180754	128763	4181919	96.92	0.03
MK	FYR of Macedonia	2071601	2071233	2071302	0.02	0.01
MR	Mauritania	5012	0	0	NA	NA
MT	Malta	370303	367727	369205	0.70	0.30
NL	Netherlands	16520676	16434312	16478071	0.52	0.26
NO	Norway	4098615	4095461	4087513	0.08	0.27
PL	Poland	38421809	38519678	38418258	0.25	0.01
PS	Gaza Strip	1498668	0	1478175	NA	1.37
PT	Portugal	10412683	9861808	10159938	5.29	2.43
RO	Romania	22205558	22128997	22198058	0.34	0.03
RU	Russia	70217411	98889	70203221	99.86	0.02
SA	Saudi Arabia	3532931	0	2181992	NA	38.24
SE	Sweden	8439104	8366266	8424304	0.86	0.18
SI	Slovenia	2001854	2008347	1997541	0.32	0.22
SK	Slovakia	5455670	5410358	5457332	0.83	0.03
SM	San Marino	27823	27345	27060	1.72	2.74
SY	Syria	20686031	87063	20657156	99.58	0.14
TN	Tunisia	9885773	0	9871903	NA	0.14
TR	Turkey	70175681	69581235	69958907	0.85	0.31
UA	Ukraine	45808482	260555	45809961	99.43	0.00
ME	Montenegro	640345	636915	642871	0.54	0.39
RS	Serbia	7407755	7423510	7414359	0.21	0.09