

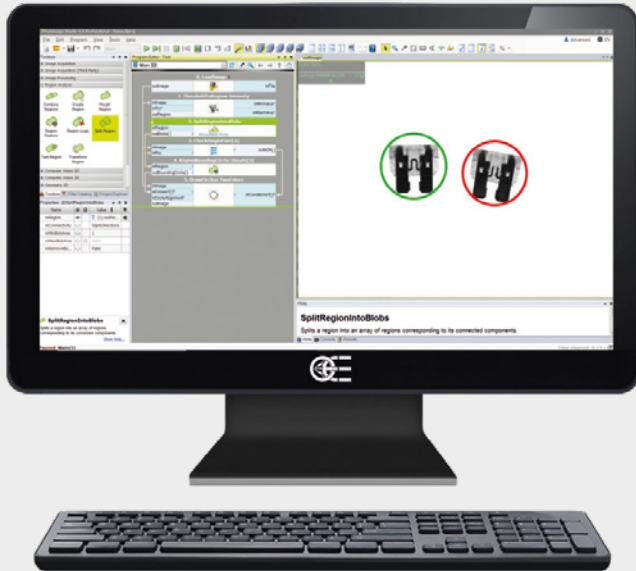
# FABIMAGE

Machine vision software and libraries



**PRELIMINARY**

## Fabimage studio professional



### KEY ADVANTAGES

**No low-level programming knowledge required.**

**Data-flow based software.**

**Fast and optimized algorithms.**

**1000+ high performance functions.**

**Custom machine vision filters.**

**FabImage Studio Professional** is **data-flow based software** designed for machine vision engineers. It does not require any programming skills, but it is still so powerful that it can win even against solutions based on low-level programming libraries.

Also, the architecture is highly flexible, ensuring that users can easily adapt the product to the way they work and to specific requirements of any project.

## FEATURES

### Intuitive

#### Drag & Drop

All programming is done by **choosing filters and connecting them with each other**. You can focus all your attention on computer vision.

#### You Can See Everything

Inspection results are visualized on multiple configurable data previews; and when a parameter in the program is changed, **you can see the previews updated in real time**.

#### HMI Designer

You can easily **create custom graphical user interfaces** and thus build the entire machine vision application using a single software package.

### Powerful

#### Over 1000 Ready-for-Use Filters

There are over **1000 ready-for-use machine filters tested and optimized** on hundreds of applications. They have many advanced capabilities such as outlier suppression, subpixel precision or any-shape region-of-interest.

#### Hardware Acceleration

The filters are **aggressively optimized for the SSE technology** and for **multicore processors**. Our implementations are ones of the fastest in the world!

#### Loops and Conditions

Without writing a single line of code, **you can create custom and scalable program flows**. Loops, conditions and subprograms (macrofilters) are realized with appropriate data-flow constructs in the graphical way.

## Adaptable

### GigE Vision and GenTL Support

FabImage Studio is a **GigE Vision compliant** product, supporting the GenTL interface, as well as a number of vendor-specific APIs. Thus, you can use it with most cameras available on the market, including models from **Matrix Vision**, Allied Vision, Basler, Baumer, Dalsa, PointGrey, Photon Focus and XIMEA.

### User Filters

You can use user filters to **integrate your own C/C++ code with the benefits of visual programming**.

### C++ Code Generator

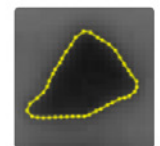
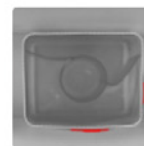
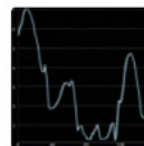
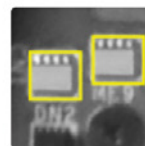
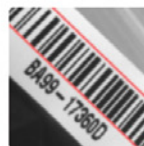
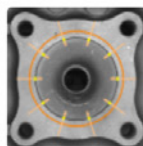
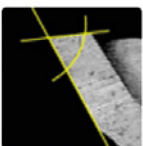
Programs created in FabImage Studio can be exported to C++ code or to .NET assemblies. This makes it very easy to **integrate your vision algorithms with applications created in C++, C# or VB programming languages**.

The screenshot displays the FabImage Studio interface. On the left, a 'Toolbox' contains various filter icons. The central 'Program Editor' shows a flowchart with filters like 'EnumerateImages', 'outImageAlign', and 'outImageLevel'. On the right, a 'Configurable data previews' window shows a video feed of four bottles with colored bounding boxes and measurement lines. A 'BOTTLE INSPECTOR' HMI window is overlaid, showing 'Bottle Count: Pass' and 'Foam Presence: Top Orientations'. Labels with arrows point to specific features: 'Daflow program' (flowchart), '1000 ready-for-use filters' (toolbox), 'Extensive parametrization' (HMI window), 'Configurable data previews' (video feed), and 'Intuitive HMI designer' (BOTTLE INSPECTOR window).

## CAPABILITIES

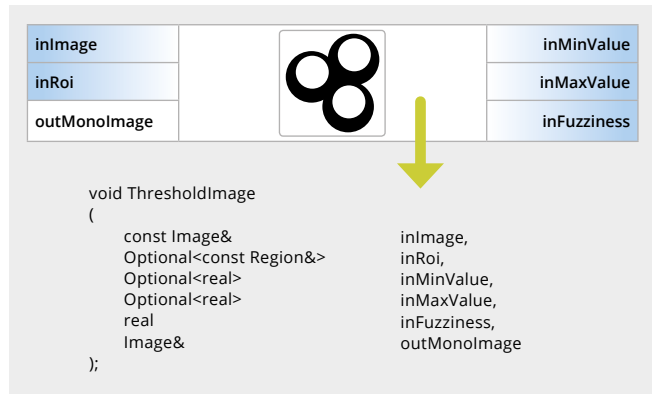
There are over 1000 filters encompassing both basic transformations and specialized machine vision tools.

- Image processing
- Shape fitting
- Barcode reading
- Template Matching
- Support vector machines
- Blob analysis
- Camera calibration
- Data code reading
- Measurements
- GigE Vision and GenTL
- Contour analysis
- Fourier analysis
- Corner detection
- Histogram analysis
- Planar geometry
- Hough transform
- 1D profile analysis
- OCR



**FabImage Library Suite** is a machine vision library for C++ and .NET programmers. It provides a comprehensive set of functions for creating industrial image analysis applications - from standard-based image acquisition interfaces, through low-level image processing routines, to ready-made tools such as template matching, measurements or barcode readers.

The main strengths of the product include the highest performance, modern design and simple structure making it easy to integrate with the rest of your code.



## Features

### Performance

In FabImage Library Suite careful design of algorithms goes hand in hand with extensive hardware optimizations, resulting in performance that puts the library among the fastest in the world. Our implementations make use of SSE instructions and parallel computations on multicore processors.

### Modern Design

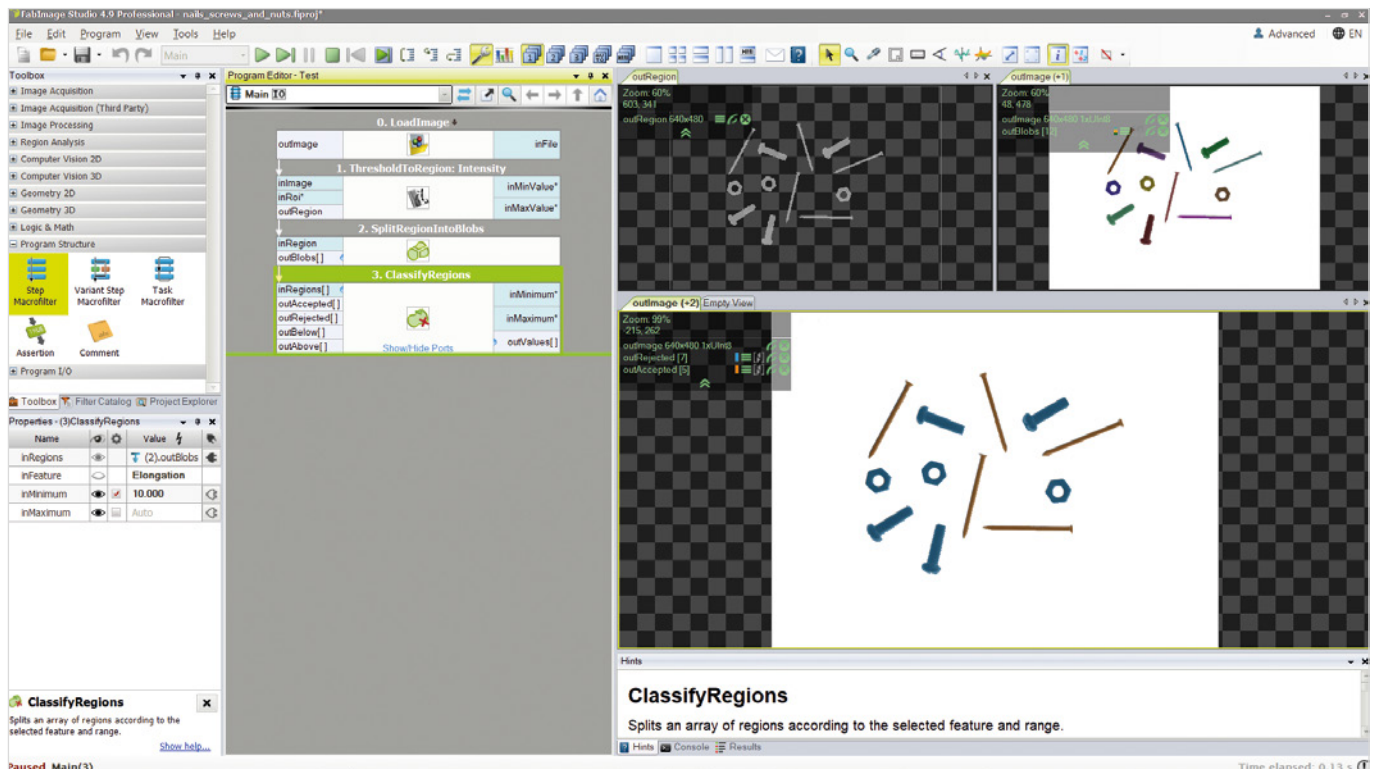
All types of data feature automatic memory management, errors are handled explicitly with exceptions and optional types are used for type-safe special values. All functions are thread-safe and use data parallelism internally, when possible.

### Simplicity & Consistency

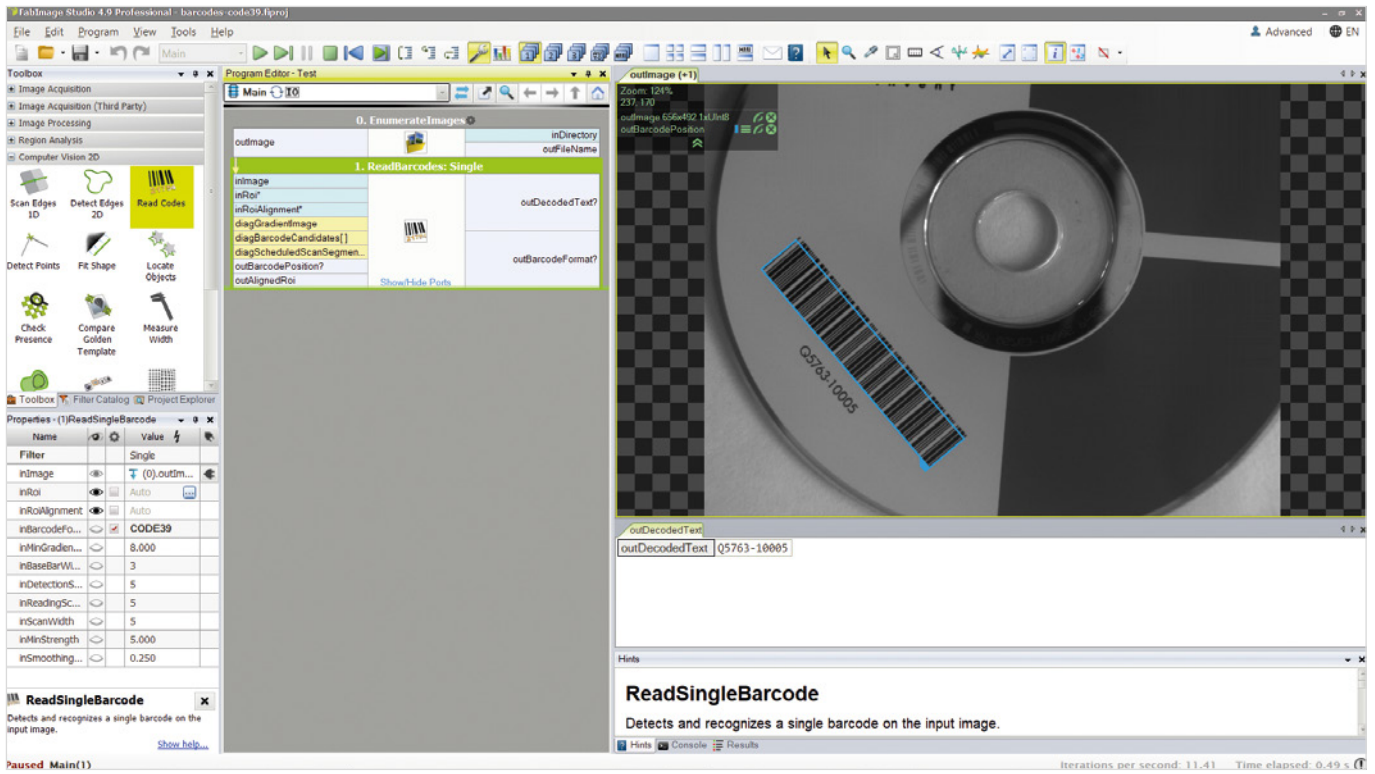
The library is a simple collection of types and functions, provided as a single DLL file with appropriate headers. For maximum readability, functions follow consistent naming convention (e.g. the VERB + NOUN form as in: SmoothImage, RotateVector). All results are returned via reference output parameters, so that many outputs are always possible.

## APPLICATION EXAMPLES

### Fabimage studio pro



In this application, we need to sort nails amongst nuts and bolts. The image is thresholded and the resulting regions are split into blobs; finally, the blobs are classified by their elongation and the nails are easily found.



This example shows a basic ReadBarcodes filter. The tool automatically find the barcode and gives as output the decoded text.

## Fabimage library suite

Below is an elementary, yet complete, example of acquiring images from a GigE Vision camera, thresholding them and saving to files on disk, all done with FabImage Library Suite (C++).

```
#include <iostream>
#include "Genicam.h"
#include "FIL.h"

using namespace ftl;
using namespace fil;

int main(void)
{
    try
    {
        // Find devices
        Array< GigEVision_DeviceDescriptor > deviceList;
        GigEVision_FindDevices(800, 1, deviceList);
        if (deviceList.Size() >= 1)
        {
            // Connect to the first found
            String addr = deviceList[0].IpAddress;
            GigEHandle hDev = GigEVision_OpenDevice(addr);
            GigEVision_StartAcquisition(hDev, "Mono8");

            // Main loop
            Image image1, image2;
            for (int i = 0; i < 100; ++i)
            {
                // Grab image
                GigEVision_ReceiveImage(hDev, image1);

                // Process image
                ThresholdImage(image1, NIL, 128.0f, NIL, 0.0f, image2);

                // Save to file
                char fileName[16];
                sprintf(fileName, "%05d.png", i);
                SaveImage(image2, ImageFileFormat::PNG, fileName, false);

                // Finalize acquisition
                fil::GigEVision_StopAcquisition(hDev);
                fil::GigEVision_CloseHandle(hDev);
                return 0;
            }
        }
        else
        {
            return -1;
        }
    }
    catch (ftl::Error& error)
    {
        std::cout << error.Kind() << ": " << error.Message() << std::endl;
        return -2;
    }
}
```

## LICENSING MODEL

### License types

There are two types of commercial licenses:

- **Development:** assigned to a single engineer. It includes one year of technical support, which can be extended with an annual fee. Valid technical support also gives you the right to upgrade the software to newer versions and provides a discount on runtime licenses.
- **Runtime:** assigned to a single vision system. You can use one license for one multi-camera system, but multiple licenses are required to control multiple independent systems, even if run on a single physical computer. There are two flavors: 'Studio Runtime' and 'Library Runtime'.

### License carriers

There are two types of license carriers:

- **Computer-ID:** these licenses are stored in a computer file and work only with a specific computer.
- **USB dongle:** these licenses are stored on a device that can be used with many different computers (but still by only one person in case of development licenses).

We suggest using the USB Dongle solution for development licenses and the Computer-Id solution for runtime licenses.