

Machine Vision Express™

enables computers to "see" like a human.



Three Development Platforms and a Vision Library

Introduction

Machine Vision Express™ is a development platform for a wide range of Machine Vision based solutions such as:

- Inspection
- Counting
- Gauging
- Positioning
- Identification
- Area surveillance
- Access control

For these and similar tasks Machine Vision is an economical solution, especially when accuracy, reliability, high speed, and consistency are essential.

Machine Vision Express™ may be used to create full custom solutions, ranging from simple to very complex. These full custom solutions do not suffer from the limitations imposed by many "canned" applications, which offer a limited set of variables to attack the same problem.

Solutions created with Machine Vision Express™ may be utilized on Personal Computers and smart cameras with "PC-Inside".

Machine Vision Solutions

A typical Machine Vision Solution can be broken down in distinct sections:

1. Image capture
2. Image enhancement
3. Image analyzes
4. Matching of test criteria with image data
5. Yes/No signal output for product routing

Section 1 requires the presence of a camera. Section 5 requires the presence of some type of digital I/O channels. When using a personal Computer, these are peripherals that need to be purchased separately, whereas in a smart camera with "PC-Inside" these peripherals are included.

Development platforms

Machine Vision Express™ includes three development platforms:

- Graphical User Interface
- Python scripting
- C++

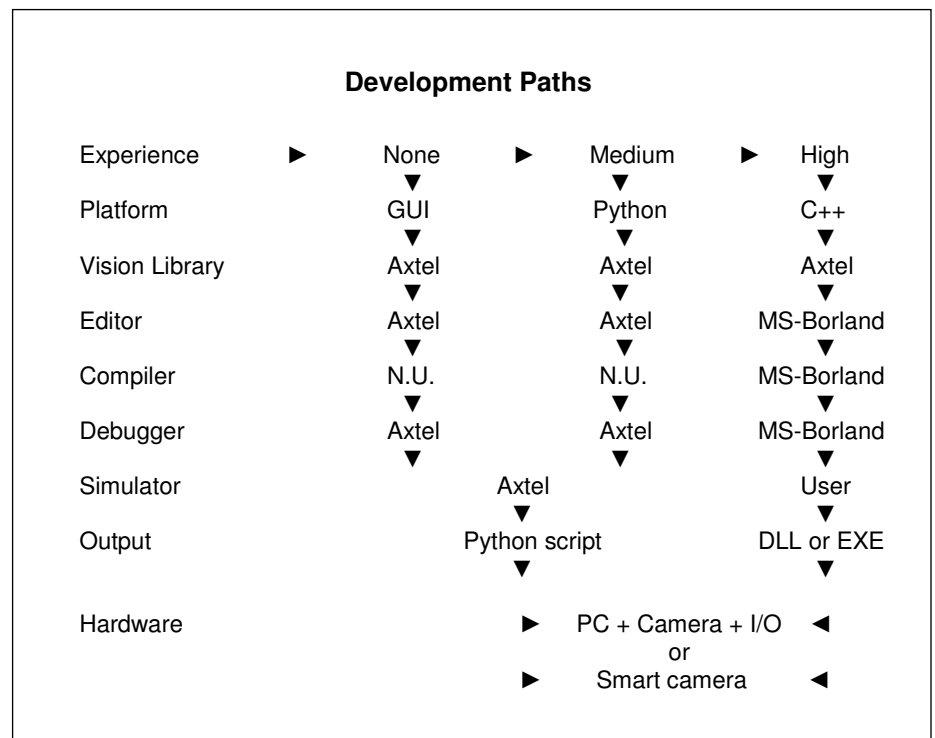
Based on personal preferences, experience and job complexity a client may choose the most suitable development platform.

Graphical User Interface (GUI)

The GUI-Studio does not require programming skills, but knowledge of the library based graphics functions is preferential.

A solution is created by adding a number of steps onto a canvas. Each step has a pull-down menu to assign a specific library function to it. The input and output cells of each step may then be interconnected through "drag and drop" of interconnection lines.

After such a visual script is completed, the engine automatically stores it in the form of a Python script, compliant with the syntax requirements of that language. This Python script may then be executed on a Personal Computer or smart camera with "PC-Inside".



Visual debug capability is built into the GUI. After a "Run" the results of each step are displayed, and images may be viewed before and after each processing step. Even the minutest differences between two images become visible when viewed in the image alternator.

Simulation mode allows for batch processing of canned test images. The engine builds a text DataBase,



to report pass / fail and other relevant details for each of the test images. The idea is to move R&D work from the field into the lab.

Python scripting

Use of the Python-Studio requires knowledge of the Python language and library based graphics functions.

A solution is created by entering a sequence of Python statements, which in their entirety form a Python script. To facilitate this task, the Python-Studio offers an Editor and a Debugger. Possible syntax errors are displayed and trouble-shooting is aided by viewing images at various stages of script execution.

After such a solution is completed the engine automatically stores it in the form of a Python script, compliant with the syntax requirements of that language. The script may then be executed on a Personal Computer or smart camera with "PC-Inside".

C++

The use of C++ requires language specific knowledge as well as knowledge of library based graphics functions.

Typically this development platform is reserved for the advanced programmer. A solution is created by using tools such as the MS-Development Studio, or Borland

compiler, for Edit, Debug, Compile and Link. When utilizing the C++ programming language, the visual aids built into the GUI can't be accessed and therefore the user must utilize his own resources for debugging.

The above mentioned compilers create .EXE or .DLL files. These files may then be executed on a Personal Computer or smart camera with "PC-Inside".

Speed

The main factors which determine maximum system performance are CPU speed and image import time. When a PC is used, then CPU speed and camera interface type are the main elements that determine performance.

When a smart camera with PC-Inside™ is used, then image import time is typically reduced to a minimum, but CPU speed may be less because of heat budget restrictions.

In Machine Vision Express™ compiled .EXE or .DLL code and Python scripts run at identical speeds, no measurable difference can be observed. This is because only a very small portion of CPU time is spent for script interpretation; most of the time is spent for executing library functions, which consist of fast and compiled code.

Library

The library included within Machine Vision Express™ contains several hundred basic graphics functions in addition to such advanced image processing features as:

- Pattern matching
- Optical Character Recognition (OCR)
- Barcode recognition

Run Time Module (RTM)

The Run Time Module is a separate program, its function is to control the environment and to handle all logistical functions.

An RTM allows users to choose between different scripts as some productions lines change articles from time to time. An additional function of an RTM is to control illumination, camera exposure, trigger selection, I/O port configuration and the setup of binning related functions.

