ToolIP

Visual Programming for Image Processing

Ali Moghiseh and Andreas Jablonski

Image Processing Department,

Fraunhofer ITWM

IPOL meeting, June 2012

ToolIP what is it ?

ToolIP stands for :

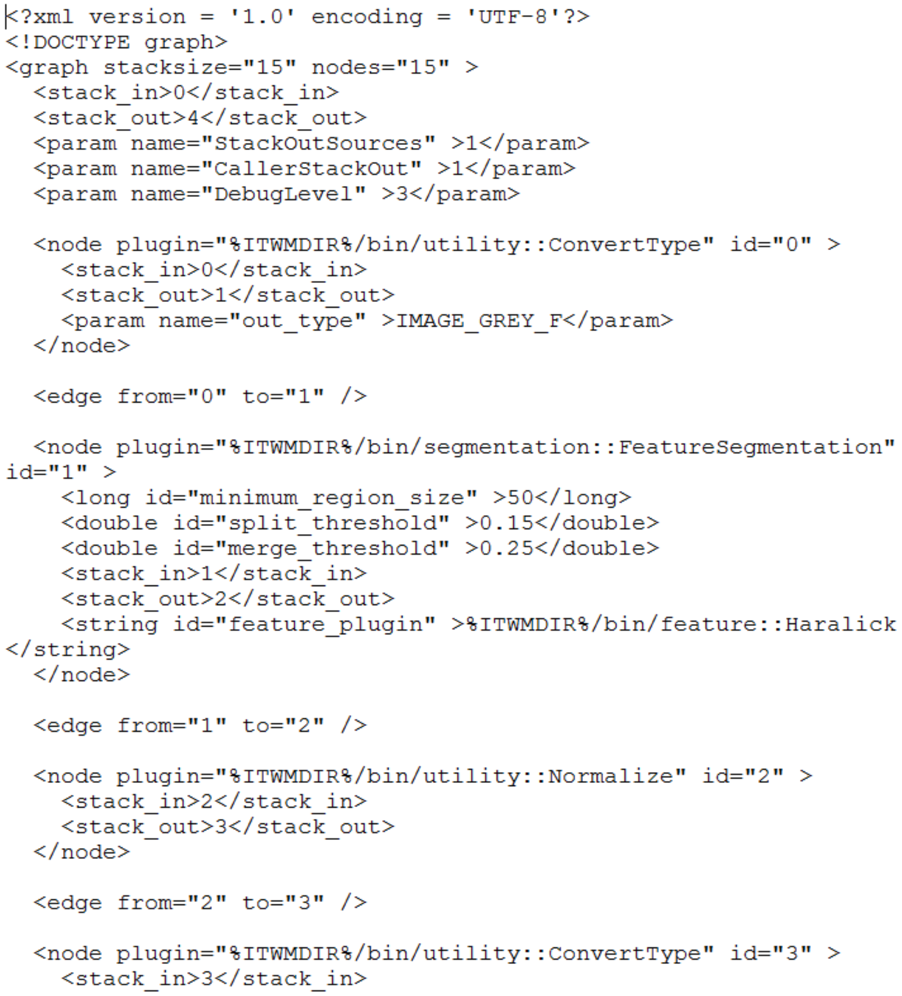
- **Tool for Image Processing**
- **Visual Programming Tool for Image Processing**
- **Editing Graphs, no Image Processing Functionality**
- **Algorithms and Visualization via Plugins**

Fraunhofer ITWM

ToolIP Origins and History

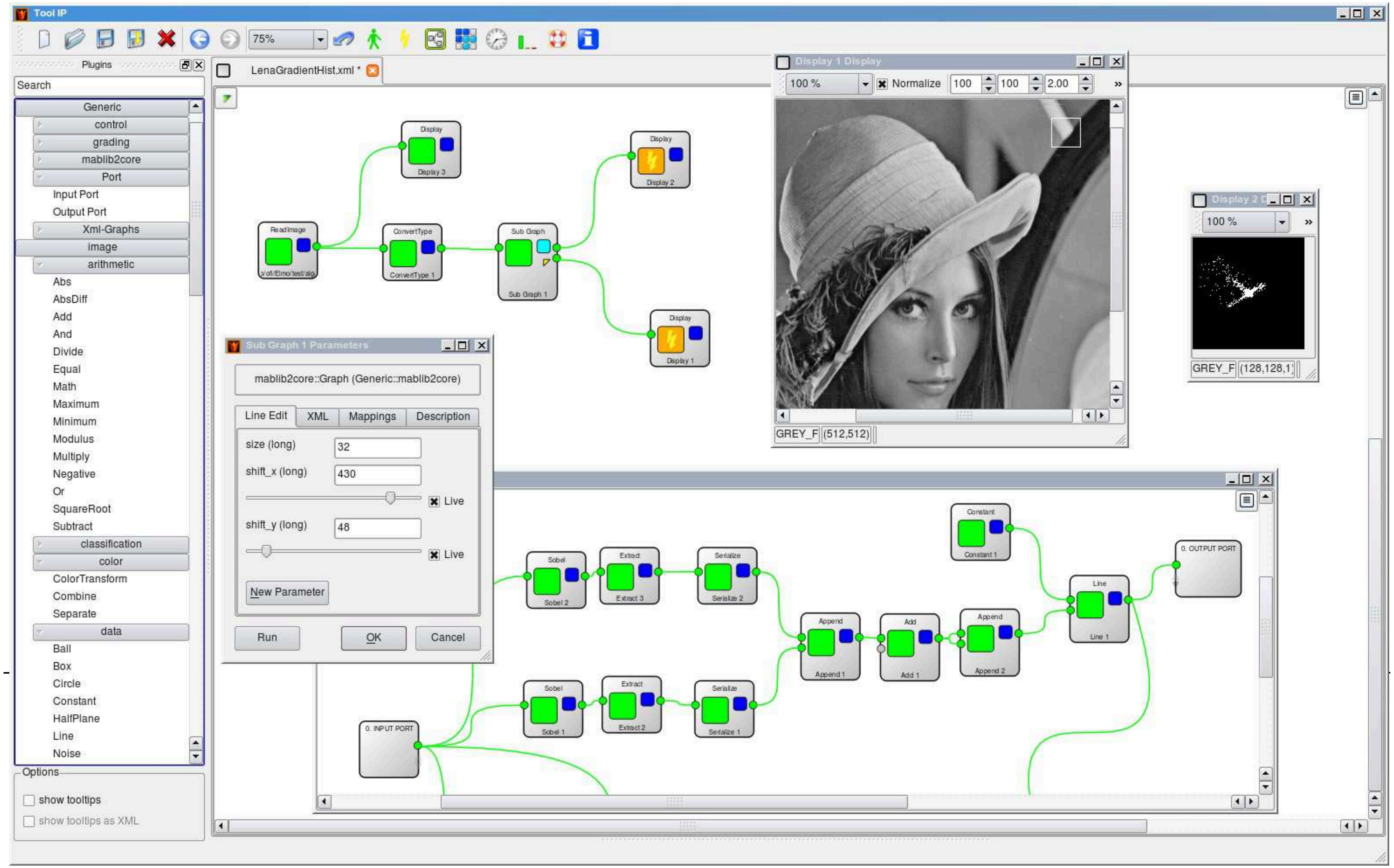
2005 and before	Surface Inspection Tasks
2005	General C++ plugin framework (MABlibCore)
2006	Algorithm nodes (MASClib) and Graph Runner (no GUI)
2008	simple GUI, step by step execution, no flow control
2009 - Today	improved GUI, automatic parallel execution, dynamic parameters, flow control, new parallel Graph Runner

Algorithm graph in xml



```
<?xml version = '1.0' encoding = 'UTF-8'?>
<!DOCTYPE graph>
<graph stacksize="15" nodes="15" >
  <stack_in>0</stack_in>
  <stack_out>4</stack_out>
  <param name="StackOutSources" >1</param>
  <param name="CallerStackOut" >1</param>
  <param name="DebugLevel" >3</param>

  <node plugin="%ITWMDIR%/bin/utility::ConvertType" id="0" >
    <stack_in>0</stack_in>
    <stack_out>1</stack_out>
    <param name="out_type" >IMAGE_GREY_F</param>
  </node>

  <edge from="0" to="1" />

  <node plugin="%ITWMDIR%/bin/segmentation::FeatureSegmentation"
id="1" >
    <long id="minimum_region_size" >50</long>
    <double id="split_threshold" >0.15</double>
    <double id="merge_threshold" >0.25</double>
    <stack_in>1</stack_in>
    <stack_out>2</stack_out>
    <string id="feature_plugin" >%ITWMDIR%/bin/feature::Haralick
</string>
  </node>

  <edge from="1" to="2" />

  <node plugin="%ITWMDIR%/bin/utility::Normalize" id="2" >
    <stack_in>2</stack_in>
    <stack_out>3</stack_out>
  </node>

  <edge from="2" to="3" />

  <node plugin="%ITWMDIR%/bin/utility::ConvertType" id="3" >
    <stack_in>3</stack_in>
```

ToolIP what does it look like ?



Sub Graph 1 Parameters

mablib2core::Graph (Generic::mablib2core)

size (long) 32

shift_x (long) 430 ☒ Live

shift_y (long) 48 ☒ Live

Display 1 Display: 100 % ☒ Normalize 100 100 2.00

GREY_F (512,512)

Display 2: 100 %

GREY_F (128,128,1)

Options

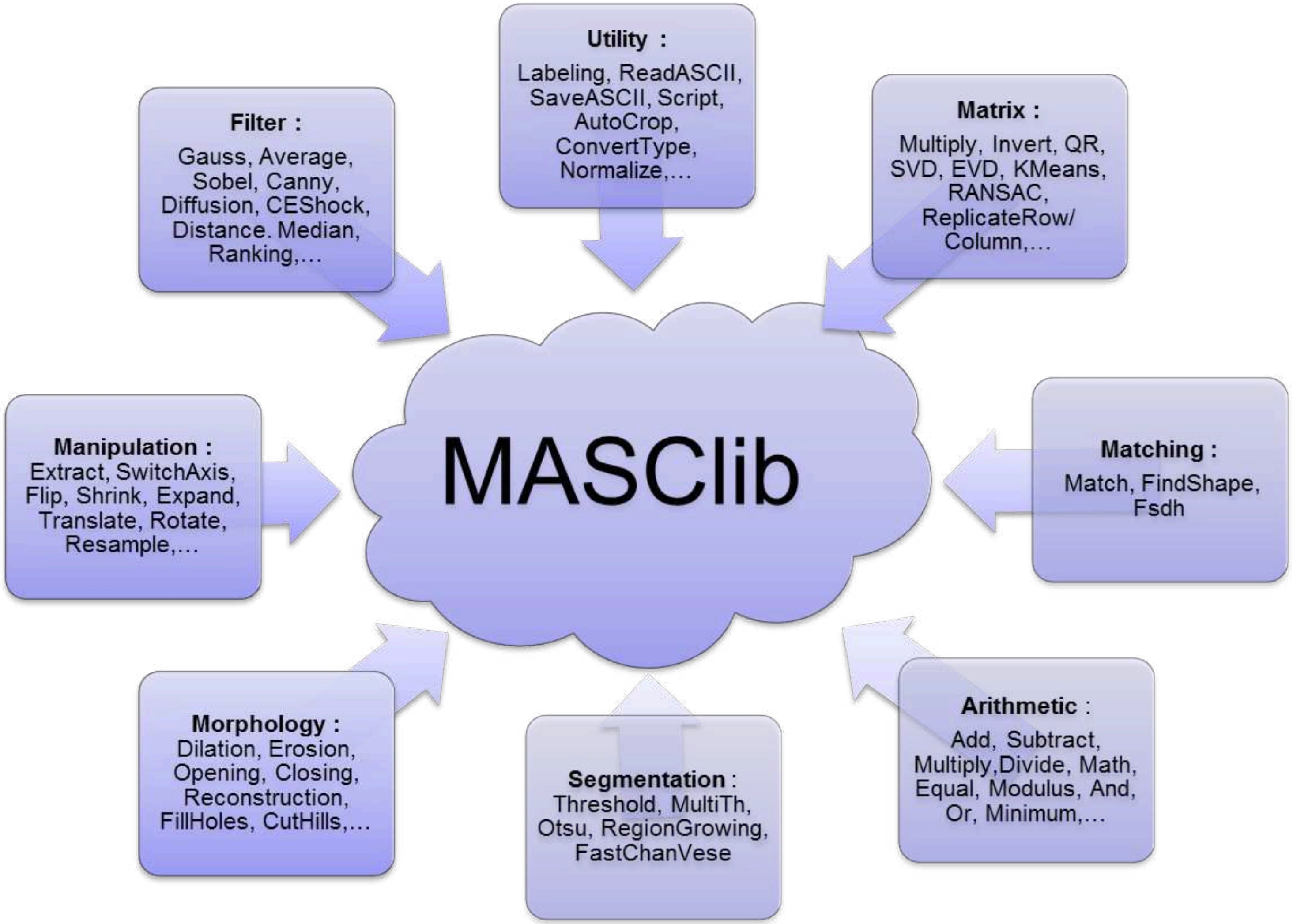
□ show tooltips

□ show tooltips as XML

ToolIP why is it interesting?

- Selfexplaining to use, no programming necessary
- Can use graphs at the customer site
- Easy to extend, add new Algorithms
- Cross platform - works on linux and windows
- Almost no overhead compared to compiled code

ToolIP includes MASClib



MASClib

Utility :
Labeling, ReadASCII, SaveASCII, Script, AutoCrop, ConvertType, Normalize,…

Filter :
Gauss, Average, Sobel, Canny, Diffusion, CEShock, Distance. Median, Ranking,…

Matrix :
Multiply, Invert, QR, SVD, EVD, KMeans, RANSAC, ReplicateRow/Column,…

Manipulation :
Extract, SwitchAxis, Flip, Shrink, Expand, Translate, Rotate, Resample,…

Matching :
Match, FindShape, Fsdh

Morphology :
Dilation, Erosion, Opening, Closing, Reconstruction, FillHoles, CutHills,…

Segmentation :
Threshold, MultiTh, Otsu, RegionGrowing, FastChanVese

Arithmetic :
Add, Subtract, Multiply,Divide, Math, Equal, Modulus, And, Or, Minimum,…

Fraunhofer
ITWM

Using Plugins and Algorithm Graphs from C++

```
// read "Lena.png" from file

CImage *pImgIn = RunPlugin("%ITWMDIR%/bin/utility::ReadImage", "filename",
   "Lena.png" );

// apply the average3d filter to pImgIn

std::string plugin_path = "%ITWMDIR%/bin/filter::Average3d";

CImage *pImgOut = RunPlugin( plugin_path,  pImgIn,  "step_x", 3, "step_y", 3);

// subtract pImgOut from pImgIn inplace

RunPlugin(pImgOut,"%ITWMDIR%/bin/arithmetic::Subtract", pImgIn,pImgOut,"upcast",
   true);
```

Using Plugins and Algorithm Graphs from C++

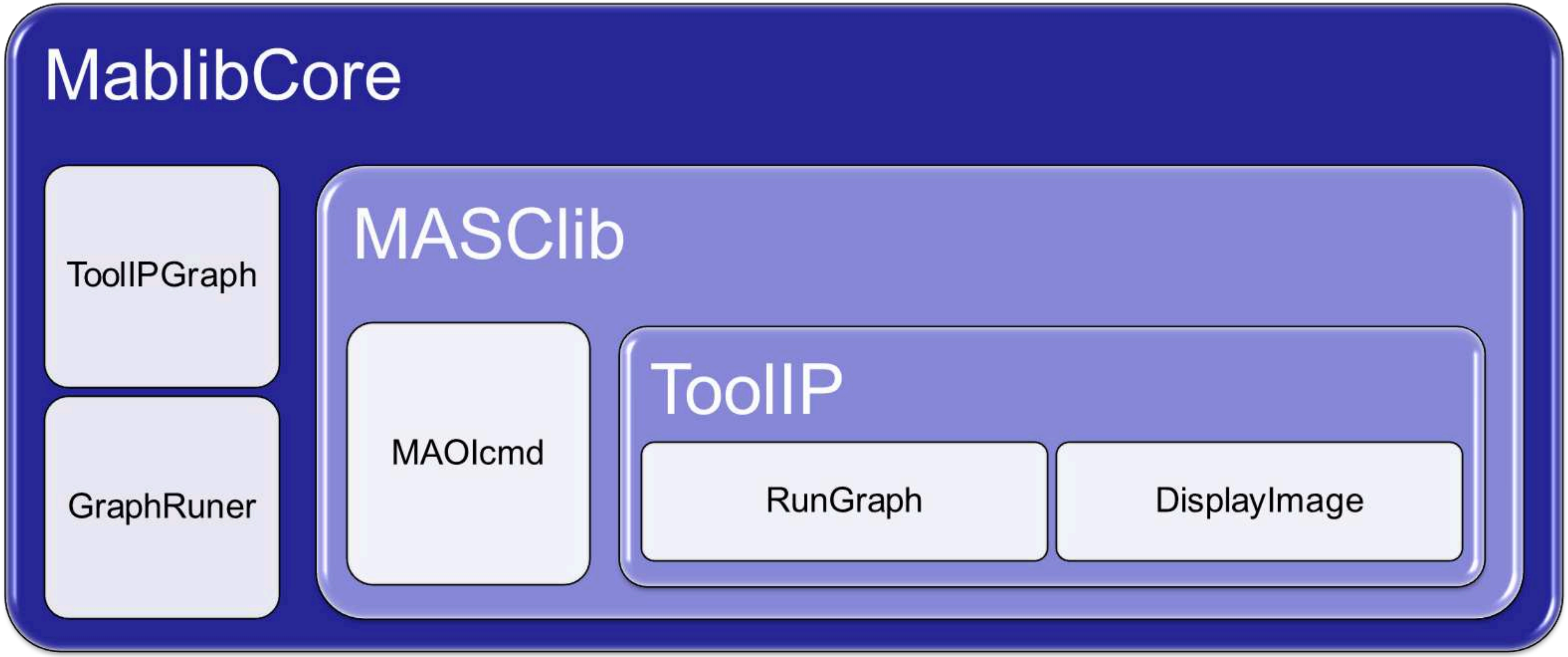
```
// save pImgOut in ASCII format

RunPlugin("%ITWMDIR
   %/bin/utility::SaveASCII",pImgOut,"filename","laplace.asc");
```

- An algorithm created in ToolIP can be called too as follows:

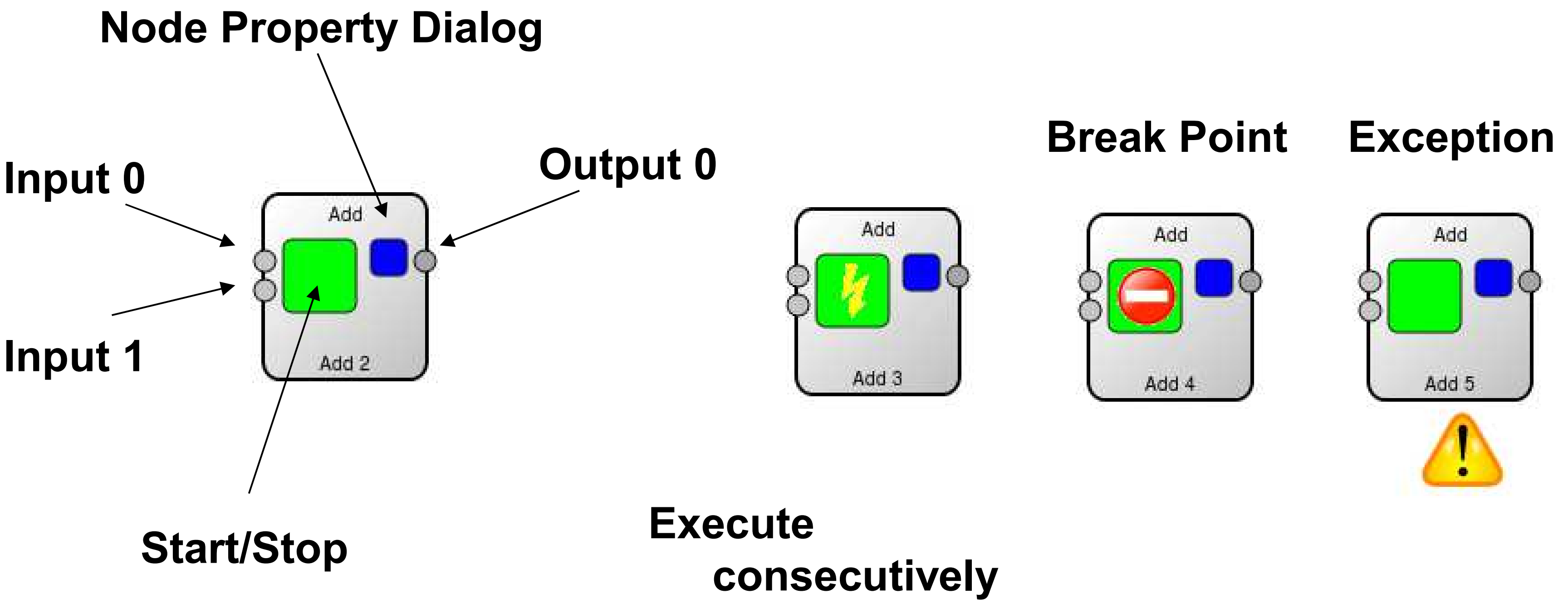
```
// rotate input image by 17 degrees using the graph Rotate.xml

CImage *pImgRot = RunGraph( "Rotate.xml" , pImgIn,  "angle", 17.0,
   "bg_value", 125.0);
```

Thank You for Your Attention!

Fraunhofer
ITWM


MablibCore
ToolIPGraph
GraphRuner
MASClib
MAOIcmd
ToolIP
RunGraph
DisplayImage

ToolIP GUI Nodes



Node Property Dialog

Input 0

Input 1

Output 0

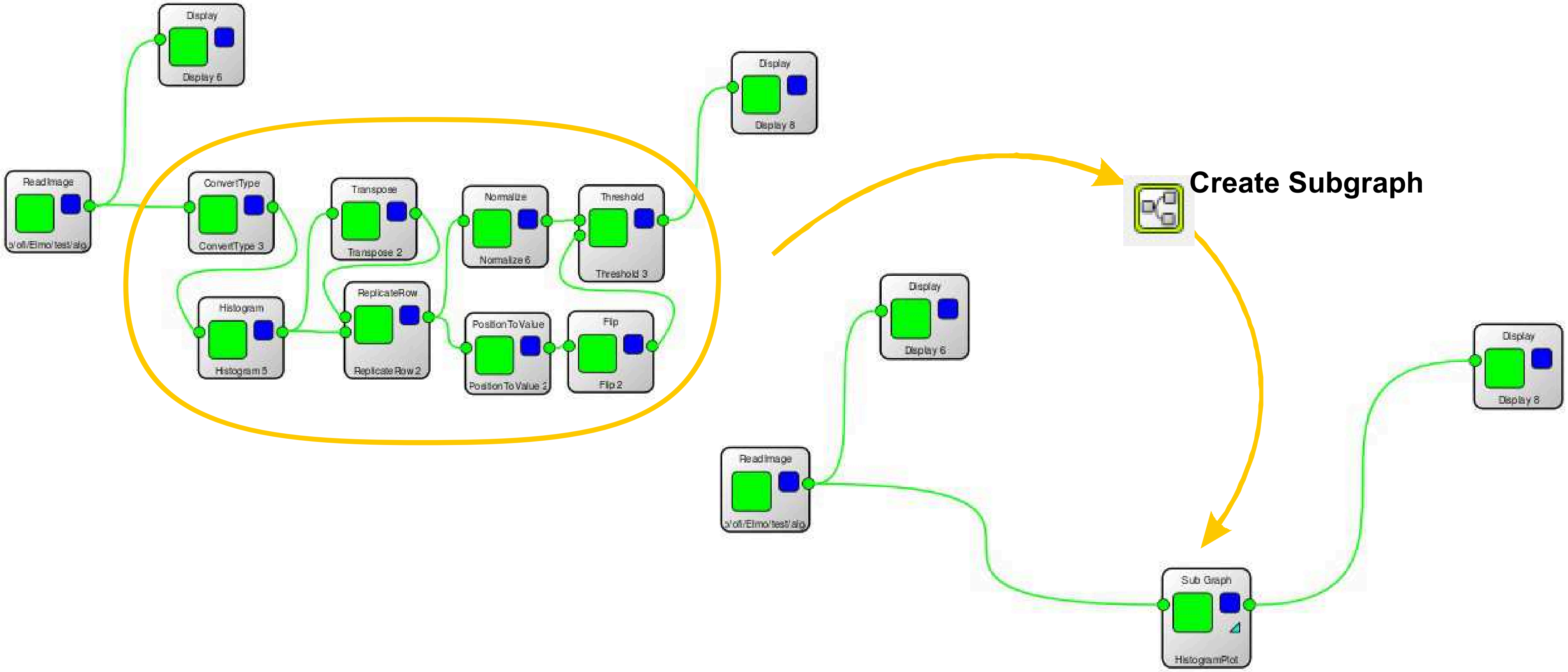
Start/Stop

Execute consecutively

Break Point

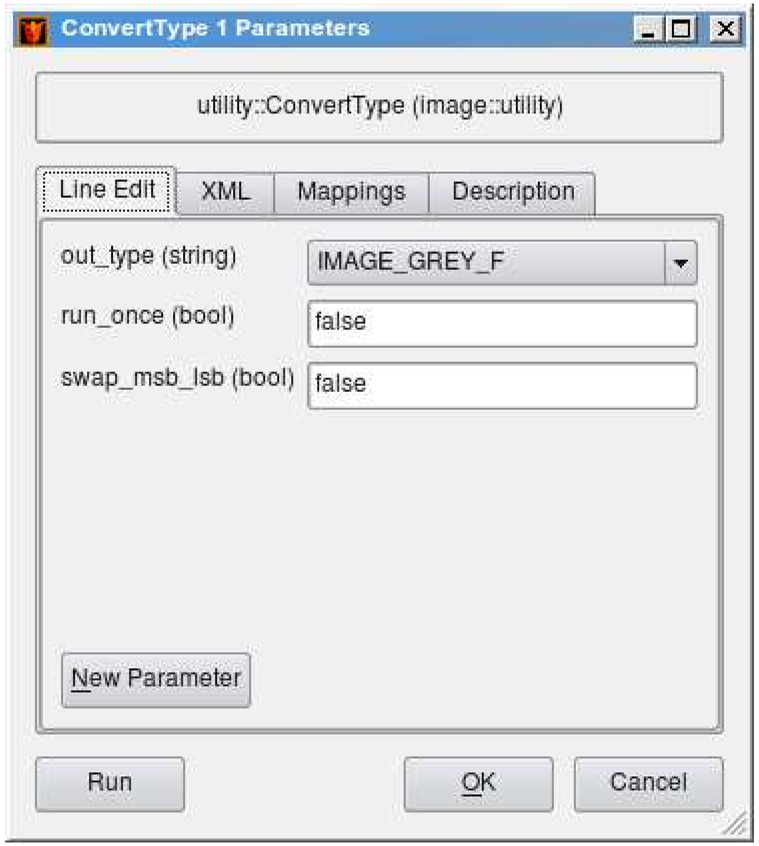
Exception

ToolIP Subgraph



Create Subgraph

ToolIP GUI Node Property Dialog

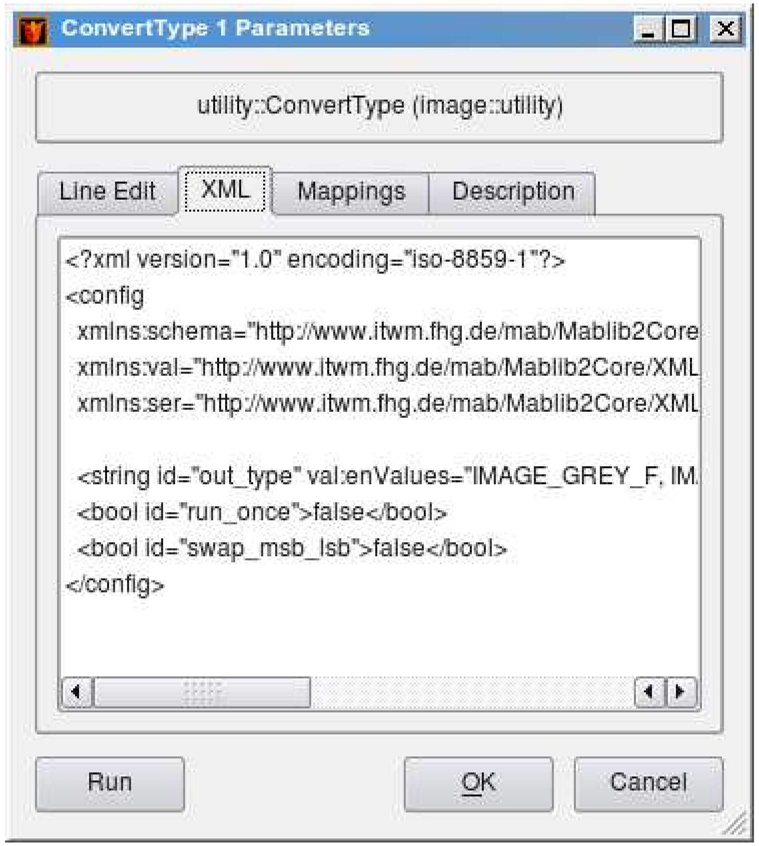


utility::ConvertType (image::utility)

out_type (string) IMAGE_GREY_F

run_once (bool) false

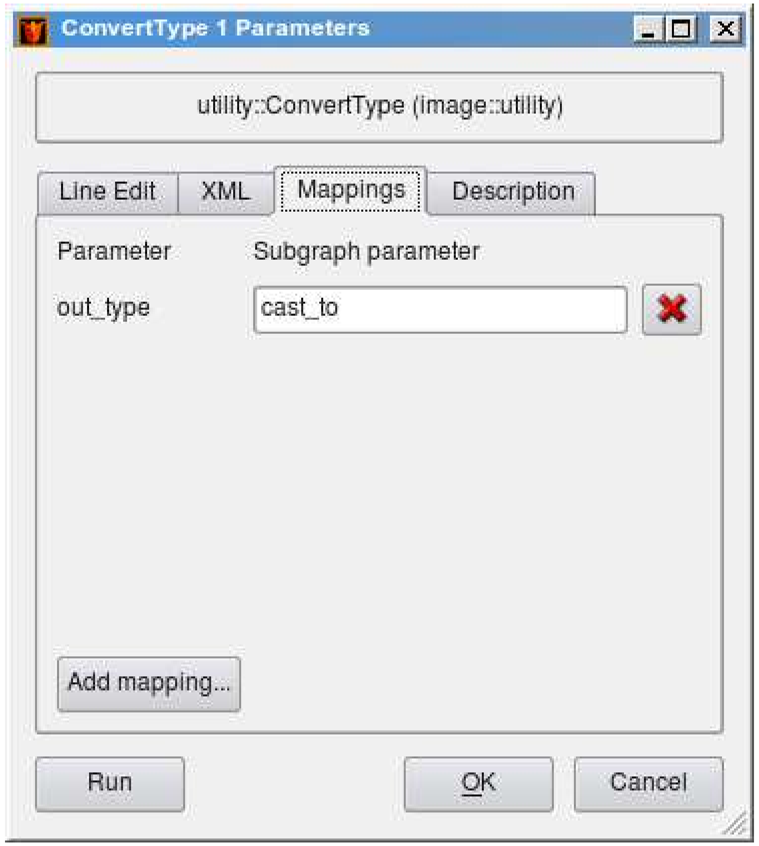
swap_msb_lsb (bool) false



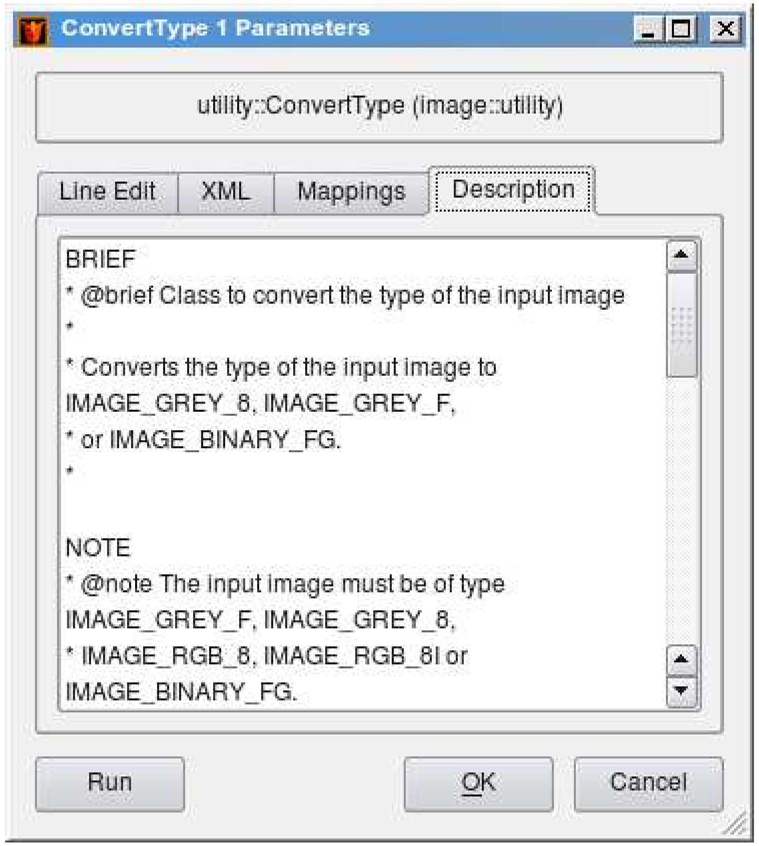
utility::ConvertType (image::utility)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<config
  xmlns:schema="http://www.itwm.fhg.de/mab/Mablib2Core
  xmlns:val="http://www.itwm.fhg.de/mab/Mablib2Core/XML
  xmlns:ser="http://www.itwm.fhg.de/mab/Mablib2Core/XML

  <string id="out_type" val:enValues="IMAGE_GREY_F, IM
  <bool id="run_once">false</bool>
  <bool id="swap_msb_lsb">false</bool>
</config>
```

ToolIP GUI Node Property Dialog

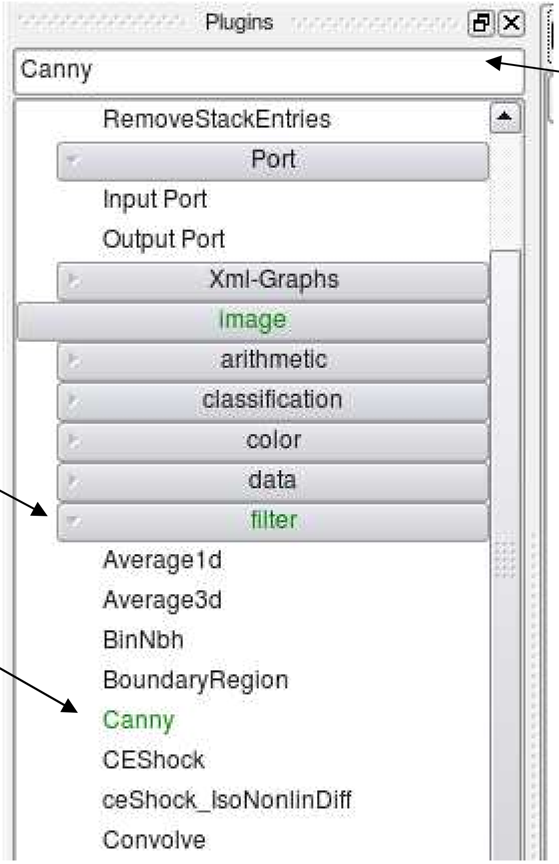


Parameter	Subgraph parameter
out_type	cast_to



```
BRIEF
* @brief Class to convert the type of the input image
*
* Converts the type of the input image to
IMAGE_GREY_8, IMAGE_GREY_F,
* or IMAGE_BINARY_FG.
*


NOTE
* @note The input image must be of type
IMAGE_GREY_F, IMAGE_GREY_8,
* IMAGE_RGB_8, IMAGE_RGB_8I or
IMAGE_BINARY_FG.
```

ToolIP GUI Plugin Window



Plugins

Canny

- RemoveStackEntries
- Port
  - Input Port
  - Output Port
- Xml-Graphs
- Image
  - arithmetic
  - classification
  - color
  - data
  - filter
    - Average1d
    - Average3d
    - BinNbh
    - BoundaryRegion
    - Canny
    - CEShock
    - ceShock_IsoNonlinDiff
    - Convolve

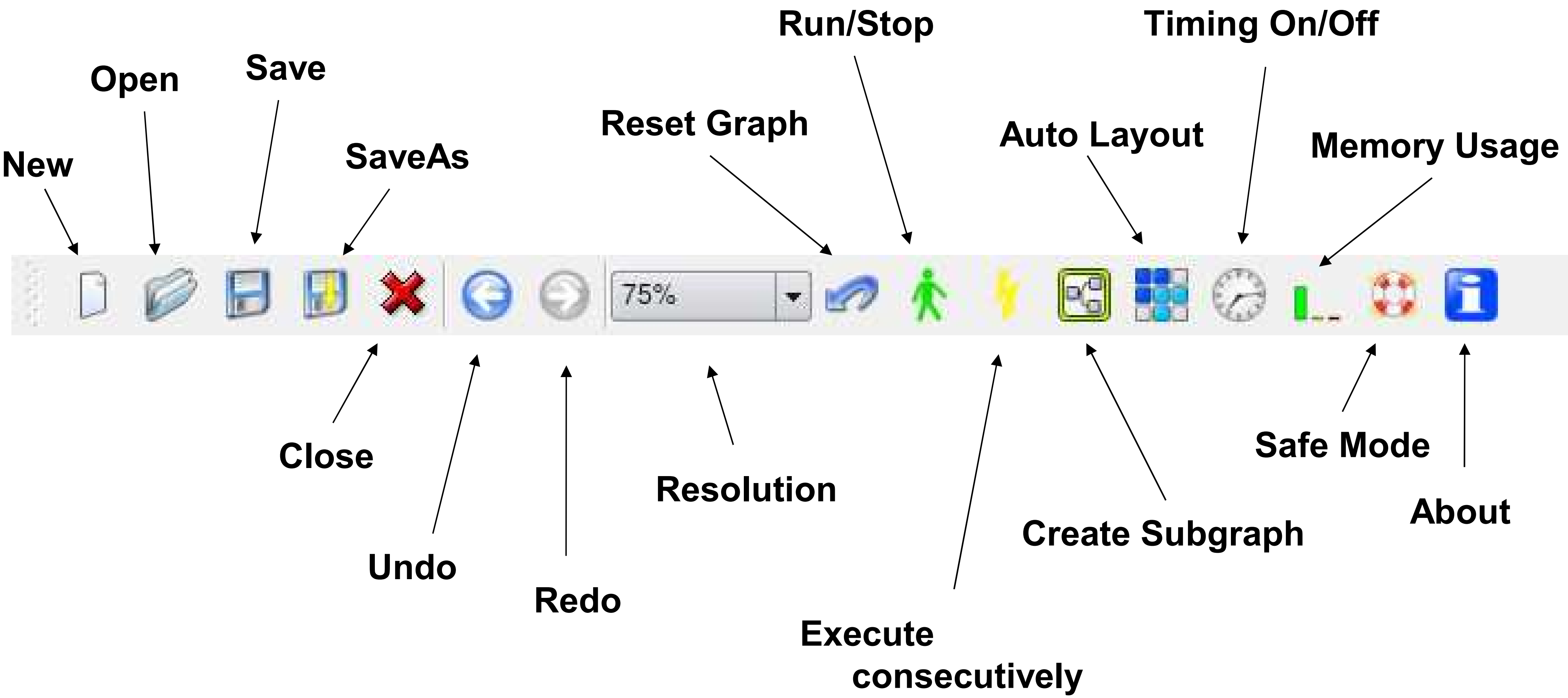
Search mask

Type

SubTyp
e

Name

ToolIP GUI Toolbar



New

Open

Save

SaveAs

Reset Graph

Run/Stop

Auto Layout

Timing On/Off

Memory Usage

Close

Undo

Redo

Resolution

Execute consecutively

Create Subgraph

Safe Mode

About