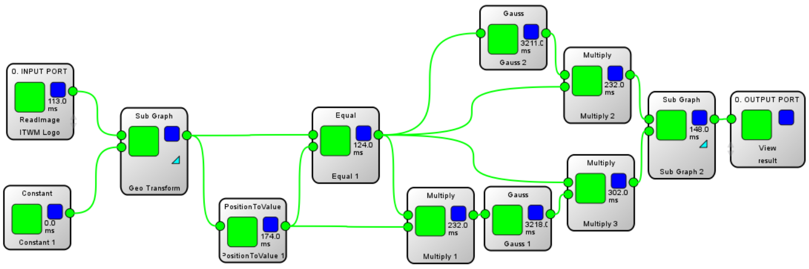
www.itwm.fraunhofer.de/toolip

ToolIP: Tool for Image Processing

Ali Moghiseh[1], Andreas Jablonski[1]

ToolIP is a visual programming environment for image processing. It was developed especially to facilitate the design of complex algorithmic solutions in a an intuitive way. Solutions are displayed as graph structures. Each node represents an algorithmic component whereas the edges describe the data flow. The resulting solution can be embedded in customer applications via a C/C++-interface or run from the command line. The design of ToolIP is very modular, the visual programming GUI (also called ToolIP) has no image processing or visualization capabilities by itself and such functionality is realized via plugins. This makes it very easy to extend the system with new algorithms. The current version of ToolIP includes the MASClib library (Modular Algorithms for Surface InspeCtion) containing standard image processing functions as well as efficient implementations of state-of-the-art algorithms. The ToolIP environment is cross platform and works on linux and windows maschines.



[1] Fraunhofer ITWM Kaiserslautern, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
ali.moghiseh@itwm.fraunhofer.de
andreas.jablonski@itwm.fraunhofer.de

MASClib Software Library

Basic Image Handling

MASClib supports a rich set of image formats and color spaces. There are many operations that allow adapting the orientation, resolution, coordinate system etc. of the image data to the specific application. Furthermore, images can be represented on multiple resolution levels via nonlinear downsampling and wavelet operators.

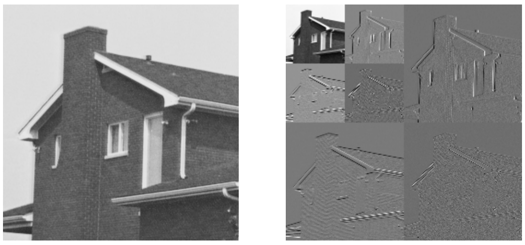


Figure 1: Multiscle wavelet image representation

Image Denoising and Enhancement

There is a whole variety of filtering operations useful for denoising and qualitative enhancement of image data. These include: linear filters (Mean, Gauss, Binomial) morphology filters (Dilation, Erosion), ranking filters and several adaptive anisotropic filtering approaches.

Edge Detection

For detecting edges, there exist many possibilities including gradient calculation with several masks, edge detection with Canny's detector and ridge detection using the Hessian matrix. Furthermore, line structures in images can be detected using the Hough transform.

Object Detection

Known objects can be easily detected by using templates. MASClib provides gradient based object models and similarity measures derived from the displacement histogram.

Image Registration

MASClib offers robust grey value based matching via maximization of statistical similarity measures derived from the joint grey level distribution.

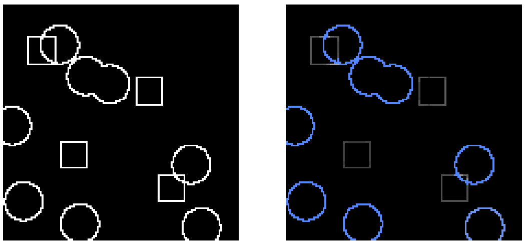


Figure 2: Detecting circles

Segmentation

Basic thresholding, Otsu's method, and the fast Chan Vese method constitute the tools for image segmentation. MASClib also supports automatic adaptive thresholding via maximization of Mutual Information and custom thresholds derived from the image histogram. Besides the image grey values, other features of the local neighbourhoods can be used as fundament of the segmentation, especially in textured regions. Combination with further building blocks allows for highly specialised and adaptive solutions.

Labeling

Once objects are segmented, working with different object labels offers a high degree of flexibility: Features can be mapped onto the single objects, allowing for the selection and classification of objects with respect to features such as shape, size and texture.

Region Features

Every identified object can be characterized using a variety of image features, like for instance average, variance, entropy, texture descriptors (Haralick, Local Binary Pattern) or region geometry descriptors. This is useful for analysing regions with highly complex characteristics.

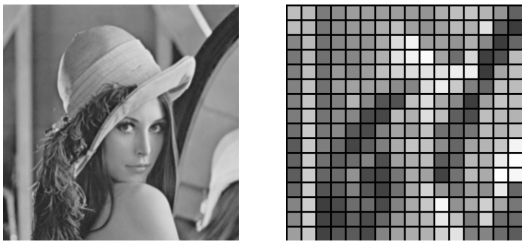


Figure 3: Calculate image features on a grid. Here mean value of each cell.

Classification

Several classification algorithms are provided to automatically distinguish objects belonging to different classes. These include simple linear discriminant and Mahalanobis distance classifiers as well as the more advanced SVM classifier.

Clustering

Automatic grouping of objects can be a useful step for many applications. Beside the basic KMeans algorithm also more sophisticated approaches can be realized using operators from the linear algebra module.

Matrix Operations

In addition to standard image processing and analysis methods, MASClib supports linear algebra operations. These operations are especially useful for developing advanced pattern recognition algorithms; there is no need to include external dedicated libraries. Supported operations include the SVD, the QR-decomposition, matrix inversion, the computation of Eigenvalues and Eigenvectors as well as algorithms for PCA and ICA. In this module, each grey value image is treated as a matrix in a natural way.

Usage Example

The following C++ code samples shows how one can use the library in his own application. MASClib plugins are used to load, save and manipulate images:

```
// 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", 3l, "step_y", 3l, "upcast", true);

// subtract pImgOut from pImgIn inplace
RunPlugin(pImgOut,"%ITWMDIR%/bin/arithmetic::Subtract", pImgIn,pImgOut);

// 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);
```