Matrox Imaging White Paper

Vision library or vision-specific IDE: Which is right for you?

Abstract

Commercial machine vision software is currently classified along two lines: the conventional vision library and the vision-specific integrated development environment (IDE). Determining which software is right for your vision project depends upon a variety of factors: ease-of-use, productivity, flexibility, performance, completeness, and maintenance. This white paper uses these factors to contrast the two software development approaches and clearly establish the merits and drawbacks of each. The discussion assumes that the vision tools available in both types of software are similar—if not identical—and does not explore possible discrepancies with these tools. Also, the discussion ignores the hardware platform that the vision applications run on as to not bias one over the other.

Matrox Imaging White Paper

Abstract

Commercial machine vision software is currently classified along two lines: the conventional vision library and the vision-specific integrated development environment (IDE). Determining which software is right for your vision project depends upon a variety of factors: ease-of-use, productivity, flexibility, performance, completeness, and maintenance. This white paper uses these factors to contrast the two software development approaches and clearly establish the merits and drawbacks of each. The discussion assumes that the vision tools available in both types of software are similar—if not identical—and does not explore possible discrepancies with these tools. Also, the discussion ignores the hardware platform that the vision applications run on as to not bias one over the other.

Ease-of-use

Developing an application using a vision library requires that you have knowledge of—some will even argue have expert knowledge of—and experience working with a traditional programming language like C/C++, C# or Visual Basic. It is also important for you to be very familiar with the associated development tools: code editor, compiler, linker and debugger. As many in the field would attest, however, acquiring and maintaining these skill sets can be elusive and costly. In contrast, working with a vision-specific IDE, requires a rudimentary knowledge of programming principles: flow control, variables and conditional/logical expressions. The required minimum skill set makes the vision-specific IDE accessible to a much broader technical audience.
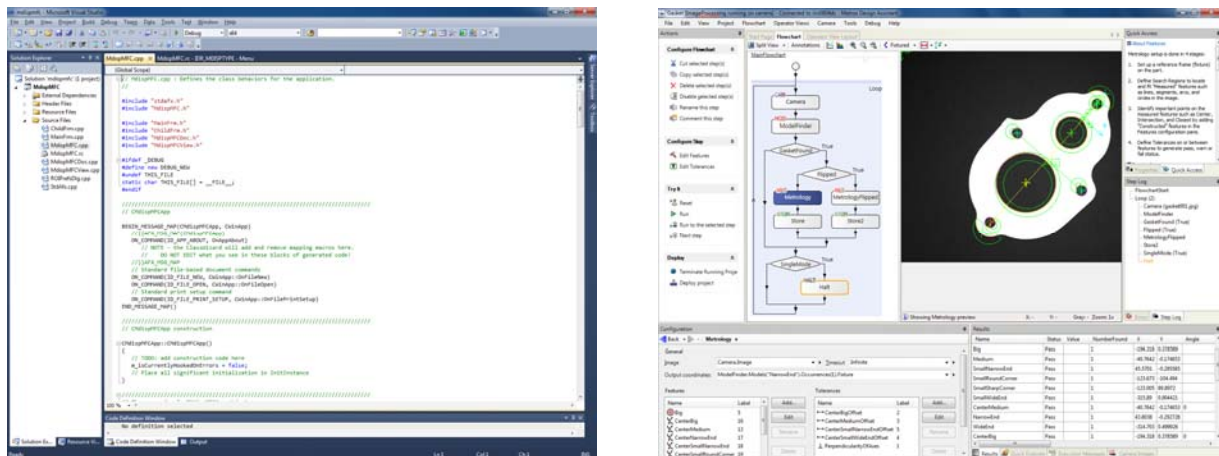

Figure 1 – Developing an application using a vision library by writing traditional program code (left). Creating an application using a vision-specific IDE by connecting and configuring operation blocks (right).

Vision library or vision-specific IDE: Which is right for you?

Matrox Imaging White Paper

Productivity

How quickly you become productive working with a vision library is highly dependent upon your knowledge of traditional programming and experience, as well as the quality and intuitiveness of the vision library's application programming interface (API) and its documentation. Making proper use of a vision library requires careful study of the supplied programming examples and documentation. And it is extremely beneficial for you to take advantage of the various training options offered by the software vendor before starting application development. You must also invest the time needed to properly design the initial application program architecture as this is essential for its effective reuse in subsequent projects. Working with a vision library generally results in an overall development time measured in weeks or months.

A vision-specific IDE is, unlike a vision library, designed to quickly tie together and configure the handful of operations needed for a typical vision application: get the next image, locate (an) object(s) or (a) feature(s) of interest, analyze/measure/read/decode, make a pass/fail decision, and communicate results. The simplicity of this approach makes starting a new project—even from scratch—straightforward. The automation of usual application requisites (i.e., fixing an analysis region based on result of a location operation) simplifies and thus accelerates project development. And, the modification of the application at a deployment site is less burdensome because of the all-inclusive nature of the software development environment. Working with a vision-specific IDE requires, on average, a development time frame measured in days or weeks.

Flexibility

A vision library provides you with the utmost flexibility to handle applications that require considerable and complex decision making, substantial use of custom vision or other algorithms (e.g., math and machine learning) alongside the ready-made vision tools, and the need to consolidate and work on multiple views from multiple cameras. To reiterate, as discussed in the previous section, a vision-specific IDE is best suited to applications that respect the intended usage model. Deviating from the intended usage model can be awkward and messy. Furthermore, the addition of custom vision or other routines basically requires traditional programming.

Vision library or vision-specific IDE: Which is right for you?

P a g e | 3

Matrox Imaging White Paper

Performance

A vision library invariably offers the best performance because it operates at a level closest to the hardware. In fact, a vision-specific IDE itself makes use of a vision library of some form or another. Working with a library also provides more opportunities for performance tuning, including manual task parallelization and offloading, and permits the most effective use of memory and the reuse of computing resources. A vision-specific IDE has an inherent performance overhead but the magnitude of this depends upon the quality of the implementation. And, typically, memory usage is not the most optimal because of the IDE's need to maintain flexibility.

Completeness

When you decide to use a vision library, the implementation of other application functions (i.e., operator interface and communication with automation and enterprise equipment) requires additional programming that is either custom or based on third-party libraries. With a vision-specific IDE, the set-up of the usual ancillary functionality (i.e., operator interface and external communication) is a key characteristic of the IDE. However, advanced vision features are purposely hidden away or not exposed to ensure simplicity and thus ease-of-use.
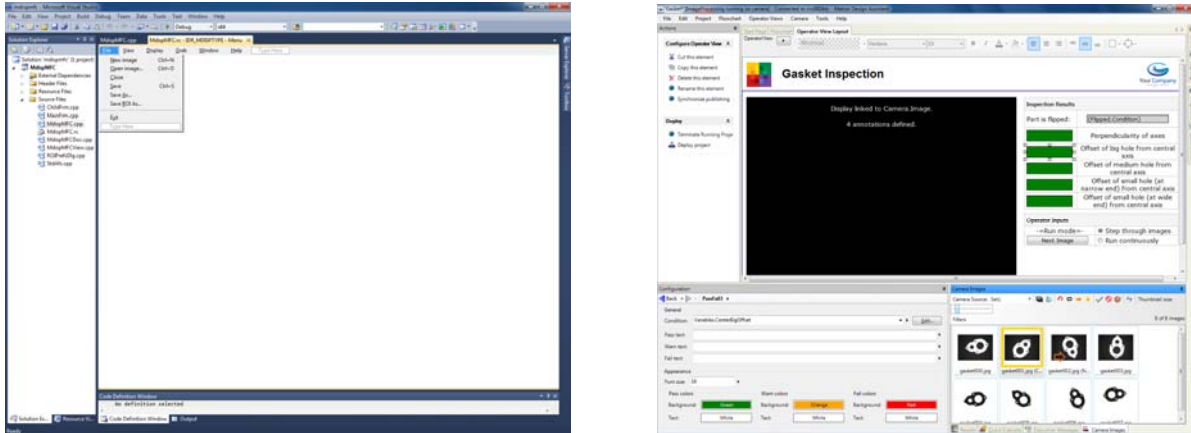

Figure 2 – Creating an operator interface when using a vision library requires the use of separate tools and additional programming (left). A vision-specific IDE integrates the facility to create an operator interface starting from a ready-made template.

Vision library or vision-specific IDE: Which is right for you?

P a g e | 4

Matrox Imaging White Paper

Maintenance

Once an application developed using a vision library is deployed, any subsequent effort needed to revise or adapt it can be substantial depending on its complexity as well as the quality of its implementation and documentation. What's more, transferring this responsibility to another programmer can be a lengthy and difficult process. This is unlike a project developed using a vision-specific IDE, which is easier to transfer or share.

The verdict

Choosing between a vision library, like the Matrox Imaging Library (MIL), or a vision-specific IDE, like Matrox Design Assistant, depends on circumstances and application objectives. If you are willing and able to invest in obtaining and retaining traditional programming know-how, and need your machine vision system to deliver unprecedented levels of performance and functionality, you will not go wrong using a vision library. A typical vision library user is an original equipment manufacturer (OEM) that embeds machine vision into an overall machine to be sold in significant quantities over many years. If instead, you need to move from one machine vision project to another quickly and often, while delivering existing levels of performance and capability, then a vision-specific IDE is best-suited to your needs. Users of vision-specific IDEs are often system integrators with multidisciplinary technical staff bidding on one-off installations or projects that have a modest number of duplicate installations. Some commercial machine vision software vendors, like Matrox Imaging, understand these diverging needs and offer products that cater to both user types.

To learn more about the Matrox Imaging Library (MIL), visit
http://www.matrox.com/imaging/en/products/software/mil/.

To learn more about Matrox Design Assistant, visit
http://www.matrox.com/imaging/en/products/smart_cameras/iris_gt/development/.

Vision library or vision-specific IDE: Which is right for you?

P a g e | 5