TreeD version 0.8

An Image Processing Application for Single Tree Detection

Kenneth Olofsson

Arbetsrapport 106 2003
Working Paper 106 2003

SWEDISH UNIVERSITY OF
AGRICULTURAL SCIENCES (SLU)
Department of Forest Resource
Management and Geomatics
S-901 83 UMEÅ
Phone: 090-786 58 25 Fax: 090-14 19 15, 77 81 16

ISSN 1401-1204
ISRN SLU-SRG-AR--106-SE

TABLE OF CONTENTS

Acknowledgements

This project is financed by the Kempe foundation and the Heureka research programme. I would like to thank Olle Hagner, Johan Holmgren, Steve Joyce, Professor Håkan Olsson and Jörgen Wallerman at the Remote Sensing Laboratory at SLU in Umeå for valuable discussions.

Introduction

The TreeD © software, version 0.8, is an application that loads an aerial or satellite image and detects trees by template matching. The software is developed at the Remote Sensing Laboratory, Swedish University of Agricultural Sciences (SLU) in Umeå, Sweden. This is beta software and the methods and algorithms are continuously being improved upon. Please report any errors you find in the software to the Remote Sensing Laboratory.

The algorithms for tree detection used in the application are based on a PhD thesis by Richard James Pollock (1996), University of British Columbia, Canada. The application is build upon two software libraries, **Intel® Image Processing Library, IPL** © and **wxWindows** ©. The IPL is used for image processing and wxWindows is used as a graphical user interface.

The report contains three major sections, a manual, a method description and a software description. The manual is for someone, without any prior knowledge of template matching, who wants to run the software. The method and software descriptions are for someone that wants to build a similar application as TreeD or as a support for in-house development at the Remote Sensing Laboratory.

PART I

Manual

Tutorial / Manual

The TreeD 0.8 application is available at the Remote Sensing Laboratory, SLU and is primarily used as a research tool for single tree detection. The software runs on a Microsoft® Windows 2000 workstation. The **application** binary depends on the **Intel® Image Processing Library, IPL** © DLLs and consequently they need to be put into the same folder as the program. All of these binaries can be found at SLU.

The input to the application is an aerial/satellite image (central or orthogonal projection), a tree library, information about the camera and solar positions, and a path to a directory to put the results in. The current status of the input variables can be viewed and changed before starting the correlation of the image. The output from the application consists of three text files, **status.txt**, **treelist.txt** and **probable_treelist.txt**. If there are old files with these names at the result directory they will be overwritten. To save a new batch you can either rename the old files or use a new result directory.

The application assumes that the terrain is fairly flat and that the camera is positioned approximately in a nadir view. If these conditions are not fulfilled the accuracy of the positioning and detection of the trees will decrease.

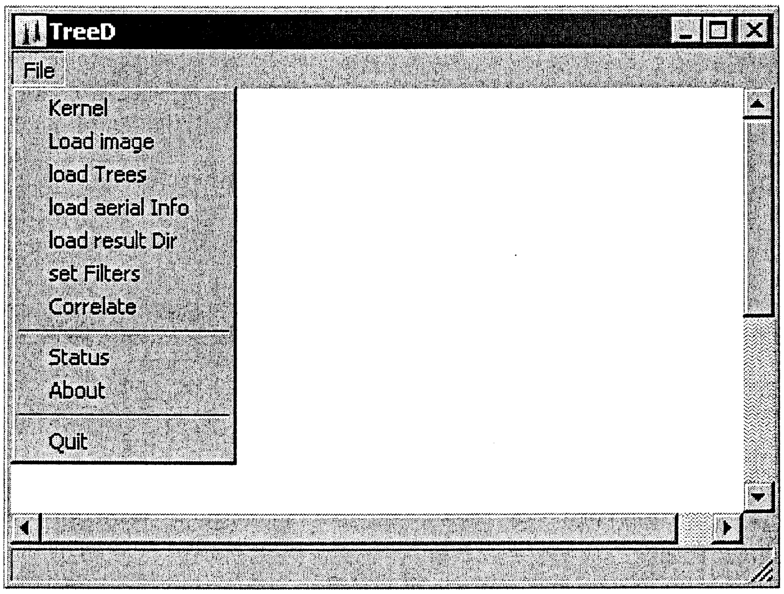

Figure 1 *Drop list for the input to the application. The aerial image and the tree and aerial information can be imported from here. When all input data is set, press correlate to start template matching a new aerial image.*

Setting up the environment

The TreeD 0.8 application and DLLs are available at the Remote Sensing Laboratory, SLU. The binary is compiled for the Microsoft® Windows 2000 platform. To set up the environment, put the exe file**, tree_d.exe**, in a directory together with the **Intel® Image Processing Library, IPL** © DLLs,

Cpuinf32.dll,
ipl.dll,
ipla6.dll,
iplm5.dll,
iplm6.dll,
iplp6.dll,
iplpx.dll
and
iplw7.dll.

Double click the **tree_d.exe** binary icon to start the application.

Pre-processing of images

The TreeD application can process aerial/satellite images both in central projection and in orthogonal projection. The program assumes that the images are close to a nadir view. If the camera angle is oblique, the program cannot choose the correct viewing angle for the templates.

You can use image software to rectify an image into for example the RT90 coordinate system. When rectifying the image it is important to save the leftmost, rightmost, lowest and highest coordinates to be used as input later in the program. The application can read 24-bit bmp image files. If the rectified image is in another format such as TIFF it has to be converted by image software before loading it.

Creating a tree library

The application use a library to create the tree templates used in the matching algorithm. Such a tree library file can be created in a standard text editor. The parameters in this library can be adjusted to be suitable for a local forest type. The library text file should contain a row for each tree the application should look for. Five predefined tags followed by a value determine the different tree parameters, see example.

```
tree list

<name> tree1 <exponent> 2.0 <radius> 2.5 <crownheight> 10.0 <stemheight> 10.0
<name> tree2 <exponent> 2.0 <radius> 1.5 <crownheight>  5.0 <stemheight>  5.0
```

Where <exponent> is a shape value. A value of 1.0 = cone, 2.0 = ellipsoid and ∞ = cylinder. It is important to use space between the tag and the value. The <radius>, `<crownheight>` and `<stemheight>` are defined as, `r`, `ch`, and `sh` in Figure 2.

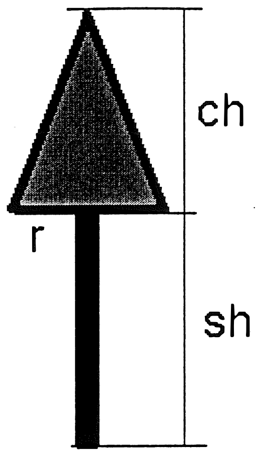

Figure 2 *The definitions of the tree size tags in the tree library, r = <radius>, ch = <crownheight> and sh = <stemheight>. These values are used when rendering the tree templates to be used in the matching algorithms.*

The values of <radius>, <crownheight> and <stemheight> should be defined in meters [m] if the global coordinate system is defined in meters [m]. No more than 30 trees can be used at a time. For performance reasons however it is advisable to keep the number of trees low. No more than five at a time.

Aerial information

The application read the aerial information from a text file with predefined tags. Such an information file can be created in a standard text editor. The input data the program needs are the flying altitude <z0> measured from the ground (not the sea level). The leftmost, rightmost, lowest and highest coordinates of the global coordinate system, `<left>`, `<right>`, `<bottom>` and `<top>` all defined in the same unit as the tree data. The solar altitude and azimuth angle in degrees, `<altitude>` and `<azimuth>`. If the image is not rectified in a north south direction the azimuth angle must be adjusted correspondingly. See Figure 3 for the angle definitions.

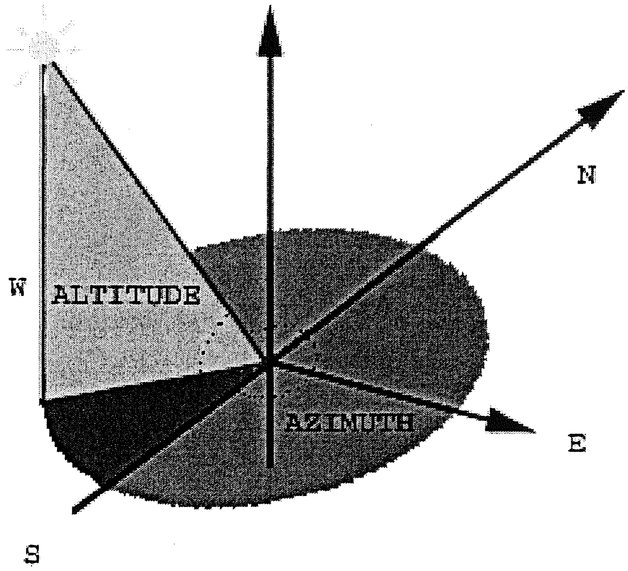

Figure 3 *Definition of the solar angles that are used in the template matching algorithms. N = north, E = east, S = south and W= west.*

An example of an **aerial information** text file can bee seen below.

```
aerial info

<z0> 4600.0

<left> 0.0 <right> 256.0 <bottom> 0.0 <top> 256.0

<altitude> 45.0 <azimuth> 30.0
```

If a global coordinate system is not known for an aerial image it is possible to set the left and bottom coordinates to zero and the right and top coordinates to the (image width / pixels per meter) and (image height / pixels per meter). That way a relative positioning of the trees is possible.

Setting the image filtering

The **correlation threshold** value sets the level where possible trees are accepted. A value of one is a perfect match. A value of zero means that the template does not correlate with the image. If the image has a high resolution it can be necessary to smooth the image before matching. **Blur** sets the size, in pixels, of an averaging filter. **Gauss** sets the standard deviation, in pixels, of a Gaussian filter. When the values are set to zero, no filtering is performed. It is not necessary to use both a Gaussian and an averaging filter at the same time.

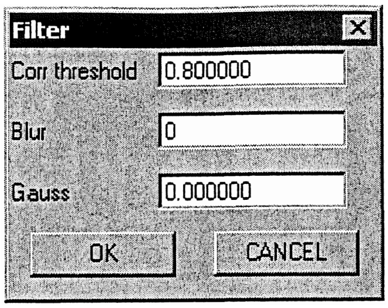

Figure 4 *The Filtering input window. A correlation value of 1.0 is a perfect match whereas a value of 0.0 means that there is no correlation at all. Blur and Gauss sets the averaging and Gaussian smoothing of the image.*

Program output

Before starting a correlation of tree templates a **result directory** must be chosen. There three text files will be saved, **status.txt**, **treelist.txt** and **probable_treelist.txt**. <u>If the result directory contains old files with these names they will be overwritten</u>. To save a new batch you can either rename the old files or use a new result directory. Before starting a new correlation it is advisable to see the status of the current input variables. When you are satisfied with the choice of the input files and settings, the correlation can be started from the file menu.

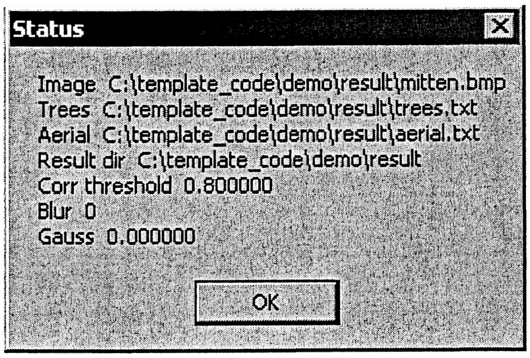

Figure 5 *The current input settings can be viewed in the status window. These are the values that will be used when starting the next correlation.*

The status text file contains information about the input settings that were used when correlating the current batch, see example.

status.txt

```
Image C:\template_code\demo\result\mitten.bmp
Trees C:\template_code\demo\result\trees.txt
Aerial C:\template_code\demo\result\aerial.txt
Result dir C:\template_code\demo\result
PixPerMeter 2.000000
x_pixPerMeter 2.000000
y_pixPerMeter 2.000000
```

```
corrThreshold 0.800000
Blur 0
Gauss 0.000000
Width 512
Height 512
Geo width 256.000000
Geo height 256.000000
Tile width 48
Tile height 48
```

The file **treelist.txt** contains all found trees with all templates and the file **probable_treelist.txt** contains the most probable candidates. The apex and root positions of the trees are saved both in local coordinates (pixels) and in global coordinates. The table can be imported to standard GIS software to study the positions of the trees overlaid upon the aerial image. The position of the apex is easiest to use because the tree tops are seen in the image. The root position is the one that is more correct to use as the global position. The accuracy of the root position depends on how well the height of the tree is estimated (if the correct tree template was chosen).

An example of a **probable_treelist.txt** file

```
correlation x_apex      y_apex      x_root      y_root
0.932431    8.000000    10.000000   9.013100    11.013100
0.943925    26.000000   14.000000   26.505447   14.505447
0.928759    40.000000   14.000000   40.505447   14.505447
...

x_apex_geo  y_apex_geo  x_root_geo  y_root_geo
4.000000    5.000000    4.506550    5.506550
13.000000   7.000000    13.252723   7.252723
20.000000   7.000000    20.252723   7.252723
...

name        N           radius      crownHeight stemHeight
tree1       2.000000    2.500000    10.000000   10.000000
tree2       2.000000    1.500000    5.000000    5.000000
tree2       2.000000    1.500000    5.000000    5.000000
...

index kernel_M    kernel_N
0     0     0
1     0     0
1     0     0
...

lightx      lighty      lightz      pos_x       pos_y       z0
-0.353553   0.612372    -0.707107   -92.000000  -92.000000  4600.000000
-0.353553   0.612372    -0.707107   -92.000000  -92.000000  4600.000000
-0.353553   0.612372    -0.707107   -92.000000  -92.000000  4600.000000
...
```

Template rendering

The application has a separate kernel rendering tool to use when experimenting with different tree sizes, solar angles or pixels per meter values. The same template rendering algorithms as in the correlation loop is used. Therefore this rendering tool can be used as a support when writing a tree library or finding an unknown solar angle.

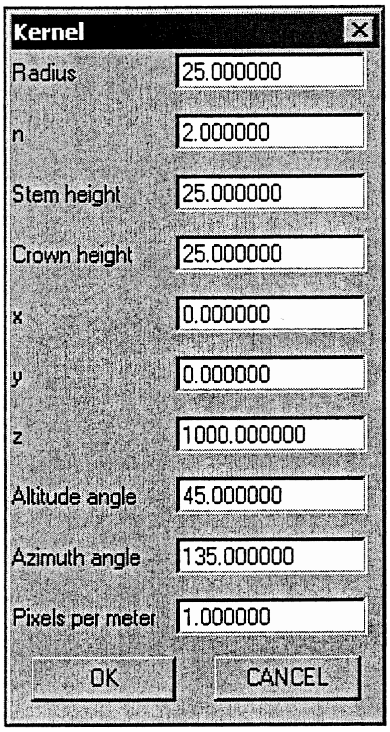

Figure 6 *The input window for the kernel rendering tool.*

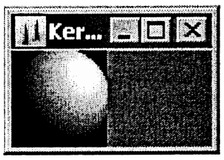

Figure 7 *The output from the kernel rendering tool show the synthetic tree in a viewing window.*

PART II

Method description

Methods

The procedures of the application are

- Divide the aerial image into tiles
- Render a template for each tree and tile
- Cross correlate the template with the tile of the aerial image to receive possible tree positions
- Search for candidates to the same position from four neighbouring tiles
- Choose the most probable tree for each position

Rendering of synthetic trees

The templates in the TreeD application are rendered to look like a shadowed tree in the correct viewing angle. The shape used is that of a generalized ellipsoid of revolution as suggested by Pollock (1996), se Figure 8. The synthetic kernels are rendered by flat shading and the viewing projection is that of a pinhole camera located z0 [m] above nadir.

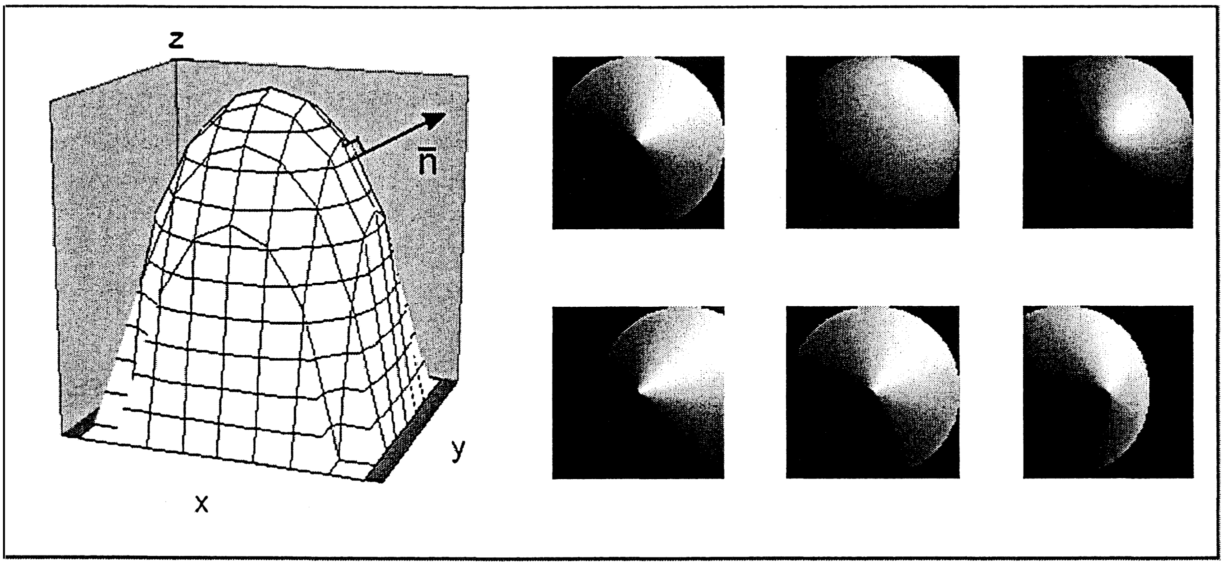

Figure 8 *To the left, a generalized ellipsoid of revolution. To the right, ellipsoids of different shapes and in different viewing angles, rendered with flat shading.*

Flat shading

When rendering a flat shaded object the direction of the illumination and the direction of the surface normal vector are needed. The intensity i at a point of the surface is given by

$$i = -I\cos(180 - \theta) = -I\frac{(\bar{n} \bullet \bar{l})}{|\bar{n}||\bar{l}|} \qquad [0° < \theta < 90°]$$

Where I is the maximum intensity, $\bar{n}$ is the surface normal vector and $\bar{l}$ is the illumination vector, see Figure 9.

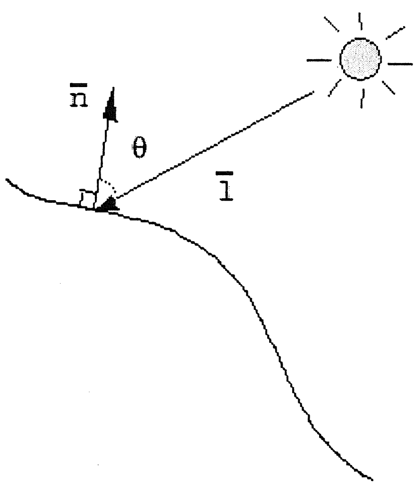

Figure 9 *When rendering a surface colour the light vector $\bar{l}$ and the normal vector $\bar{n}$ of the surface have an angular difference θ. In a flat shading an angle $\theta = 0°$ gives full intensity whereas an angle $\theta = 90°$ gives the lowest intensity.*

Deriving the normal surface vector of generalized ellipsoids

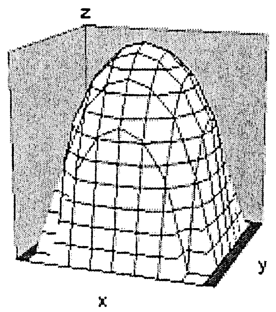

Figure 10 *A generalized ellipsoid of revolution*

The equation for a generalized ellipsoid is as follows

$$\left(\frac{z}{a}\right)^n + \left(\frac{r}{b}\right)^n = 1 \qquad [1.1]$$

Substituting $r = \sqrt{x^2 + y^2} = \left(x^2 + y^2\right)^{1/2}$, in [1.1] gives

$$\frac{z^n}{a^n} + \frac{\left(x^2 + y^2\right)^{n/2}}{b^n} = 1 \qquad [2.1]$$

Solving for z gives

$$z = a\left(1 - \frac{\left(x^2 + y^2\right)^{n/2}}{b^n}\right)^{1/n} \qquad [3.1]$$

The derivative of z for x, y and z is

$$\frac{dz}{dx} = \frac{a}{n}\left(1 - \frac{\left(x^2 + y^2\right)^{n/2}}{b^n}\right)^{1/n-1}\left(-\frac{n2x\left(x^2 + y^2\right)^{n/2-1}}{2b^n}\right) = -\frac{ax\left(x^2 + y^2\right)^{\frac{n-2}{2}}}{b^n}\left(1 - \frac{\left(x^2 + y^2\right)^{n/2}}{b^n}\right)^{\frac{1-n}{n}} \qquad [4.1]$$

$$\frac{dz}{dy} = -\frac{ay\left(x^2 + y^2\right)^{\frac{n-2}{2}}}{b^n}\left(1 - \frac{\left(x^2 + y^2\right)^{n/2}}{b^n}\right)^{\frac{1-n}{n}} \qquad [5.1]$$

$$\frac{dz}{dz} = 1 \qquad [6.1]$$

The tangent vector of a curve is $\begin{bmatrix} Dx \\ Df \end{bmatrix}$ according to Figure 11. Therefore the normal vector of a curve is $\begin{bmatrix} -Df \\ Dx \end{bmatrix}$.

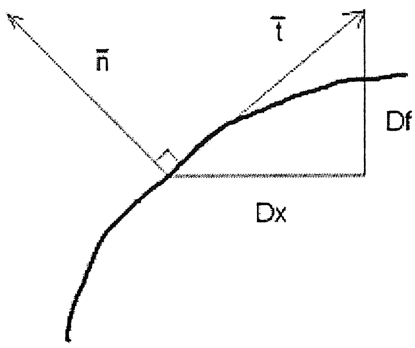

Figure 11 *The normal and tangent vectors of a curve*

Normalize the normal vector by Dx $\frac{1}{Dx}\begin{bmatrix} -Df \\ Dx \end{bmatrix} = \begin{bmatrix} -\frac{Df}{Dx} \\ \frac{Dx}{Dx} \end{bmatrix} = \begin{bmatrix} -\frac{Df}{Dx} \\ 1 \end{bmatrix} \approx \begin{bmatrix} -\frac{df}{dx} \\ 1 \end{bmatrix}$. By similar reasoning the normal of a surface is $\begin{bmatrix} -\frac{df}{dx} \\ -\frac{df}{dy} \\ 1 \end{bmatrix}$. Please observe that the normal vector is not normalized to unit length in this form. If necessary that can be done numerically after differentiation. Then use the pre calculated derivatives in equation [4.1], [5.1] and [6.1] to obtain the surface normal of generalized ellipsoids.

Ray tracing

When generating templates the shape of the object is known in a local coordinate system whereas the ray tracing of the object is performed in a world coordinate system. Hence there is a need to transform the different coordinates. Figure 12 shows a system where the camera is located at *(0, 0, Z0)* in the *X, Y* and *Z* coordinates whereas the template is positioned at (*XtemplatePos*, *YtemplatePos*, *0*). The shape of the template is defined in the local system *xLocal, yLocal* and *Z*.

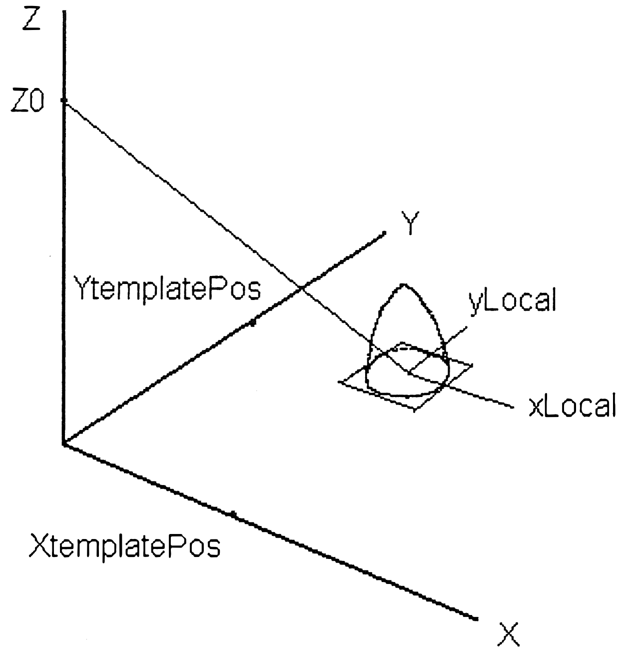

Figure 12 *A coordinate system showing the imaging of aerial photos. The camera at position (0, 0, Z0) and the template at position XtemplatePos and YtemplatePos.*

The transform from *(X, Y, Z)* to *(xLocal, yLocal, Z)* is simply

$$\begin{aligned} xLocal &= X - XtemplatePos \\ yLocal &= Y - YtemplatePos \\ Z &= Z \end{aligned} \qquad [1.2]$$

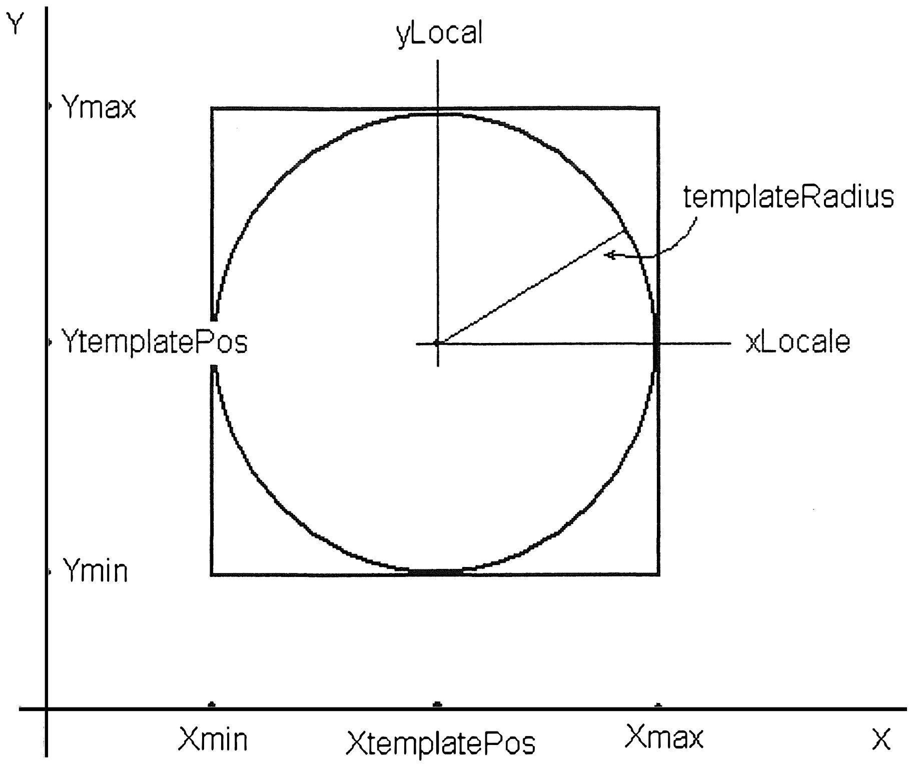

Figure 13 *A template with definitions in two coordinate systems*

A template size must be chosen when ray tracing the template in the world coordinate system. If for instance

```
templateWidth  = Xmax - Xmin
templateHeight = Ymax - Ymin.
```

Xmax, *Xmin, Ymax* and *Ymin* can for example be chosen as

```
Xmin = XtemplatePos - templateRadius
Xmax = XtemplatePos + templateRadius
Ymin = YtemplatePos - templateRadius
Ymax = YtemplatePos + templateRadius
```

When oversampling, the *samplingWidth* and *samplingHeight* is chosen to larger numbers than the corresponding template sizes. The ratio between the two is

```
RatioWidth  = samplingWidth  / templateWidth
RatioHeight = samplingHeight / templateHeight
```

The increments used in the over sampling is then

```
DeltaX = 1 / RatioWidth
```

```
DeltaY = 1 / RatioHeight
```

And thus the sampling coordinates is

```
X = x * deltaX + Xmin
Y = y * deltaY + Ymin
```

Where *x* and *y* is the coordinates in the new ray traced image

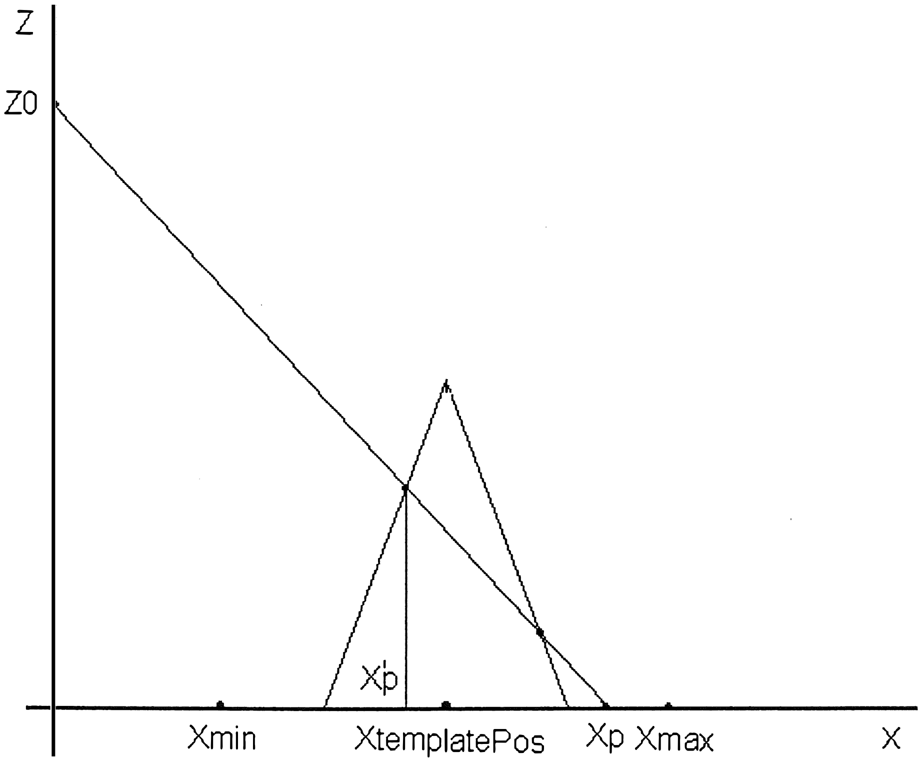

Figure 14 *Ray tracing a template. The grey value at position Xp in the image (world zero plane) is collected from position X´p at the rendered template.*

When ray tracing the template the grey value in position *Xp* is collected from position *X´p* in the rendered template, see Figure 14. The equation for the ray tracing line on parametric form is

$$\begin{cases} d = \sqrt{X_p^2 + Y_p^2 + Z_0^2} \\ X = X_p - \dfrac{X_p t}{d} \quad [0 \le t \le d] \\ Y = Y_p - \dfrac{Y_p t}{d} \quad [d \ne 0] \\ Z = \dfrac{Z_0 t}{d} \end{cases} \qquad [2.2]$$

Solve [2.2] for t results in

$$t = (X_p - X)\frac{d}{X_p} \quad [X_p \neq 0]$$
$$t = (Y_p - Y)\frac{d}{Y_p} \quad [Y_p \neq 0] \qquad [3.2]$$
$$t = \frac{Zd}{Z_0}$$

The intersection point $X'p$ in Figure 14 must be found by a numerical method. One way can be to sample along the ray tracing line and see when the difference in Z-values between the ray and the object (*rayZ – templateZ*) is positive or negative. When there is a shift from negative to positive values there is an intersection point.

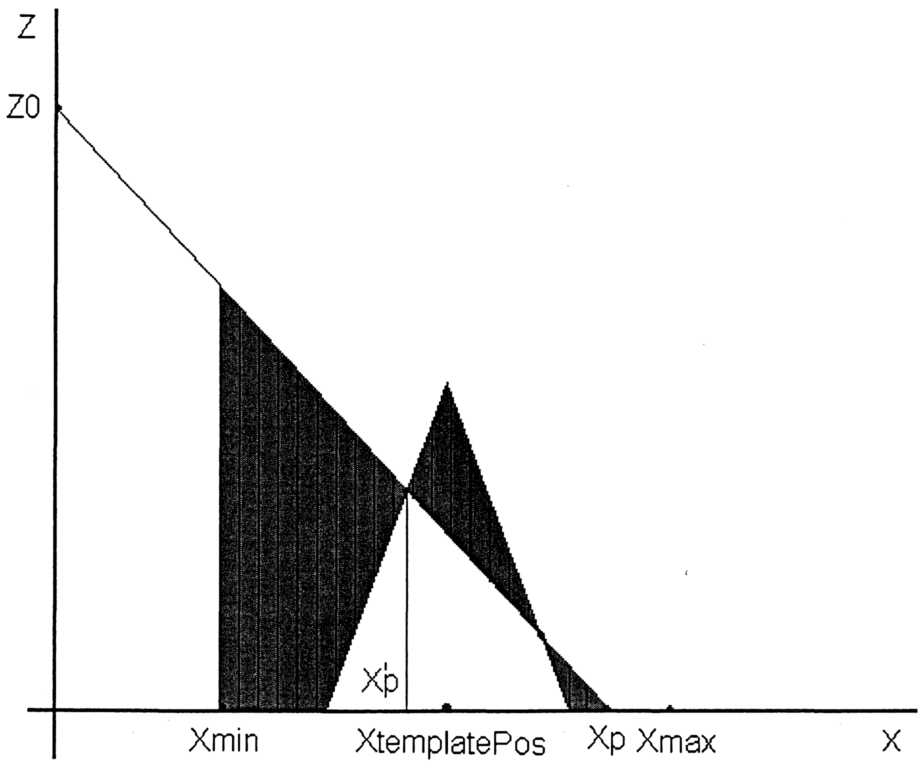

Figure 15 *Diagram showing when the ray tracing line is inside or outside the object. An intersection point can be found where the difference rayZ – templateZ shifts from negative to positive.*

Figure 15 shows such a ray intersecting an object. When the height difference shifts it indicates that there is an intersection point. The template can be inscribed in a box to decrease the number of sampling points along the ray. To find the value of t at X=*Xmin* use [3.2].

`tXmin = (Xp - Xmin)d / Xp`

Each increment of *t* can be found by dividing with the interesting number of samples

`deltaT = tXmin/N`

Where *N* = number of samples.

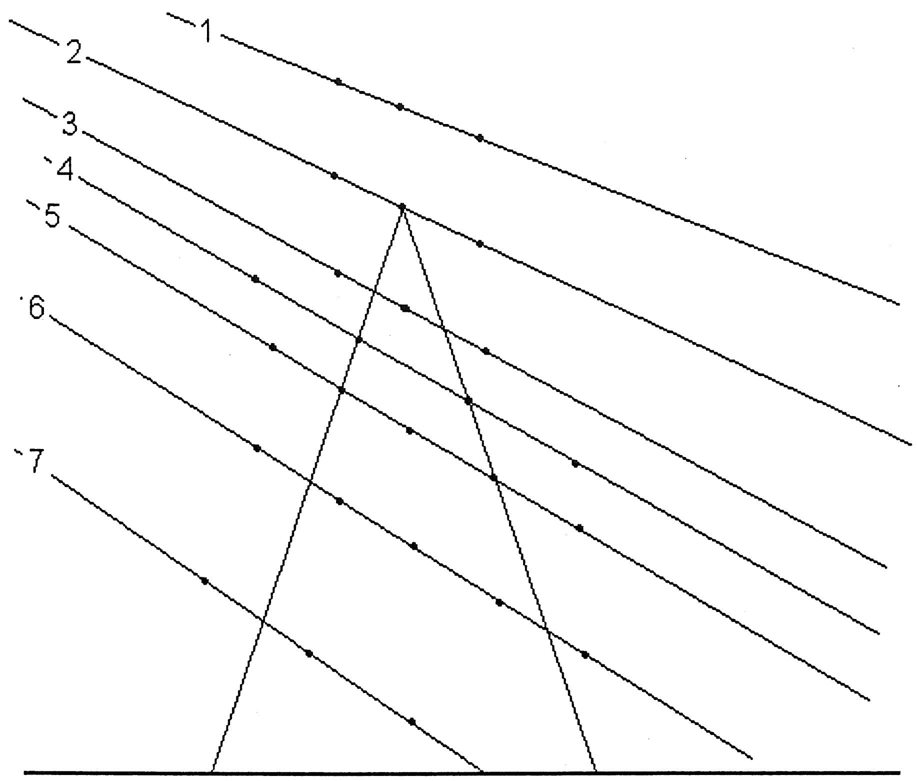

Figure 16 *Seven different ray tracing lines sampling the height difference between the object and the ray.*

The binary logic for finding positions where the ray pass from the inside to the outside of the tree crown is as follows.

Height difference	Code (binΔZ)
`Zero (on crown surface)`	`0`
`Positive (above crown surface)`	`1`
`Negative (below crown surface)`	`-1`

Table 1 Height difference encoding

The difference between the present and the last height difference (*diffBin*ΔZ) is also of interest.

Going from right to the left in Figure 16 results in the following sequences

	Ray tracing line 1		
BinΔZ	1	1	1
diffBinΔZ	---	0	0

	Ray tracing line 2		
BinΔZ	1	0 *	1
diffBinΔZ	---	-1	1

	Ray tracing line 3		
BinΔZ	1	-1	1
diffBinΔZ	---	-2	2 *

	Ray tracing line 4			
BinΔZ	1	0 *	0 *	1
diffBinΔZ	---	-1	0	1

	Ray tracing line 5				
BinΔZ	1	0 *	-1	0 *	1
diffBinΔZ	---	-1	-1	1	1

	Ray tracing line 6				
BinΔZ	1	-1	-1	-1	1
diffBinΔZ	---	-2	0	0	2 *

	Ray tracing line 7		
BinΔZ	-1	-1	1
diffBinΔZ	---	0	2 *

Positions where the ray pass from the inside to the outside of the tree crown, are marked with a *. If *binΔZ* = 0 the intersection point is found directly. If *diffBinΔZ* = 2 the points are passing from the inside to the outside of the object and is therefore a good starting point for interpolating the value.

Interpolating an intersection point works best when the two surrounding points both is located inside the object base radius. If one of them is outside it is better to move it closer to the object. Figure 17 shows such a case.

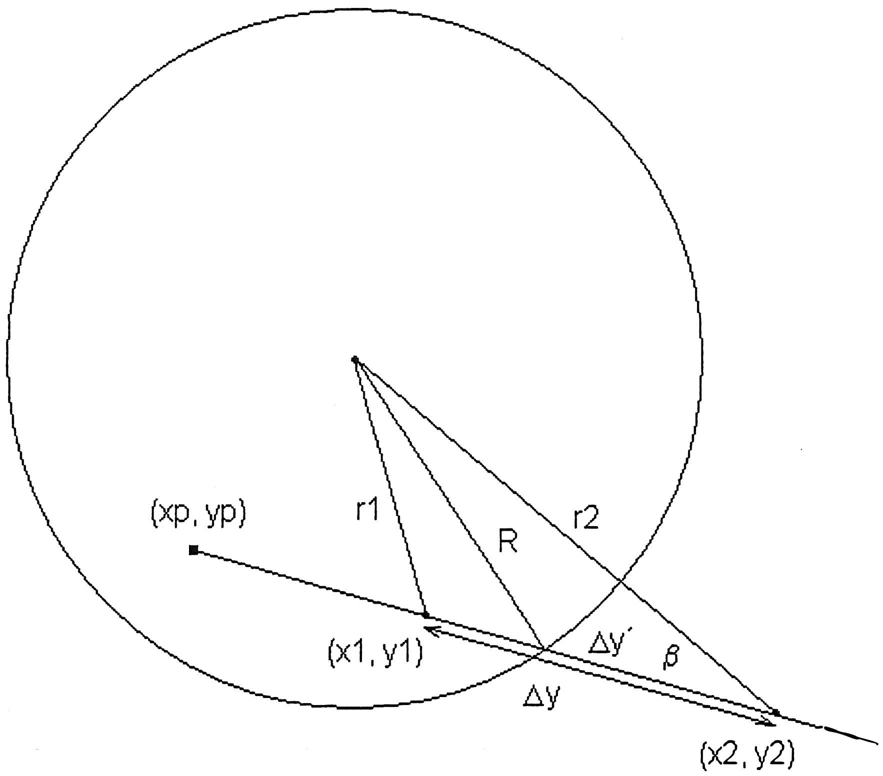

Figure 17 *Two interpolating points (x1, y1) and (x2, y2) where the second is outside the template base radius. The ray tracing line starts at (xp, yp).*

The following relationship is found in Figure 17.

$$r_1^2 = r_2^2 + \Delta y^2 - 2r_2 \Delta y \cos\beta \qquad [4.2]$$

$$R^2 = r_2^2 + \Delta y'^2 - 2r_2 \Delta y' \cos\beta \qquad [5.2]$$

Solve for $\cos\beta$ in 4.2

$$\cos\beta = \left(\frac{r_2^2 - r_1^2 + \Delta y^2}{2r_2 \Delta y} \right) \qquad [6.2]$$

Rearrange [5.2] into a second order equation.

$$\Delta y'^2 + (-2r_2 \cos\beta)\Delta y' + \left(r_2^2 - R^2\right) = 0 \qquad [7.2]$$

Solve for $\Delta y'$

$$\Delta y' = r_2 \cos\beta \pm \sqrt{r_2^2 \cos^2\beta - r_2^2 + R^2} =$$

$$r_2 \cos\beta \pm \sqrt{r_2^2\left(\cos^2\beta - 1 + \frac{R^2}{r_2^2}\right)} \quad , \qquad [8.2]$$

$$\left(\cos^2\beta - 1 + \frac{R^2}{r_2^2}\right) \geq 0$$

The smallest root in [8.2] is the one that is searched for.

The new value for *t2* is found by simple interpolation

$$t_{2new} = \frac{(t_2 - t_1)(\Delta y - \Delta y')}{\Delta y} + t_1 \qquad [9.2]$$

Where *t1* and *t2* corresponds to the points (x1, y1) and (x2, y2) in the ray tracing line.

The intersection point is found by interpolating with the new *t2* value.

$$t_x = \frac{\left(\frac{t_{2new}}{\Delta z_{2new}} - \frac{t_1}{\Delta z_1}\right)}{\left(\frac{1}{\Delta z_{2new}} - \frac{1}{\Delta z_1}\right)} \qquad [10.2]$$

Where *Δz2new* and *Δz1* is the difference between the height of the ray and the height of the object at points *(x1, y1)* and *(x2new, y2new)*.

Template matching

Cross correlation with normalized image matrixes

Consider the case when a kernel is correlated with an image matrix.

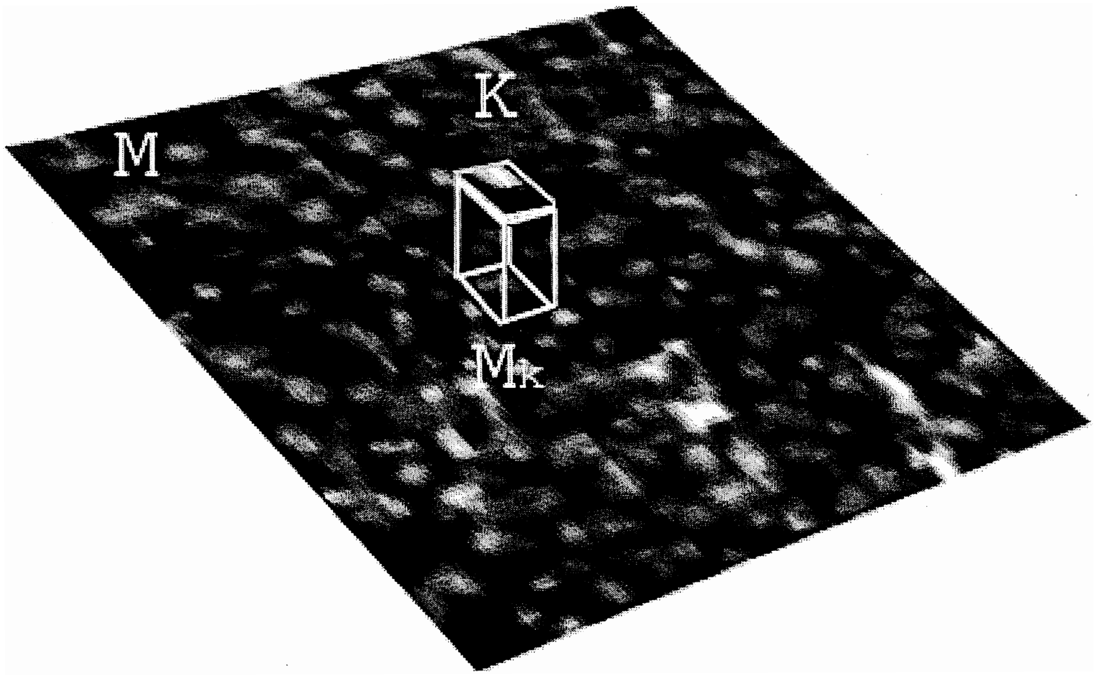

Figure 18 *The kernel K is correlated with the image matrix M. The current position of the correlation is at the sub image Mk.*

Rearrange the content in matrixes K and M_k into vectors

$$\overline{K} \quad and \quad \overline{M}_k$$

Where

$$K_i = K_{xy} - \frac{1}{mn}\sum_x \sum_y K_{xy} = K_{xy} - K_{average}$$

And

$$M_{ki} = M_{kxy} - \frac{1}{mn}\sum_x \sum_y M_{kxy} = M_{kxy} - M_{average}$$

And m and n are the matrix width and height:

These can be seen as vectors in an n-dimensional space. The dot product of those two is

$$\overline{K} \bullet \overline{M}_k = |\overline{K}||\overline{M}_k|\cos\theta \qquad [1.3]$$

Rewriting [1.3] in index form gives

$$\sum_i K_i M_{ki} = \sqrt{\sum_i K_i^2}\sqrt{\sum_i M_{ki}^2}\cos\theta \qquad [2.3]$$

Rearranging [2.3] gives

$$\cos\theta = \frac{\sum_i K_i M_{ki}}{\sqrt{\sum_i K_i^2}\sqrt{\sum_i M_{ki}^2}} \qquad [3.3]$$

If $M_{ki} = K_i$ then [3.3] becomes

$$\cos\theta = \frac{\sum_i K_i K_i}{\sqrt{\sum_i K_i^2}\sqrt{\sum_i K_i^2}} = \frac{\sum_i K_i K_i}{\sum_i K_i^2} = \frac{\sum_i K_i^2}{\sum_i K_i^2} = 1 \qquad [4.3]$$

If $M_{ki} = -K_i$ then [3.3] becomes

$$\cos\theta = \frac{\sum_i K_i(-K_i)}{\sqrt{\sum_i K_i^2}\sqrt{\sum_i (-K_i)^2}} = \frac{-\sum_i K_i K_i}{\sqrt{\sum_i K_i^2}\sqrt{\sum_i K_i^2}} = \frac{-\sum_i K_i K_i}{\sum_i K_i^2} = \frac{-\sum_i K_i^2}{\sum_i K_i^2} = -1 \qquad [5.3]$$

Thus if you have a perfect match between the two vectors the ratio is 1 and if you have an inverted relationship the ratio is –1. Since ($-1 < \cos\theta < 1$) all other non perfect matches lies between the two extremes.

Rewrite [3.3] in matrix index form

$$\cos\theta = \frac{\sum_x \sum_y \left(K_{xy} - K_{average}\right)\left(M_{kxy} - M_{kaverage}\right)}{\sqrt{\sum_x \sum_y \left(K_{xy} - K_{average}\right)^2} \sqrt{\sum_x \sum_y \left(M_{kxy} - M_{kaverage}\right)^2}} \qquad [6.3]$$

Equation [6.3] describes how to correlate the kernel K with the image matrix M at position k.

Choosing probable trees

When correlating an aerial image with several templates corresponding to different tree sizes and shapes there will be many candidates to the same position. To decide whether two template correlations are the same tree, the mask coverage ratio is measured for the two kernels, see Figure 19. The coverage ratio is the intersection area divided by the total mask area. This is measured for both of the investigated templates. The larger of the two values is chosen for comparison. A small kernel can for instance be inside a large one, giving100 % coverage whereas the larger template is only covered by a small amount lower than the threshold by the small template. When a number of candidates to the same position are found the one with the highest correlation value is chosen. It is possible to use weights on the correlation values if for instance one would like to pick a large tree rather than a small one.

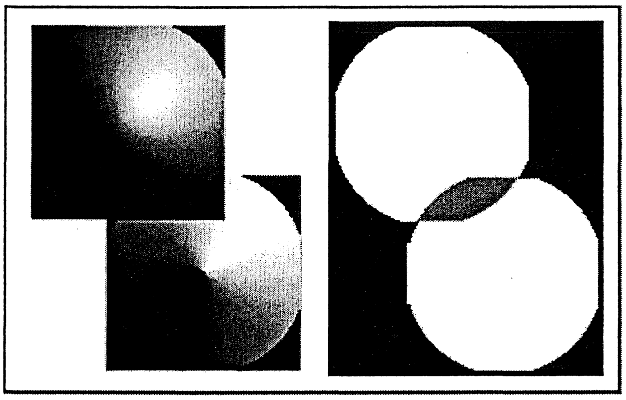

Figure 19 *To the left, two templates which partly cover each other. To the right, the template masks. The intersection of the template masks is marked with grey. If the grey area ratio is larger than a chosen threshold the templates are considered to be candidates to the same tree position.*

Tiling

If large aerial images are used it is not possible to perform the calculations on the entire image at the same time, because the computer memory will be too small. The image needs to be split into several tiles. This will also enhance the search for the most probable trees since only the neighbouring tiles need to be searched in. Not the whole image. If the tiles are chosen small enough it is possible to use the same viewing angle when rendering the templates. A value of two times a chosen max crown diameter for a forest could for instance be chosen. If the size and the anchor of the correlation kernel are known it is easy to calculate the size of the image window corresponding to the tile window which is to be investigated, see Figure 20.

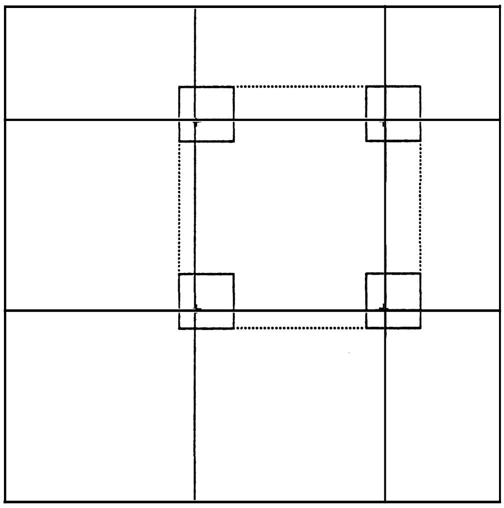

Figure 20 *An image consisting of nine tiles. Four positions of a template are shown. The anchors of the templates are marked with a cross. The dotted lines circumscribe the image window needed to investigate every anchor position of the middle tile.*

The tile width in the TreeD 0.8 application is chosen as

```
tileWidth  = 2 * MAX_CROWN_DIAMETER * pixelsPerMeter
tileHeight = tileWidth
```

```
MAX_CROWN_DIAMETER = 12
```

The maximum number of tiles in the width and height direction can be calculated by

```
Mmax = floor( imageWidth  / tileWidth )
Nmax = floor( imageHeight / tileHeight )
```

The tile position in the image coordinate system can be calculated by

```
tileXmin =  M      *tileWidth
tileXmax = (M + 1) *tileWidth  - 1
tileYmin =  N      *tileHeight
tileYmax = (N + 1) *tileHeight - 1
```

, where *M* and *N* are the indexes to the current tile. To calculate the window needed for the correlation kernel to cover a tile, the kernel size and anchor must be known. The circumscribing coordinates is calculated by

```
tileImageXmin = tileXmin - anchorX
tileImageXmax = tileXmax + kernelWidth  - anchorX - 1
tileImageYmin = tileYmin - anchorY
tileImageYmax = tileYmax + kernelHeight - anchorY - 1
```

Height and position errors due to altitude differences in the aerial image

Central projection

In aerial images the ground level is usually not flat and that results in errors when estimating the size and position of objects.

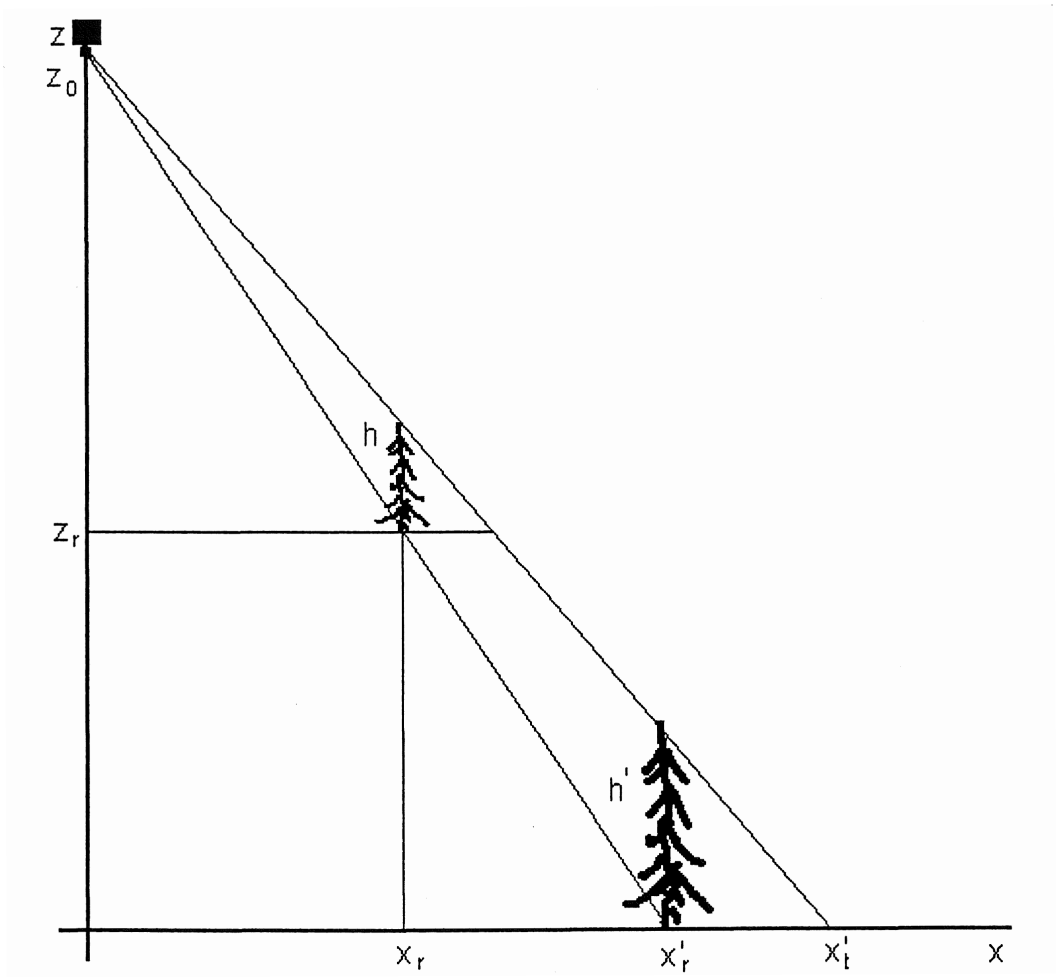

Figure 21 *Projection of a tree, with height h and positioned at level zr, to the zero level in an aerial image.*

In Figure 21 the following relationships can be shown

$$\frac{x_r'}{z_0} = \frac{x_r}{z_0 - z_r}, \; x_r' = \frac{z_0 x_r}{z_0 - z_r} \qquad [1.4]$$

$$\frac{x_t'}{z_0} = \frac{x_r}{z_0 - z_r - h}, \; x_t' = \frac{z_0 x_r}{z_0 - z_r - h} \qquad [2.4]$$

$$\frac{h'}{x_t' - x_r'} = \frac{z_0}{x_t'}, \quad h' = z_0\left(\frac{x_t' - x_r'}{x_t'}\right) = z_0\left(1 - \frac{x_r'}{x_t'}\right) \qquad [3.4]$$

[1.4] and [2.4] in [3.4] gives

$$h' = z_0\left(1 - \frac{z_0 x_r (z_0 - z_r - h)}{(z_0 - z_r) z_0 x_r}\right) = z_0\left(1 - \frac{(z_0 - z_r - h)}{(z_0 - z_r)}\right) = z_0\left(1 - 1 + \frac{h}{(z_0 - z_r)}\right) = \frac{z_0 h}{(z_0 - z_r)} = \frac{h}{1 - \frac{z_r}{z_0}} \qquad [4.4]$$

The position and height error is

$$\Delta x_r = x_r' - x_r, \quad \Delta h = h' - h \qquad [5.4]$$

[1.4] and [4.4] in [5.4] gives

$$\Delta x_r = \frac{z_0 x_r}{z_0 - z_r} - x_r = \left(\frac{1}{1 - \frac{z_r}{z_0}} - 1\right) x_r, \quad \Delta h = \left(\frac{1}{1 - \frac{z_r}{z_0}} - 1\right) h \qquad [6.4]$$

Thus the errors depends on the error factor *e*

$$e = \left(\frac{1}{1 - \frac{z_r}{z_0}} - 1\right), \quad \Delta x_r = e x_r, \quad \Delta h = eh \qquad [7.4]$$

Solve for z_r / z_0 in [7.4] results in

$$e + 1 = \frac{1}{1 - \frac{z_r}{z_0}}, \quad 1 - \frac{z_r}{z_0} = \frac{1}{e+1}, \quad \frac{z_r}{z_0} = 1 - \frac{1}{e+1} \qquad [8.4]$$

With error factor $e = 0.1$, [8.4] gives

$$\frac{z_r}{z_0} = 0.09090909091 \approx 0.1$$

Thus if you want an error smaller than 10 % in the estimates the highest altitude difference compared to the camera height cannot be larger than 10%. The position error is larger further out in the image and the height error is larger with high trees.

Orthogonal projection

To avoid errors depending on altitude differences sometimes the photographs are converted to an orthogonal projection, when a digital terrain model of the photographed area is known. These conversions are however often crude using a large grid containing several trees.

Figure 22 shows an orthogonal projection of a tree.

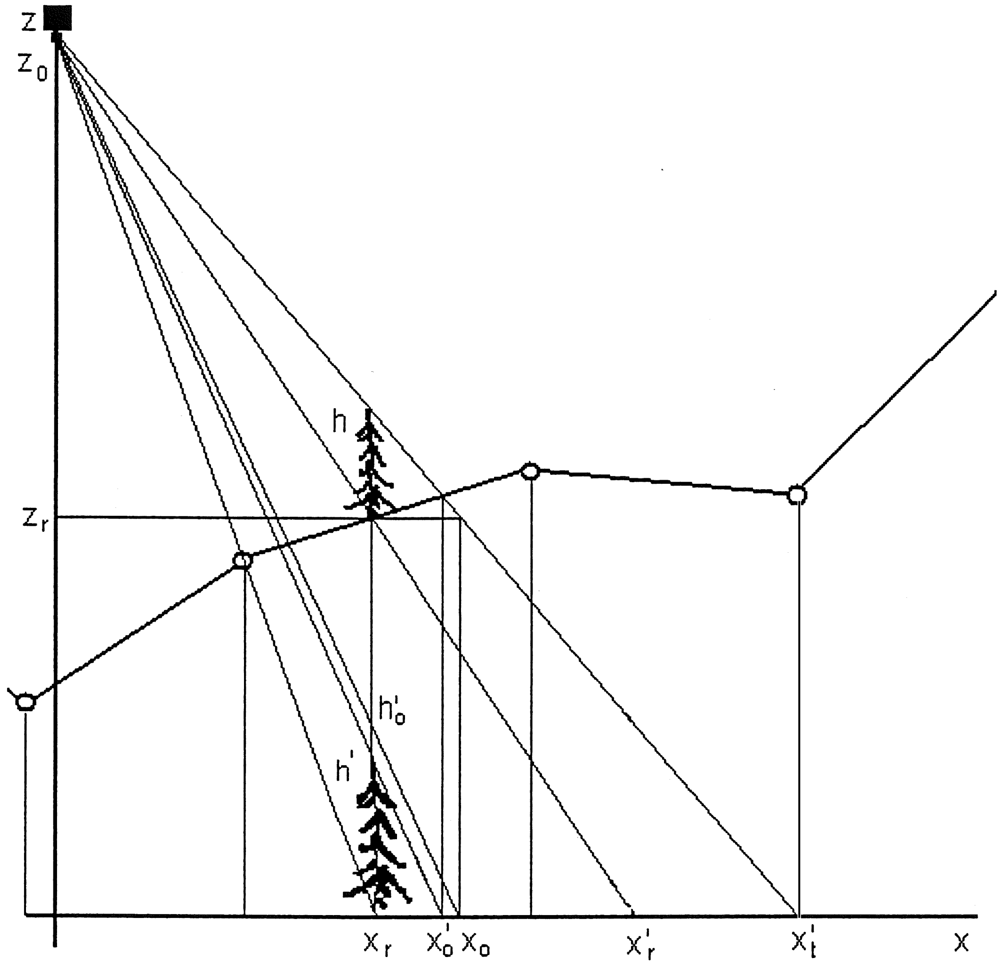

Figure 22 *An orthogonal projection of a tree with height h and position xr, zr, The tree root and top are projected to xr, x'o and the projected height is h'. The tree would have the projected height h'o if the grid element was horizontal.*

The position of the root is correctly projected onto x_r. The projection of the top x'_o depends on the current slope of the ground grid element, leading to an error in the height estimate. If the current grid element is horizontal the top is projected to x_o.

In Figure 22 the following relationship can be seen

$$\frac{x_0}{z_0 - z_r} = \frac{x_t'}{z_0}, \quad x_0 = \frac{(z_0 - z_r)x_t'}{z_0} \qquad [9.4]$$

$$\frac{h_0'}{x_0 - x_r} = \frac{z_0}{x_0}, \quad h_0' = \frac{(x_0 - x_r)z_0}{x_0} = \left(1 - \frac{x_r}{x_0}\right)z_0 \qquad [10.4]$$

$$\frac{h'}{x_0' - x_r} = \frac{z_0}{x_0'}, \quad h' = \frac{(x_0' - x_r)z_0}{x_0'} = \left(1 - \frac{x_r}{x_0'}\right)z_0 \qquad [11.4]$$

Inserting [2.4] in [9.4] gives

$$x_0 = \frac{(z_0 - z_r)z_0 x_r}{z_0(z_0 - z_r - h)} = \frac{(z_0 - z_r)x_r}{(z_0 - z_r - h)} = \frac{x_r}{\left(1 - \frac{h}{z_0 - z_r}\right)} \qquad [12.4]$$

Inserting [12.4] in [10.4] gives

$$h_0' = \left(1 - \frac{x_r\left(1 - \frac{h}{z_0 - z_r}\right)}{x_r}\right)z_0 = \left(1 - 1 + \frac{h}{z_0 - z_r}\right)z_0 = \frac{h}{z_0 - z_r}z_0 = \frac{h}{1 - \frac{z_r}{z_0}} \qquad [13.4]$$

Which is the same result as in [4.4].

For a terrain with gentle slopes

$$x_0' \approx x_0 \Rightarrow h' \approx h_0' \qquad [14.4]$$

This means that the magnitude of the height error in orthogonal projections is of the same order as for normal projections in landscapes without steep hillsides. The position of the tree will be correct.

For a grid segment with a steep slope

$$\frac{\Delta h}{s_2} = \frac{z_0 - z_r - h}{x_r}, \frac{\Delta h}{x_0 - x_r - s_1} = \frac{z_0 - z_r - h}{x_r} \qquad [15.4]$$

See Figure 22 and Figure 23. For a grid element with slope θ

$$\frac{\Delta h}{s_1} = \tan\theta, \Delta h = s_1 \tan\theta \qquad [16.4]$$

[15.4] in [14.4] gives

$$\frac{s_1 \tan\theta}{x_0 - x_r - s_1} = \frac{z_0 - z_r - h}{x_r} \qquad [17.4]$$

Solving for *s1* gives

$$s_1 = \frac{(z_0 - z_r - h)(x_0 - x_r)}{(z_0 - z_r - h) + x_r \tan\theta} = \frac{(x_0 - x_r)}{1 + \dfrac{x_r \tan\theta}{(z_0 - z_r - h)}} \qquad [18.4]$$

Figure 23 gives

$$x_0' = x_r + s_1 = x_r + \frac{x_0 - x_r}{1 + \dfrac{x_r \tan\theta}{(z_0 - z_r - h)}} \qquad [19.4]$$

[12.4] in [18.4] gives

$$x_0' = \frac{x_r + \dfrac{z_0 - z_r}{\tan\theta}}{1 + \dfrac{(z_0 - z_r - h)}{x_r \tan\theta}} \qquad [20.4]$$

[18.4] in [11.4] gives

$$h' = \left(1 - \frac{x_r}{\dfrac{x_r + \dfrac{z_0 - z_r}{\tan\theta}}{1 + \dfrac{z_0 - z_r - h}{x_r \tan\theta}}} \right) z_0 \qquad [21.4]$$

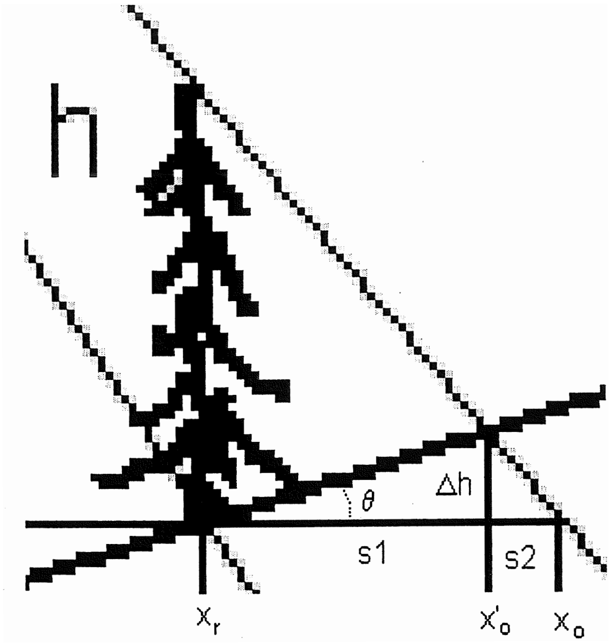

Figure 23 *Magnification of Figure 22*

With a tree height h = 20 [m], a flying height z_0 = 400 [m], a root position z_r = 0 [m], a position x_r = 200 [m] and a negative slope θ = -30 [°],

h' = 28.12

which gives

$h'/h = 28.12 / 20 = 1.41$

PART III

Software description

Application function API

Template matching program

The program consists of a main loop, main functions and a number of supporting functions written at SLU and from the IPL© library. All written in the C programming language. The C-algorithms are called from a Graphical User Interface, GUI, which is using the wxWindows© C++ library.

The main C-loop is

correlate_dib

The main functions are;

create_kernel,
correlate_tree,
find_local_maxD_add_origo,
read_tile,
get_intersect_list_D.

The definition of the main loop is,

correlate_dib

```
BITMAPINFO* correlate_dib(
     const char* fname,
     const char* fnametrees,
     const char* fname_aerial,
     const char* resultdir,
     double      externPixPerMeter,
     double      corrThreshold,
     int         blur,
     double      gauss).
```

Input variables:

`const char* fname,`	The path to the aerial image file
`const char* fnametrees,`	The path to the tree library
`const char* fname_aerial,`	The path to the aerial information
`const char* resultdir,`	The path to the directory where the results will be saved

`double`	`externPixPerMeter,`	Not implemented (for future use)
`double`	`corrThreshold,`	Correlation threshold for accepted match
`int`	`blur,`	Amount of average filtering
`double`	`gauss,`	Amount of Gaussian filtering

Output variables:

`BITMAPINFO* dibPtr,`	The filtered centre part of the aerial image and the templates used

The definitions of the main functions are

create_kernel

```
void create_kernel(
     SluTree   tree,
     SluPoint  templatePos,
     double    Z0,
     SluVector light,
     int       showImage,
     SluPoint* rootPtr,
     SluPoint* apexPtr,
     IplImage* imgKernel,
     IplImage* imgKernelMask),
```

Create kernel is a function that renders a synthetic image of a tree. The illumination and viewing directions can be set. This in order to be used as a correlation kernel when detecting single trees in aerial images.

Input variables:

`SluTree`	`tree,`	The tree geometry (pixels)
`SluPoint`	`templatePos,`	Tree position relative to the camera nadir (pixels)
`double`	`Z0,`	Height from tree root (ground) to camera (pixels)
`SluVector`	`light,`	Illumination vector pointing from the sun (parallel light)
`int`	`showImage,`	0 = false, 1 = true. Shows an IPL window of the kernel. Used for debugging purposes

Output variables:

`SluPoint* rootPtr,`	Position of the tree root in the kernel coordinate system (pixels)
`SluPoint* apexPtr,`	Position of the tree apex in the kernel coordinate system (pixels)
`IplImage* imgKernel,`	Pointer to rendered kernel image
`IplImage* imgKernelMask,`	Pointer to rendered kernel mask This mask is used to remove the surroundings of the kernel in the image

correlate_tree

```
void correlate_tree(
     IplImage* img,
     IplImage* kernel,
     IplImage* kernelMask,
     double    Xanchor,
     double    Yanchor,
     IplImage* resultImg32S),
```

Uses a kernel of a tree to cross correlate with an aerial image of a forest.

Input variables:

`IplImage*`	`image`	The aerial image of interest
`IplImage*`	`kernel`	The current kernel
`IplImage*`	`mask`	The current mask
`double`	`Xanchor`	x-position of the anchor in the kernel coordinate system
`double`	`Yanchor`	y-position of the anchor in the kernel coordinate system

Output variables:

`IplImage*`	`resultImg32S`	Correlation image map for the current kernel. It is necessary to find local maxima in this image to decide the position of possible trees.

find_local_maxD_add_origo

```
void find_local_max_D_add_origo(
     IplImage*      correlationMap,
     double         corrThreshold,
     SluPointI      origo,
     SluPointI      anchor,
     SluPoint       apex,
     SluPoint       root,
     SluTemplate    templateInfo,
     SluKernelIndex tileInfo,
     SluListDHit*   resultsPtr)
```

Finds the local maxima in a cross correlation image, which correspond to possible tree positions. The origo value of the current tile is added to the results.

Input variables:

`IplImage*`	`correlationMap,`	A correlation map image received from the **correlate_tree** function
`double`	`corrThreshold,`	Correlation threshold value of how good match that is allowed for a possible tree, 1 = perfect match, 0 = non matching kernel.
`SluPointI`	`origo,`	Origo of the current tile of the image compared to the aerial image origo
`SluPointI`	`anchor,`	Kernel anchor in local coordinate system (pixels)
`SluPoint`	`apex,`	Kernel apex in local coordinate system (pixels)
`SluPoint`	`root,`	Kernel root in local coordinate system (pixels)
`SluTemplate`	`templateInfo,`	Information about the used template
`SluKernelIndex`	`tileInfo,`	Information about current image tile

Output variables:

`SluListDHit*`	`resultsPtr,`	Dynamic list where the results are saved

read_tile

```
void read_tile(
     SluDiscImage* discImagePtr,
     SluIplImage*  iplImagePtr,
     int M,
     int N,
     int imageXmin,
     int imageXmax,
     int imageYmin,
     int imageYmax,
     int tileWidth,
     int tileHeight,
```

```
int kernelWidth,
int kernelHeight,
SluPointI kernelAnchor)
```

Reads a tile from an image on disc. To save memory if large images are used.

Input variables:

`SluDiscImage* discImagePtr,`		Object with information about the aerial image on disc
`int`	`M,`	The current tile x-index
`int`	`N,`	The current tile y-index
`int`	`imageXmin,`	Aerial image x-minimum value (usually zero)
`int`	`imageXmax,`	Aerial image x-maximum value (width - 1)
`int`	`imageYmin,`	Aerial image y-minimum value (usually zero)
`int`	`imageYmax,`	Aerial image y-maximum value (height - 1)
`int`	`tileWidth,`	Width of current tile
`int`	`tileHeight,`	Height of current tile
`int`	`kernelWidth,`	Width of current kernel
`int`	`kernelHeight,`	Height of current kernel
`SluPointI`	`kernelAnchor,`	Anchor of the kernel in local coordinate system (pixels)

Output variables:

`SluIplImage*`	`iplImagePtr,`	Pointer to the image object of the tile of the aerial image

get_intersect_list_D

```
void get_intersect_list_D(
     SluListDHit*       treeListPtr,
     SluListDHit*       treeList2Ptr,
     SluListDHit*       treeList3Ptr,
     SluListDHit*       treeList4Ptr,
     SluMatrixIndex     tileIndex,
     SluKernelData      kernelMatrix[][4],
     SluListDGraphHit*  resultListPtr)
```

Reads the result lists from four neighboring tiles and returns the most probable trees.

Input variables:

`SluListDHit*`	`treeListPtr,`	Result from current tile
`SluListDHit*`	`treeList2Ptr,`	Result from neighbouring tile
`SluListDHit*`	`treeList3Ptr,`	Result from neighbouring tile
`SluListDHit*`	`treeList4Ptr,`	Result from neighbouring tile
`SluMatrixIndex`	`tileIndex,`	Information about current tile

SluKernelData	kernelMatrix[][4],	Rendered kernels for the four neighbouring tiles

Output variables:

SluListDGraphHit*	resultListPtr,	Result list with most probable trees

Graphical user interface

The graphical user interface uses the wxWindows© C++ cross platform GUI library. The class declarations can be seen below.

```
/* ---------------------------------------------------------------------
|
|    Name:          wx_declare.h
|    Purpose:   -
|    Context:   class declarations for wxWindows application
|    Returns:   -
|    Parameters:     -
|
|    Notes:          -
|
|
|                   Kenneth Olofsson
|
|                   Remote Sensing Laboratory
|                   Department of Forest Resource Management and
|                   Geomatics
|                   Swedish University of Agricultural Sciences,
|                   SLU, Umeå
|
|                   copyright (c) 2003
 ---------------------------------------------------------------------
*/


#ifndef __WX_DECLARE__
#define __WX_DECLARE__


// ImageApplication

class ImageApplication : public wxApp {

public:

    virtual bool OnInit();

};

DECLARE_APP(ImageApplication)


// ScrollCanvas

class ScrollCanvas : public wxScrolledWindow {
```

```
public:

     ScrollCanvas(wxWindow* parentPtr);

     void OnPaint(wxPaintEvent& event);
     void SetImagePath(wxString theString);
     void SetBitmap(wxBitmap bitmap);
     bool LoadBitmap();
     void SetCanvasScrollSize();
     void SetCorrelateStatus(bool is_correlating);
     bool GetCorrelateStatus();

     wxString GetImagePath();

     DECLARE_EVENT_TABLE()

private:

     wxString imagePath;
     wxImage  theImage;
     wxBitmap bitMap;

     bool     isCorrelating;

};


// ImageFrame

class ImageFrame : public wxFrame {

public:

     ImageFrame(
          const wxString& title,
          const wxPoint&  position,
          const wxSize&   size);

     void OnKernel(         wxCommandEvent& event);
     void OnLoad(           wxCommandEvent& event);
     void OnLoadTrees(      wxCommandEvent& event);
     void OnLoadRenderInfo(wxCommandEvent& event);
     void OnLoadResultDir( wxCommandEvent& event);
     void OnSetFilters(     wxCommandEvent& event);
     void OnCorrelate(      wxCommandEvent& event);
     void OnStatus(         wxCommandEvent& event);
     void OnAbout(          wxCommandEvent& event);
     void OnQuit(           wxCommandEvent& event);

     // methods for correlation

     double GetCorrThreshold();
     int    GetBlur();
```

```
    double GetGauss();

    // methods for kernel rendering

    double GetRadius();
    double GetN();
    double GetStemHeight();
    double GetCrownHeight();
    double GetX();
    double GetY();
    double GetZ();
    double GetAltitudeAngle();
    double GetAzimuthAngle();
    double GetPixelsPerMeter();

    ScrollCanvas *theScroll;

    DECLARE_EVENT_TABLE()

private:

    wxMenu    *menuPtr;
    wxMenuBar *menuBarPtr;

    // attributes for correlation

    wxString imagePath;
    wxString treePath;
    wxString renderPath;
    wxString resultPath;

    double   externPixPerMeter;
    double   corrThreshold;
    int      blur;
    double   gauss;

    // attributes for kernel rendering

    double radius;
    double n;
    double stemHeight;
    double crownHeight;
    double x;
    double y;
    double z;
    double altitudeAngle;
    double azimuthAngle;
    double pixelsPerMeter;

    enum {

        EVENT_KERNEL,
        EVENT_LOAD,
        EVENT_LOAD_TREES,
```

```
        EVENT_LOAD_RENDER,
        EVENT_LOAD_RESULT,
        EVENT_SET_FILTERS,
        EVENT_CORRELATE,
        EVENT_STATUS,
        EVENT_FOOBAR,
        EVENT_ABOUT,
        EVENT_QUIT
    };
};


// BitmapFrame

class BitmapFrame : public wxFrame {

public:

    BitmapFrame(
        wxWindow*       parentPtr,
        const wxString& title,
        const wxBitmap& bitMap);

    ~BitmapFrame();

    void OnPaint(wxPaintEvent &event);

    DECLARE_EVENT_TABLE()

private:

    wxBitmap bitmap;
};


// SluKernelInput

class SluKernelInput : public wxDialog {

public:

    SluKernelInput(ImageFrame* parentPtr);

    wxTextCtrl *radiusTextCtrl;
    wxTextCtrl *nTextCtrl;
    wxTextCtrl *stemHeightTextCtrl;
    wxTextCtrl *crownHeightTextCtrl;
    wxTextCtrl *xTextCtrl;
    wxTextCtrl *yTextCtrl;
    wxTextCtrl *zTextCtrl;
    wxTextCtrl *altitudeAngleTextCtrl;
    wxTextCtrl *azimuthAngleTextCtrl;
    wxTextCtrl *pixelsPerMeterTextCtrl;
```

```
private:

     wxButton *theOkButton;
     wxButton *theCancelButton;

};


// SluCorrelateInput

class SluCorrelateInput : public wxDialog {

public:

     SluCorrelateInput(ImageFrame* parentPtr);
     wxTextCtrl *corrTextCtrl;
     wxTextCtrl *blurTextCtrl;
     wxTextCtrl *gaussTextCtrl;

private:

     wxButton *theOkButton;
     wxButton *theCancelButton;

};


// CorrelateThread

class CorrelateThread : public wxThread {

public:

     CorrelateThread(
          ScrollCanvas* parentScrollCanvasPtr,
          wxString imagePath,
          wxString treePath,
          wxString renderPath,
          wxString resultPath,
          double   externPixPerMeter,
          double   corrThreshold,
          int      blur,
          double   gauss);

     virtual void* Entry();

     virtual void  OnExit();

private:

     ScrollCanvas* prntScrollPtr;
     wxString      imagePath;
     wxString      treePath;
```

```
    wxString        renderPath;
    wxString        resultPath;
    double          externPixPerMeter;
    double          corrThreshold;
    int             blur;
    double          gauss;

};

#endif
```

Program libraries

IPL

The Intel® Image Processing Library is a C-programming library constructed to give a high performance on Intel® Pentium processors. Standard image processing algorithms like Fast Fourier Transforms, image filtering and colour space conversion is included. The TreeD application use function-calls to this library in the main C-image-processing-loop.

http://developer.intel.com/

wxWindows

wxWindows is an open source, cross platform, C++ programming library for graphical user interface (GUI) development. The TreeD application use class declarations from this library in the user interface code. The application calls the C-main loop from a windowing environment, where the paths to the current aerial image and result directories have been set. Since the application is modular, it is possible to remove the GUI and make a pure text based interface.

http://www.wxwindows.org/

Appendix

Bibliography

Brandtberg, T. and Walter, F. (1999) An Algorithm for Delineation of Individual Tree Crowns in High Spatial Resolution Aerial Images using Curved Edge Segments at Multiple Scales. *Proceedings of Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, Victoria, British Columbia, Canada, February 10-12, 1998. (eds. Hill, D.A., and Leckie, D.G.) Natural Resources Canada, Canadian Forest Service, Pacific Forestry Centre, Victoria, British Columbia. pp.41-54.

Dralle, K. (1997) Locating Trees by Digital Image Processing of Aerial Photos. *PhD thesis.* Royal Veterinary and Agricultural University. Frederiksberg, Denmark.

Erikson, M. (2001) Structure-Keeping Colour Segmentation of Tree Crowns in Aerial Images. *Scandinavian Conference on Image Analysis (SCIA 2001),* Bergen, Norway. NOBIM, Norwegian Society for Image Processing and Pattern Recognition, pp. 185-191.

Gonzalez, R. C., Wintz, P. (1987) Digital Image Processing. Second edition. Addison-Wesley Publishing Company. pp 425-427.

Gougeon, F. A. (1999) Automatic Individual Tree Crown Delineation Using a Valley-Following Algorithm and a Rule-Based System. *Proceedings of Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, Victoria, British Columbia, Canada, February 10-12, 1998. (eds. Hill, D.A., and Leckie, D.G.) Natural Resources Canada, Canadian Forest Service, Pacific Forestry Centre, Victoria, British Columbia. pp.11-23.

Larsen, M. (1999) Finding an Optimal Match Window for Spruce Top Detection on an Optical Tree Model. *Proceedings of Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, Victoria, British Columbia, Canada, February 10-12, 1998. (eds. Hill, D.A., and Leckie, D.G.) Natural Resources Canada, Canadian Forest Service, Pacific Forestry Centre, Victoria, British Columbia. pp.55-66.

Pollock, R. J. (1996) The Automatic Recognition of Individual Trees in Aerial Images of Forests Based on a Synthetic Tree Crown Image Model, *PhD Thesis*, University of British Columbia, Vancouver, Canada.

Rudemo, M. (1999) Spatial Tree Pattern Analysis from Maxima of Smoothed Aerial Photographs, *Proceedings of Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, Victoria, British Columbia, Canada, February 10-12, 1998. (eds. Hill, D.A., and Leckie, D.G.) Natural Resources Canada, Canadian Forest Service, Pacific Forestry Centre, Victoria, British Columbia. pp.35-40.

Serien Arbetsrapporter utges i första hand för institutionens eget behov av viss dokumentation. Rapporterna är indelade i följande grupper: Riksskogstaxeringen, Planering och inventering, Biometri, Fjärranalys, Kompendier och undervisningsmaterial, Examensarbeten samt internationellt. Författarna svarar själva för rapporternas vetenskapliga innehåll.

This series of Working Papers reflects the activity of this Department of Forest Resource Management and Geomatics. The sole responsibility for the scientific content of each Working Paper relies on the respective author.

Riksskogstaxeringen: (*The Swedish National Forest Inventory*)

1995 1 Kempe, G. Hjälpmedel för bestämning av slutenhet i plant- och ungskog. ISRN SLU-SRG-AR--1--SE

2 Riksskogstaxeringen och Ståndortskarteringen vid regional miljöövervakning. - metoder för att förbättra upplösningen vid inventering i skogliga avrinningsområden. ISRN SLU-SRG-AR--2--SE.

1997 23 Lundström, A., Nilsson, P. & Ståhl, G. Certifieringens konsekvenser för möjliga uttag av industri- och energived. - En pilotstudie. ISRN SLU-SRG-AR--23--SE.

24 Fridman, J. & Walheim, M. Död ved i Sverige. - Statistik från Riksskogstaxeringen. ISRN SLU-SRG-AR--24--SE.

1998 30 Fridman, J. & Kihlblom, D. & Söderberg, U. Förslag till miljöindexsystem för naturtypen skog. ISRN SLU-SRG-AR--30--SE.

34 Löfgren, P. Skogsmark, samt träd- och buskmark inom fjällområdet. En skattning av arealer enligt internationella ägoslagsdefinitioner. ISRN SLU-SRG-AR--34--SE.

37 Odell, G. & Ståhl, G. Vegetationsförändringar i svensk skogsmark mellan 1980- och 90-talet. -En studie grundad på Ståndortskarteringen. ISRN SLU-SRG-AR--37--SE.

38 Lind, T. Quantifying the area of edge zones in Swedish forest to assess the impact of nature conservation on timber yields. ISRN SLU-SRG-AR--38--SE.

1999 50 Ståhl, G., Walheim, M. & Löfgren, P. Fjällinventering. - En utredning av innehåll och design. ISRN SLU-SRG--AR--50--SE.

52 Riksskogstaxeringen inför 2000-talet. - Utredningar avseende innehåll och omfattning i en framtida Riksskogstaxering. Redaktörer: Jonas Fridman & Göran Ståhl. ISRN SLU-SRG-AR--52--SE.

54 Fridman, J. m.fl. Sveriges skogsmarksarealer enligt internationella ägoslagsdefinitioner. ISRN SLU-SRG-AR--54--SE.

56 Nilsson, P. & Gustafsson, K. Skogsskötseln vid 90-talets mitt - läge och trender. ISRN SLU-SRG-AR--56--SE.

	57	Nilsson, P. & Söderberg, U. Trender i svensk skogsskötsel - en intervjuundersökning. ISRN SLU-SRG-AR--57--SE.
1999	61	Broman, N & Christoffersson, J. Mätfel i provträdsvariabler och dess inverkan på precision och noggrannhet i volymskattningar. ISRN SLU-SRG-AR--61--SE.
	65	Hallsby, G m.fl. Metodik för skattning av lokala skogsbränsleresurser. ISRN SLU-SRG-AR--65--SE.
	75	von Segebaden, G. Komplement till "RIKSTAXEN 75 ÅR". ISRN SLU-SRG-AR--75--SE.
2001	86	Lind, T. Kolinnehåll i skog och mark i Sverige - Baserat på Riksskogstaxeringens data. ISRN SLU-SRG-AR--86--SE

Inventering och planering: ***(Forest inventory and planning)***

1995	3	Holmgren, P. & Thuresson, T. Skoglig planering på amerikanska västkusten - intryck från en studieresa till Oregon, Washington och British Columbia 1-14 augusti 1995. ISRN SLU-SRG-AR--3--SE.
	4	Ståhl, G. The Transect Relascope - An Instrument for the Quantification of Coarse Woody Debris. ISRN SLU-SRG-AR--4--SE
1996	15	van Kerkvoorde, M. A sequential approach in mathematical programming to include spatial aspects of biodiversity in long range forest management planning. ISRN SLU-SRG-AR--15--SE.
1997	18	Christoffersson, P. & Jonsson, P. Avdelningsfri inventering - tillvägagångssätt och tidsåtgång. ISRN SLU-SRG-AR--18--SE.
	19	Ståhl, G., Ringvall, A. & Lämås, T. Guided transect sampling - An outline of the principle. ISRN SLU-SRGL-AR--19--SE.
	25	Lämås, T. & Ståhl, G. Skattning av tillstånd och förändringar genom inventeringssimulering - En handledning till programpaketet "NVSIM". ISRN SLU-SRG-AR--25--SE.
	26	Lämås, T. & Ståhl, G. Om dektering av förändringar av populationer i begränsade områden. ISRN SLU-SRG-AR--26--SE.
1999	59	Petersson, H. Biomassafunktioner för trädfraktioner av tall, gran och björk i Sverige. ISRN SLU-SRG-AR--59--SE.
	63	Fridman, J., Löfstrand, R & Roos, S. Stickprovsvis landskapsövervakning - En förstudie. ISRN SLU-SRG-AR--63--SE.
2000	68	Nyström, K. Funktioner för att skatta höjdtillväxten i ungskog. ISRN SLU-SRG-AR--68--SE.

70 Walheim, M. & Löfgren, P. Metodutveckling för vegetationsövervakning i fjällen. ISRN SLU-SRG-AR--70--SE.

73 Holm, S. & Lundström, A. Åtgärdsprioriteter. ISRN SLU-SRG-AR--73--SE.

76 Fridman, J. & Ståhl, G. Funktioner för naturlig avgång i svensk skog. ISRN SLU-SRG-AR--76--SE.

2001 82 Holmström, H. Averaging Absolute GPS Positionings Made Underneath Different Forest Canopies - A Splendid Example of Bad Timing in Research. ISRN SLU-SRG-AR--79--SE.

2002 91 Wilhelmsson, E. Forest use and its economic value for inhabitants of Skröven and Hakkas in Norrbotten. ISRN SLU-SRG-AR--91--SE.

94 Eriksson, O. m fl. Wood Supply From Swedish Forests Managed According to the FSC-standard. ISRN SLU-SRG-AR--94--SE.

Biometri: ***(Biometrics)***

1997 22 Ali, Abdul Aziz. Describing Tree Size Diversity. ISRN SLU-SRG-AR--22--SE.

1999 64 Berhe, L. Spatial continuity in tree diameter distribution. ISRN SLU-SRG-AR--64--SE

2001 88 Ekström, M. Nonparametric Estimation of the Variance of Sample Means Based on Nonstationary Spatial Data. ISRN SLU-SRG-AR--88--SE.

89 Ekström, M. & Belyaev, Y. On the Estimation of the Distribution of Sample Means Based on Non-Stationary Spatial Data. ISRN SLU-SRG-AR--89--SE.

90 Ekström, M. & Sjöstedt-de Luna, S. Estimation of the Variance of Sample Means Based on Nonstationary Spatial Data with Varying Expected Values. ISRN SLU-SRG-AR--90--SE.

2002 96 Norström,F. Forest inventory estimation using remotely sensed data as a stratification tool - a simulation study. ISRN SLU-SRG-AR--96--SE.

Fjärranalys: ***(Remote Sensing)***

1997 28 Hagner, O. Satellitfjärranalys för skogsföretag. ISRN SLU-SRG-AR--28--SE.

29 Hagner, O. Textur till flygbilder för skattning av beståndsegenskaper. ISRN SLU-SRG-AR--29--SE.

1998 32 Dahlberg, U., Bergstedt, J. & Pettersson, A. Fältinstruktion för och erfarenheter från vegetationsinventering i Abisko, sommaren 1997. ISRN SLU-SRG-AR--32--SE.

43 Wallerman, J. Brattåkerinventeringen. ISRN SLU-SRG-AR--28--SE.

1999 51 Holmgren, J., Wallerman, J. & Olsson, H. Plot - Level Stem Volume Estimation and Tree Species Discrimination with Casi Remote Sensing.
ISRN SLU-SRG-AR--51--SE.

53 Reese, H. & Nilsson, M. Using Landsat TM and NFI data to estimate wood volume, tree biomass and stand age in Dalarna. ISRN SLU-SRG-AR--53--SE.

2000 66 Löfstrand, R., Reese, H. & Olsson, H. Remote Sensing aided Monitoring of Non-Timber Forest Resources - A literature survey. ISRN SLU-SRG-AR--66--SE.

69 Tingelöf, U & Nilsson, M.Kartering av hyggeskanter i pankromaötiska SPOT-bilder.
ISRN SLU-SRG-AR--69--SE.

79 Reese, H & Nilsson, M. Wood volume estimations for Älvsbyn Kommun using SPOT satellite data and NFI plots. ISRN SLU-SRG-AR--79--SE.

2003 106 Olofsson, K. TreeD version o.8. An Image Processing Application for Single Tree Detection.
ISRN SLU-SRG-AR--106--SE.

Kompendier och undervisningsmaterial: ***(Compendia and educational papers)***

1996 14 Holm, S. & Thuresson, T. samt jägm.studenter kurs 92/96. En analys av skogstill-ståndet samt några alternativa avverkningsberäkningar för en del av Östads säteri.
ISRN SLU-SRG-AR--14--SE.

21 Holm, S. & Thuresson, T. samt jägm.studenter kurs 93/97. En analys av skogsstill-ståndet samt några alternativa avverkningsberäkningar för en stor del av Östads säteri. ISRN SLU-SRG-AR--21--SE.

1998 42 Holm, S. & Lämås, T. samt jägm.studenter kurs 93/97. An analysis of the state of the forest and of some management alternatives for the Östad estate.
ISRN SLU-SRG-AR--42--SE

1999 58 Holm, S. samt studenter vid Sveriges lantbruksuniversitet i samband med kurs i strate-gisk och taktisk skoglig planering år 1998. En analys av skogsstillståndet samt några alternativa avverknings beräkningar för Östads säteri. ISRN SLU-SRG-AR--58--SE.

2001 87 Eriksson, O. (Ed.) Strategier för Östads säteri: Redovisning a v planer framtagna under kursen Skoglig planering ur ett företagsperspektiv HT2000, SLU Umeå.
ISRN SLU-SRG-AR--87--SE..

2002 93 Lind, T. (Ed.) Strategier för Östads säteri: Redovisning av planer framtagna under kursen Skoglig planering ur ett företagsperspektiv HT2001, SLU Umeå.
ISRN SLU-SRG-AR--93--SE..

Examensarbeten: *(Theses by Swedish forestry students)*

1995 5 Törnquist, K. Ekologisk landskapsplanering i svenskt skogsbruk - hur började det?. Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--5--SE.

1996 6 Persson, S. & Segner, U. Aspekter kring datakvaliténs betydelse för den kortsiktiga planeringen. Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--6--SE.

7 Henriksson, L. The thinning quotient - a relevant description of a thinning? Gallringskvot - en tillförlitlig beskrivning av en gallring? Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--7--SE.

8 Ranvald, C. Sortimentsinriktad avverkning. Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--8--SE.

9 Olofsson, C. Mångbruk i ett landskapsperspektiv - En fallstudie på MoDo Skog AB, Örnsköldsviks förvaltning. Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--9--SE.

10 Andersson, H. Taper curve functions and quality estimation for Common Oak (Quercus Robur L.) in Sweden. Examensarbete i ämnet skogsuppskattning och skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--10--SE.

11 Djurberg, H. Den skogliga informationens roll i ett kundanpassat virkesflöde. - En bakgrundsstudie samt simulering av inventeringsmetoders inverkan på noggrannhet i leveransprognoser till sågverk. Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--11--SE.

12 Bredberg, J. Skattning av ålder och andra beståndsvariabler - en fallstudie baserad på MoDo:s indelningsrutiner. Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--12--SE.

13 Gunnarsson, F. On the potential of Kriging for forest management planning. Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--13--SE.

16 Tormalm, K. Implementering av FSC-certififering av mindre enskilda markägares skogsbruk. Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--16--SE.

1997 17 Engberg, M. Naturvärden i skog lämnad vid slutavverkning. - En inventering av upp till 35 år gamla föryngringsytor på Sundsvalls arbetsomsåde, SCA. Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN-SLU-SRG-AR--17--SE.

20 Cedervind, J. GPS under krontak i skog. Examensarbete i ämnet skogsuppskattning och skogsindelning. ISRN SLU-SRG-AR--20--SE.

27 Karlsson, A. En studie av tre inventeringsmetoder i slutavverkningsbestånd. Examensarbete. ISRN SLU-SRG-AR--27--SE.

1998 31 Bendz, J. SÖDRAs gröna skogsbruksplaner. En uppföljning relaterad till SÖDRAs miljömål, FSC's kriterier och svensk skogspolitik. Examensarbete. ISRN SLU-SRG-AR--31--SE.

33 Jonsson, Ö. Trädskikt och ståndortsförhållanden i strandskog. - En studie av tre bäckar i Västerbotten. Examensarbete. ISRN SLU-SRG-AR--33--SE.

35 Claesson, S. Thinning response functions for single trees of Common oak (Quercus Robur L.) Examensarbete. ISRN SLU-SEG-AR--35--SE.

36 Lindskog, M. New legal minimum ages for final felling. Consequences and forest owner attitudes in the county of Västerbotten. Examensarbete. ISRN SLU-SRG-AR--36--SE.

40 Persson, M. Skogsmarksindelningen i gröna och blå kartan - en utvärdering med hjälp av riksskogstaxeringens provytor. Examensarbete. ISRN SLU-SRG-AR--40--SE.

41 Eriksson, F. Markbaserade sensorer för insamling av skogliga data - en förstudie. Examensarbete. ISRN SLU-SRG-AR--41--SE.

45 Gessler, C. Impedimentens potientiella betydelse för biologisk mångfald. - En studie av myr- och bergimpediment i ett skogslandskap i Västerbotten. Examensarbete. ISRN SLU-SRG-AR--45--SE.

46 Gustafsson, K. Långsiktsplanering med geografiska hänsyn - en studie på Bräcke arbetsområde, SCA Forest and Timber. Examensarbete. ISRN SLU-SRG-AR--46--SE.

47 Holmgren, J. Estimating Wood Volume and Basal Area in Forest Compartments by Combining Satellite Image Data with Field Data. Examensarbete i ämnet Fjärranalys. ISRN SLU-SRG-AR--47--SE.

49 Härdelin, S. Framtida förekomst och rumslig fördelning av gammal skog. - En fallstudie på ett landskap i Bräcke arbetsområde. Examensarbete SCA. ISRN SLU-SRG-AR--49--SE.

1999 55 Imamovic, D. Simuleringsstudie av produktionskonsekvenser med olika miljömål. Examensarbete för Skogsstyrelsen. ISRN SLU-SRG-AR--55--SE

62 Fridh, L. Utbytesprognoser av rotstående skog. Examensarbete i skoglig planering. ISRN SLU-SRG-AR--62--SE.

2000 67 Jonsson, T. Differentiell GPS-mätning av punkter i skog. Point-accuracy for differential GPS under a forest canaopy. ISRN SLU-SRG-AR--67--SE.

71 Lundberg, N. Kalibrering av den multivariata variabeln trädslagsfördelning. Examensarbete i biometri. ISRN SLU-SRG-AR--71--SE.

72 Skoog, E. Leveransprecision och ledtid - två nyckeltal för styrning av virkesflödet. Examensarbete i skoglig planering. ISRN SLU-SRG-AR--72--SE.

74 Johansson, L. Rotröta i Sverige enligt Riksskogstaxeringen. Examens arbete i ämnet skogsindelning och skogsuppskattning. ISRN SLU-SRG-AR--74--SE.

77 Nordh, M. Modellstudie av potentialen för renbete anpassat till kommande slut-avverkningar. Examensarbete på jägmästarprogrammet i ämnet skoglig planering. ISRN SLU-SRG-AR--77--SE.

78 Eriksson, D. Spatial Modeling of Nature Conservation Variables useful in Forestry Planning. Examensarbete. ISRN SLU-SRG-AR--74--SE.

81 Fredberg, K. Landskapsanalys med GIS och ett skogligt planeringssystem. Examensarbete på skogsvetarprogrammet i ämnet skogshushållning. ISRN SLU-SRG-AR--81--SE.

2001 83 Lindroos, O.Underlag för skogligt länsprogram Gotland. Examensarbete i ämnet skoglig planering. ISRN SLU-SRG-AR--83-SE.

84 Dahl, M. Satellitbildsbaserade skattningar av skogsområden med röjningsbehov. Examensarbete på akogsvetarprogrammet i ämnet skoglig planering. ISRN SLU-SRG-AR--84--SE.

85 Staland, J. Styrning av kundanpassade timmerflöden - Inverkan av traktbankens stlorlek och utbytesprognosens tillförlitlighet. Examensarbete i ämnet skoglig planering. ISRN SLU-SRG-AR--85--SE.

2002 92 Bodenhem, J. Tillämpning av olika fjärranalysmetoder för urvalsförfarandet av ungskogsbe stånd inom den enkla älgbetesinventeringen (ÄBIN). Examensarbete på skogsvetarprogram met i ämnet fjärranalys. ISRN SLU-SRG-AR--92--SE.

95 Sundquist, S. Development of a measure of production density for the Swedish National Forest Inventory. ISRN SLU-SEG-AR--95--SE.

98 Söderholm, J. De svenska skogsbolagens system för skoglig planering. *The planning system of Swedish forest companies*. Examensarbete på skogsvetarprogrammet i ämnet skoglig planering. ISRN SLU-SRG-AR--98--SE.

99 Nordin, D. Fastighetsgränser. Del 1. Fallstudie av fastighetsgränsers lägesnoggrannhet på fastighetskartan. Examensarbete på skogliga magisterprogrammet i ämnet skoghushållning med inriktning skoglig planering. ISRN SLU-SRG-AR--99--SE.

100 Nordin, D. Fastighetsgränser. Del 2. Instruktion för gränsvård. Examensarbete på skogliga magisterprogrammet i ämnet skoghushållning med inriktning skoglig planering. ISRN SLU-SRG-AR--100--SE.

101 Nordbrant, A. Analyser med Indelningspaketet av prIvata skogsfastigheter inom Norra Skogsägarnas verksamhetsområde. Examensarbete på skogsvetarprogrammet i ämnet skoglig planering. ISRN SLU-SRB-AR--101--SE.

2003 102 Wallin, M. Satellitbildsanalys av gremmeniellaskador med skogsvårdsorganisationens system. Examensarbete på skogsvetarprogrammet i ämnet skogshushållning med insriktning fjärranalys. ISRN SLU-SRG-AR--102--SE.

103 Hamilton, A. Effektivare samråd mellan rennäring och skogsbruk - förbättrad dialog via ett utvecklat samrådsförfarande. More effective consultations between reindeer herding and forestry - improved dialog by a developed constultation process. Examensarbete på skogsvetarprogrammet i ämnet skoghushållning. ISRN SLU-SRG-AR--103--SE.

104 Hajek, F. Mapping of Intact Forest Landscapes in Sweden according to Global Forest Watch methodology. MSc Thesis in forest Resource management, specialization in remote sensing. ISRN SLU-SRG-AR--104--SE.

105 Anerud, E. Kalibrering av ståndortsindex i beståndsregister - en studie åt Holmen Skog AB. Examensarbete på skogsvetarprogrammet i ämnet skoglig planering.
ISRN SLU-SEG-AR--105--SE.

Internationellt: ***(International issues)***

1998 39 Sandewall, Ohlsson, B & Sandewall, R.K. People´s options on forest land use - a research study of land use dynamics and socio-economic conditions in a historical perspective in the Upper Nam Nan Water Catchment Area, Lao PDR.
ISRN SLU-SRG-AR--39--SE.

44 Sandewall, M., Ohlsson, B., Sandewall, R.K., Vo Chi Chung, Tran Thi Binh & Pham Quoc Hung. People´s options on forest land use. Government plans and farmers intentions - a strategic dilemma. ISRN SLU-SRG-AR--44--SE.

48 Sengthong, B. Estimating Growing Stock and Allowable Cut in Lao PDR using Data from Land Use Maps and the National Forest Inventory (NFI). Masters thesis.
ISRN SLU-SRG-AR--48--SE.

1999 60 Inter-active and dynamic approaches on forest and land-use planning - proceedings from a training workshop in Vietnam and Lao PDR, April 12-30, 1999.
Edited by Mats Sandewall ISRN SLU-SRG-AR--60--SE.

2000 80 Sawathvong. S. Forest Land Use Planning in Nam Pui National Biodiversity Conservation Area, Lao P.D.R. ISRN SLU-SRG-AR--80--SE.

2002 97 Inter-active and dynamic approaches on forest and land-use planning in Southern Africa - proceedings from a training workshop in Botswana, December 3-17, 2001.
Edited by Mats Sandewall. ISRN SLU-SRG-AR--97--SE.