

# Hidden Treasures

Hardware Constructors - or the Art of Designing a Frame Grabber

„Everything must change that everything will remain as it is“. Unpleasant but daily life for a requirement change to a Machine Vision system. If the performance of the computer is exploited and restrictions in quality exhausted, there are only a few remaining possibilities of reacting to new conditions.

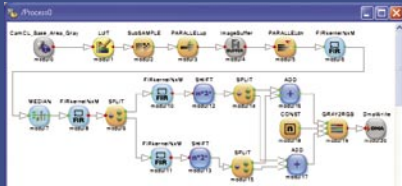


Fig. 1: Dataflow elements in VisualApplets

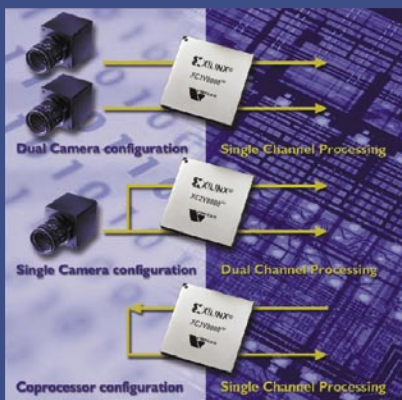


Fig. 2: Basic configuration schemes for Visual Applets

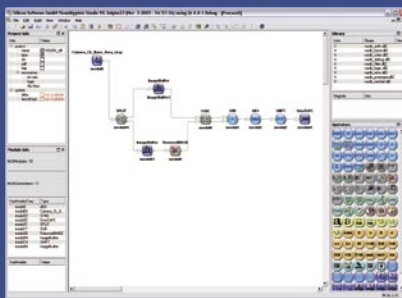


Fig. 3: Visual Applets surface with data flow of "difference images" application

Who thinks of image processing turns his look towards software packages for the PC first. But even the impressive feature variety is not able to handle all tasks at the current amounts of data in real time. The evacuation of pre-processing tasks on the frame grabber hardware might be one solution. Since the frame grabber doesn't use any classic CPU generally, the access to this processing resource is more complicated. In the case of an efficient FPGA technology it is necessary to have a look into hardware programming. However, in many cases hardware programmers aren't available in any office.

## Designing Hardware Without Soldering Iron

The software solution Visual Applet isn't addressed to hardware programmers as a matter of priority. Visual Applet eases the first steps of creating a FPGA design for a software programmer.

Visual Applet is a graphically oriented programming tool for FPGA hardware. The central window is the construction area. This is the place, where the data flow elements are combined with each other. The library windows and information bars are arranged besides (Fig. 1). The extensive library is separated in hardware modules and algorithmic elements. The separation corresponds to a hardware-level function replica of the frame grabber and the access to the FPGA for the image pre-processing. The algorithms are represented by data flow types and models. The resulting data flow is stored and processed. These data flow elements are used as a describing construct for the definition of the pre-processing tasks. The modules can be parameterized, e.g., in the function, behaviour or data format.

The data flow scheme is converted to function-related blocks for a FPGA layout.

This operation is processed without any additional entries of the user. There is no configuration necessary for synchronization, timing or problems with side-effects.

A VHDL compiler isn't needed as well. Both a synthesis and a high level simulation are integrated in the software. The user keeps control over effects by changes and the final visual result of his application at any time. The simulation can be integrated as independent module into the data flow and displays the progressing result at any point of the processing. Since the simulation runs on software base, the

calculation of resulting images is not hardware accelerated. The user controls the complexity of the pre-processing related to the expected result by the display of processing resources, load of resources and expected data bandwidth.

The Place & Route process converts the hardware design into a FPGA layout. An additional tool of the FPGA manufacturer is needed here. Once installed, the software is automatically integrated, configured and executed. The complete process of the hardware design creation lasts 10-15 minutes on an average. After this process, a so-called hardware applet is automatically created, which can be loaded in microDisplay. microDisplay is a configuration and viewer program for the FPGA hardware. The functions of the hardware applet, which are defined as „adjustable“ can be configured in microDisplay or with use of the SDK. A documentation with listing of the parameters and value ranges is automatically created. In this combination a complete software interface is made available for the user.

In first approximation, Visual Applet is hardware-independent. At present microEnable III product line is the only supported hardware platform. Visual Applets supports Windows 2000 and XP.

## Calculated in Hardware

The hardware programming in Visual Applets is the combination and parameterization of hardware-level modules and algorithmic elements in a data flow structure. The interfaces to the frame grabber work hardware-dependently, however the pre-processing on the FPGA hardware-independently.

The representation of the hardware are the modules CameraLink, Trigger, Image Buffer and DMA.

The CameraLink interface supports up to two Base cameras (single and dual) or one Medium camera. Standard as well as Non-Standard cameras are supported by the opportunity of the individual configuration. Arbitrary combinations of Lines and Area Cameras or Grey Scale and Colour Cameras are allowed.

The Trigger is eligible as a module for Area or Line Cameras. The DMA modules support Write and Read function. The Image Buffer module contains function of the onboard RAM access and the sensor correction for the Tap sorting.

Hardware designs can be configured as a two channel data flow. This enables a setup of two different and independent processing units. This

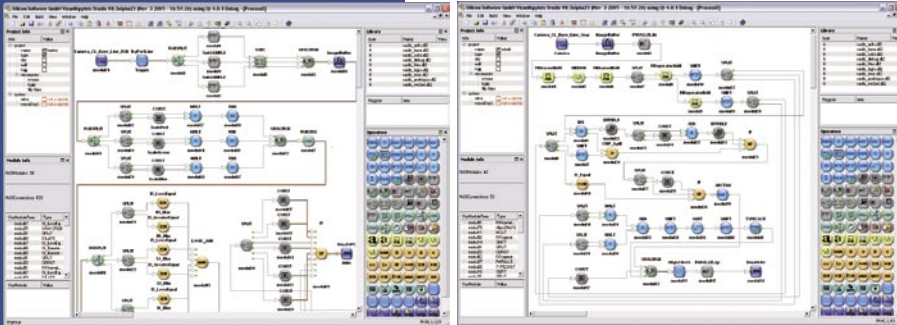


Fig. 4: Visual Applets surface with data flow of HSI segmentation and Sobel edge detection application

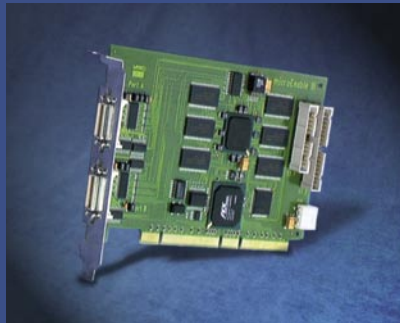


Fig. 5: Visual Applets supporting hardware platform microEnable III I-XXL

opportunity enables three different types of configuration schemes in general (Fig. 2):

- The use of two cameras, each with an individual processing channel.
- The use of one camera with two independent processing channels, e.g. one original image acquisition stream as protocol channel and one channel as pre-processing stream.
- The use of the coprocessor capability with one DMA Read channel, which reads data from the PCI bus, and one DMA write channel.

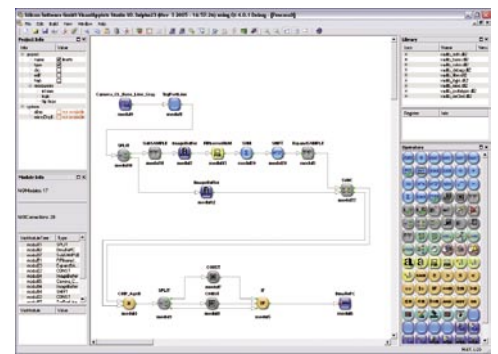
There is an extensive library provided for the implementation of the algorithms. A base library supports the processing of pixel fields by extraction functions. It can be used for images, colour channels, components, bit field, or pixels. Split, sampling and synchronization functions complete the content of the library. The library of the arithmetic operators supports the pixel manipulation. It contains mainly operators for type conversions and base operators like the addition, subtraction, multiplication and division, further mathematical functions, usage of trigonometrical functions, or the arithmetic with powers. The Filter Factory offers different kinds of the pixel field manipulation, e.g. determining minima, maxima, median and morphological operators. The libraries with more complex functions are pre-combined modules and can be used as a block module by the user. One library is dedicated to the colour processing. A focus of the library is the colour space conversion from RGB to HSI,  $L^*a^*b$  or YUV. The other focus is the colour interpolation, which is used for the Bayer filter reconstruction. Visual Applets supports the high-quality, edge-directed LaPlace  $5 \times 5$  conversion as well as the standard bilinear  $3 \times 3$  conversion. Another library contains modules for Lookup and Knee-Lookup tables, threshold, binarisation and counter functions.

For the use of classification tasks logical operators are integrated. Boole operators are implemented as well as selector operators.

Further libraries are in preparation. Outstanding features will be a compression library with RLE and JPEG support and one library with BLOB analysis tools.

With the increasing number of combinations and complexity of the pre-processing, the automatically performed rule check is an important part of the software. A visual control by the software based simulation and bit-based image checks allows the evaluation of the partial and final result at any time.

The hardware applets are generally operation system independent. Only the availability of suitable drivers for the frame grabber limit the use in application.



**SILICONSOFTWARE**

Laetia hent nim dolendigna facipit illi lore con et lamcoore eummy nit ipsostrud tiscin henim qui at. Loreet augait volobore magnibh el ut wisi. Lore dolendre modipisim in et prate vulpore facum ad eros delendion volore conulput wisi te velit vullian henim volorem velendre min velor austrud tio conseniam, vercidis dolere delatata velid in henim do odiam istrud magnibh assim dit alit, quicid do cons nonto od ting cogait naim dignibh et utpat, volore

**SILICONSOFTWARE**

Laetia hent nim dolendigna facipit illi lore con et lamcoore eummy nit ipsostrud tiscin henim qui at. Loreet augait volobore magnibh el ut wisi. Lore dolendre modipisim in et prate vulpore facum ad eros delendion volore conulput wisi te velit vullian henim volorem velendre min velor austrud tio conseniam, vercidis dolere delatata velid in henim do odiam istrud magnibh assim dit alit, quicid do cons nonto od ting cogait naim dignibh et utpat, volore

Fig. 6: Visual Applets surface with data flow and example of an adaptive threshold application

### Hardware control

A flexibility even on hardware-level will become a more and more important part of advanced applications in Machine Vision. Just by the tendency to integrate the frame grabber into camera systems with PC standard interfaces, will enhance the meaning of pre-processing and data compression. A transparent and simple control of the hardware gives a better chance to use available resources and operate more flexible with existing components.