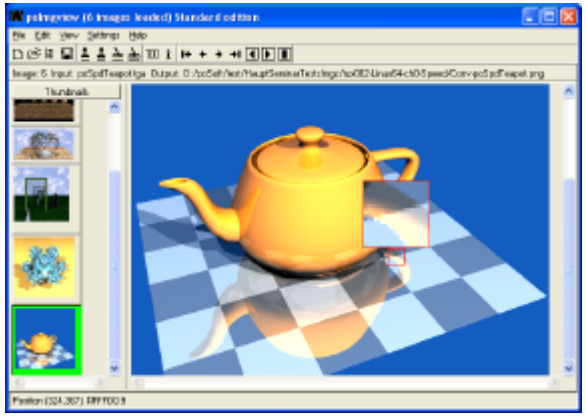
tkImg Application Examples



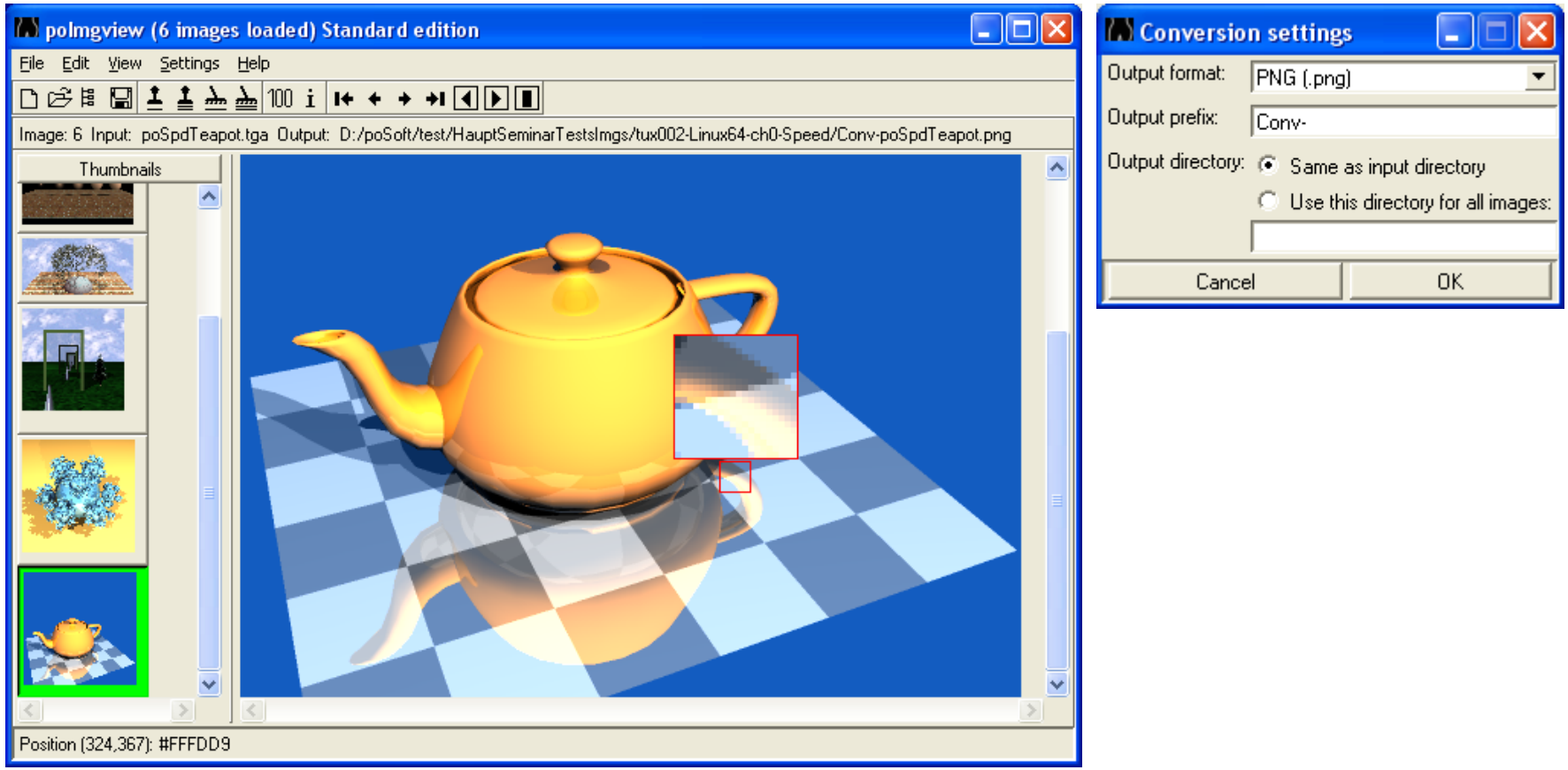
poImgview
A portable image viewer.

poImgdiff
A portable image comparison program.

tkImgTests
A series of new tkImg tests.

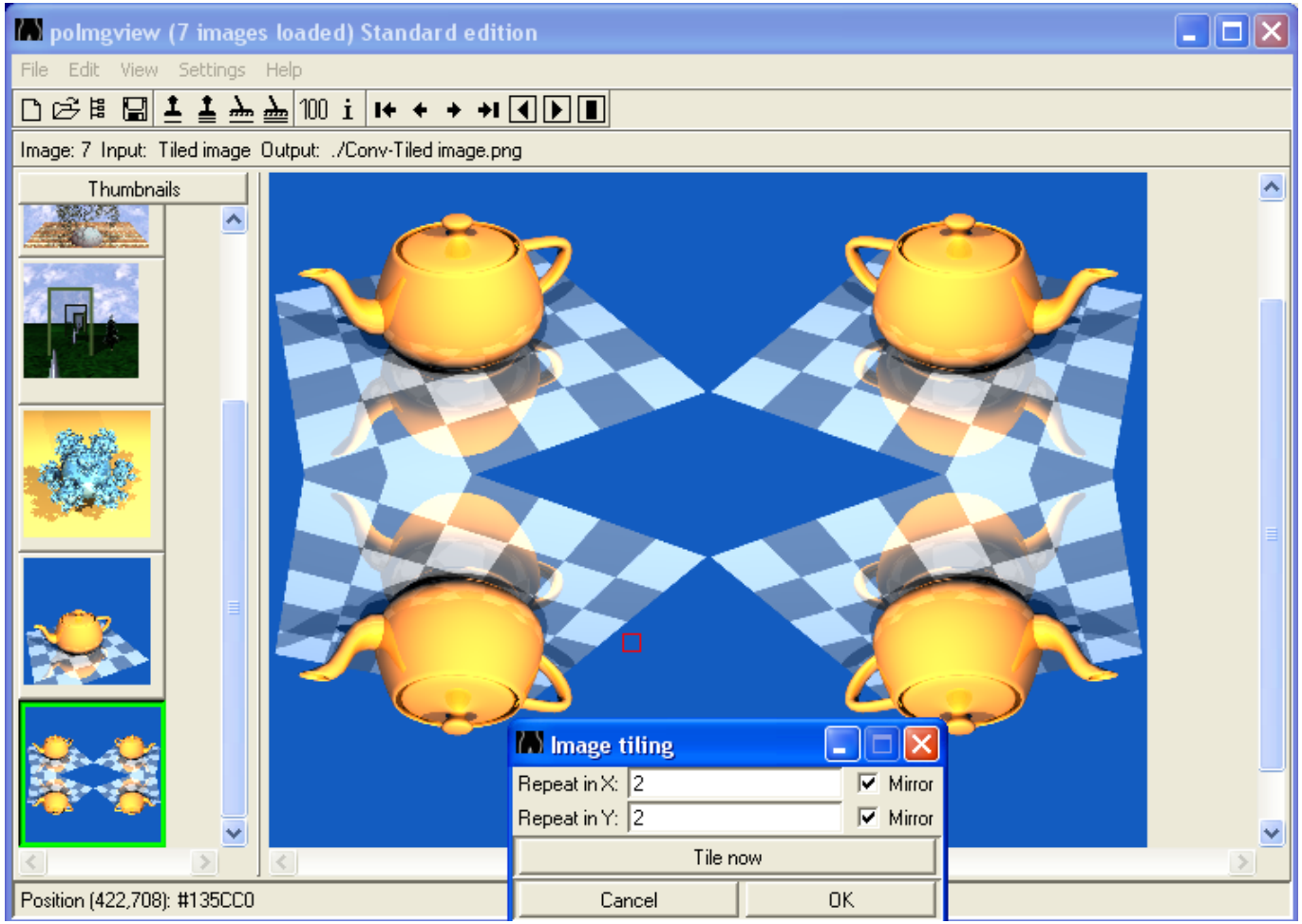
poImgview: A portable image viewer

Main window containing the scrollable image window, the preview list of loaded images and the zoom rectangle.
poImgview also supports batch format conversion.



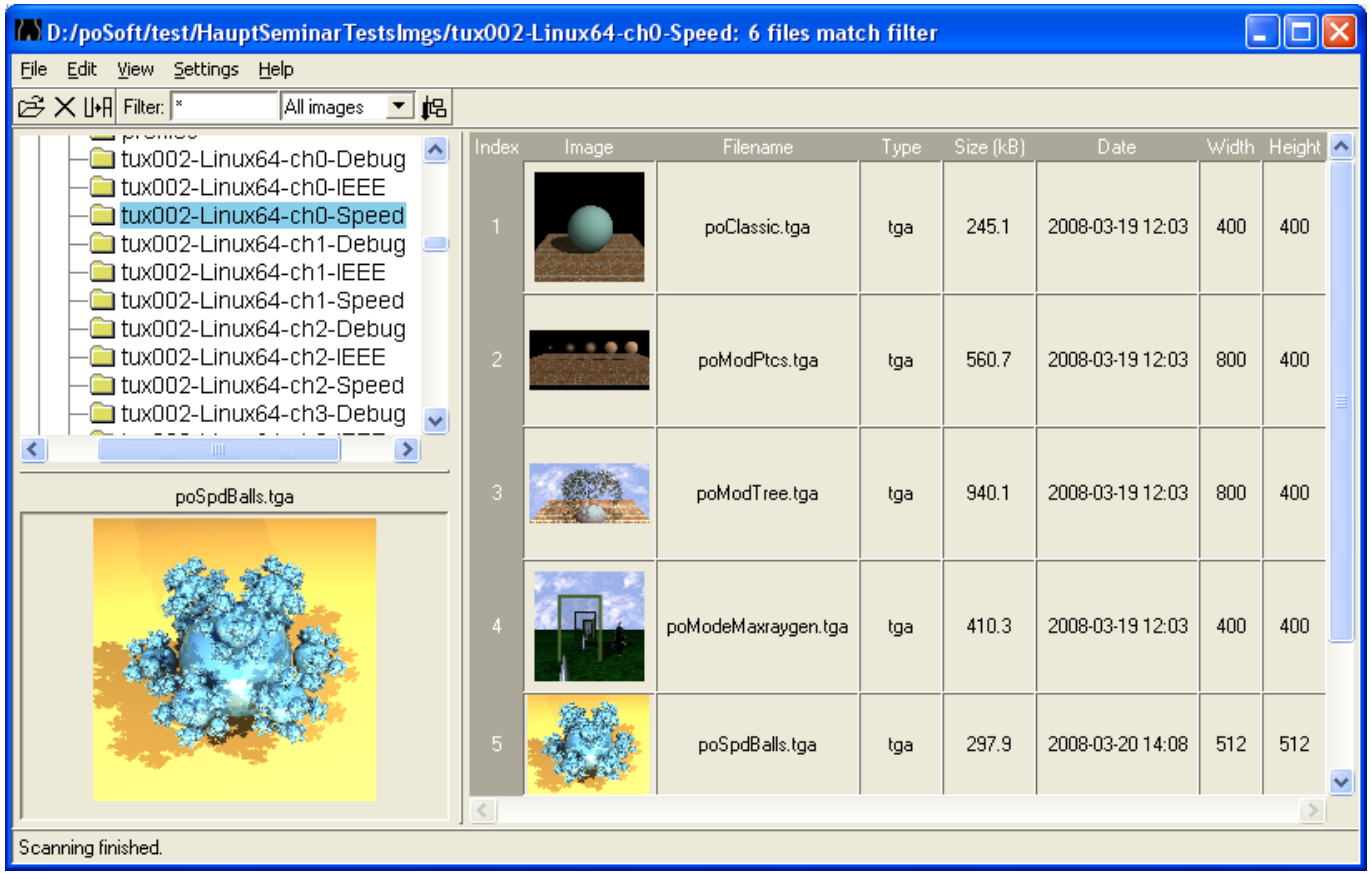
poImgview: Tiling functionality

poImgview allows generation of tiled images.



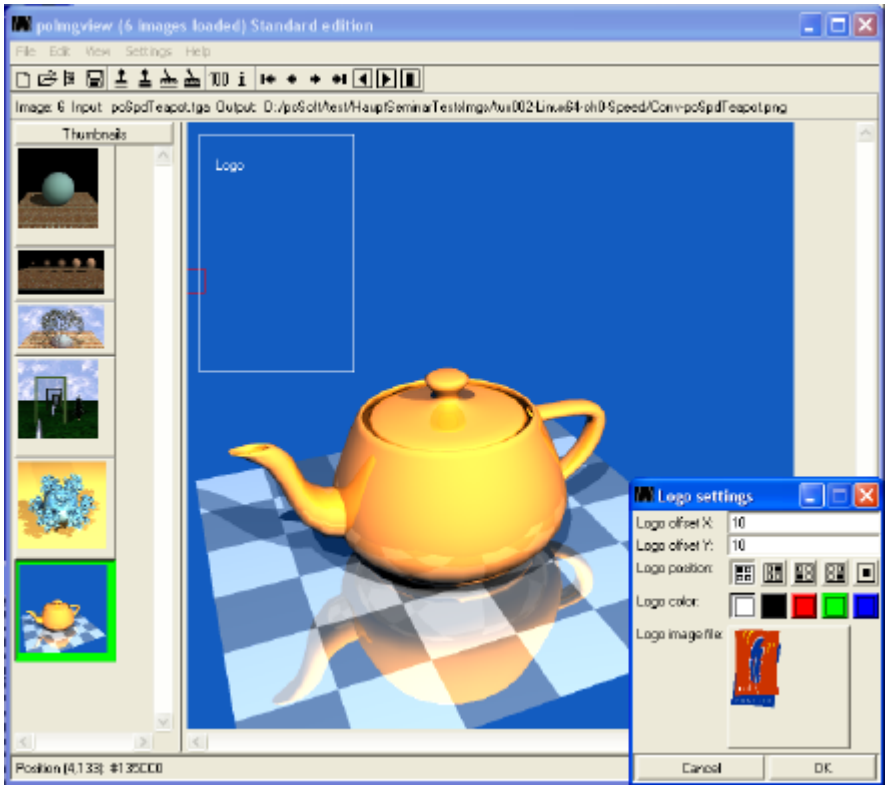
poImgview: Image browser

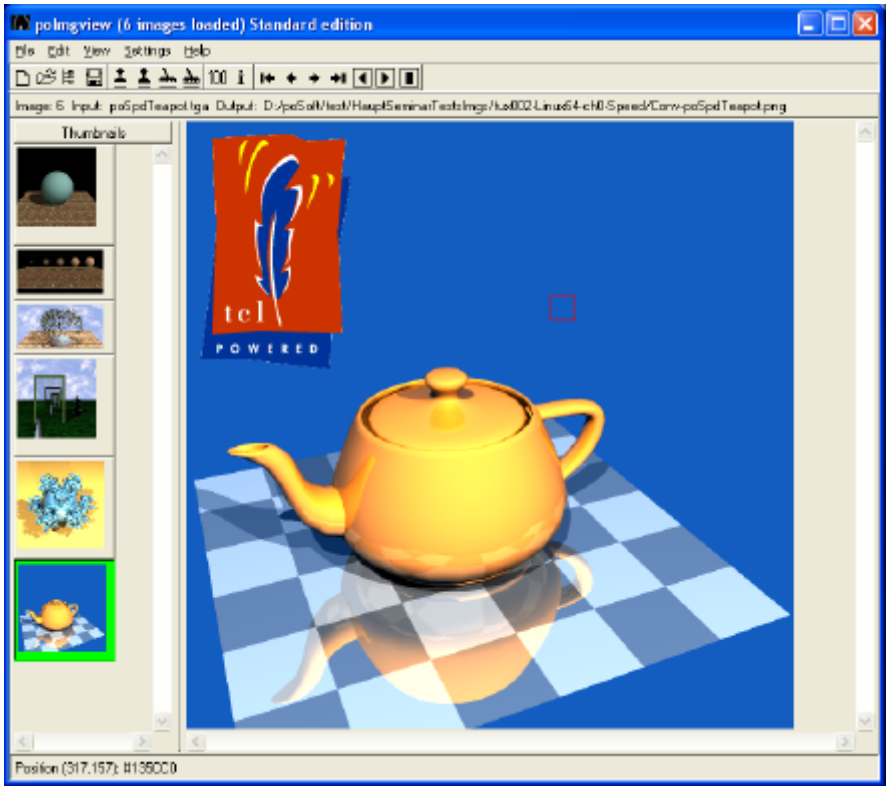
The image browser window contains a directory tree view, a preview window and a TkTable based browse window.
Additional features via context menu: Fullscreen slideshow and reorganization (copy/move).



poImgview: Logo addition

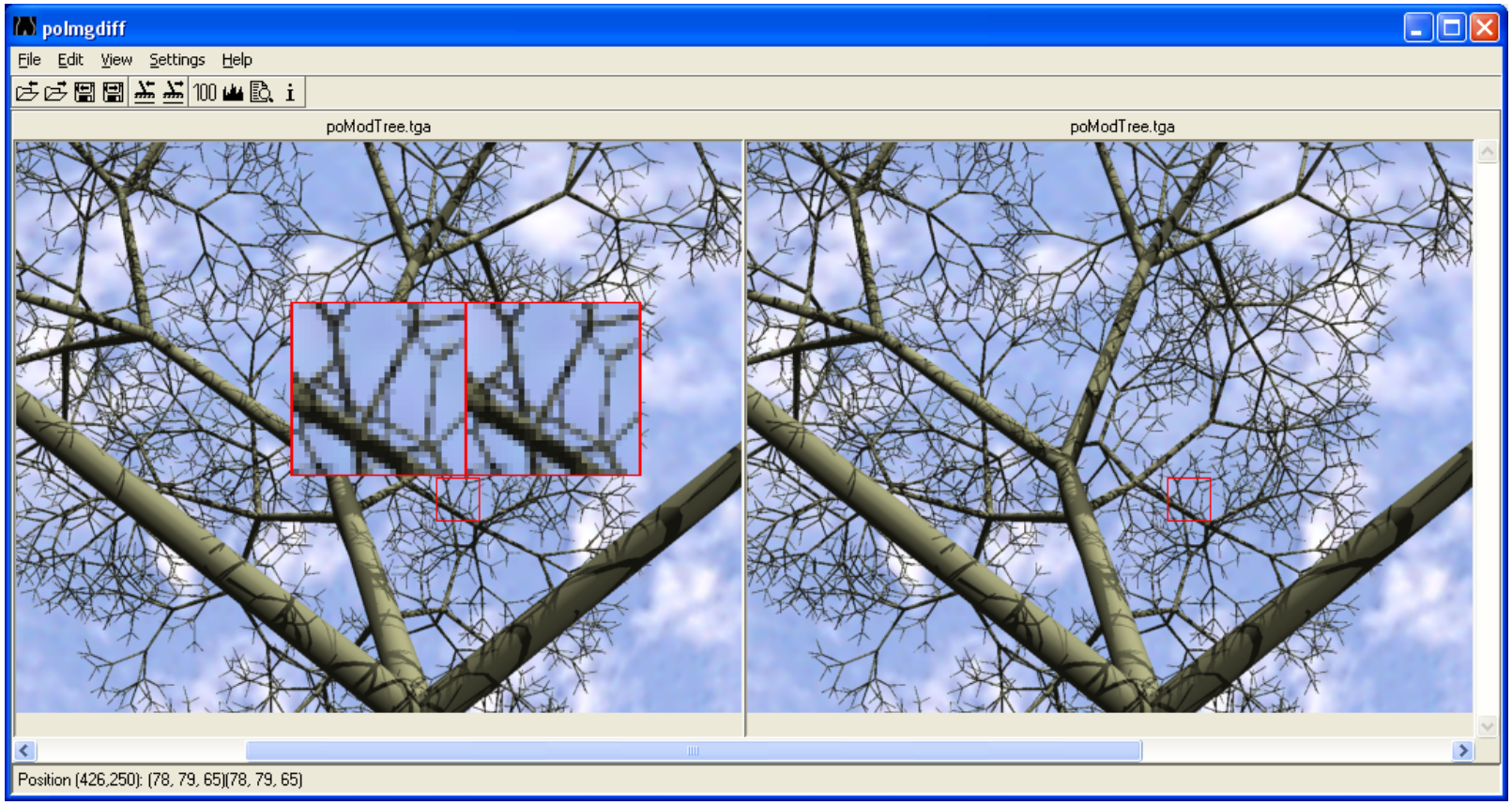
poImgview contains a batch mode function for adding logos to a series of images.





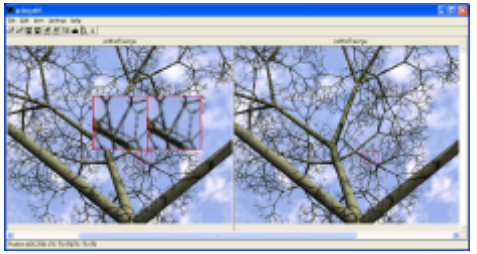
poImgdiff: A portable image comparison program

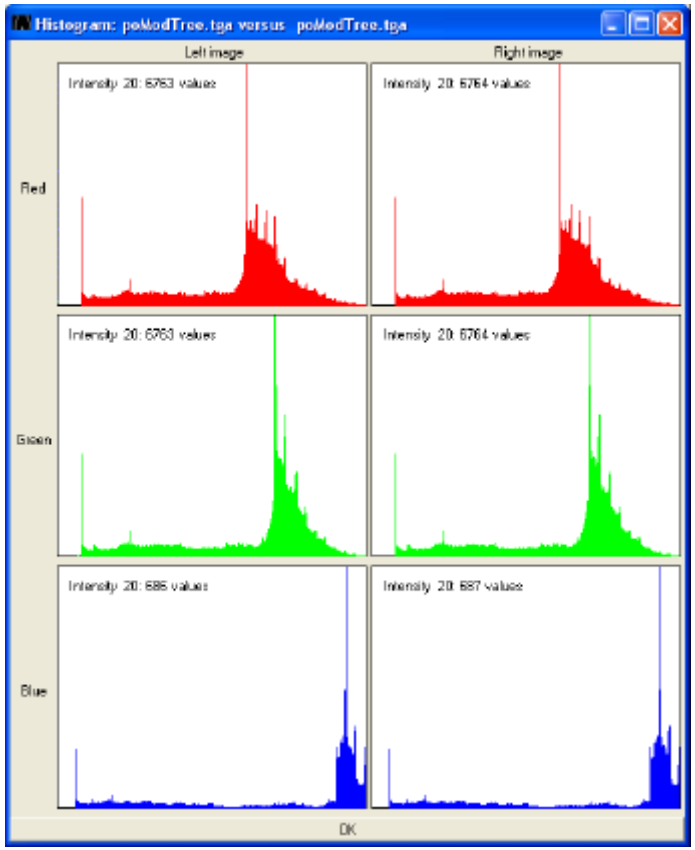
Main window allows a side-by-side view of two images.
For detailled inspection two side-by-side zoom rectangles are available.

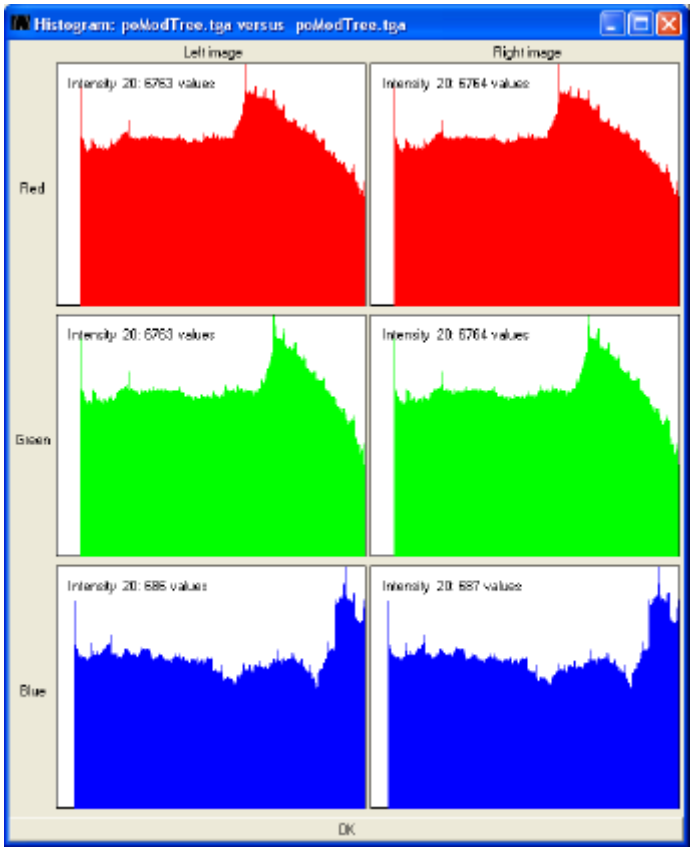


poImgdiff: Histograms

Histograms of the two images can be generated.
The histograms can be scaled uniformly or logarithmic.

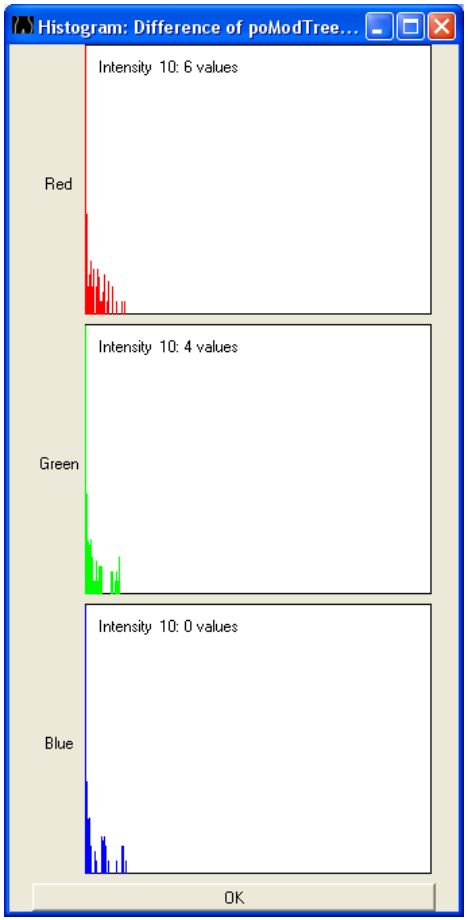






poImgdiff: Difference images

A difference image can be generated and adjusted interactively.
A histogram of the difference image gives an overview and detailed knowledge about the number of different pixels.



tkImgTest: A series of new tkImg tests.

t_imgio1.tcl: Read and write full images
t_imgio2.tcl: Read and write images with option "-from"
t_imgio3.tcl: Read and write images with option "-to"
t_imgio4.tcl: Read and write images with sizes from 1x1 to 4x4

The following ways to read image data are tested:

```
Read from file 1:          image create photo -file $fileName
Read from file 2:          set ph [image create photo] ; $ph read $fileName
Read as binary 1:          image create photo -data $imgData
Read as binary 2:          set ph [image create photo] ; $ph put $imgData
Read as uuencoded string: set ph [image create photo] ; $ph put $imgData
```

The following ways to write image data are tested:

```
Write to file:             $ph write $fileName -format $fmt
Write to uuencoded string: $ph data -format $fmt
```