Internal Report

CORILIS (Smoothing of CLC data)

Technical Procedure

Prepared by:
Ferran Páramo

March 2006

European Topic Centre
Terrestrial Environment

Universitat Antònoma de Barcelona
Edifici C – Torre C5
4ª planta
08193 Bellaterrra (Barcelona)
Spain

Contact: etcte@uab.es

European Environment Agency

TABLE OF CONTENTS

1. CORILIS (SMOOTHING OF CLC DATA)

1.1. OVERVIEW

Spatial smoothing has been proved to be a suitable technique for Land Cover data generalization and analysis.

CORILIS, from CORIne and LISsage (smoothing in French), is a methodology developed jointly by the French Environment Institute (IFEN), the Hypercarte Research Group and the French National Institute for Statistics and Economic Studies (INSEE) that provides technical specifications for the smoothing of CORINE Land Cover Data.

The purpose of CORILIS is to calculate “intensities” or “potentials” of a given theme in each point of a territory. A Gaussian type statistical function (called BiWeight) is used to weight this information according to the distance from the considered point in kilometers.

CORILIS results into probability surfaces (varying from 0 to 1) for the presence of a certain CLC class within a smoothing radius. Individual CORILIS layers from a given level can be aggregated to upper levels by simple addition.

First CORILIS tests used hexagon tessellation as the starting generalization for the smoothing process. This first generalization is needed to reduce the weight of raw data. Direct implementation of smoothing algorithms on pixels of 100 m is not feasible due to the heaviness of the computation.

Later tests carried out at ETCTE changed the hexagon tessellation by regular square grids. Last CORILIS implementation has used the recently defined European Reference Grid of 1 x 1 km as generalization framework.

Table 1.— History of CORILIS methodology implementations using various software and data sources. First implementation used hexagon tessellation which was changed to square reference grids in the following ones. The attempt of using raw CLC data (100 m pixels) as input for smoothing resulted into a very time consuming process.

Developer	Land Cover	Tessellation	Software
IFEN: Lacaze (2000)	Raster 250 m	Hexagons (6km base)	SAS
ETCTE: Salvador, R. (2001)	Raster 100 m	100 m pixels	C++
ETCTE: Páramo, F. (2002)	Raster 100 m	3 Km square cells	Visual Basic 6
ETCTE: Zapata, J.L. (2005)	Raster 100 m	1 Km square cells	MatLab 7

1.2. PROCEDURE STEPS

The following steps describe the last version of CORILIS methodology set up at ETCTE during 2005. Main goals of this reviewed methodology are the use of a newly defined European Reference Grid and the migration of the smoothing algorithm to MatLab 7 powerful image processing software.

1) Preparing Land Cover information and creating the Reference Grid

This step is the starting point both for CORILIS and LEAC (Land and Ecosystems Accounting) methodologies. Land Cover information for 1990 and 2000 is combined in a single layer that will be crossed with the Reference Grid in the following step.

Gaps in Switzerland and Andorra were covered using auxiliary data. In the case of Switzerland available Level 2 data for 1990 was integrated in the CLC90 layer. In the case of Andorra CLC90 was in fact available due to the overlapping of Spanish and French datasets. Since no data was available for 2000 neither for Switzerland nor for Andorra we assumed that no changes occurred since 1990 in order to produce continuous smoothed layers. However, it should be taken into account that smoothed data for 2000 in these two countries is not real.

The Reference Grid of 1 x 1 km was created using a modified version of the Generate Fishnet Tool for ArcMap (Robert Nichola's). This customized tool, named "EEA Reference Grid Fishnet Tool for ArcGIS 9" can be downloaded from ESRI download pages. It creates fishnet grids accordingly to the new European Standards.

2) Tabulating Land Cover data with the European Reference Grid 1 x 1 km.

The combination of CLC90 and CLC00 layers was tabulated into the Reference Grid using Tabulate Areas function form ArcView 3.3 Spatial Analyst. The resulting dbf files were imported to MS Access and structured in the following format:

GRIDCODE	CHANGE	AREA
10232	111111	120
10232	211111	25
10232	323111	500
10233	111111	200
10233	323211	75

From this Land Cover Change table we obtained CLC90 and CLC00 stock tables which had the format described below.

GRIDCODE	VALUE-111	VALUE-211	VALUE-323	...
10232	120	25	500	
10233	200	0	75	

GRIDCODE	VALUE-111	VALUE-211	VALUE-323	...
10232	645	0	0	
10233	200	75	0	

3) Export CLC data to text files

Land cover data aggregated by Grid cell was exported to an ASCII file format defined by ESRI for the import/export of raster datasets. This ASCII file contains a header with information about the extent of the geographic layer as well as the resolution for the image pixels:

```
<ncols xxx>
<nrows xxx>
<xllcenter xxx | xllcorner xxx>
<yllcenter xxx | yllcorner xxx>
<cellsize xxx>
{nodata_value xxx}
row 1
row 2
.
.
.
row n
```

We created a text file for each CLC level 3 class and one representing the Total Land surface by grid cell (excluding Sea and Oceans class). This information was used as an input for the MatLab script.

4) Smoothing data in MatLab

The spatial smoothing consists on determining for each point of the land the potential information present in its neighborhood. A Gaussian type statistical function (called BiWeight) is used to weight (w) this information according to the distance from the considered point in kilometers.

$$\mathbf{w = (1 - (d/R)^2)^2}$$

w : weight

d: distance in Km

R: Smoothing radius

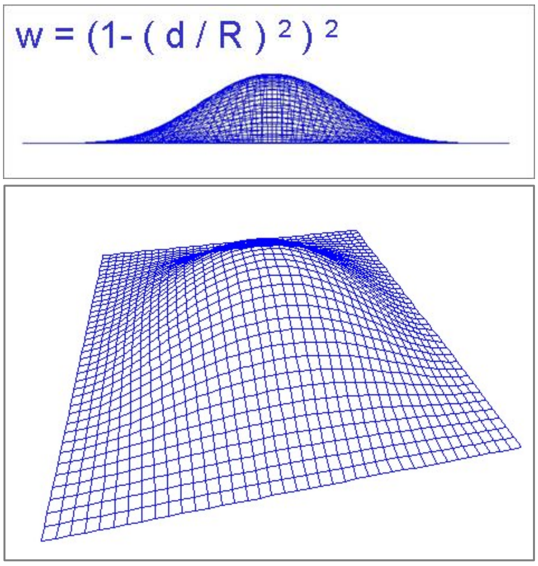

Figure 1.— Gaussian function used to weight land cover values in the smoothing process.

This algorithm was implemented in MatLab software which allows high processing rates working with image datasets (See annex A for details).

5) Import resulting text files to ArcInfo Grid format.

The smoothed data is stored in the same format than the input data. This ASCII files can be imported directly to ArcGIS using the ASCIIGRID command.

Before using this layers they should be masked with a Land/Sea layer to eliminate the border effect. Another mandatory step is to define the projection for the new files. The projection should be the same ETRS 1989 LAEA which is currently implemented in ArcGIS and that was used to define the Reference Grid.

1.3. FURTHER DEVELOPMENT

The bottle neck of the smoothing process in MatLab is to read/write data from text files. We propose an improvement of the script that allow to read and write data from/ to a Database Management System such as MS SQL Server. These functionality is implemented in MatLab through the Database Toolbox.

Another limitation of the smoothing process is that resulting values should be stored as integer. This is because using Floating Point values make output useless due to its size.

In the conceptual side we propose to smooth raw data instead of surfaces aggregated to the Reference Grid. This smoothing would work directly on Boolean values for each of the CLC classes and the results would maintain the spatial resolution of the input Land Cover layers (100 m).

1.4. REFERENCES

Geographic Information Management (2000) *CORILIS Technical Report.*

Hypercarte Project Website: http://www.parisgeo.cnrs.fr/cg/hyperc/

Weber J.L., Páramo F., Breton F., Haines-Young R. (2003) *Integration of geographical and statistical data in the environmental accounting framework; methodological development based on two case studies*

ANNEX A: MATLAB SCRIPTS USED IN CORILIS

The following code is subject to General Public License (GPL). Please refer to European Topic Centre on Terrestrial Environment (etcte@uab.es) for further information.

CorilisInt.m

```
% initial parameters
%%%%%%%%%%%%%%%%%%%%%
% R => Smoothing radius
% AreaFile => Total area file e.g. data/gis/ASCII/TOTAL.txt
% DataFiles => Cell array with file names
% OutputDir => Output directory for smoothing results

function corilis(R, DataFiles, AreaFile, OutputDir);
%% Disable division by zero warning
warning off MATLAB:divideByZero;

%% Starting time
timestart = uint8(clock());
disp(sprintf('Smoothing started at
%d:%d:%d',timestart(4),timestart(5),timestart(6)));

%% Reading header information
f = fopen(AreaFile);
line = fgetl(f);
TotalCols = str2num(line(7:size(line,2))); % Number of columns
line = fgetl(f);
TotalRows = str2num(line(7:size(line,2))); % Number of rows
% Skip xllcorner, yllcorner, cellsize and ndata_value
for i=1:4
    fgetl(f);
end
% Reading Total area values
ftotales = uint8(fscanf(f, '%d',[TotalCols,Inf]));
ftotales = ftotales';
fclose(f);
% Create convolution window of radius R
win = single(createWindow(R));
% Smoothing Total Area file and clearing
stotales = single(conv2(single(ftotales),win,'same'));
clear ftotales;
% For each file:
for i = 1:size(DataFiles,2)
    % Open source file
    f = fopen(DataFiles{i});
    % Open target file
    [path,filename,extension] = fileparts(DataFiles{i});
    fo = fopen([OutputDir filename 'S' extension],'wt');
    % skip first 4 header lines and print them in the output file
    for j=1:5
        fprintf(fo,'%s\n',fgetl(f));
    end
    fgetl(f); % skip last header line
    fprintf(fo,'%s\n','nodata_value -100'); % print last header line with nodata
flag
    fdata = uint8(fscanf(f, '%d',[TotalCols,Inf])); % read source data
    fdata = fdata'; % transpose matrix
    fclose(f); % close source file
    sdata = single(conv2(single(fdata),win,'same')); % perform convolution analysis
    clear fdata; % clear data
    % Divide results by total area smoothed.
    % If a division by zero id done a Nan value is stored: 0/0 = NaN, N/0 = Inf
    datasuavizada = (sdata ./ stotales) * 100;
    clear sdata % clear sdata
```

```
    datasuavizada(isnan(datasuavizada)) = -1; % Replace NaN by -1
    datasuavizada = datasuavizada'; % Transpose the matrix again
    datasuavizada = int8(datasuavizada); % Converting to integer
    % formatting output values
    format = ' %d';
    for j=2:TotalCols
        format = [format ' %d'];
    end
    format = [format '\n'];
    % Writing smoothed data in the output file
    fprintf(fo,format, datasuavizada);
    fclose(fo);
    timenow = uint8(clock());
    disp(sprintf('Smoothing of file %s completed successfully at
%d:%d:%d',DataFiles{i},timenow(4),timenow(5),timenow(6)));
end

%% Finishing time
timeend = uint8(clock());
disp(sprintf('Smoothing completed at %d:%d:%d',timeend(4),timeend(5),timeend(6)));
```

run.m

```
function run()
% DataPrefix => Prefix in the source files ej. data/gis/ASCII1/VALUE_
% InputDir => Folder name for the source files

DataPrefix = 'VALUE_';
InputDir = 'C:\WORKSPACE\WK\INPUT\';
OutputDir = 'C:\WORKSPACE\WK\OUTPUT\';
AreaFile = 'C:\WORKSPACE\WK\INPUT\TOTAL.txt';
R = 5;
% Reading source files
DataFilesStruct = dir([InputDir DataPrefix,'*']);
% Pick only the file names
for i = 1:size(DataFilesStruct,1)
    DataFiles{i} = [InputDir DataFilesStruct(i).name];
end
% pasarlos al suavizado
corilis(R,DataFiles,AreaFile,OutputDir);
```

createWindow.m

```
% Radius => if Radius = 5 it creates a 9x9 window, etc

function window = createWindow(Radius)
for i = 1:Radius * 2 - 1
    for j = 1:Radius * 2 - 1
        dist = sqrt((radio - i)^2 + (Radius -j)^2);
        %distance([Radius,Radius],[i,j]);
        if dist <= radio
            window(i,j) = (1 - (dist/Radius)^2)^2;
        end
    end
end
```