IMAGING INSIGHT
Solutions, products and news from Matrox Imaging

Vol. 9 No. 2

OUR SOFTWARE ISSUE!

- The Vision Squad Files: Register images to see the big picture
- MIL 9.0 operates at full throttle
- Case Study: MIL and Iris smart camera bring glass inspection out of the dark
Contents
IMAGING INSIGHT Vol. 9 No. 2

VP's Commentary ........................................ 03
Vice President of Sales & Marketing François Bertrand offers a few words about our latest software innovations

Recent News ............................................. 04
MIL Training Update: sharpen your skills

Featured Product: Matrox Imaging Library .......... 05
Behind the scenes of the newest features of MIL 9.0

Product News: Design Assistant ...................... 08
Q & A: What our smart camera/software package can do for you!

Case Study: Spectral Process .......................... 10
A German engineer uses custom lighting, MIL and the Matrox Iris P-Series smart camera to inspect glass bottle openings

The Vision Squad Files .................................. 13
How to effectively use MIL’s registration module

Trade Show Calendar
Visit Matrox Imaging on the road!
For tickets or more info, contact Joanna Puccio: jpuccio@matrox.com.

EMAF (Representative: UNITRONICS, S.A.)
Oporto, Portugal
November 12-15, 2008

PHOTONICS (Representative: UNITRONICS, S.A.)
Valencia, Spain
November 26-27, 2008

HOMSEC (Representative: UNITRONICS, S.A.)
Madrid, Spain
December 1-4, 2008
Welcome to ‘The Software Issue’ of Imaging Insight. It’s true we have lots to say about software - the momentum keeps building! We recently announced Matrox Imaging Library (MIL) version 9.0 and Matrox Design Assistant version 2.0, our software platforms. We’re proud that we can offer new and innovative image processing tools for your application.

Let’s put this into a context: the PC and chip manufacturers have given us machines that we could only dream of in the recent past. The processing power we can achieve today would have cost thousands of dollars a few years ago. Not only are we seeing multiple-core PCs as the norm, we can also employ 64-bit operating systems and take advantage of yet-to-be-tapped processing resources in DirectX 9-compliant GPUs.

MIL 9.0 supports multi-core processing, offloading to GPUs, and makes it easy to distribute MIL tasks across a network. It doesn’t matter if you call it distributed processing or cluster-based processing; achieving higher performance gains has never been easier.

But there’s more to talk about than MIL. We’re also proud of Matrox Design Assistant 2.0, the non-coding development environment for our Matrox Iris E-Series smart cameras. The latest version supports communications protocols for automation equipment such as PLCs, and features a tool to integrate custom algorithms. And speaking of smart cameras, this issue’s case study hails from Germany, where a customer inspects glass bottle openings with MIL and a Matrox Iris P-Series smart camera. Finally, our Vision Squad offers readers some tips for using MIL’s Registration module, one of our newer MIL processing tools.
Applications reach new heights with Matrox Imaging software training

Machine vision is not an exact science – and this means there can be several ways to process your images. But a little knowledge can go a long way. That’s why we have created a variety of training programs that help developers increase productivity, reduce development and engineering costs, and bring projects to market sooner.

Here’s what we’re offering:

**Introduction to the MIL/MIL-Lite Environment (classroom environment)**
This training offers a general overview of MIL/MIL-Lite. Students learn how to set up their development environment and manage image buffers for image capture and display. With “hands-on” workshops and the MIL Development and Applications Engineers available to answer questions, participants leave the training with the knowledge needed to solve real-world applications.

**Matrox Imaging Library (MIL) Processing (classroom environment)**
This training shows students how to choose and apply the correct image processing tool for your next application! With “hands-on” workshops and the MIL Development and Applications Engineers available to answer questions, attendees leave the training with the knowledge needed to solve real-world applications. This in-depth training course covers all of MIL’s processing tools and includes new MIL 9.0 features:
- Multi-processing with MIL
- GPGPU with MIL
- MIL Registration
- MIL color analysis tools
- MIL 3D calibration and mapping tools

**Matrox Imaging FPGA Developer’s Toolkit Parts 1 and 2 (webinars and classroom environment)**
Part 1 is created for software developers who need to interface and control FPGA configurations from the MIL/MIL-Lite environment. The training covers fundamental FPGA concepts such as offloading selected standard MIL/MIL-Lite functions, and optimizing FPGA configurations using the MIL/MIL-Lite Mfpga API.

Part 2 is created for FPGA/hardware developers who need to create custom FPGA configurations for the Matrox Solios and Matrox Odyssey Xpro+ boards. This training explores the components provided in the Matrox SOPC library as well as Matrox and third-party development tools.

Be sure to visit matroximaging.com/training or email imaging.training@matrox.com for more information, course pre-requisites, and dates!
Source code:

Behind the scenes of MIL 9.0

The Matrox Imaging Library (MIL) 9.0 has a lot to brag about. Here we bring you an insider’s look to explain why the new features in MIL 9.0 are so beneficial.
Multiprocessing
Why is multiprocessing so important? Virtually all new PCs are equipped with multi-core processors. Making actual use of them is one of the most obvious ways to improve the performance of your application. However, what is not so obvious is how to code for multiprocessing. Matrox Imaging has a team of developers who specialize in optimizing code for speed, and they’ve made MIL 9.0 automatically detect the multiple cores in a system and judiciously dispatch operations to them for maximum performance. When accelerated processing is transparent to the developer, the application will automatically perform faster when the system has multiple cores. Of course, developers are given the leeway to assign the number of cores available to and actually used by MIL 9.0.

“Programming the communication between multiple threads and multiple cores is complex and requires in-depth knowledge – not only of CPU architectures but also the operating system used to run the application. Multiprocessing performance gains vary from function to function. An algorithm that runs fast on a single processor may not be suitable for splitting across multiple cores. Simple addition functions behave this way, showing only fractional improvement when split over multiple cores; they are limited by memory bandwidth. Operations such as large convolutions, however, demonstrate almost linear improvement over multiple cores.” Stephane Maurice, Software Development Director.

Distributed processing and monitoring
Today, the means to interconnect computing devices is ever-present. This not only enables remote process monitoring but it also allows for scaling performance outside of a single computing box. Effectively coding for it is no easy feat. Matrox Imaging’s developers designed Distributed MIL (DMIL) to simplify the programming of applications that run across multiple computers. A task that cannot be performed on a single computer, or one that would require very expensive workstations/servers can easily be distributed across multiple computers using Gigabit Ethernet or other high-speed interconnects – this provides an almost infinite scaling to many applications. DMIL transparently dispatches commands, transfers data, sends and receives event notifications, and performs function callbacks across systems.

“General-purpose inter-PC communications must serve a whole range of applications, and don’t necessarily consider the intricacies of the image processing domain. To implement DMIL so that it would perform transparently was a great challenge. DMIL was designed and optimized for image processing and high-throughput data transfer. In the end, DMIL is the ideal platform for image processing applications that need to run on multiple PCs.” Stephane Maurice.

Speed-up factor versus multiple cores

Large convolutions demonstrate almost linear improvement over multiple cores, while addition operations hardly improve at all over multiple cores.
GPU processing
Originally designed to satisfy the demands of the video game industry, Graphics Processor Units (GPUs) are now viewed as an additional, but powerful, general-purpose processing resource. MIL 9.0 supports offloading under Windows® with DirectX® 9-compliant GPUs. The Matrox Imaging developers chose to base GPU processing on DirectX® (DX) 9 because it is vendor neutral and supports the largest number of existing graphics boards. Developers who code in DX9 are coding for all DX9-compliant GPUs. Graphic boards are designed to receive data for display purposes, so programming GPUs as an imaging co-processor requires a solid understanding of I/O functions and system bottlenecks. The Matrox Imaging developers implemented both an underlying synchronization mechanism and a protocol to push data to and pull it from the graphics board. And not unlike multiprocessing, not all algorithms can be accelerated by a GPU. Neighborhood and point-to-point operations respond very well to GPU acceleration, but again, it’s often the bus bandwidth to the GPU that limits overall performance.

“It’s important to remember that a graphics card is not and does not behave like a processing board. We needed to rethink our implementation strategies function by function in order to get the best performances.” Stephane Maurice.

Color analysis
Color tools can reveal the extent of color differences in and between images, separate features from an image based on their colors, identify objects based on their color, and offer enhanced color-to-grayscale conversion.

“With the intensifying demands placed on vision systems, the use of color is a growing avenue to discern features otherwise difficult or impossible to detect with monochrome setups. Matrox Imaging now has and will continue to pursue tools to make use of the additional information offered by color.” Arnaud Lina, Processing Software Development Manager.

3D calibration and reconstruction
Pinpoint a camera, measure features and locate an object in 3D space. When the work plane is at a known place but different from the calibration plane, MIL 9.0’s 3D calibration tool makes the necessary adjustments. MIL can also position an object of known geometry and dimension or a known feature using a monocular or stereo vision setup respectively. These techniques enable vision-based robot guidance. MIL also has a tool for performing laser-based 3D profiling. The tool generates a calibrated depth map of a surface or the fully calibrated 3D cloud of points of an object for subsequent analysis. It can also be used to create the 3D reproduction of a scanned object using a third-party tool.

“It was a real challenge to keep the same portable API and level of performance for MIL under Windows and Linux while fully and efficiently supporting all the features of each OS. We ported MIL to 64 bits with great care. We studied, benchmarked, and tailored 32-bit MIL to use all benefits of 64-bit technology. All the MIL objects were generalized to 64-bit, and all the processing functions reoptimized for it. We’re proud of having minimized portability requirements and workload for the existing 32-bit MIL users – we can help them move forward without significant application rework.” Stephane Maurice.
Integrated development environment lets you build vision applications one step at time!

The machine vision industry is still buzzing about smart cameras. It's a hot topic, and Matrox Imaging continues to fuel the fire with Design Assistant 2.0. The interactive software package designed for our Matrox Iris E-Series smart camera has all the necessary tools that system integrators need to develop their own custom, robust vision applications. Read on to learn what Matrox Design Assistant 2.0 has in store for you.
When should I use a smart camera?
There are no rules that define when you should use a smart camera. There are, however, many benefits to using one, especially if you have a straightforward inspection application. Smart cameras offer simple system integration – camera, processor, and software all come in a single package from a single vendor.

What makes Matrox Design Assistant different from other smart camera packages?
Design Assistant’s intuitive interface lets you build your application graphically with a flowchart. Grabbing an image, locating a feature on a part, and analyzing the feature are procedures that are created by adding “steps” to your application: a grab step, a geometric model finder step, a metrology step, and so on. But if your application requires a custom filter or uses proprietary algorithms, Design Assistant allows you to add them. Finally, Design Assistant has its roots in the Matrox Imaging Library (MIL), Matrox Imaging’s field-proven development kit for imaging and machine vision applications.

But what about traditional programming or “coding”?
There’s no traditional programming (i.e., writing code) required to use the standard steps in Design Assistant. While you’re developing your application in Design Time, each step displays all the information you need to configure that step: the parameters, context-sensitive help, and the results after each step. Finally, when your application has been tested and fine-tuned, you can configure your Operator View. The Operator View is the GUI that is displayed when the application is running. During deployment, the instructions associated with the flowchart are automatically generated and downloaded to the Matrox Iris E-Series camera along with the operator interface. In your web browser, enter the camera’s network name on any PC to access the Operator View and run!

Can Design Assistant use my own proprietary algorithms?
Yes – that’s what makes Design Assistant so flexible. Design Assistant 2.0 provides advanced users with a tool to create a custom step in your application, a flowchart step for extra processing, communication protocols, and logging and reporting. You add the custom step with the Custom Step SDK. In Microsoft® Visual Studio®, you write C# code, compile and add the resulting dll files to the library of steps in Design Assistant.

I need to communicate with other automation devices. Does Design Assistant allow me to do that?
Absolutely. The Matrox Iris E-Series camera features a 10/100 Mbit Ethernet interface, and supports MODBUS® over TCP/IP and EtherNet/IP™ to control robots or PLCs. You can also rely on the RS-232 serial interface and 16 general purpose I/Os.

Does Design Assistant really have the tools to get the job done?
Yes! Design Assistant 2.0 has a complete set of image processing and analysis tools for calibrating, enhancing and transforming images, locating objects, extracting and measuring features, reading character strings, and decoding and verifying identification marks. All the Design Assistant steps are built upon the Matrox Imaging Library (MIL), which has a 15-year, field-proven history.
The bright spot

Smart camera and MIL shine a light on glass inspection
The melting of sand and forming it into glass is a manufacturing process that dates back to ancient days. Though contemporary production methods benefit from greater scientific knowledge of the properties of various additives to improve the durability of the glass itself, the manufacturing process is far from perfect. The finished glass can contain bubbles or cracks, so each piece must be inspected. Thorsten Gonschior, President and founder of Spectral Process (Erkelenz, Germany) says that formal inspection of glass is as old as glass production itself. “And there are still some regions today where inspection is performed manually, by the human eye.”

Some plants rely on mechanical inspection systems that have some kind of direct contact with the container. The container might be rotated, be filled with compressed air, or plugged with a gauge. There are also many high-end camera-based inspection systems, but Gonschior explains that even though manufacturers are under pressure to maintain strict quality control (especially for food and beverage containers), there are plant owners worldwide who are not capable or willing to spend €250 000 to 500 000 for an inspection system. Cost is clearly an issue, but manufacturers understand they have little choice but to keep up with high quality standards. They must also improve the production process if they want to maintain a competitive edge.

Gonschior was confident that he could provide a lower-cost alternative when one of his customers asked him to retrofit an existing machine. “That original system had an optical component to locate the defects, but no processor,” he says. “The original machine was no longer available on the market, so we had the idea of replacing one part of the full inspection process with a scalable subsystem.”

**Hot stuff**

Glass is made when a mixture of silicon oxide, sodium carbonate, calcium oxide, magnesium oxide and other minerals are heated to a temperature well over 1000°C. If the glass is intended for containers, the molten glass is pressed into molds and then annealed, that is, slowly cooled in a temperature-controlled kiln. At this point, cold end inspection is performed on the glass. Gonschior calls his system the Opening Inspector, as it checks the opening of hollow glass containers (i.e., bottles) for cracks, enclosures (bubbles), and pressed artifacts. “Keeping defective bottles off a production line is essential for the beverage industry. Anything but a smooth surface can cause injuries. Carbonated beverages might go flat if the bottles have defects and can’t be sealed properly,” he says. The container, then, must be free of burrs, chips, or sharp edges if it is to pass inspection.

**Keep it simple**

The Opening Inspector can be retrofitted for a wide variety of glass inspection machinery. The “sub-system” consists of a Matrox Iris P-Series smart camera, a power supply and a custom-designed illumination device. Gonschior opted for the Matrox Iris P-Series camera for a number of reasons. “It’s cost-efficient, small, yet powerful for its range,” he says. A smart camera also eliminates a lot of what he calls “additional engineering”: housing, computer, electrical connections, etc. More importantly, “After testing the Iris [camera], I realized Matrox had done a good job... I found no flaws or non-functioning components in the hardware or software like I usually do with [off-the-shelf] components.”

As the core of the system, the Matrox Iris P-Series camera not only performs the visual inspection, but also reads the sensors and updates actuators through digital I/O. The software application uses a number of modules in the Matrox Imaging Library (MIL), in particular Blob Analysis, Edge Finder, and Metrology, to measure the container’s inner and outer diameters and locate the enclosures, cracks, and over-pressed structures. The eye can easily distinguish over-pressed structures, but the software requires clear definitions...
to detect them. In theory the Opening Inspector works with glass of any color; clear, brown and green glass are considered standard. Adjusting the intensity of the illumination device or the camera’s amplifier creates the right conditions to acquire a useable image. Clear glass is the most challenging of all to inspect.

A number of complex sub systems can be integrated into the Opening Inspector with Ethernet connections. For example, Gonschior is planning to use a 2D actuator with high way resolution to control a labeling arm. “Integrating network-capable third party devices into the Matrox Iris network is easy and straightforward,” notes Gonschior.

Reflections on illumination
“Glass has a bad reputation when it comes to illumination,” notes Gonschior. Indeed, both the material (the glass) and its shape create illumination challenges. On a microscopic level, glass is non-uniform, has a moon-like surface, with craters. These irregularities affect the way the light reflects and refracts through the surface; the resulting images can show strong contrasts in the background micro structure. The roundish bottle shape and deviations in the bottle’s wall thickness only compounds the problem. Gonschior was able to resolve many illumination issues with his custom solution. The light source uses diffuse light and makes the glass reflect light into the camera at those spots that contain the damage, (the inspection vector). “It sounds easy but the development was anything but,” recalls Gonschior.

A bright idea
Gonschior believes the Opening Inspector has an edge over the competition. By developing his system with Matrox Imaging’s Iris P-Series, he can offer the system at a fraction of the cost of many “established” glass inspection systems. Moreover, manufacturers want to inspect over 400 containers per minute. With his scalable design, Gonschior simply adds more cameras to the existing inspection line when a higher throughput is needed.

What’s next?
Once Gonschior is satisfied with the Opening Inspector’s robustness and stability in the industrial environment, he plans to offer additional inspection stations. These will inspect side walls and bottoms for defects, and read dot codes and human readable text. Developing these new inspection stations should be straightforward for Gonschior, who already has experience with laser-based code readers. As Gonschior sells and installs more systems, he will develop remote software tools for central control, statistics, supervision, and set up.

For more information:
Thorsten Gonschior
Spectral Process
info@spectral-process.de
www.spectral-process.de

Top to bottom: A bottle that passes inspection; an over-pressed bottle opening; the binary image of the over-pressed bottle with the defect highlighted. All images courtesy of Spectral Process.
Get the big picture with registration
In vision applications, sometimes the scene requiring inspection is too large for the camera’s field of view (FOV). The scene can be acquired by moving a single camera or with multiple cameras. Either method results in several images—which can be assembled and “stitched” together to create a single image of the entire scene. This process is called “registration”, and is useful when:

- high resolution is needed
- physical constraints prevent the camera from being positioned so that it can capture the entire scene
- the panoramic view of a scene has to be generated

A global coordinate system is usually required for further inspection and analysis of a composed scene, but each image acquired has its own coordinate system. The registration process transforms the different coordinate systems into a single coordinate system. The calculated transformations can then be used to generate a “mosaic” of the scene. (Figure 1)

Images are always registered relative to another image’s coordinate system except one: the image whose coordinate system is set relative to the global coordinate system. Typically, the image registration and the resulting mosaic are performed this way:

1. Set the coordinate system of one of the images relative to the global coordinate system with a call to MregSetLocation(). The target parameter must be set to M_REGISTRATION_GLOBAL.
2. Set the approximate relative location of the other images with successive calls to MregSetLocation(). For each call, the target parameter specifies the reference image.
3. Set additional registration controls with MregControl() (optional). See the section “Camera position is key: good setups for registration”.
4. Call MregCalculate() to perform the registration.
5. Call MregTransformImage() to compose a mosaic.

Need for common distinctive features

- The images being registered must share common areas containing meaningful and non-ambiguous pixel data. In other words, the region where the images overlap must contain some unique grayscale information that can easily be used to align the two images correctly. The quality of the content is the key here; uniform grayscale or repetitive patterns must be avoided because they can yield multiple registrations. (Figure 2)
Camera position is key: good setups for registration

- Registration can be used in a setup where a single scene is imaged with a moving camera or with multiple cameras. In either case, for registration to work, the scene must be flat (or the camera must be positioned far enough away so that the scene appears to be flat like in aerial photography). Specify the transformation type with MregControl() with M_TRANSFORMATION_TYPE, but use caution:
  - select M_TRANSLATION only when camera movements are translations on a plane that is parallel to the scene (this setting offers the fastest and most robust performance)
  - select M_TRANSLATION_AND_ROTATION when the camera movements are translations and rotations on a plane parallel to the scene
  - select the more general M_PERSPECTIVE when other kinds of camera movements are also present (e.g. tilting, zooming)

- A more advanced setup allows the registration of scenes containing objects with 3D features by making use of a telecentric lens that eliminates the parallax effect. In this case, translations and rotations within a single plane are the only camera movements permitted, and the plane must be perpendicular to the optical axis of the camera. Set the M_TRANSFORMATION_TYPE to M_TRANSLATION if the camera does not rotate or M_TRANSLATION_ROTATION if it does. (Figure 3)

The not-so-good camera setup

- With a standard optical lens, uneven deformations (from the center of the FOV outwards) will appear due to the parallax effect. In the dual-camera system illustrated, region R1 occupies a larger portion of the field of view of camera A than of camera B. It will therefore appear larger for camera A than for camera B. Inversely, Region R2 will appear larger for camera B than for camera A. There are no global transformations than can correct these local deformations, so it is not possible to generate identical views in the overlap region. In this case registration can not be performed. (Figure 4)

Generating the mosaic

- Once the registration has been successfully performed, the image mosaic can be generated. Choose how the mosaic operation will handle the pixel intensities of the overlapping regions with MregControl() with M_MOSAIC_COMPOSITION:
  - use the values of either overlapped data
  - use the average value of the overlapped data
  - use a progressive blend of the overlapped data (new in MIL 9.0)

When the M_MOSAIC_COMPOSITION parameter is set, the mosaic can be performed by calling the MregTransformImage() MIL function.

- In a fixed camera setup, the registration needs to be performed only once. Since the relationship between each image does not change, the same registration result can be used by MregTransformImage() to generate the mosaic of subsequent series of images.

Take note:

- Set the initial approximate location as close as possible to the estimated perfect match position with MregSetLocation().
- Set the maximum allowable displacement to a value that is as small as possible with MregControl() with M_LOCATION_DELTA; larger search regions increase the processing time.
- Set the minimum overlap area between the two images with MregControl() with M_MIN_OVERLAP; the match will be optimized in the overlapping region only.
- If subpixel accuracy is not required for the application, specify an accuracy setting with MregControl(); set M_ACCURACY to M_LOW.