PINK image processing library

M. Couprie

Université Paris-Est - LIGM-A3SI - ESIEE, France

27/06/2012

History

- Started as a personnal project (15 years ago)
- Inspired by Khoros/Cantata (Pink Is Not Khoros!)
- Demand-driven development

Users/public

- ESIEE students
- PhD students
- Researchers in LIGM/ESIEE
- Academic partners (INSERM, Hôpital Henri Mondor, ICMCB, CSIRO. . .)
- Industrial partners (CEA, SANOFI, Saint Gobain, EDF, Lafarge SA. . .)

Goals

- Fast/easy development
- Easy handling for students
- Up-to-date algorithms in our expertise field
- Supporting our applicative/collaborative projects

Authors

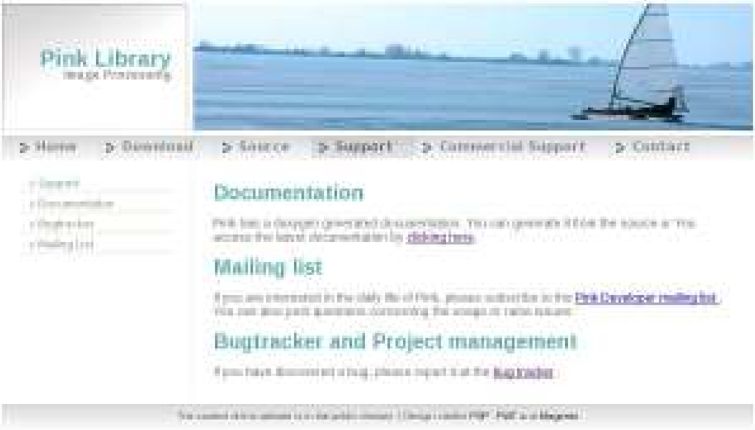
Many contributors, including ESIEE students and PhD students, the main developpers are:

- Michel Couprie
- Jean Cousty
- Laszlo Marak
- Laurent Najman
- Hugues Talbot

Distribution, license, support

- CeCILL license
- web site: pinkhq.com
- source code available from Mercurial repository
- Linux distributions
- OSX
- Microsoft Windows
- Doxygen generated documentation, mailing list, bug tracker

Web site (created and maintained by L. Marak)



Documentation

Pink has a doxygen generated documentation. You can generate it from the source or you access the latest documentation by clicking here.

Mailing list

If you are interested in the daily life of Pink, please subscribe to the Pink Developer mailing list. You can also post questions concerning the usage or raise issues.

Bugtracker and Project management

If you have discovered a bug, please report it at the bugtracker.

[illegible]

Implementation

- Core library in standard C language
- C++ for some operators and wrappers
- Scripts in bash, TCL
- Python front end
- Python-TK GUI
- Python-VTK for 3D visualization

Content

$>$ 200 operators - main modules:

- Mathematical morphology (45)
- Digital connectivity (41)
- Digital topology, binary (48)
- Digital topology, grayscale (33)
- Orders topology (27)
- Geometrical operators (68)

Content

Mathematical morphology

- Basics: erode, dilate, open, close. . .
- Binary and grayscale, 2D and 3D
- Arbitrary structuring elements
- Higher level operators (alternate sequential filter. . .)
- Distance maps, medial axis, morphological skeletons

Content

Digital connectivity

- Component labelling, geodesic operators
- Watershed transformations (2D, 3D, 4D)
- Connective filtering, component tree building/manipulation

Content

Digital topology

- Topology-preserving thinning (2D, 3D)
- Proven topology-related properties (Gilles Bertrand)
- Constrained/guided topological transformations
- Sequential, parallel transformations
- Detection of local topological features
- Controlled modifications of topology (3D hole closing...)

Content

Grayscale digital topology

Binary digital topology is generalized to grayscale images by considering the level sets of the image:
Functions F and G are homotopic if F_k and G_k are homotopic (in the binary sense), for all k

- Grayscale thinning, skeletons
- Topological filtering
- Crest restoration...

Content

Orders topology

- Cubical(/simplicial) complexes (points, edges, squares, cubes...)
- Sound and rich framework for topology in discrete spaces
- Models and extends digital topology
- Easier handling of some topogical notions (dimension...)
- New results, algorithms, properties...

Content

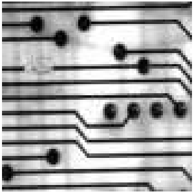
Geometrical operators

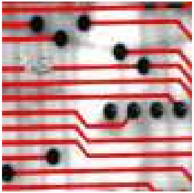
- Shape analysis (measurements, moments. . .)
- Discrete/continuous conversions (splines. . .)
- Geometric primitives detection (lines/planes, circles, ellipses. . .)

Illustrations

Python script for a segmentation scheme

```
def FindTheWires(image, threshold):
    binary = pink.threshold(image, threshold, 0, 255)
    inv = pink.inverse(binary)
    eros = pink.erosball(inv, 2)
    filtered = pink.geodilat(eros, inv, 8)
    filled = fill_the_holes(filtered)
    open = pink.openball(filled, 6)
    joints = pink.geodilat(open, filled, 8, 1)
    result = filled - joints
    return result
```

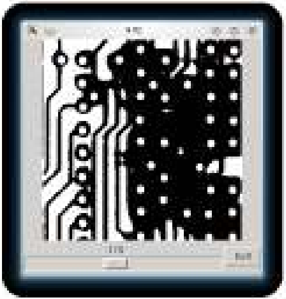




Illustrations

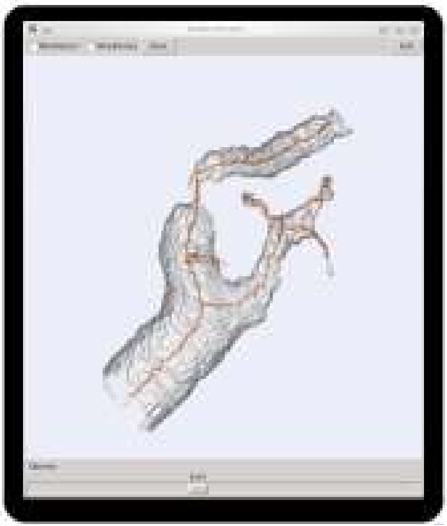
Python script for interactive manipulation of parameter

```
Im = pink.cpp.readimage(”circuit2.pgm”)

def binarise(value)
    global Im
    return pink.threshold(Im, value)

pink.manipulate(binarise, 0, 255)
```



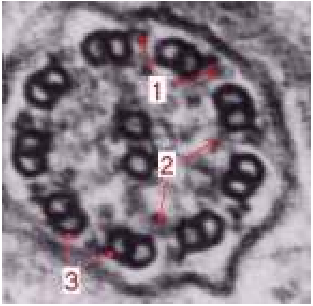
Illustrations

3D visualisation with Python-VTK



Examples of projects

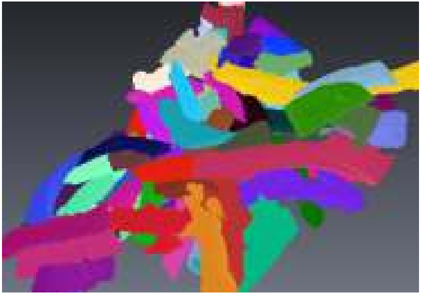
Analysis of biomedical images - computer aided diagnosis



Collaboration with INSERM

Examples of projects

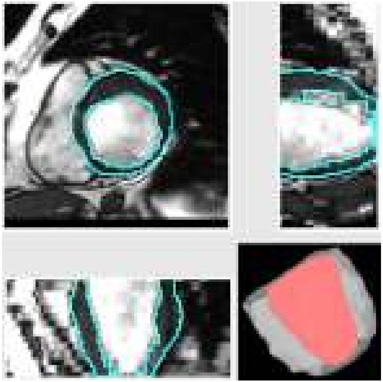
Analysis of fibrous material from 3D microtomography images



Collaboration with Lafarge SA, ICMCB and ITASCA

Examples of projects

4D segmentation of the beating heart



Collaboration with Henri Mondor Hospital