

Versteckte Schätze

Hardware Konstrukteure,
oder die Kunst, einen Framegrabber zu bauen

„Alles muss sich ändern, damit es bleibt, wie es ist“. Für eine Anforderungsänderung an ein System in der industriellen Bildverarbeitung eine unangenehme, aber täglich eintretende Tatsache. Wenn die Leistungsreserven des Computers ausgeschöpft und qualitative Einschränkungen ausgereizt wurden, bleiben nur wenige Möglichkeiten, auf neue Rahmenbedingungen zu reagieren.

Wer an Bildverarbeitung denkt, richtet seinen Blick zuerst auf Softwarepakete für den PC. Aber selbst bei beeindruckenden Funktionsvielfalt können nicht alle Aufgaben bei den heute anfallenden Datenmengen in Echtzeit bewältigt werden.

Hier bietet sich die Auslagerung von Vorverarbeitungsaufgaben auf die Framegrabber-Hardware an. Da der Framegrabber im Allgemeinen keine klassische CPU besitzt, ist der Zugang zu dieser Processing-Ressource komplizierter. Im Falle einer leistungsstarken FPGA Technologie muss man sich mit Hardwareprogrammierung auseinandersetzen.

In vielen Fällen stehen jedoch Hardwareprogrammierer nicht zur Verfügung.

Hardwarebau ohne Lötcolben

Die Softwarelösung VisualApplet richtet sich vorrangig nicht an den Hardwareprogrammierer. Bewusst wurde auf den Ansatz geachtet, einem Softwareprogrammierer den Einstieg in die Erstellung eines FPGA-Designs zu erleichtern.

VisualApplet ist ein graphisch orientiertes Programmierwerkzeug für FPGA Hardware. Der zentrale Bereich ist das Konstruktionsfenster. Hier werden die Datenflüsselemente miteinander kombiniert. Seitlich gruppieren sich Bibliotheksfenster und Informationsbereiche.

Die umfangreiche Bibliothek ist in Hardwaremodule und Algorithmelemente eingeteilt.

Der Aufbau entspricht einer hardwarenahen Funktionsnachbildung des Framegrabbers und dem Zugriff auf den Bildprozessor für die Bildvorverarbeitung. Die Algorithmik wird in Daten- und Flussmodelle abgebildet. Der resultierende Datenfluss wird aufgezeichnet und abgespeichert. Diese Datenflüsselemente dienen als Sprachkonstrukt für die Definition der Vorverarbeitungsaufgaben. Die Module können parametrisiert werden, z.B. in der Funktion oder im Datenformat.

Aus dem Datenflussschema werden funktionale Blöcke für das FPGA Layout erstellt.

Dieser Prozess erfolgt ohne weiteren Eingriff des Anwenders. Eine Auseinandersetzung mit der notwendigen Synchronisation, dem Timing oder der Problematik einer Seiteneffektfreiheit entfällt.

Ein VHDL Compiler wird ebenso nicht benötigt. Sowohl eine Synthese als auch eine High Level Simulation ist in der Software integriert. Hierdurch behält der Anwender stets die Kontrolle über Auswirkung durch Änderungen und das visuelle Endergebnis seiner Applikation. Die Simulation wird als eigenständiges Modul in den Datenfluss integriert und kann somit an jeder Stelle innerhalb der Verarbeitung das Teilergebnis anzeigen. Da die Simulation auf Softwarebasis läuft, sind die resultierenden Bilder nicht hardwarebeschleunigt zu sehen.

Über die Anzeige und Ausgabe von Ressourcenreserven, -verbrauch und dem geschätzten Datendurchsatz kann der Anwender die Komplexität seiner Vorverarbeitung im Abgleich mit dem zu erwarteten Ergebnis steuern.

Für den Place & Route Vorgang, also der Abbildung einer Hardwareschaltung in ein FPGA-Layout, werden zusätzliche Werkzeuge des FPGA-Herstellers benötigt. Einmal installiert, werden diese automatisch aufgerufen, konfiguriert und eingebunden.

Der gesamte Vorgang der Erstellung eines Hardware-Designs dauert durchschnittlich 10-15 Minuten.

Nach diesem Durchlauf wird ein so genanntes Hardware-Applet erstellt, das automatisch in microDisplay aufgerufen werden kann. microDisplay ist ein Konfigurations- und Viewer-Programm für die FPGA-Hardware. Die als parametrisierbar definierten Funktionen des neu erstellten Hardware-Applets können über microDisplay oder Verwendung mit dem SDK konfiguriert werden.

Für die Verwendung in eigenen Applikationen erstellt das Programm automatisch eine Dokumentation mit verwendeten Parametern und dazugehörigen Bereichsgrenzen.

Visual Applet ist in erster Näherung hardwareunabhängig. Derzeit existiert nur eine Anbindung an die microEnable III Hardwareplattform. Visual Applets ist lauffähig unter Windows 2000 und XP.

Rechnen in Hardware

Die Hardwareprogrammierung erfolgt in Form eines Datenflusses mit Modulen für hardwarenahe und algorithmische Elemente. Die Schnittstellen zum Framegrabber sind hardwareabhängig, hingegen die Vorverarbeitung auf dem FPGA hardwareunabhängig.

Die Hardwarerepräsentation erfolgt durch die Module CameraLink, Trigger, Bildspeicher und DMA.

Das CameraLink Interface unterstützt den Anschluss bis zu zwei Base-Kameras (Single und Dual) oder einer Medium-Kamera. Standard sowie non-standard Kameras werden durch die Möglichkeit der individuellen Konfiguration unterstützt. Zeilen- und Flächenkameras oder auch Grauwert- und Farbkameras können beliebig kombiniert werden.

Der Trigger ist als Funktionseinheit für Flächen- oder Zeilenkameras auswählbar.

Die DMA Module unterstützen sowohl die Write-, als auch Read-Funktion.

Der Bildspeicher beinhaltet Module zur Ansteuerung des onboard RAMs und der Sensorkorrektur zur Tap-Sortierung.

Die Designs können zweikanalig aufgebaut werden. Dieses ermöglicht zwei unterschiedliche und unabhängige Verarbeitungsstränge. Ebenso



Abb. 1: Datenflussmodell in VisualApplets für microUSB2



Abb. 2: Konfigurationsschemata in VisualApplets



Fig. 3: Bedienoberfläche in Visual Applets mit Design Flow für eine Differenzbild-Anwendung

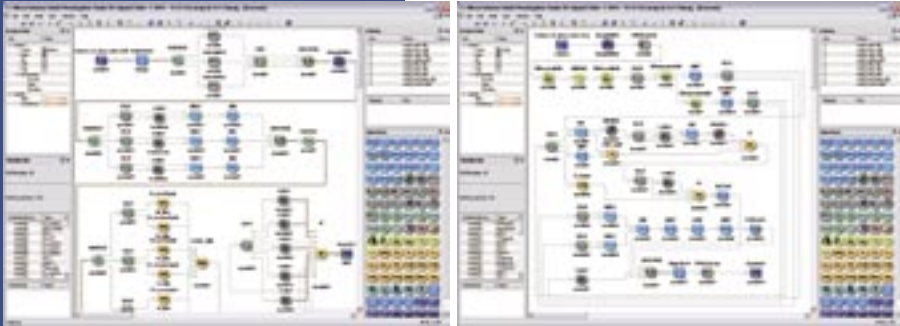


Fig. 4: Bedienoberfläche in Visual Applets mit DesignFlow und Beispiel für kombinierte HSI Segmentierung und Sobel Kantenfilterung.

ist es möglich, die Kanäle umzukonfigurieren, um einen Protokollkanal und einen vorverarbeitenden Kanal zu erhalten oder über die Nutzung der Funktionen DMA Read und DMA Write einen Koprozessor zu konfigurieren, der Daten aus dem PCI Bus einliest und verarbeitet. Generell gibt es drei Schemata für die Konfiguration: zwei Kameras mit je einer unabhängigen Vorverarbeitung, eine Kamera mit zwei unabhängigen Vorverarbeitungen oder eine kamera-lose Vorverarbeitung.

Für die Implementierung der Algorithmen wird ein umfangreiches Instrumentarium bereitgestellt.

Eine Basisklasse dient der Bearbeitung von Pixelfeldern über Extraktionsfunktionen, auf Basis von Bildinhalten, Farbkanälen, Komponenten, Bitfelder oder Pixeln. Split-, Sampling- und Synchronisationsfunktionen ergänzen diese Klasse. Die Bibliotheksklasse der arithmetischen Operatoren dient zur Pixelmanipulation. Neben Konvertierungstypen, beinhaltet sie hauptsächlich Basisoperatoren zur Addition, Subtraktion, Multiplikation und Division, den Umgang mit Winkelfunktionen oder dem Rechnen mit Potenzen.

Die Filter Factory bietet verschiedene Arten der Pixelfeldmanipulation, dem Bestimmen von Minima, Maxima, Median und morphologischen Operatoren.

Die Bibliotheken mit komplexeren Funktionen sind ihrerseits aus Modulen aufgebaut und können vom Anwender als Block verwendet werden.

Eine Bibliothek ist der Farbverarbeitung gewidmet. Hier werden Farbraumkonvertierungen aus der RGB-Farbraum und die Rekonvertierung in den RGB-Raum vorgenommen. Die Farbinterpolation dienen der Bayerfilterumrechnung. Neben der Standard bilinearen 3x3 Umrechnung kommt auch die hochwertige LaPlace 5x5 Umrechnung zum Einsatz.

Eine weitere Bibliothek bietet die Verwendung von Modulen für Lookup- und Knee-Lookup-

Tabellen an, Schwellwertberechnungen, Binarisierung und Zählerfunktionen.

Zur Verwendung von Klassifizierungsaufgaben können logische Operatoren herangezogen werden. Boolesche Operatoren sind ebenso wie Selektionsoperatoren in die Klasse implementiert. Weiter Bibliotheken sind in Vorbereitung. Hier sind die Kompression mit RLE und JPEG, aber auch BLOB Analysewerkzeuge zu nennen.

Bei der steigenden Anzahl der Möglichkeiten und Komplexität der Vorverarbeitung ist die Überprüfung des Regelwerks ein wichtiger Bestandteil. Die softwarebasierte Simulation ermöglicht zu jeder Zeit die Kontrolle über das Ergebnis. Somit besteht eine visuelle und bit-basierte Kontrolle. Schließlich wird ein lauffähiges Applet erzeugt, das umgehend einsatzfähig ist und in microDisplay geladen oder im SDK verwendet werden kann. In dieser Kombination wird ein vollständiges Softwareinterface für den Anwender zur Verfügung gestellt.

Die erzeugten Hardware Applets sind generell betriebssystemunabhängig und nur von der Bereitstellung geeigneter Treiber für den Framegrabber abhängig.

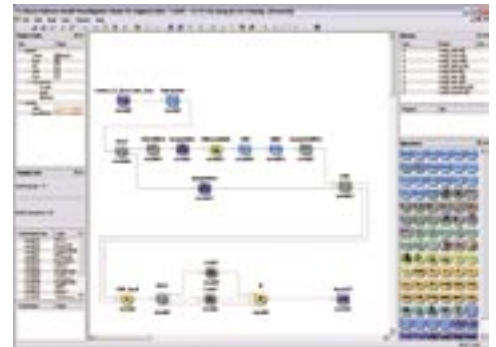


Fig. 6: Bedienoberfläche in Visual Applets mit Design Flow für eine "adaptive Threshold" Anwendung.

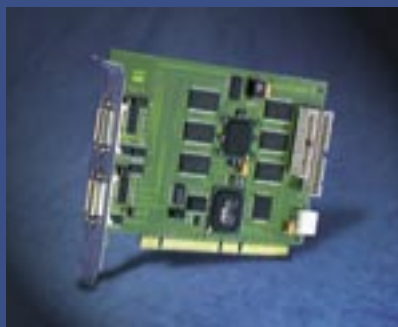


Abb. 5: microEnable III/-XXL als unterstützten Hardwareplattform

Hardware-Kontrollreure

Die Flexibilität auf Hardwarebasis wird ein immer wichtigerer Teil anspruchsvoller Anwendungen in der industriellen Bildverarbeitung werden. Gerade durch die Tendenz, die Aufgaben eines Framegrabbers in Kamerasysteme mit PC-Standardchnittstellen zu integrieren, werden Vorverarbeitungsaufgaben, die zudem auch Daten komprimieren, wichtig bleiben. Die Kontrolle über die Hardware ermöglicht dem Anwender eine bessere Ausnutzung vorhandener Ressourcen und einen flexibleren Einsatz von Komponenten.

Aktualisiertes Reprint aus inspect 03/2005 „Versteckte Schätze“, GIT Verlag

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 Visual Applets

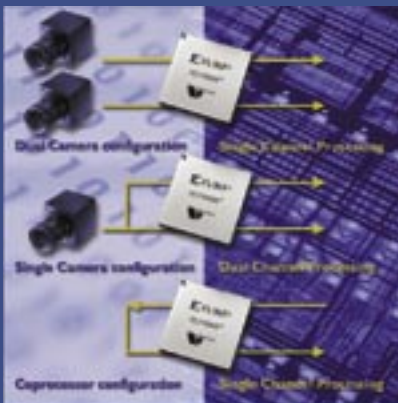


Fig. 2: Basic configuration schemes for Visual Applets

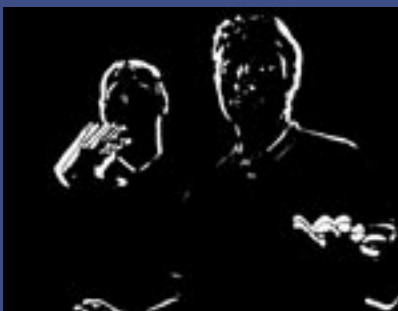


Fig. 3: Visual Applets surface with data flow of "difference images" application

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

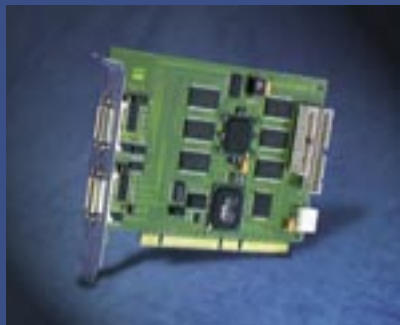


Fig. 5: Visual Applets supporting hardware platform microEnable III /-XXL

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 5×5 conversion as well as the standard bilinear 3×3 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.

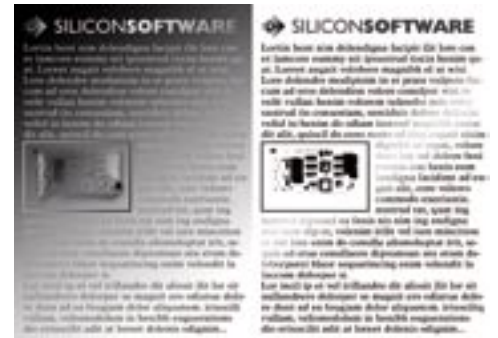
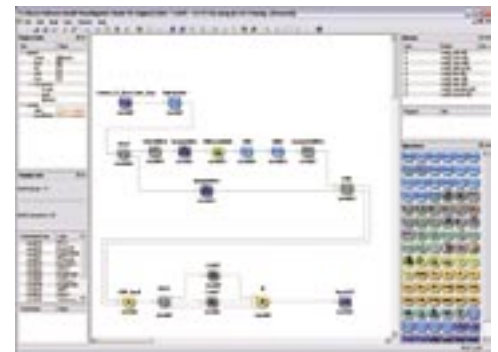


Fig. 6: Visual Applets surface with data flow and example of an adaptive threshold application

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.