



Pink is an image processing and image analysis library developed at ESIEE Engineering for research and teaching purposes. It is build upon Boost Python, the Python Imaging Library, and uses plug-ins that like python-vtk, numpy and matplotlib. While the core foundation has been in development for over a decade, its Python front-end is recent work.



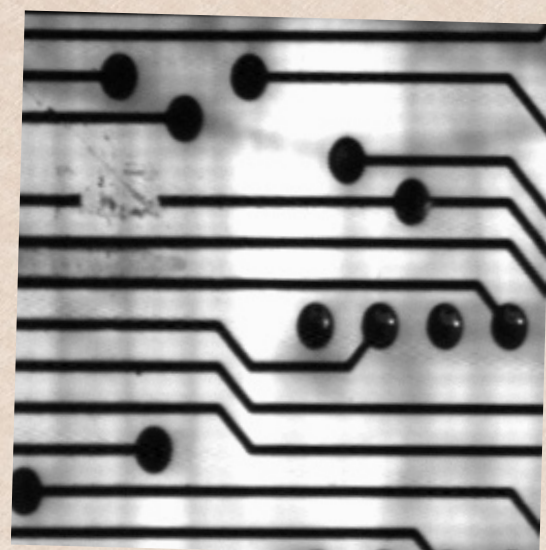
The Pink Image Processing Library

<http://pinkhq.com>

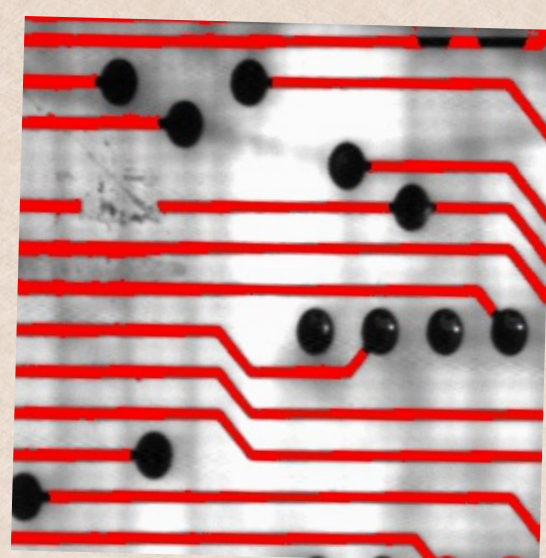
Point of departure

- Given a set of image processing operators
- Using peer-reviewed algorithms
- Implemented in C/C++

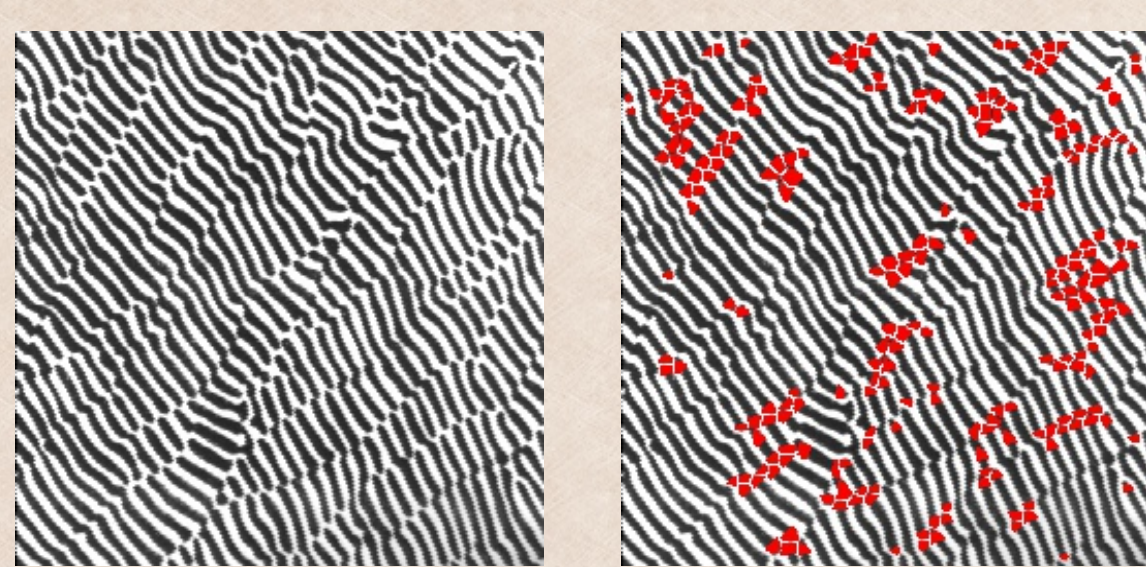
Right. With mathematical morphology the wires on the printed circuit can be easily segmented. With Python the necessary operators are simply chained. The corresponding code can be seen below.



```
def extract_fractures(image):
    seuil = pink.seuil(image, 123)
    inv = pink.inverse(seuil)
    skeleton1 = pink.skelcurv(inv, 0, 8)
    endpoints = pink.ptend(skeleton1, 8)
    dilated = pink.dilatball(endpoints, 1, 8)
    sub = skeleton1 - dilated
    adt = endpoints + sub
    inv = pink.inverse(adt)
    skeleton2 = pink.skeleton(inv, 0, 4)
    inv = pink.inverse(skeleton2)
    dilated = pink.geodilat(endpoints, inv, 4)
    return dilated
```



Right. The breaks in this lamellar eutectic material form discontinuity surfaces that are hard to detect by the human eye, but they can be discriminated by connectivity.



Interest

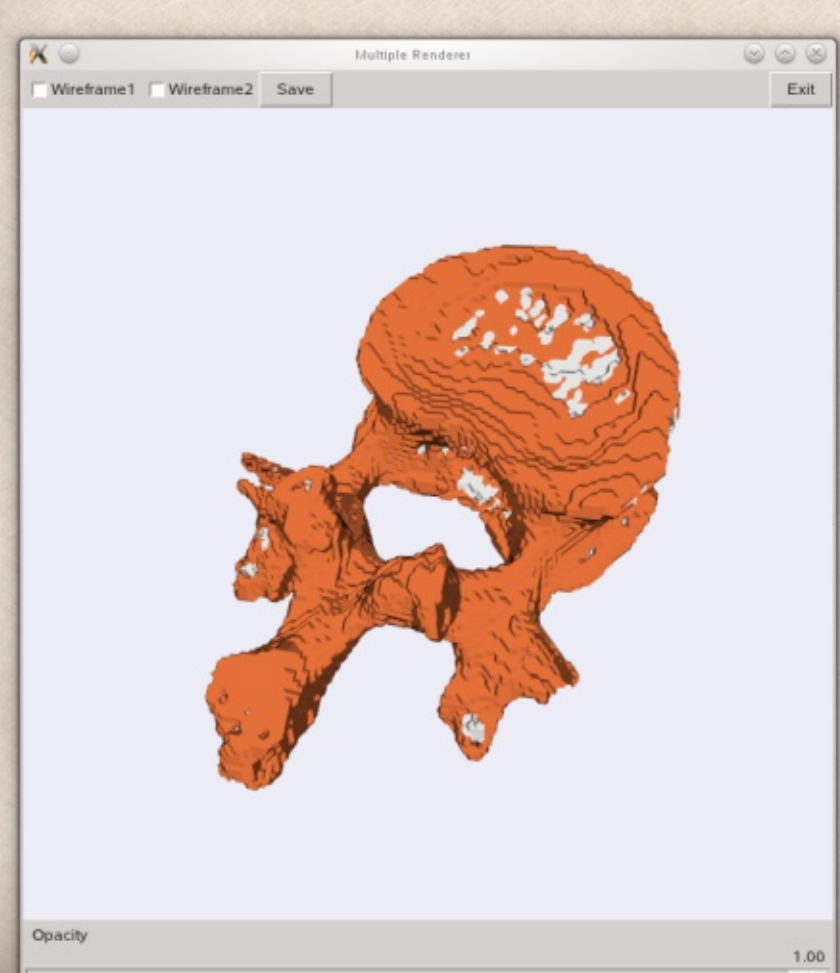
- Use the operators to solve image processing problems
- Simple operators chained can solve problems easily
- Parameters should be adjustable
- Accelerate development time compared to C/C++
- No C/C++ knowledge should be necessary for the "users"

Notion

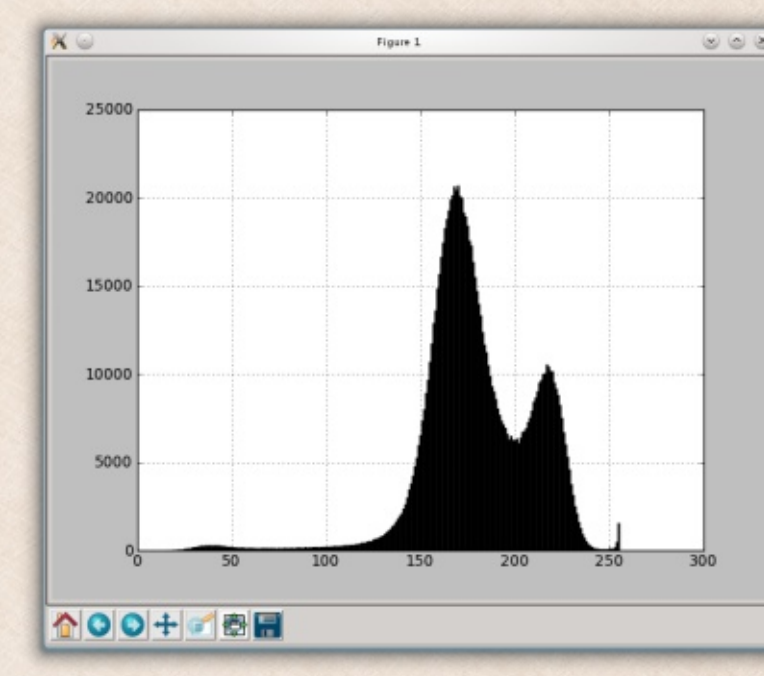
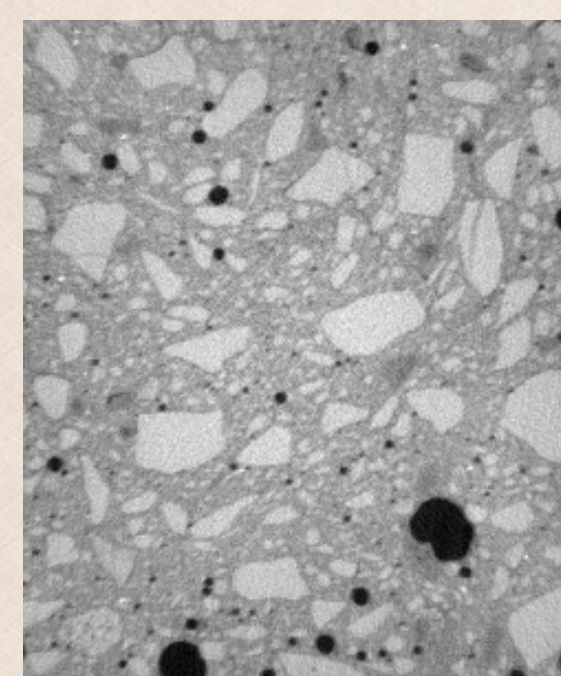
- Use Python as the scripting language
- Follow the functional programming paradigm
- Never leave C/C++ and Python

Down-Right. Pink supports 3D visualisation with vtk-python. Pink can generate marching cubes for binary segmentation which are then superimposed with the original object.

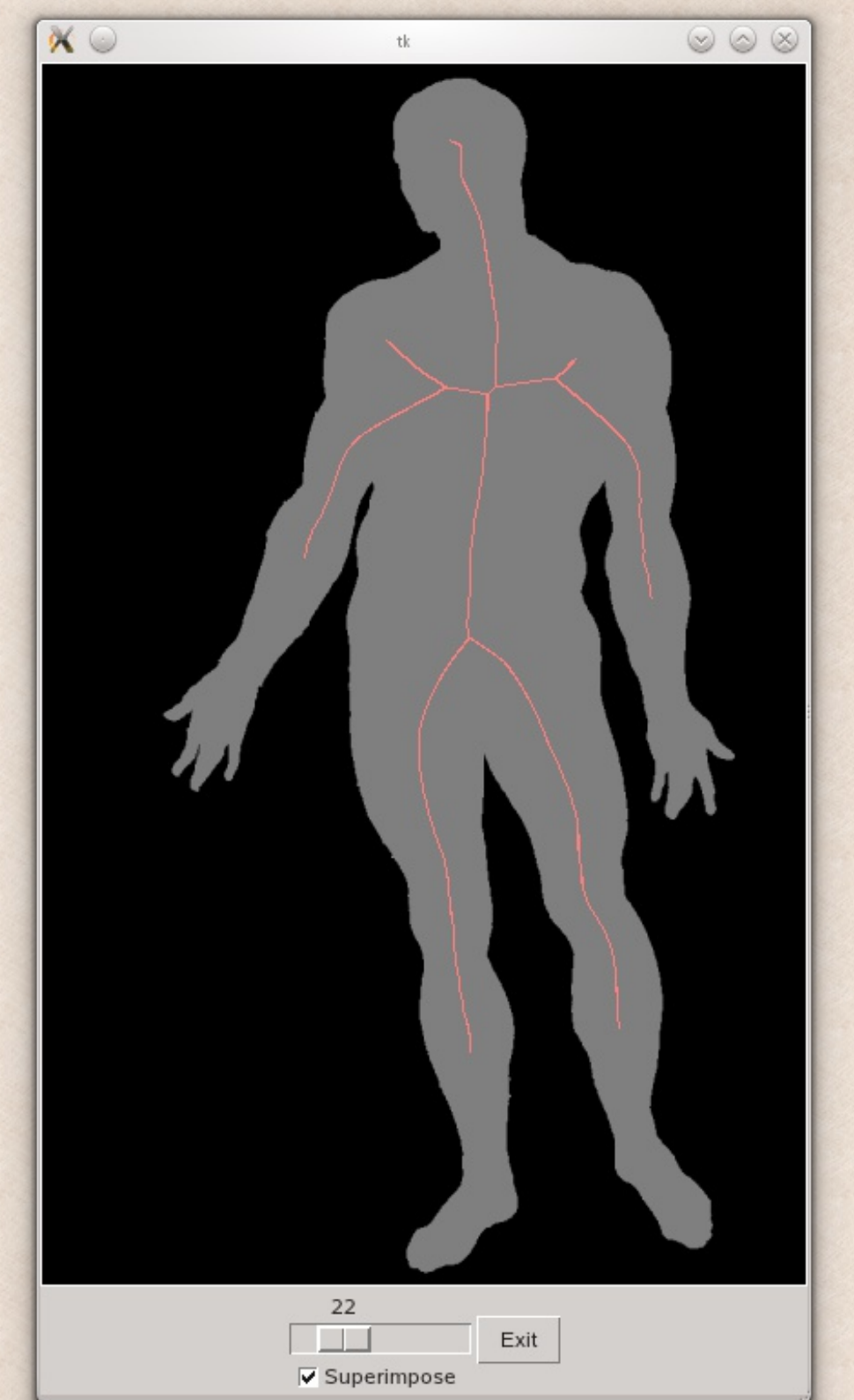
Down-Left. With the assumption that the thoracic vertebrae is one solid piece with a single large tunnel we can reconstruct the incomplete acquisition by topologically filling the holes in the segmentation.



Down. In this application we would like to know the distribution of particle size in the composite material. With python-matplotlib the results can be visualised as a histogram. *Image courtesy of EDF.*

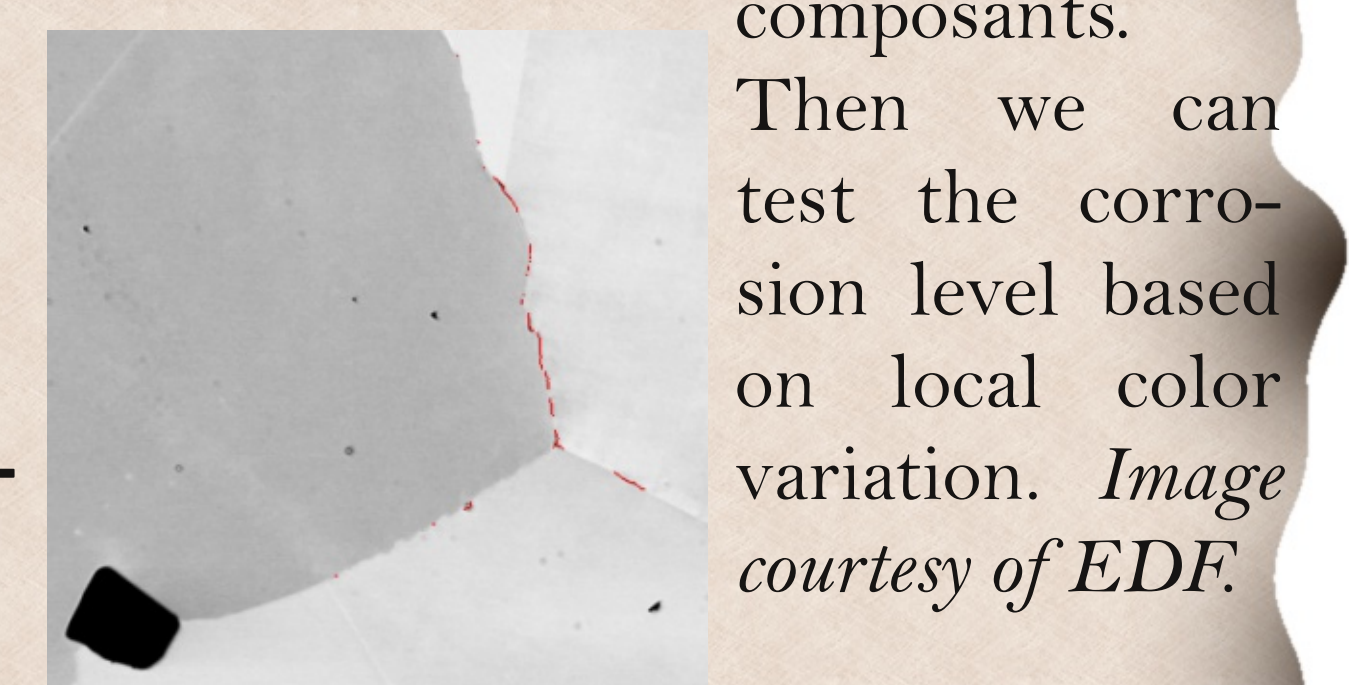
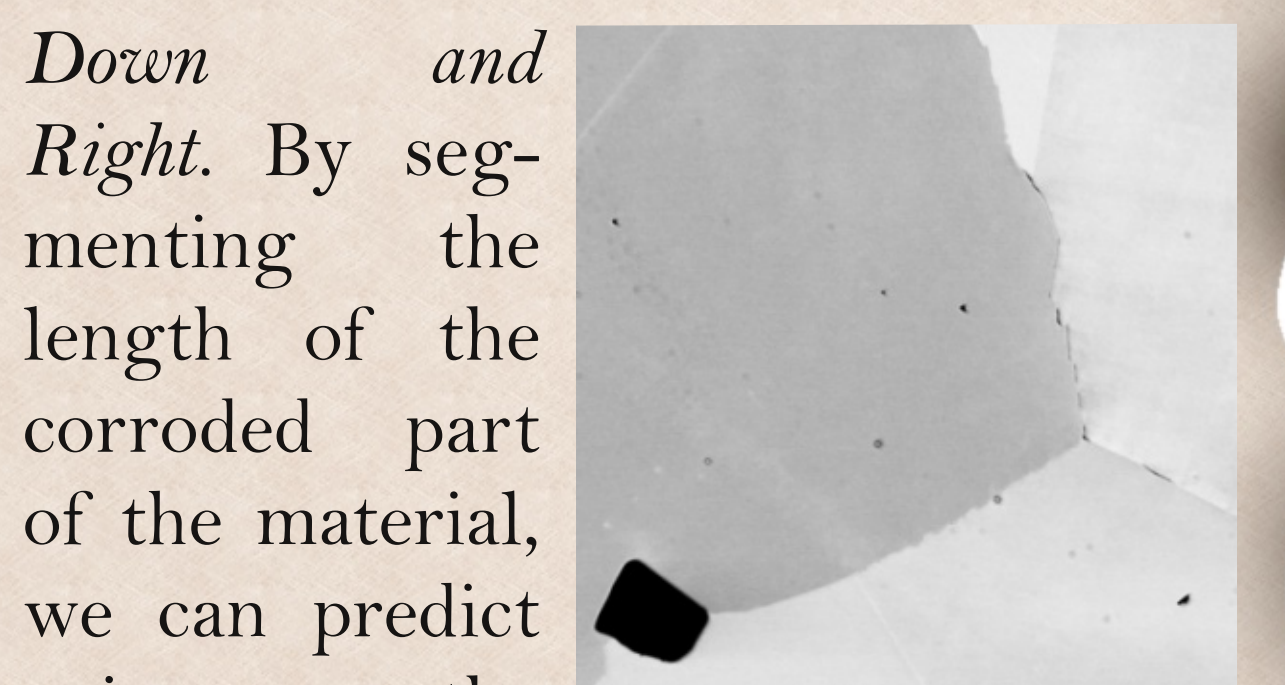


Up. There are many operators for skeletonization and skeleton filtering. They are useful for segmentation or registration.



Down and Right. By segmenting the length of the corroded part of the material, we can predict aging or the material quality.

With image segmentation tools, we can partition the image into regions belonging to different components. Then we can test the corrosion level based on local color variation. *Image courtesy of EDF.*



- Operators behave natively in Python
- They are easily wrapped by the programmer
- All the wrapping remains in C/C++
- The overhead for wrapping is O(1)

```
// function declaration in C
int32_t Idilatball(struct xvimage* ob, int32_t r, int32_t mode);

// Pink-Python wrapper (in C++)
ULWRAP_FUNCTION(
    "dilatball",
    Idilatball,
    (arg("image"), arg("radius"), arg("mode")=0),
    doc_dilatball_c_
);
# include BOOST_PP_UPDATE_COUNTER()
```