

dgc_tool v1.0. brief instructions

Yll Haxhimusa

Pattern Recognition and Image Processing Group 183/2

Institute for Computer Aided Automation, Vienna University of Technology

yll@prip.tuwien.ac.at

First *dgc_tool* and LEDA library must be installed.

- - Download *dgc_tool* from the ftp://ftp.prip.tuwien.ac.at/pub/dgc_tool/ and read carefully the instruction how to install the LEDA Library. More information about *dgc_tool* can be found in the technical report TR-69-PRIP, download it from <http://www.prip.tuwien.ac.at/yll/publication.html/>

The command synopsis for the *dgc_tool* is :

```
dgc_tool option [filename]
```

We have these options:

- -i – Internal. A small picture is used for testing purposes.
- -t – TIFF¹ images. Only no compressed tiff images can be used.
- -o – Output. To down-project the basins into an output file.

During the executions of the *dgc_tool* a menu is given to navigate throught the program. To make the usage of the *dgc_tool* more clearly some examples are given.

- To test that everything is properly installed² the user should type the command:

```
dgc_tool -i
```

followed by the input 1 for gray level onto borders, 1 for the maximum independent algorithm and 1 for the application.

- Tiff images can be put into the *dgc_tool* with the command:

```
dgc_tool -t tiff_filename1 [-o tiff_filename2]
```

¹Tag Image File Format. More info <http://www.adobe.com>

² To check for example if LEDA libraries are properly installed.

- For example:

- To do connected component analysis on tiff image for example the user should type:

```
dgc_tool -t circle.tif
```

followed by the input 1 (gray levels on borders), 1 (maximal independent algorithm) and 2 (connected component analysis) yields a graph consisting of two self-loops - one for the border of the circle, the other one for the edges of the image.

- Or another example:

```
dgc_tool -t circle_soebel.tif
```

followed by the input 1 (gray levels on borders), 1 (maximal independent algorithm) and 3 (watershed segmentation) yields a graph consisting of two self-loops - one for the border of the circle, the other one for the edges of the image.

- To do watershed segmentation the options `-t` plus the filtered image and `-o` plus the name of the image (makes sense for monotonic dual graph contraction, only) are chosen.

```
dgc_tool -t circle_sobel.tif -o circle.tif
```

followed by the input 1 (gray levels on borders), 1 (maximal independent algorithm) and 3 (watershed segmentation) yields down-projection of basins into *Output.pgm*. The basins are filled with the mean gray levels from original image *circle.tif*.

- In both examples the dual of the graph is shown, if "done" is pressed in the graph window. In the dual graph the background vertex is depicted by the large square.
- Upon completion you can manipulate the graphs and save it in *.GML*, *.GW* (LEDA based graph formats) or in PostScript *.PS* format.

For more insight into *dgc_tool* see TR-69-PRIP.