IMPACT OF ERROR

IMPLEMENTATION AND EVALUATION OF A SPATIAL MODEL FOR ANALYSING LANDSCAPE CONFIGURATION

MARIKA WENNBOM

Master’s thesis
Biology-Earth Sciences, 30 HECs

BA 14
2012
Cover illustration:
Preface

This Master’s thesis is Marika Wennbom’s degree project in Biology-Earth Science, at the Department of Physical Geography and Quaternary Geology, Stockholm University. The Master’s thesis comprises 30 HECs (one term of full-time studies).

Supervisors have been Ian Brown, Wolter Arnberg and Maj-Liz Nordberg at the Department of Physical Geography and Quaternary Geology, Stockholm University. Examiner has been Peter Schlyter, at the Department of Physical Geography and Quaternary Geology, Stockholm University.

The author is responsible for the contents of this thesis.

Stockholm, 16 March 2012

[Signature]

Lars-Ove Westerberg
Director of studies
**ABSTRACT**

Quality and error assessment is an essential part of spatial analysis which with the increasing amount of applications resulting from today’s extensive access to spatial data, such as satellite imagery and computer power is extra important to address. This study evaluates the impact of input errors associated with satellite sensor noise for a spatial method aimed at characterising aspects of landscapes associated with the historical village structure, called the Hybrid Characterisation Model (HCM), that was developed as a tool to monitor sub goals of the Swedish Environmental Goal “A varied agricultural landscape”. The method and error simulation method employed for generating random errors in the input data, is implemented and automated as a Python script enabling easy iteration of the procedure. The HCM is evaluated qualitatively (by visual analysis) and quantitatively comparing kappa index values between the outputs affected by error. Comparing the result of the qualitative and quantitative evaluation shows that the kappa index is an applicable measurement of quality for the HCM. The qualitative analysis compares impact of error for two different scales, the village scale and the landscape scale, and shows that the HCM is performing well on the landscape scale for up to 30% error and on the village scale for up to 10% and shows that the impact of error differs depending on the shape of the analysed feature. The Python script produced in this study could be further developed and modified to evaluate the HCM for other aspects of input error, such as classification errors, although for such studies to be motivated the potential errors associated with the model and its parameters must first be further evaluated.
# Table of contents

1. Introduction .......................................................................................................................... 1
2. Background ............................................................................................................................. 2
   2.1. The HCM ......................................................................................................................... 2
   2.2. Potential error sources for the HCM .............................................................................. 4
   2.3. The Swedish agricultural landscape .............................................................................. 5
   2.4. Study area ....................................................................................................................... 5
3. Method and data ...................................................................................................................... 7
   3.1. Software and data ............................................................................................................ 7
   3.2. Implementation of the HCM .......................................................................................... 7
   3.3. Introducing errors .......................................................................................................... 8
   3.4. Running the Python script ............................................................................................ 9
   3.5. Evaluation ...................................................................................................................... 10
4. Results .................................................................................................................................. 13
   4.1. The Python script .......................................................................................................... 13
   4.2. Interpreting the HCM .................................................................................................... 13
   4.3. Qualitative evaluation, landscape scale ........................................................................ 14
   4.4. Qualitative evaluation, village scale ............................................................................. 18
   4.5. Quantitative evaluation – uncertainty analysis .............................................................. 22
   4.6. Quantitative evaluation – sensitivity analysis ................................................................. 23
5. Discussion .............................................................................................................................. 24
   5.1. The HCM – general comments ....................................................................................... 24
   5.2. The HCM – sensitivity to error ...................................................................................... 25
   5.3. Error simulation ............................................................................................................. 25
6. Conclusions ........................................................................................................................... 27
Acknowledgements ..................................................................................................................... 28
References .................................................................................................................................. 29
Appendix A. LCIs and transects for Bjursås, Siljansnäs, Stumsnäs and Våmhus ................. 30
Appendix B. Python script ....................................................................................................... 35
1. **INTRODUCTION**

The Hybrid Characterisation Model (HCM) was developed as a reproducible and qualitative tool for monitoring subgoals of the Swedish environmental goal “A Varied Agricultural Landscape” as a part of the project “Tillämpning av fjärranalys i kulturmiljövården” (“Application of remote sensing in heritage protection”) within the Mistra-supported project “Remote Sensing for the Environment” (Wästfelt, Nordin, et al. 2004). The HCM is a raster-based spatial model aimed at enhancing aspects of agricultural landscapes associated with the historical composition of villages for two spatial scales; the local village scale and the broader landscape scale (Wästfelt and Arnberg 2004). The output of the method is a Landscape Configuration Image (LCI) (Wästfelt and Arnberg Unpublished).

For a model, such as the HCM, to be applicable it is here stated that at least the two following criteria need to be fulfilled:

- a. The model should be evaluated with respect to potential errors and their impact.
- b. The model should be sufficiently documented and/or formalised as an automated implementation.

Criterion ‘a’ concerns all potential errors throughout the execution of the model including different kinds of input error and errors associated with the model and its parameters (Karssenberg and De Jong 2007). Evaluation of model errors (i.e. potential errors in the HCM’s ability to produce valid LCI’s that show the intended landscape values) is partly subjective and requires knowledge within the field of Human Geography and will not be addressed here; this study will instead focus on evaluating the impact of input errors. The input to the HCM is a mosaic of Landsat TM/ETM+ images. Potential sources of input error include sensor noise (including thermal and electronic noise) and/or file errors in the satellite imagery.

To evaluate the impact of input error an appropriate evaluation method needs to be identified. For some simple Boolean overlay models an analytical evaluation method is possible (Heuvelink 1989) but for evaluating impact of input error for more complex spatial models a Monte Carlo approach is suitable where the input data is manipulated to simulate defined amounts of error for a chosen number of input data realisations (Karssenberg and De Jong 2007).

To be able to employ a Monte Carlo approach for evaluating impact of input error criterion ‘b’ needs to be fulfilled. In this case no automated implementation of the HCM was available; hence the implementation and automation of the HCM was included in the study.

Today high quality satellite imagery with global coverage as well as computer power to process large amounts of data are easily accessible which means virtually everyone can download spatial data and use it for a wide range of applications. The easy access and data power also sometimes leads to the sacrifice of sufficient quality assessment for the benefit of rapid development of new applications emphasising the need for thorough investigations of error propagation in geospatial analyses and models.

The aim of this study is to implement and automate the HCM and to investigate the impact of input error associated with sensor noise and other potential errors.
2. BACKGROUND

2.1. The HCM

Cultural qualities of landscapes in Sweden have traditionally been studied by looking at the discrete parts, or zones, and objects that they are made up of e.g. stonewalls, buildings and arable fields. Around the turn of the millennium new needs for describing and monitoring landscapes arose, largely because of the introduction of the Swedish environmental goals, in particular the 13th goal “A varied agricultural landscape”. During the early 2000s some effort was put on investigating the potentials of remotely sensed data for looking at the cultural landscape lead by the Swedish National Heritage Board (Frisk and Moström 2003, Frisk, Moström and Landeholm 2003).

The HCM was designed to detect and emphasise qualities in the modern landscape originating from the village landscape and is based on the human perception of this landscape. The HCM uses a series of focal statistics with differently sized kernels (Wästfelt and Arnberg, Hybrid characterisation of local landscapes 2004). Figure 1 shows a conceptual model of the HCM, for further details of the method see Appendix B.

A static spatial model, such as the HCM, can be described as

\[ Z_{1,m} = f(I_{1,n}, P_{1,l}) \]

where \( Z_{1,m} \) represents the model variables resulting from a function or functions, \( f \), with associated inputs, \( I_{1,n} \), and parameters, \( P_{1,l} \) (Karssenberg and De Jong 2007). Table 1 shows a list of all inputs, functions, parameters and model variables for the HCM as shown in Figure 1. The model variable of main interest for the HCM is the resultant landscape classification product, the LCI. The intermediate output, referred to here as the Internal Class Context (ICC), has been used to identify present and historic sites for shielings (fäbodar) (Wästfelt, Jansson, et al. 2007) but will not be further addressed in this study.

Table 1 List of Inputs, functions, parameters and model variables for the HCM

| Inputs \((I_{1,n})\): | 
|---------------------|---|
| Mosaic of Landsat TM/ETM scenes: | \( I_1 \) |
| Landsat 5 TM 194/18 2000-07-28 | |
| Landsat 7 ETM+ 195/17, 2001-08-15 | |
| Landsat 7 ETM+ 196/17, 2001-07-05 | |

| Functions \((f)\): | 
|-------------------|---|
| The Hybrid Characterisation method (HCM) | \( f \) |

| Parameters \((P_{1,l})\): | 
|-------------------|---|
| Algorithm for unsupervised classification | \( P_1 \) |
| Identity of desired input classes from unsupervised classification | \( P_{2-4} \) |
| Size of kernel for focal analysis | \( P_{5-12} \) |
| Signatures for maximum likelihood classification | \( P_{13} \) |

| Model variables \((Z_{1,m})\): | 
|-------------------|---|
| Internal Class Context (ICC) | \( Z_1 \) |
| Landscape Configuration Image (LCI) | \( Z_2 \) |
Figure 1. Overview of HCM exemplified with pictures showing subarea Stumsnäs.

* Step 1 and 2 were not executed in this study, the classified image and identified land cover classes from the original study (Wästfelt and Arnberg 2004) were used.
2.2. Potential error sources for the HCM

The total error of the final output of the HCM, i.e. the LCI, is the sum of all input errors and model errors and their propagation through the model. Input error is associated with errors in the model input variables \( (I_{1:n}) \) and model error is associated with how well the model, made up of the function \( f \) and the parameters \( P_{1:t} \) describes reality (Karssenberg and De Jong 2007).

As stated earlier, this study will consider errors associated with input data, i.e. mosaic of Landsat TM/ETM scenes \( (I_s) \) which are then propagated through the HCM via the unsupervised classification of the satellite data.

2.2.1. Errors associated with acquisition of satellite imagery

The input for the HCM, as defined by Wästfelt and Arnberg (2004), is a mosaic of three Landsat TM/ETM+ scenes. A spatial subset of the mosaic is shown in Figure 2 where the two Principal Component Analysis (PCA) bands (Figure 2c and d) illustrate noise present in the mosaic. The seam between two of the mosaiced Landsat images is also visible in figure Figure 2c and d.
The image recorded by a satellite sensor is a combination of the actual signal of brightness intensity from the measured land cover and noise, where the noise is a combination of accumulated errors from components of the sensor (e.g. thermal noise, electronic noise) and interference from the atmosphere (Campbell 2006). The relation between signal and noise is measured as the signal to noise ratio (S/N or SNR) (Campbell 2006).

2.2.2. Errors associated with the initial unsupervised classification
The manner in which the initial unsupervised classification and selection of the three land cover classes to be extracted (P1–4) are performed is an important error source for the HCM. Since the selection of land cover classes is user dependant this step will be larger if the user lacks the knowledge of the landscape held by the original authors. Although the error evaluation approach employed here aims at simulating errors associated with sensor noise, the result will also have bearing on the errors associated with the initial unsupervised classification and selection of land cover classes while it gives an indication of the impact of error of different magnitude. This motivates evaluation of high error levels (about 30% and above) that would normally not result from sensor noise.

2.3. The Swedish agricultural landscape
The composition and usage of the Swedish agricultural landscape has always been strongly affected by physical factors such as bedrock and landforms, soil conditions, hydrology, and climate (Ihse 1995). Three main types of Swedish agricultural landscapes can be identified through history; the village landscape that existed for about 1000 years starting around 800 AC, the scattered farm landscape that dominated for around 100 years up to the end of the Second World War and lastly the industrial farm landscape prevailing today, characterised by intensification and homogenisation of land use (Ihse 1995). The village landscape was organised around small villages with enclosed areas of fields and meadows close to the village centre followed by pastures and grazed woodlands further away from the village (Ihse 1995).

2.4. Study area
The study area (Figure 3) is located in the Siljan area, Dalarna county, Sweden. The round shape of lake Siljan is the result of a meteorite impact during the Devonian and the circular depression eventuated in the preservation of Silurian sedimentary rock that does otherwise not exist in the area (SNA 2009), these special geological conditions makes the land around lake Siljan more fertile than would be expected from an area just above the highest shore line (Sporrong, Ekstam and Samuelsson 1995).

The agrarian history of Dalarna county is unique in Sweden largely due to the practice of partial inheritance where the land have been divided between all children resulting not only in a very fragmented field structure but also in villages consisting of a great number of characteristic red houses as every generation built their own home on their inherited land (Juhlin Dannfelt 1929). Efforts were made during the 17th and 18th century to reorganize the field structure, since, at this time, the size and shape of the numerous fields made farming inefficient, but it was not until the 19th century that the state financed reorganisation of fields (storskifte) was undertaken in Dalarna county (Juhlin Dannfelt 1929). The practice of partial inheritance, however, continued even after the reorganisation and today the remnants of a fragmented agrarian landscape, the numerous red houses and remaining shielings (fäbodar) combine into the unique landscape of the Siljan area and are parts of a rich cultural heritage that many consider worth preserving (Sporrong, Ekstam and Samuelsson 1995).
The analysis of the HCM focuses on five subareas namely Bjursås, Boda, Siljansnäs, Stumsnäs, and Våmhus (Figure 4), these areas have previously been analysed using the HCM and described in detail by Wästfelt et al. (2007).

Figure 3. Study area in Dalarnas county, Sweden shown as a false colour composite (R:4 G:3 B:2) of the mosaic of Landsat scenes used as input for the HCM. The study area intersects Orsa, Rättvik, Falun, Leksand and Mora parish and comprises the lake Siljan.

Figure 4. Subareas are, from left to right, Våmhus, Siljansnäs, Stumsnäs, Boda and Bjursås shown with original LCI as background.
3. **Method and Data**

3.1. **Software and data**

All data processing was performed in ArcGIS 9.3 using a Python command script. As described in Figure 1, step 1 (unsupervised classification) and step 2 (manual identification of land cover classes) were not executed in this study. The land cover map used here is the same that was used by Wästfelt and Arnberg (2004) that was derived from a mosaic of Landsat TM/ETM+ images (Table 2 List of data). The input mosaic of Landsat TM/ETM images is, in this study, only used as illustration in Figure 2 and Figure 3.

Table 2 List of data

<table>
<thead>
<tr>
<th><strong>Mosaic of Landsat TM/ETM scenes (input for the HCM, used only for visualisation in this study)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Landsat 5 TM</td>
</tr>
<tr>
<td>Landsat 7 ETM+</td>
</tr>
<tr>
<td>Landsat 7 ETM+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Land cover map</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Land cover map derived from unsupervised classification (Step 2 of the HCM) (Wästfelt och Arnberg, Hybrid characterisation of local landscapes 2004)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Parameters retrieved directly from authors not explictely defined in Wästfelt &amp; Arnberg (2004)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity of desired input classes from unsupervised classification ( (P_{2-4}) )</td>
</tr>
<tr>
<td>Forest = 17</td>
</tr>
<tr>
<td>Transition land = 18</td>
</tr>
<tr>
<td>Cultivated land = 20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>**Size of kernels for focal analysis ( (P_{5-15}) ) **</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPOT5 HRG 053-225 2008-06-06</td>
</tr>
<tr>
<td>SPOT5 HRG 050-225 2005-08-05</td>
</tr>
</tbody>
</table>

3.2. **Implementation of the HCM**

The HCM was implemented as described by Wästfelt and Arnberg (2004) and Wästfelt et al. (2007). It was not possible to exactly recreate the HCM in its original form (Wästfelt and Arnberg 2004), despite using the same input data, the uncertainty is partly due to the use of different software (the original method was carried out using ArcView) and because the technical details of the final maximum likelihood classification was lacking. Nevertheless the HCM as implemented in this study conforms to the method described by Wästfelt and Arnberg (2004) and Wästfelt et al. (2007) and the resulting LCIs are considered close enough to the original for the error evaluation to be valid.

The implementation required some considerable testing of different setups of the method before a result similar to that described by Wästfelt and Arnberg (2004) could be obtained. The basic outlines of the HCM were first implemented as an ArcGIS model that was exported as a python script. The python script was then reorganised to better structure the parameters of the HCM and make them easy to change. Different parameterisation of the HCM python script was then produced and run until a satisfactory result was reached. The main issues that had to be tested out this way were whether the images should be normalised after each processing of the focal statistics, whether to use division or multiplication in step 4 (see Figure 1) and the size of the kernels used in step 5.
3.3. Introducing errors

The error simulation approach used (Figure 5) is a Monte Carlo approach where an increasing amount of random error is generated using a raster containing linearly distributed random values (between 0 and 1) and a conditional statement (Figure 6).

The basis for the error simulation was modelled using ArcGIS model builder and exported as a python script. The HCM python script was then extended with the error simulation script. The combined HCM and error simulation script was further developed so that the script could be iterated for a chosen number of times, were the number and magnitude of error levels could be defined in a list by the user.

![Figure 5. Schematic description of the python script including generating error from a random raster, running the HCM and cross tabulating output and the original. *The blue frame contains the original HCM, with no introduced errors. This part is only run once, while the remaining part of the schematic model is run for every iteration of the script.](image)

3.3.1. Two error evaluation approaches

Since the HCM is based on three land cover classes and each class is handled separately and differently in the first part of the HCM it is interesting to investigate to what degree errors in the different classes have different impact on the final result. Therefore two different approaches to evaluation of impact error where employed in this study; uncertainty analysis and sensitivity analysis and are, according to Jager and King (2004) defined as follows:

- Uncertainty analysis deals with investigating how uncertainties in input data affect the uncertainty in model output.
- Sensitivity analysis deals with investigating which input variable errors the model is most sensitive to.
Impact of error - Implementation and evaluation of a spatial model for analysing landscape configuration

<table>
<thead>
<tr>
<th></th>
<th>Cultivated land</th>
<th>Transition land</th>
<th>Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cultivated land</td>
<td>$1 - x$</td>
<td>$x / 2$</td>
<td>$x / 2$</td>
</tr>
<tr>
<td>Transition land</td>
<td>$x / 2$</td>
<td>$1 - x$</td>
<td>$x / 2$</td>
</tr>
<tr>
<td>Forest</td>
<td>$x / 2$</td>
<td>$x / 2$</td>
<td>$1 - x$</td>
</tr>
</tbody>
</table>

Figure 6. Conditional statement for generating error shown as cross tabulation and as pseudo code where $x$ equals the desired amount of error as a value between 0 and 1 (for actual python statement see Appendix B, row 171).

The uncertainty analysis was executed as described above. For the sensitivity analysis the effect of errors for the three extracted land cover classes ($P_{2-4}$) separately where investigated. To introduce error for one land cover class at a time the conditional statement presented in Figure 6 needs to be altered to:

```python
IF random raster <= x
    IF classified image == forest
        IF random raster < (x/2)
            transition land
        ELSE
            cultivated land
    ELSE
        IF classified image == transition land
            IF random raster > (x/2)
                cultivated land
            ELSE
                forest
        ELSE
            IF classified image == cultivated land
                IF random > (x/2)
                    forest,
                ELSE
                    transition land
            ELSE
                classified image
        ELSE
            classified image
```

3.4. Running the Python script

The process of generating error and running the HCM needs to be repeated a large number of times. Each run, including error simulation and the entire HCM processing generates 1 LCI image and 1 error matrix (table in *dbf-format) comparing the resulting LCI with the original LCI (0% error), the script also generates a number of intermediate results that are per default saved. One LCI and its intermediates add up to about 750 MB of data which makes about 12.5 GB for 17 iterations (0 – 80% error with 5% resolution). Running the script for 17 iterations takes about 5 hours. To effectively run the script, different configurations were saved as separate files and then run sequentially over nights and weekends (by running a *.bat-file containing the different filenames).

The script is designed so that a number of parameters, such as number of error levels and magnitude of error (as mentioned earlier), name of input and output files, identity of land cover classes (defined in step 2 of the HCM) and size of the different kernels can easily be defined by the user.

For further details on the python script see Appendix B.
3.5. Evaluation

3.5.1. Transects – qualitative evaluation
To visualize the changes and identify a breakpoint for the HCM’s vulnerability to errors in input data, transects for 5 subareas were defined. The subareas Bjursås, Boda, Siljansnäs, Stumsnäs and Våmhus correspond to the ones described by Wästfelt et. al. (2004). Each transect stretches from the middle of the village and northward for 2500 m. The transect was created as a line vector, transformed to raster and then to points using, the Howt's tools extension to ArcGIS (Beyer 2004), resulting in 101 points (one point per pixel) per subarea. The points were then used to sample the LCI images and the resulting profiles were plotted as profiles (one profile per error step and area). This way, each subarea gets a specific signature and by looking at the different profiles for an increasing amount of error it’s possible to define when the specific signature of the area is lost and thereby determining the breakpoint for what can be considered an acceptable amount of error in the input data.

3.5.2. Accuracy assessment – quantitative evaluation
The result of the cross tabulation produced from Python script as shown in Figure 5 are two error matrices per iteration. The error matrices were compiled as kappa values (one kappa value per cross table). The KHAT estimate of kappa value is calculated as (Congalton and Green 1999):

\[ \hat{\kappa} = \frac{n \sum_{i=1}^{k} n_{ii} - \sum_{i=1}^{k} n_{i+} n_{+i}}{n^2 - \sum_{i=1}^{k} n_{i+} n_{+i}} \]

Where \( n \) is the total number of samples, \( k \) the number of classes, \( n_{ii} \) the diagonal cells in the error matrix, \( n_{i+} \) the row total and \( n_{+i} \) the column total which for the error matrix for 40% introduced error shown in Table 3 results in a kappa value of 0.389 and for
Table 4 a kappa value of 0.259. To quantitatively evaluate the impact of input error the calculated kappa values were plotted in one graph showing the result of the uncertainty analysis and one showing the result of the sensitivity analysis.

Table 3. Example of error matrix for input (40% error), $\bar{K}$=0.389

<table>
<thead>
<tr>
<th></th>
<th>forest</th>
<th>transition</th>
<th>cultivated</th>
<th>Row total</th>
</tr>
</thead>
<tbody>
<tr>
<td>forest</td>
<td>298735000</td>
<td>99633750</td>
<td>99291875</td>
<td>497660625</td>
</tr>
<tr>
<td>transition</td>
<td>107159375</td>
<td>320220000</td>
<td>106662500</td>
<td>534041875</td>
</tr>
<tr>
<td>cultivated</td>
<td>48611875</td>
<td>48611875</td>
<td>146080000</td>
<td>243303750</td>
</tr>
<tr>
<td>Col. total</td>
<td>454506250</td>
<td>468466625</td>
<td>352034375</td>
<td>1275006250</td>
</tr>
</tbody>
</table>
Table 4. Example of error matrix for LCI (40% error in input data), $\hat{R}=0.259$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>63678125</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>63678125</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>312375</td>
<td>10000</td>
<td>0</td>
<td>52433125</td>
<td>40309375</td>
<td>341875</td>
<td>3125</td>
<td>13809375</td>
<td>36250</td>
<td>0</td>
<td>110095000</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>14131875</td>
<td>524375</td>
<td>3125</td>
<td>0</td>
<td>1389375</td>
<td>210575625</td>
<td>2854375</td>
<td>1250</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>10000</td>
<td>1090795625</td>
<td>111875</td>
<td>0</td>
<td>128125</td>
<td>13181875</td>
<td>625</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>6875</td>
<td>693053750</td>
<td>106897500</td>
<td>0</td>
<td>1276875</td>
<td>29760000</td>
<td>919375</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>10000</td>
<td>26571875</td>
<td>38749675</td>
<td>88750</td>
<td>940000</td>
<td>4281875</td>
<td>13860000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1328125</td>
<td>47605625</td>
<td>21344375</td>
<td>0</td>
<td>138083125</td>
<td>8331250</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>296875</td>
<td>197993125</td>
<td>258750</td>
<td>0</td>
<td>555000</td>
<td>229395000</td>
<td>1750</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>10625</td>
<td>1034375</td>
<td>37256875</td>
<td>16403125</td>
<td>14375</td>
<td>177500</td>
<td>126645000</td>
<td>37418125</td>
<td>47500</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>86078125</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>47500</td>
<td>50508750</td>
<td>4554375</td>
<td>363750</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>62870000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>305000</td>
<td>40000</td>
<td>13554375</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>3534375</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>28384375</td>
<td>2264375</td>
<td>0</td>
</tr>
</tbody>
</table>

0 | 3134375 | 169310625 | 209380125 | 584948750 | 40412500 | 206617500 | 673015625 | 73473750 | 42387500 | 2264375 | 0 | 3889366250 |
4. Results

4.1. The Python script

The python script (Appendix B) comprises all steps of the complete HCM and the simulation of different amounts of error. The input required to run the script is an image showing vegetation classes (in this case an unsupervised classification of Landsat TM images). Before running the script it is also necessary to define which classes in the input image represent the land cover types of interest (cultivated land, transition land and forest). A number of other variables can also be defined or altered in the python script, such as the number of and amount of error (equal to number of iterations). The script has been run for random errors from 0 to 80% equally distributed between the classes and for 0 – 80% error for one land cover class at a time. By changing the variables of the script it would also be possible to investigate other aspects of error effects e.g. what would happen if other land cover classes were used as input or to run the HCM and do the same error simulations as presented here but for other locations.

4.2. Interpreting the HCM

To understand and interpret the effect of introducing error in the HCM it is important to know what the desired original output looks like and what the different classes represent. As described by Wästfelt and Arnberg (2004), the HCM is based on and intended to reflect the zones of a traditional Swedish village. Figure 7 shows the LCI with the transition from cultivated land in the centre of the traditional village to forest at its outskirts illustrated by the legend (colours correspond to those used by Wästfelt and Arnberg (2004)).

![Figure 7. Original LCI with legend](image)
The red areas (class 1), a concentration of cultivated land, represent the centre of a village and are typically surrounded by a pink border of class 2 areas which represent areas of mixed composition but with a strong influence of cultivated land. From the centre and out the red areas should, for a traditional village structure, be followed by some or all blue and then green classes. The landscape shown in Figure 7 is dominated by class 3 (not counting the “other” class shown in black) as illustrated by the pie chart to the right. Class 3 lies between concentration of cultivated land and concentration of transition land. The least common LCI-class in the study area is concentration of forest.

4.3. Qualitative evaluation, landscape scale
Figure 8 and Figure 9 illustrate the effect of increased amounts of linearly distributed random error (white noise) in the classification used as input for the HCM. Since the error simulation is conducted so that only the three input classes are affected (a forest pixel will either be wrongly classified as transition land or cultivated land, never as other) the general structure of the area will be preserved even with 100% error, hence the lake Siljan is clearly distinguishable in all error maps.

Up to 60% error the LCI classes generally decrease (are replaced by the black “other” areas) or classes that are close to the original class and most village centres (red areas in the original LCI) can still be discerned as pink class 2 areas although all red class 1 areas have disappeared at 40% as shown in Figure 10. The first class that disappears as the amount of error increases is class 10, concentration of forest, which is gone after 30% error. For around 30% error the general landscape character of the original LCI is still well represented (Figure 9) although a bit paler (the classes of concentrated land cover have been replaced by mixed classes). Looking at Figure 10 it is also noticeable that class 3, the dominating class in the original LCI, decreases rapidly at 30% after a slight initial increase while class 11 is the only class that increases with increasing error. For up to 60% the respective groups of classes (red classes, blue classes and green classes) are still in the right location. From 65% error and above LCI classes start to appear in new locations. The three classes representing concentrated land cover (1, 5 and 10) are among the first classes to disappear.
Figure 8. Effect of introducing linearly distributed random error (0 – 35 %) in input to HCM for Siljan area shown as maps and pie charts.
Figure 9. Effect of introducing linearly distributed random error (40 – 75 %) in input to HCM for Siljan area shown as maps and pie charts.
Figure 10. Distribution of LCI classes for increasing amount of introduced error.
4.4. Qualitative evaluation, village scale

To look at the effect of error for individual villages five subareas corresponding to those studied and described by Wästfelt et al. (2004) have been studied closer.

The transect plots (Figure 12, Figure 13 and Appendix A) illustrate the transition of LCI classes from the centre of the villages and 2500 m north. The transect plots provide a unique signature for each subarea and is a helpful tool for determining what amount of error that alter the specific character of the village. Although the different subareas differ in character, all transects start changing around 10% introduced error. The signature for the original LCI is shown as a dashed line in the transect plots for each subarea for reference. For all subareas below it is noticeable how the mixed classes, especially 2 (pink), 3 (light blue) and 11 (light green), preserve their shape and replace the vanishing core areas within them as the amount of error increases. The graph in Figure 10 shows that the amount of these three classes are relatively constant for the error span described in the subarea examples below (0 – 30%). After 30%, however, the light blue class decreases while the other two stay constant up to 80%.

4.4.1. Boda

The LCI image (Figure 11) shows the village centre of Boda as an area elongated in the north-south direction. Because of this a transect stretching from the middle and east- or westwards would have resulted in a quite different signature. The Boda area together with the Siljansnäs area differs from the other subareas in that very few LCI classes are represented in the transect plots (Figure 12) and in that, as mentioned above, the village centre have elongated shape, the class representing concentrated cultivated land (red) and its closest mixed class (pink) cover a large portion of the zoomed in subarea (about 5000 by 5000 m). In the Boda area the red and pink classes cover almost the entire north-south stretch in the middle of the area. Worth noticing is the “dual core” or “8” shaped character of the Boda village centre and that the two red areas of concentrated cultivated land are separated already at 10% error. The quarry, seen in the Spot image (Figure 11), shows up in the LCI image as a patch of class 2 (pink).

4.4.2. Bjursås

The Bjursås area is in the LCI (Appendix A) characterised as an area that seems to correspond well to the traditional village model with a centre (red) surrounded by the different mixed classes in consecutive order that show up as distinct steps in the transect plot. Six LCI classes are represented along the transect in the original image. Class 4, a mixed class close to concentration of transition land, appear as two bigger and several smaller patches around the village centre but since the transect only shows the northern direction from the centre they are not included in the transect plots.

The LCI images in Figure 16 shows that the village centre, class 1, decreased with increased error and disappeared at 30% error. At 10% error the general character was still preserved, as shown by the transect plots, but at 20% the village centre was considerably diminished and the different mixed forest classes in the small green patch in the outskirts of the area have changed place which is shown in the transect plots.

The LCI images and the two classes represented in the transect plot (Figure 12) show that the character of the Boda area starts changing at 10% and is considerably changed at 20%. Even though the village centre is larger than for Bjursås all red class 1 areas have disappeared at 30% error. The patches of concentrated forest in the eastern part of the first three LCI images are all gone at 20%. 
Impact of error – Implementation and evaluation of a spatial model for analysing landscape configuration

4.4.3. Siljansnäs
The LCI for the Siljansnäs area (Appendix A) shows only four major classes with a village centre that is larger than any of the other areas. The large red and dark blue patches of class 1 and 5 indicate that the area is characterised by large homogenous areas of cultivated land and transition land. The general character is relatively well preserved for up to 20% error as shown in the LCI and transect plots in Appendix A. The airstrip visible in the upper left corner of the Spot Image shows as a distinct “tip” which is still discernible at 30% introduced error.

4.4.4. Vâmhus
The Vâmhus area, like Boda, has a north-south stretch hence few classes are crossed by the transect (Appendix A) and represented in the transect plots. As for the other subareas the character is preserved up to 10%, altered at 20% error and at 30% almost all classes showing concentrated land-cover types (1, 5 and 10) are gone. The very small green patch of the mixed class 6 surrounded by an area of dark blue class 5 in the lower left corner increases considerably in size when error is introduced.
Figure 12. Input, LCI and transect through LCI from centre and 2500 m north for increasing error, Boda
Impact of error - Implementation and evaluation of a spatial model for analysing landscape configuration

Figure 13. Input, LCI and transect through LCI from centre and 2500 m north for increasing error, Stumsnäs
4.4.5. Stumsnäs

The Stumsnäs village centre is the smallest of the five subareas and has the largest coherent area of concentrated transition land (Figure 14). The transect stretches through the collected village centre (red) through a section of class 2 (pink), a very small portion of class 3 (light blue) and on to a rather large are of mixed forest classes. The LCI images in Figure 13 show that the character of the area is preserved for up to 10% error but is considerably changed at 20% where the red city centre is smaller with a jagged perimeter and the core of concentrated transition land in the lower left corner of the image is broken up in smaller patches. At 30% error the red class 1 is almost entirely gone and the dark blue (concentrated transition land) just remains as one small patch. The green forest classes have, at 30% error, turned from darker to lighter green colours where the darker colours are closer to concentrated forest. The transect plots of shows that the first small changes in the LCI image have appeared at 5% error and further changes (shown as spikes in the plot) show up for 10% but first at 20% the signature curve starts deviating from the reference curve.

![Figure 14. Transects Stumsnäs from center and 2500 m north, on (a) Spot image and (b) original LCI](image)

4.5. Quantitative evaluation – uncertainty analysis

Kappa values were calculated for all degraded input and LCI images (as compared to the correct image). Figure 15 shows the kappa values for the input image (only the three classes used in the method) and LCI image per error introduced with the kappa value limit between strong, moderate and poor classification (Landis and Koch 1977).

From Figure 15 it is apparent that the kappa values of the LCI images decrease faster with introduced error than the input images but do not decrease linearly. The kappa values for the LCI images are within the “strong” range for up to 10% introduced error and within the “moderate” range for up to 30%. The series of kappa values cross the input series just before 60% introduced error.
4.6. Quantitative evaluation – sensitivity analysis

Figure 16 shows the HCM’s response to introduced errors for one input class at the time. The slope of the respective input series indicates the total amount of the class in the input image. Error in the least common class, cultivated land, has the least impact on the result. Since the amount of forest in the input images are about the same as the amount of transition land (the input series for these two classes are very similar) comparing the curves for these two classes shows that errors in the transition land class has a larger impact than errors in the forest class. Comparing the results presented in Figure 16 to the flowchart of the HCM (Figure 1) presented on page 3 we see that the impact of error for the respective classes seems to correspond to the radii (R) of step 4; the smaller the focal window the larger the impact of error for that class.
5. DISCUSSION

5.1. The HCM – general comments

As stated before it was not possible to exactly recreate the final maximum likelihood classification step of the HCM (and consequently the result is not identical to the original). The HCM as implemented here conforms to the description provided in the published literature and additional comments from the original authors Wästfelt and Arnberg (2004) and Wästfelt et al. (2007). Hence it is deemed similar enough to the original to be valid. That said, for the HCM to be repeatable over time the algorithm of the final maximum likelihood classification needs to be firmly defined e.g. by defining crisp thresholds for each output class.

Two parts of the HCM are user dependant. The first manual step is the identification of the three land cover classes of interest. The other is the size of the kernels in the HCM that was established through field observations in the Siljan area. Thus the HCM as implemented here is only valid for this particular area and input data and not generally applicable. If the HCM was to
be made applicable for other areas and other input data clear guidelines for the choice of land cover classes and kernel sizes would be needed.

Except for the two manual steps of the HCM the final maximum likelihood classification would have to be standardised, e.g. by defining specific signatures for the separate LCI classes to make the HCM applicable for comparison between different areas and over time.

5.2. The HCM – sensitivity to error

5.2.1. Shape of villages
When comparing the transect series for Boda (Figure 11) and for Stumsnäs (Figure 14) their difference in shape is apparent. The Stumsnäs village like Bjursås and Siljansnäs have a collected and relatively round village centre while the village centre Stumsnäs has an outstretched elongated shape. As shown the two parts of the elongated Stumsnäs village centre are separated already at 10% error which illustrates that LCI representation of villages with an elongated, narrow, shape are more sensitive to error than LCI representations of round villages. This effect depends on that the HCM is based on circular moving windows (focal statistics). Furthermore, this questions the validity of the landscape pattern that the HCM method is attempting to represent.

5.3. Error simulation

5.3.1. Error simulation method results in overrepresentation of some classes
The error simulation method used in this study is focused on error from a producer’s point of view i.e. the method controls the percentage of pixels for one land cover class in the original image that will be correctly classified in the simulated error image. Errors from a producer’s point of view can be quantified as producer’s accuracy (Congalton and Green 1999):

\[
\text{producer's accuracy}_i = \frac{n_{ji}}{n_{+j}}
\]

It is also possible to look at error from a user’s point of view, i.e. the percentage of the pixels of one land cover class in the error image that are actually that land cover class in the original classified image. Errors from a user’s point of view can be quantified as user’s accuracy (Congalton and Green 1999):

\[
\text{user's accuracy}_i = \frac{n_{ii}}{n_{i+}}
\]

Which means that producer’s accuracy will be directly controlled by amount of introduced error (x in Figure 6, page 9 showing the conditional statement for error simulation) but user’s accuracy will depend on the distribution of the different land cover classes in the original image. Consequently Figure 6 is really only correct if \(n_a = n_b = n_c\) but since this is not the case here (see Figure 16 for distribution of the three land cover classes of interest in the original classified image) the distribution of error will really be as shown in table 5. The effective result of this is an over representation in the error image of the land cover class(es) that cover the smallest area in the original image, i.e. an over representation of pixels incorrectly classified as cultivated land.
Table 5. Actual result of simulating error using the conditional statement described in Figure 5, where \(n_x\) is the number of pixels for each class.

<table>
<thead>
<tr>
<th></th>
<th>Original classified image (reference)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cultivated land (a)</td>
</tr>
<tr>
<td>Cultivated land</td>
<td>(n_a(1-x))</td>
</tr>
<tr>
<td>Transition land</td>
<td>(\frac{x}{n_a})</td>
</tr>
<tr>
<td>Forest</td>
<td>(\frac{x}{n_a})</td>
</tr>
<tr>
<td>Row sums</td>
<td>(n_a(1-x) + \frac{x}{n_b} + \frac{x}{n_c})</td>
</tr>
<tr>
<td>User's accuracy</td>
<td>(\frac{n_a(1-x)}{n_a(1-x) + \frac{x}{n_b} + \frac{x}{n_c}})</td>
</tr>
<tr>
<td>Column sums</td>
<td>(n_a)</td>
</tr>
<tr>
<td>Producer's acc.</td>
<td>(1-x)</td>
</tr>
</tbody>
</table>

The effect of the over representation of cultivated land in the simulated error will most likely not have a large impact on the result of this study at least not for simulated error of about 60% and below. Because of the methods three (or for forest two) focal statistics steps (effectively different kinds of filtering) the wrongly classified pixels would not have much impact on the HCM, i.e. it is likely that a similar result as the one shown here would be obtained if the error simulation procedure would simply erase pixels from the different land cover classes and not incorrectly classify them as one of the other land cover classes. After 60% simulated error, though, new patches of LCI classes starts to appear and the over representation of pixels incorrectly classified as cultivated land will most likely have an impact.

The error simulation of this study strives to simulate errors associated with sensor noise but can also be seen as a rough estimation of the general trends for impact of input error. As long as potential model errors (i.e. errors associated with the function \(f\) and the parameters \(P_{\text{L,L}}\)), like the choice of input land cover classes, size of kernels and specification of supervised classification have not been evaluated it is not fruitful to further investigate the impact of input error. If further simulation of input error using the method and Python script presented here would be desired a suitable next step would be to alter the script to enable simulation of classification errors taking membership probability vectors (Goodchild and Wang 1989) and, in some manner, spatial auto correlation in to consideration (Goodchild, Guoqing and Shiren 1992, Canters, de Genst and Dufourmont 2002). Nevertheless, it is not unreasonable to assume that uncorrected sensor noise may create an error that is propagated through the classification stage introducing an amplified random error when classification inaccuracies are considered. Classifications rarely achieve >90% accuracy and therefore errors on the order of 10-30% are not inconceivable. For errors greater than 30% poor sensor performance and poor classification accuracy would be required (conceivable with older sensors such as Landsat MSS or highly
noisy sensors such as Hyperion). Lower spatial resolution data or less homogenous landscapes may also promote increases in errors.

The output of the HCM is also highly depending on the manual identification of land cover classes (Step 2) why it would also be relevant to investigate how the choice of which land cover classes that will represent cultivated land, transition land and forest affect the result. This, as opposed to above mentioned suggested developments of the error simulation method which require some considerable work to incorporate, could easily be achieved by altering the existing parameters of the Python script.

6. CONCLUSIONS

In this study a spatial model for characterising landscapes, the HCM, is implemented, automated and evaluated with respect to input errors associated with sensor noise.

The HCM was designed to study landscapes in two different scales i.e. characterising entire landscapes as well as individual villages. Chapter 4.3 shows the effect of introduced error when looking at the whole study area and Chapter 4.4 shows the effect of introducing error on a village scale. The general landscape pattern can be discerned for up to 60% error and that the first LCI class disappears at 35% error.

This study shows that the LCI is reliable for up to 10% sensor noise when looking at the character of single villages. The character of the entire landscape is still relatively well preserved and all original classes are still represented in the LCI for up to 30% error. After 30% error the LCI pattern is thinned out, largely due to a decrease of class 4. The groups of classes (red classes, blue classes and green classes) remain in the right position for up to 60% error where after the groups of classes start switching places.

The breakpoints identified through qualitative evaluation correspond very well to the breakpoints for kappa values shown in the qualitative evaluation which indicates that kappa is a valid measurement of quality for the HCM and similar methods.

From a user point of view this means that when looking at the LCI for the Siljan area there is no risk that an area that shows up as concentration of agricultural land will in reality be forest or transition land but when looking at a single village small differences is character might be caused by error.

The comparison of errors for one land cover class at a time shows that errors in succession land have a larger impact than errors in forest and that errors in agricultural land has the smallest impact. It is likely that the impact is governed by the size of the largest focal window in the HCM, the smaller the kernel the bigger the impact.
ACKNOWLEDGEMENTS

I would like to thank

Wolter Arnberg for everlasting faith in my potentials, I am sorry you could not be there to see the final result.

Maj-Liz Nordberg for support and encouragement during the initial part of the project.

Ian Brown for dedication to my project and inspiring ideas during the final part of my thesis.

Helle Skånes for reading and giving valuable comments.

Elisabeth Sturesson for proofreading and motivation.

Petra for being my sister in arms.

Moa and Andrew for well needed distraction and countenance back in the T407 days.

My colleagues, friends and family for listening and making it possible.

Örjan for being there at all times.
Impact of error - Implementation and evaluation of a spatial model for analysing landscape configuration

REFERENCES


Appendix A. LCIs and transects for Bjursås, Siljansnäs, Stumsnäs and Våmhus

Transects Bjursås from center and 2500 m north, on (a) Spot image and (b) original LCI

Transects Siljansnäs from center and 2500 m north, on (a) Spot image and (b) original LCI
Impact of error - Implementation and evaluation of a spatial model for analysing landscape configuration

(a) Transects Våmhus from center and 2500 m north, on (a) Spot image and (b) original LCI
<table>
<thead>
<tr>
<th>Error</th>
<th>Input (classified image)</th>
<th>Result (LCI)</th>
<th>Transect, result (LCI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td><img src="image-0" alt="Image" /></td>
<td><img src="image-1" alt="Image" /></td>
<td><img src="image-2" alt="Image" /></td>
</tr>
<tr>
<td>5%</td>
<td><img src="image-3" alt="Image" /></td>
<td><img src="image-4" alt="Image" /></td>
<td><img src="image-5" alt="Image" /></td>
</tr>
<tr>
<td>10%</td>
<td><img src="image-6" alt="Image" /></td>
<td><img src="image-7" alt="Image" /></td>
<td><img src="image-8" alt="Image" /></td>
</tr>
<tr>
<td>20%</td>
<td><img src="image-9" alt="Image" /></td>
<td><img src="image-10" alt="Image" /></td>
<td><img src="image-11" alt="Image" /></td>
</tr>
<tr>
<td>30%</td>
<td><img src="image-12" alt="Image" /></td>
<td><img src="image-13" alt="Image" /></td>
<td><img src="image-14" alt="Image" /></td>
</tr>
</tbody>
</table>

Input, LCI and transect through LCI from centre and 2500 m north for increasing error, Bjursás
Impact of error - Implementation and evaluation of a spatial model for analysing landscape configuration

<table>
<thead>
<tr>
<th>Error</th>
<th>Input (classified image)</th>
<th>Result (LCI)</th>
<th>Transect, result (LCI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td><img src="image1" alt="Input" /></td>
<td><img src="image2" alt="Result" /></td>
<td><img src="image3" alt="Transect" /></td>
</tr>
<tr>
<td>5%</td>
<td><img src="image4" alt="Input" /></td>
<td><img src="image5" alt="Result" /></td>
<td><img src="image6" alt="Transect" /></td>
</tr>
<tr>
<td>10%</td>
<td><img src="image7" alt="Input" /></td>
<td><img src="image8" alt="Result" /></td>
<td><img src="image9" alt="Transect" /></td>
</tr>
<tr>
<td>20%</td>
<td><img src="image10" alt="Input" /></td>
<td><img src="image11" alt="Result" /></td>
<td><img src="image12" alt="Transect" /></td>
</tr>
<tr>
<td>30%</td>
<td><img src="image13" alt="Input" /></td>
<td><img src="image14" alt="Result" /></td>
<td><img src="image15" alt="Transect" /></td>
</tr>
</tbody>
</table>

Input, LCI and transect through LCI from centre and 2500 m north for increasing error, Siljansnäs
Marika Wennbom

Input (classified image) | Result (LCI) | Transect, result (LCI)

<table>
<thead>
<tr>
<th>Error</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>5%</td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
<tr>
<td>10%</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>20%</td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
</tr>
<tr>
<td>30%</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Input, LCI and transect through LCI from centre and 2500 m north for increasing error, Våmhus
Appendix B. Python script

```python
# Error_and_HCM_for_ArcGIS_9.3.py
# is a python command script for ArcGIS 9.3
# but will run for ArcGIS 10.0 if:
# * imported system modules under "#Import system modules" are changed to
#   "import arcpy"
# * Extension check under "#Check out any necessary licenses" is changed to
#   "arcpy.CheckOutExtension("spatial")"
# * Create the Geoprocessor object gp = arcgisscripting.create()" and
#   gp.AddToolbox("C:/Program/ArcGIS/ArcToolbox/Toolboxes/Spatial...
# are removed
# * all calls to gp (e.g. "gp.GetRasterProperties") throughout the script are
#   changed to
#   arcpy.gp (e.g. "arcpy.gp.GetRasterProperties")

# Created by Marika Wennbom
# from scripts generated by ArcGIS/ModelBuilder
# Latest major update: 2011-12-04
# Minor cosmetic updates: 2012-02-27
# As part of Master's thesis at
# Department of Physical Geography and Quaternary Geology,
# Stockholm University
# This script performs a hybrid classification method (steps 1 - 3 in list below)
# developed by
# Anders Wastfelt
# Department of Human Geography
# Stockholm University
# and
# Wolter Arnberg
# Department of Physical Geography and Quaternary Geology,
# Stockholm University
# and simulates error in input data (step A-D + step 4 in list below)
# Error_and_HCM_for_ArcGIS_9.3.py is a template to be ***completed*** with the
# following variables:
# work_dir (working directory, e.g. "J:\Master_thesis\Method")
# study area (name of study area e.g. "Siljan")
# series (series ID for particular execution), e.g. "1"
# class_img (classified land cover image containing 12 land cover classes , e.g. work_dir="Resource/Resource/imgs_class.txt")
# cons_val (image containing constant value 1, e.g. work_dir="Resource/cons val_1_2")
# ref_img (original LCI without introduced error, e.g. work_dir="Result/"+studyarea+ "/0/mlc"
# signature (signature file for maximum likelihood classification, e.g. work_dir="Result/"+studyarea+ "/signature.gsg")
# tL (to be completed with letter and amount of errors, items in tL equals
# number of iterations, e.g. ["a", 0.00], ["b", 0.05], ["c", 0.10], ["d", 0.15],
# ["e", 0.20], ["f", 0.25], ["g", 0.30])
# This version is updated with the option to include more than one class from the
# original image
# per class in analysis (agri, succ, frst).
# As stated in the list of required parameters above:
# This version of the script requires that a reference image with no error is available
# in folder
# work_dir="Result/"+studyarea+"/0/mlc. It also requires "a constant value image" which
```
is basically a raster comprised of only 1's. This image is used when calculating zonalmax and should really be generated by the script (but isn’t at this point). The script also requires a predefined signature file for the final classification.

# Loop for x n.o. iterations with different amount of error (see tL):
A. Get extent from input image
B. Generate random raster
C. Generate classified error matrix for constructed error image and export
d. Create error matrix as dBase-file.

1. Loop for three classes (agricultural land, succession land, forest)
   a. Extract class and reclass (1/0)
   b. Run focal window, large radius (specific r for each class)
   c. Normalize with max value (full window)
   d. Run focal window, small radius (specific r for each class)
   e. Normalize with max value (full window)
   f. Multiply normalized images
   g. Apply final focal window (specific r for each class)
   h. To visualize contextual distribution i.e. amplitude of impact on landscape for each class

2. Composite the three resulting images into an RGB image
3. Perform IsoClus and Maximum Likely Hood Classification on RGB image.
4. Create an error matrix for the resulting image and a reference image and export as dBase-file.

# Import system modules
import sys, string, os, arcgisscripting
# Create the Geoprocessor object
gp = arcgisscripting.create()
# Check out any necessary licenses
gp.CheckOutExtension("spatial")
# Load required toolboxes...
gp.AddToolbox("C:/Program/ArcGIS/ArcToolbox/Toolboxes/Spatial Analyst Tools.tbx")
gp.AddToolbox("C:/Program/ArcGIS/ArcToolbox/Toolboxes/Data Management Tools.tbx")
gp.AddToolbox("C:/Program/ArcGIS/ArcToolbox/Toolboxes/Conversion Tools.tbx")
# Turn on overwrite
gp.overwriteoutput = 1
# Input images: (to be defined by user, se explanation above)
work_dir = "" 
studyarea = ""
series = ""
classimg = ""
cons_val = ""
ref_img = ""
signature = ""
# Loop list, iterations with different amount of errors (tL[1]):
T.L = [["a", 0.00]]
for i in tL:
   # Paths:
   out_path = work_dir+"Result/"+studyarea+"/"+series+"/"+i[0]+"/"
tmp_path = out_path+"temp/"
   # Intermediates:
   tmp_path = tmp_path+"random"
   # Output:
   felbild = out_path+"felbild"
### Description: Create directory if it does not exist...
if not os.path.exists(out_path):
    print "creating new directory: "+out_path
    os.makedirs(out_path)
print "creating new directory: "+tmp_path
os.makedirs(tmp_path)

### Description: Create Random Raster with same extent as input...
print "creating random raster with same extent as input..."
print "extent:"
#A. Process: GetRasterProperties
xmin = str(gp.GetRasterProperties (classimg, "LEFT"))
ymin = str(gp.GetRasterProperties (classimg, "BOTTOM"))
xmax = str(gp.GetRasterProperties (classimg, "RIGHT"))
ymax = str(gp.GetRasterProperties (classimg, "TOP"))
print "xmin= ", xmin
print "ymin= ", ymin
print "ymax= ", ymax
print "xmax= ", xmax
# concatenate coordinates to extent string
extent = xmin+" "+ymin+" "+xmax+" "+ymax
#B. Create Random Raster
gp.CreateRandomRaster_sa(random, "", classimg, extent.replace(\.", ")")
print "random raster created"
#C. Description: Uses the original image and the random image and a
# conditional expression
# to create "a misclassified image".
print "executing conditional statement to create image with error = "
   "+str(i[1]*100)+"...
conditional = "CON (+random <= +str(i[1])+", CON (+classimg == 17,
   CON (+random < +str(i[1]/2)+", 18, 20), CON (+classimg == 18,
   CON (+random > +str(i[1]/2)+", 20, 17), CON (+classimg == 20,
   CON (+random > +str(i[1]+1)/2)+", 17, 18), +classimg)),"+classimg")"
gp.SingleOutputMapAlgebra_sa(conditional, felbild)
print "image created"
#D. Create an error matrix for the resulting image and a reference image
# Process: Tabulate Area...
print "tabulating area"
error_felbild = tmp_path+"error_class"
gp.TabulateArea_sa(classimg, "VALUE", felbild, "VALUE", error_felbild, "25")
# Process: Table to dBASE (multiple)...
print "exporting crosstab to dBase file"
gp.TableToDBASE_conversion(error_felbild, out_path)
#parameters:
#list_class = [classname, reclass spec, small r, large r, final r]
#classes from land cover map to be used as input can be changed here, note
#that simulated #errors have already been generated and are not
#connected to this list. Connecting #error generation to this list would be a
#practical improvement of the script
re_cult = ["1 0;2 0;3 0;4 0;5 0;6 0;7 0;8 0;9 0;10 0;11 0;12 0;13 0;14 0;15 0;16 0;17 0;18 0;19 0;20 1"]
re_trns = ["1 0;2 0;3 0;4 0;5 0;6 0;7 0;8 0;9 0;10 0;11 0;12 0;13 0;14 0;15 0;16 0;17 0;18 1;19 0;20 0"]
re_frst = ["1 0;2 0;3 0;4 0;5 0;6 0;7 0;8 0;9 0;10 0;11 0;12 0;13 0;14 0;15 0;16 0;17 1;18 0;19 0;20 0"]
list_cult = ["cultivated", re_cult, "3", "10", "0"]
list_trns = ["shrubbery", re_trns, "3", "10", "25"]
list_frst = ["forest", re_frst, "5", "15", "0"]
L = [list_cult, list_trns, list_frst]

# 1. Loop for three classes (cultivated land, transition land, forest)
for item in L:
    print "entering loop for item "+item[0]+"..."
    # Intermediates:
    reclass = tmp_path+item[0]+"_0_1"
large_r = tmp_path+item[0]+"_r"+item[3]
large_r_dummy = tmp_path+"dummy_r"+item[3]
large_r_max = tmp_path+item[0]+"_r"+item[3]
large_r_norm_0 = tmp_path+item[0]+"_r"+item[0] +"_0"
large_r_max = tmp_path+item[0]+"_r"+item[2]+"_max"
large_r_dummy = tmp_path+"dummy_r"+item[0]
large_r_norm = tmp_path+item[0]+"_r"+item[2]+"_1"
large_r_norm_0 = tmp_path+item[0]+"_r"+item[2]+"_1_0"
multiply = tmp_path+item[0]+"_mul"
masked = tmp_path+item[0]+"_masked"
final_r = tmp_path+item[0]+"_r"+item[4]+"_1"
final_r_dummy = tmp_path+"dummy_r"+item[4]
final_r_max = tmp_path+item[0]+"_r"+item[4]+"_max"
final_r_norm = tmp_path+item[0]+"_r"+item[4]+"_norm"
final_r_dummy = tmp_path+"dummy_r"+item[4]
icc = tmp_path+item[0]+"_icc"

# a. Extract class and reclass (1/0)
print "extracting +item[0]+"...
gp.Reclassify_sa(felbild, "VALUE", item[1], reclass, "NODATA")

# b. Run focal window, large radius (specific r for each class)
print "Executing focal statistics with large window..."
gp.FocalStatistics_sa(reclass, large_r, "Circle "+item[3]+" CELL", "SUM", "DATA")
gp.FocalStatistics_sa(cons_val, large_r_dummy, "Circle "+item[3]+" CELL", "SUM", "DATA")

c. Normalise with max value (full window)
print "Normalising (large window)..."
gp.SingleOutputMapAlgebra_sa("zonalmax(+cons_val+, "+large_r_dummy+")", large_r_max)
gp.SingleOutputMapAlgebra_sa("float(+large_r/) / + large_r_max, large_r_norm, large_r_max")
print "Setting NoData to zero (large window)..."
gp.SingleOutputMapAlgebra_sa("con (ISNULL ("+large_r_norm+"), 0, "+large_r_norm+"), large_r_norm_0, large_r_norm")

# d. Run focal window, small radius (specific r for each class)
print "Executing focal statistics with small window..."
gp.FocalStatistics_sa(reclass, small_r, "Circle "+item[2]+" CELL", "SUM", "DATA")
gp.FocalStatistics_sa(cons_val, small_r_dummy, "Circle "+item[2]+" CELL", "SUM", "DATA")

e. Normalise with max value (full window)
print "Normalising (small window)..."
gp.SingleOutputMapAlgebra_sa("zonalmax(+cons_val+, "+small_r_dummy+")", small_r_max)
gp.SingleOutputMapAlgebra_sa("float(+small_r/) / + small_r_max, small_r_norm, small_r_max")
print "Setting NoData to zero (small window)..."
gp.SingleOutputMapAlgebra_sa("con (ISNULL ("+small_r_norm+"), 0, "+small_r_norm+"), small_r_norm_0, small_r_norm")

# f. Multiply normalised images
print "Multiplying small and large window images..."
gp.SingleOutputMapAlgebra_sa(large_r_norm_0 +"*" + small_r_norm_0, multiply, large_r_norm_0,";", small_r_norm_0)

# g. Apply final focal window (specific r for each class)
if item[4] != "0":
    print "Executing focal statistics with final window ("+item[4]+" cells)..."
gp.FocalStatistics_sa(multiply, final_r, "Circle "+item[4]+" CELL", "SUM", "DATA")
gp.FocalStatistics_sa(cons_val, final_r_dummy, "Circle "+item[4]+" CELL", "SUM", "DATA")
Impact of error - Implementation and evaluation of a spatial model for analysing landscape configuration

```python
# h. Normalise with max value (full window)
print "Normalising (final window)..."
gp.SingleOutputMapAlgebra_sa("zonalmax("+cons_val+"," +final_r_dummy+"), final_r_max")
print "Setting NoData to zero (final window)..."
gp.SingleOutputMapAlgebra_sa("con (ISNULL (" +final_r_norm+"), 0," +final_r_max+"), icc, final_r_norm)"

else:
print "No value for final window for class " +item[0] +", multiplying " +multiply+ " with 255 and renaming it " +item[0] +"_icc."
gp.SingleOutputMapAlgebra_sa(multiply+" * 255", icc, multiply)

# 2. Composite the three resulting images into an RGB image
# Process: Composite Bands...
composite = tmp_path+"composite"
print "Compositing bands...
" gp.CompositeBands_management(L[0][5]+";"+L[1][5]+";"+L[2][5], composite)

# 3. Perform IsoClust and Maximum Likely Hood Classification on RGB image.
# Process: Maximum Likelihood Classification...
mlc = out_path+"mlc"
confidence = tmp_path+"confidence"
# gp.IsoCluster_sa(composite, signature, "12", "20", "20", "10")
print "Performing Maximum Likelyhod Classification"
gp.MLClassify_sa(composite, signature, mlc, "0.0", "EQUAL", "", confidence)

# 4. Create an error matrix for the resulting image and a reference image
# Process: Tabulate Area...
error_mlc = tmp_path+"error_mlk"
print "tabulating area"
gp.TabulateArea_sa(ref_img, "VALUE", mlc, "VALUE", error_mlc, "25")

# Process: Table to dBASE (multiple)...
print "exporting crosstab to dBase file"
gp.TableToDBASE_conversion(error_mlc, out_path)
```