

# Choosing a frame grabber for performance & profitability

Gail Marshall  
Imagination Corporation

# Choosing a frame grabber for performance & profitability

The purpose of this paper is to help you specify a vision system by providing an introduction to video frame grabbers and their specifications. As you'd expect, the demands of your imaging application will determine the emphasis you place on the various aspects of frame grabber operation. Here, we'll focus on choosing frame grabbers in industrial machine vision applications. Specifically, we'll try to answer these two questions:

- What features should you look for when choosing a frame grabber?
- How can you use published specifications when evaluating frame grabbers?

When you develop low-volume products, such as custom applications, the software engineering investment far outweighs the hardware cost. For such applications, there is little consequence to purchasing a frame grabber with extra features that might not be used in the final product design.

When you develop a high-volume product, total engineering costs are spread over the number of units shipped, lessening their impact on per-unit profits. Savings in hardware costs on each unit go directly to your company's profits. The frame grabber is often the most expensive single component in vision based products. So, making the appropriate choice requires finding a frame grabber that meets the product's technical requirements at the minimum cost. Often this choice is left to the project engineer due to the technical issues involved. However, the major impact of this decision on product profitability means that understanding this technology can be very important to managers as well.

Initial uncertainty about system requirements may lead to the choice of a frame grabber with extra capabilities—just in case they're needed. The cost of such insurance can be very high, because employing a frame grabber usually requires custom software development. By the time you realize you don't need all those extra features, the initial frame grabber choice is often irrevocable due to the high cost of software re-engineering and time-to-market considerations.

The solution is early product definition and the participation of product managers in the frame grabber selection. We hope that this article will help both engineers and managers make better product choices. In the following pages, we'll take a look at these general categories of features and associated specifications:

- Color and gray-scale
- Digitizing accuracy
- Video signal handling
- System integration
- Reliability

# Color vs. Gray-Scale

Why are gray-scale (black and white) frame grabbers still used when color looks so much better? Color frame grabbers are often used in multimedia applications, where the goal is to produce images that look good to the human eye. To meet this goal, multimedia frame grabbers automatically make adjustments to the image to improve appearance. These image adjustments result in good-looking images with the least amount of post-processing—just the ticket when publishing a document.

For industrial or scientific applications, the goal is to produce an accurate, digitized version of the original video signal. For these applications, having the frame grabber make automatic adjustments can be disastrous.

The disadvantage of color is the larger image size and resulting longer image processing time. Color consists of red, green, and blue components, each requiring a separate image. The larger color images require more time to process and more space for storage. While a few applications, such as sorting fruit, might require color, gray-scale images are preferred for most industrial applications because gray-scale contains all the required image features in a more compact form.

## Digitizing Accuracy

Digitizing accuracy affects the fidelity of the features in the digitized image, which determines whether you'll be able to make accurate and repeatable measurements of those features. Most manufacturers publish two specifications that you can use to compare products: pixel jitter and gray-scale noise. They are reviewed in this section, along with square pixels, progressive scan, cropping, scaling and on-board video memory.

### Pixel Jitter

Did you know that you could measure distance with a frame grabber to a precision of about 1/10th of a pixel? Even though an image has a horizontal resolution of 640 pixels, each pixel has brightness information that can be used to increase the precision of measurements. This is because an object's edge is represented by a brightness difference between an object and its background. If a pixel straddles an edge, its brightness is a combination of foreground and background. Simple interpolation can be used to locate an edge to within a fraction of a pixel. The downside of this measurement technique is that fractional errors in pixel position will distort distance measurements. The measure of this error is called pixel jitter.

A frame grabber must sample the analog video signal at uniform intervals on each video line to measure the gray-scale level of each pixel. Video signals have an embedded timing synchronization pulse at the beginning of each scan line. This is called the horizontal sync pulse. Pixel jitter is a measure of how precisely a frame grabber can adjust the sampling points relative to the horizontal sync pulse, even when the sync pulse varies in time.

Pixels are sampled about every 80 nanoseconds (ns) in an image with 640 pixels per line. A good frame grabber has a jitter spec of less than  $\pm 5$  ns, which represents a pixel position error of 12.5 percent. Near an edge, the brightness of an image can change from zero to maximum in about one pixel period. So, when measuring a video waveform with such high slopes, a position error of 12 percent will translate to a similar 12 percent error in brightness measurement.

# Gray-Scale Noise

The process of digitizing an image involves amplifying and measuring the brightness (gray-scale value) of an analog video signal. Like a good stereo system, a frame grabber should handle an analog signal without introducing noise that would degrade the image. Any gray-scale noise will introduce errors in distance measurements, just like pixel jitter.

In practice, there is always a certain amount of noise, or random variation, in the frame grabber circuits, so that you don't actually get the same result every time. If you measured the gray-scale value many times for a given input voltage, you'd end up with a distribution of gray-scale values that would be roughly a bell-shaped curve, centered at the correct gray-scale value and dropping off rapidly to either side. The spread of the results is a measure of the gray-scale noise and is usually specified as the standard deviation of the distribution of sampled values.

A precision frame grabber will typically have a gray-scale noise spec of 0.7 gray-scale units or less. You might see this spec written as 0.7 LSB. The LSB stands for *least significant bit* and refers to a binary number representing the gray-scale value. For example, an 8-bit digitizer represents each gray-scale value as an 8-bit binary number, giving a range of 0-255. This notation comes from the fact that a change in the LSB of  $\pm 1$  represents a change of  $\pm 1$  gray-scale unit.

Gray-scale noise can be a problem when you're dealing with low-contrast images. Trying to differentiate between adjacent areas with similar gray-scale values is obviously a problem when noise is distorting the gray-scale values. Analyzing medical images or inspecting surfaces for irregularities are examples of applications where gray-scale noise can literally make or break the application. Gray-scale noise can also cause problems in edge detection applications.

# Square Pixels

The image geometry should be such that a number of pixels represents the same distance in the image horizontally and vertically. This is called square pixels. A video image is not square; it is wider than it is high by the ratio of 4 to 3, but that is just the image shape originally chosen for video screens. Regardless of the image shape, the pixels should be like perfectly square tiles representing the same distance horizontally as vertically.

Generally, a frame grabber that has a specified resolution of 640 by 480 pixels (or 768 by 576 pixels in Europe) will have square pixels. However, this assumes the frame grabber samples 640 pixels and exactly covers the whole horizontal video line. A faster sampling rate would still produce 640 pixels, but over a shorter horizontal distance, and not produce square pixels. The sampling rate is key. Make sure the frame grabber specifications expressly state that it generates square pixels.

# Progressive Scan

Progressive scan is an improved, non-standard video format in which all the pixels in an image are exposed at the same time and then read out of the image from top to bottom, line by line in numerical order. Regular television, and therefore almost all video cameras, use a format called interlaced video in which the even image rows are exposed and sent out of the camera, then the odd rows are exposed and read out. This usually has no consequence except when an object in the image is moving. A moving object produces two exposures at two different locations. Unlike a film camera, an interlaced video camera's image doesn't represent a single

snapshot in time. Progressive scan cameras overcome this problem but are quite expensive. Fortunately, recent developments in the technology are driving progressive scan camera prices down quickly.

Few reasonably priced frame grabbers can handle progressive scan, but the wider availability of these cameras is changing this. It is likely that in the near future almost all industrial and scientific imaging will be progressive scan. By the time you read this, you may find that the extra performance of progressive scan cameras and frame grabbers are quite worth the price.

## Cropping and Scaling

When an object in the image doesn't occupy the whole image, it makes sense to read in only the region of interest (ROI). The amount of data in an image is quite large. System performance can be improved by reading only a portion of the image. Frame grabbers can be set up to transfer to the computer's memory only the ROI defined by software. This is called cropping, and it simply discards unwanted data outside the ROI.

Scaling is usually used for display purposes. It is the modification of the image so it will fit within a region on the video display. For example, if the whole image is to fit within an area of 1/4th of the display screen, it must be compressed horizontally and vertically by a factor of two. In this example, the best way to do this is to average two adjacent pixels to compute the new displayed pixel. This form of scaling preserves some information from all pixels in the original image, although fine detail will be lost. Another form of scaling, called decimation, simply discards pixels (every other one in the previous example) to produce a smaller image. Without the averaging used in true scaling, decimation loses fine image details which were in the discarded pixels, producing a less natural image.

## On-board Video Memory

The advent of the fast PCI bus has allowed data to be transferred into the computer's memory at rates faster than video data rates (12 to 15 megabytes/second). In fact, the burst data rate of the PCI bus is 132 Mbytes/sec! The truth is that most computers have an average data rate of only about 50 to 90 Mbytes/sec. This is because the PCI bus must share its access to the computer's RAM with the CPU. Also, the computer's RAM itself isn't designed to support the full PCI burst rate. Many frame grabber manufacturers have economized by deleting on-board video memory from their designs. Video data always comes from the camera at a steady rate. If the PCI bus can sustain an average data rate equal to the video data rate, things should be OK. But, PCI bus cards don't get continuous access to the bus; instead, they take turns sending data in bursts. Between each burst the frame grabber must store the unstoppable flow of pixels. Therefore, all PCI bus frame grabbers have some form of memory, usually enough to store one horizontal video line. On a crowded PCI bus, such as one with multiple frame grabbers, it is very possible to clog the PCI bus long enough to overrun the frame grabber's temporary storage. When this occurs, the image will be corrupted.

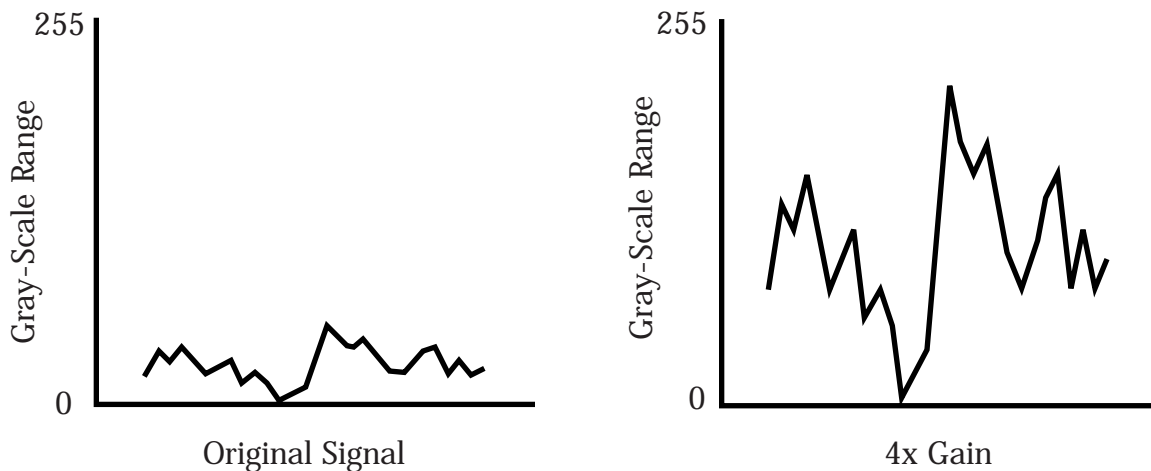
The only solution is to use image storage on the frame grabber. This ensures that the image will be received eventually, but this situation means you will not be sustaining real time video transfers of 30 frames per second. Even though this situation only occurs in a minority of applications, it is nice to have an option to install image RAM on the frame grabber.

# Video Signal Handling

Differences in the incoming video signal can complicate the job of a frame grabber. Poor lighting conditions for the camera or losses due to long cables can produce a signal that uses only a narrow band of the full amplitude range. Irregular sync signals in the incoming video can also cause problems by making it harder for the frame grabber to accurately position the image. Most cameras also have some gray-scale non-linearity (called gamma) which you might need to correct. Well-designed frame grabbers can compensate for many of these problems, so they can be used in a wide range of applications. In this section, we'll look at gain and offset adjustments, sync timing, and lookup tables for dealing with some of these signal problems. We'll also look at the need to handle different video standards for products designed to be sold worldwide.

## Gain

Sometimes the amplitude of a dark video signal uses only a fraction of the total signal range. Consider the original signal shown on the left in Figure 1.



**Figure 1. Using Gain to Adjust a Signal**

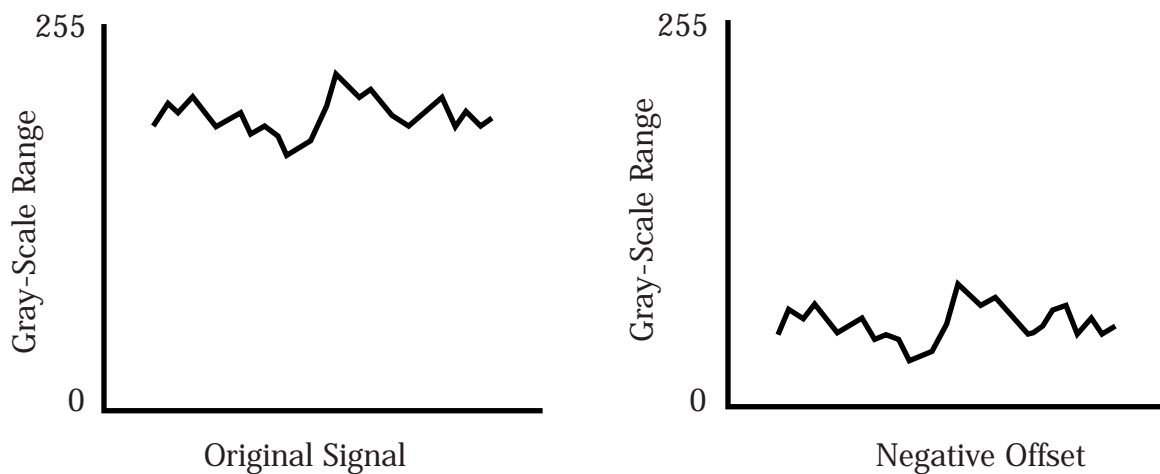
When the frame grabber digitizes this signal, you'll get a narrow range of gray-scale values near the bottom end of the scale. Two pixels could easily end up being assigned identical gray-scale values, even though they don't have the same brightness. If you could expand the amplitude of this signal, you could take advantage of more of the gray-scale range, and you'd be able to distinguish between more features in the image. This is where having a gain control is useful. The signal on the right is the same signal with a gain of four (400%). Each amplitude value in the original signal has been multiplied by four. The signal now uses almost the entire gray-scale range. Gain values below one are also useful for bringing down a signal that is too bright.

The obvious solution to poor signal amplitude is to try to correct it at the source; for example, by adjusting illumination. Unfortunately, this isn't always possible. In passive infrared imaging applications, for example, you're dependent on the infrared energy emitted by the object. In applications that depend on natural lighting, the intensity of the light won't be under your control and will even vary throughout the day. Some objects are also inherently low-contrast, such as the surface of an integrated circuit.

Since you can't always control the quality of your video source, a gain control on your frame grabber makes it much more versatile.

## Offset

Offset is useful for solving a related problem with incoming video signals. The signal on the left in Figure 2 makes no use of the lower portion of its measurement range. Applying gain to this signal, by itself, won't solve the problem—you'll just end up pushing the high-intensity portions of the signal above the gray-scale range.



**Figure 2. Using Offset to Adjust a Signal**

The solution is to use a combination of offset and gain. The signal on the right in Figure 2 shows the same signal with a negative offset applied. With the offset applied, you can now apply gain, as in Figure 1, to get the desired result.

Offset control in a frame grabber is particularly important if the camera you're using doesn't let you compensate for low or high lighting conditions. Less expensive cameras don't always provide good gain control, but might be the right choice for saving overall system cost. Also, in situations where you must adjust the gain dynamically for changing conditions, it might be easier to make the adjustments in the frame grabber rather than adding programmable controls for the camera.

For best results, look for a frame grabber that will let you offset the video signal by  $\pm 100\%$ , in small, precise increments.

# Sync Timing

Video contains both horizontal and vertical sync signals. Vertical sync occurs at the beginning of each new field (there are two fields in an image). A horizontal sync pulse marks the beginning of each horizontal scan line in an image. The frame grabber must remain synchronized with the incoming signal or the digitized image will be distorted. Unfortunately, this isn't as easy as it might seem. In some applications, you can't depend on the horizontal and vertical sync pulses having regular intervals. VCRs, for example, often produce irregular video signals.

An even worse problem occurs in applications that use resettable cameras. These cameras, which are common in industrial applications, can be reset by an external signal to expose the image precisely when a moving object comes into position. This is typical of applications for inspecting parts on conveyor belts. The camera can be reset at any time, including right in the middle of an image. The frame grabber should be able to abandon what it is doing and resynchronize to the new video stream without error. Failure to faithfully grab every new image can result in false rejects at an inspection station.

Ideally, the frame grabber would detect each horizontal sync pulse and instantly re-synchronize its pixel timing. Frame grabbers that use older style phase-locked loop (PLL) pixel clock generators can't re-synchronize instantly. To minimize pixel jitter, a PLL circuit is designed to resist changes in timing. A frame grabber with a PLL circuit might take up to a field time (half the image) to re-synchronize. To make matters worse, the better a PLL circuit is at minimizing pixel jitter, the worse it is at re-synchronizing; these two design objectives are simply in conflict for PLL circuits.

For most industrial applications, you'll want a frame grabber with a crystal-controlled digital clock that can re-synchronize instantly and, once synchronized, will be extremely stable in generating pixel intervals. If your application requires a resettable camera, look for a frame grabber that can re-synchronize to the first field after reset.

## Lookup Tables (LUTs)

After a pixel is digitized, it is represented by a numerical value for the brightness of the pixel's location in the image. A Lookup Table (LUT) uses the pixel's value as an index into a table to locate a new value that is assigned to the pixel before it is stored. For example, suppose you have an image that consists of lighter objects in a dark background. Software would have an easier job locating the objects if all the background had a numerical value of zero—then locating an object simply requires searching for non-zero pixels. For example, this can be done by loading the table with zeros in the lowest 30 locations. Then any dark pixel with a value less than 30 is translated to zero, and the darker regions in the image become completely black. This technique is often referred to as thresholding.

To disable the LUT, you can make it "transparent" by loading it with values from zero on up in numerical order (0,1,2,...). To produce a negative image, you just reverse the numerical order (255,254,253,...). Of course, you could do these gray-scale translations in software after capturing an image, but it's much faster to do them in hardware. In addition to thresholding, lookup tables are useful for gamma correction (correcting for gray-scale non-linearity in cameras) and other gray-scale translations.

Frame grabbers with built-in displays can also have an output LUT. The output LUT is used to make adjustments to an image's gray-scale or color as the data goes to the display, without changing the original pixel data. On gray-scale frame grabbers, you can use a color output LUT to display false-color images, mapping gray-scale values to different colors. False-color images are used in satellite images and thermal imaging, where differences in color are much easier to see than differences in gray-scale.



# International Video Standards

If you're designing a product to be sold internationally, you'll want to choose a frame grabber that supports video standards for many countries. The video standard used in North America is known as NTSC, while most of Europe uses a standard known as PAL and France uses SECAM. NTSC video has a frame rate of 30 frames per second (fps), while SECAM/PAL use 25 fps. For black and white images, the two European standards are identical; they differ only in how they handle color.

# System Integration

Another important set of features is associated with integrating the frame grabber into an overall system. These features include bus options, on-board memory, video output support, digital I/O and application development support. We'll look at these features below.

## Bus Design

Several different bus designs are used on the factory floor, including:

**ISA**—The Industry Standard Architecture (ISA) bus is essentially the original IBM PC AT bus introduced in 1984.

**PCI**—Most PC-compatible computers based on the Intel® Pentium® processor use the new Peripheral Component Interconnect (PCI) bus for high-speed devices, such as display controllers. The 32-bit PCI bus can achieve burst transfer rates of 132 megabytes per second. Currently most computers that use the PCI bus still include an ISA bus for additional boards that don't require such high performance. The PCI bus is so fast that video data can be sent directly to system RAM instead of being stored on the frame grabber.

**CompactPCI**—This is a system which is physically identical to a VME computer; however, the VME back plane has been replaced by the PCI bus. The VME, and now the *CompactPCI*, are used extensively in industrial systems. CPU cards used in *CompactPCI* computers are generally Pentiums.

**PC/104**—In addition to off-the-shelf systems, the basic PC architecture is used in many embedded PC designs where the computer is built into a larger system. The PC/104 bus was designed for embedded system applications, and although it is physically much smaller and more rugged, it conforms to the ISA bus standard.

**PC/104-Plus**—Take the PCI bus and add it to the PC/104 system, and you have PC/104-Plus. This dual bus architecture is similar to desktop computers, although some PC/104-Plus cards may not include an ISA bus. PC/104-Plus is ideal for applications requiring speed in a small package at a reasonable price.

**STD**—The STD bus is almost exclusively for industrial systems. Physically it is an open-frame, convection-cooled design that is extremely rugged. Today's STD-bus computers are software compatible with ISA-bus computers. One advantage of STD bus is its support for multiple, concurrently operating CPU cards, each equivalent to a motherboard. Recently, STD-based computers have incorporated the PCI bus, just like the ISA/PCI computers.

As you can tell from the above descriptions, one of your most basic decisions is whether to use an off-the-shelf PC or an embedded system. The selection of frame grabbers for the PC/104 and STD buses is limited but growing. But the choice in industrial systems will probably be dictated by other design considerations.

# Video Output Support

For most applications you need to display real-time video, even if only for setup and calibration of the system. Many frame grabbers use a separate interlaced monitor to display video. This is particularly true of frame grabbers designed with lower-performance buses, where the limited bus speed doesn't support real-time capture and display. Since the computer typically already has a VGA monitor, the requirement for a separate monitor means added expense for the system.

With PCI-based frame grabbers, you can often eliminate the need for a separate monitor. You can use the same VGA monitor you're using for your application interface to view real-time video. Support for this single-monitor solution is another cost-saving feature you should consider when choosing a frame grabber. On the other hand, if there is no need for a VGA monitor, the price of an interlaced, black and white monitor is about one-third that of a color VGA monitor.

## Graphics Overlay

When using frame grabbers with a video display output, graphics overlay capability lets you add text and graphics to the displayed image without changing the image data itself. You can add text annotations or alignment marks to an image to assist the system operator in performing tasks. The frame-grabbed image is merged during display with graphics written into an overlay memory. The captured image data is not affected because the overlay data is stored in a separate graphics memory in the frame grabber.

## Digital I/O

In industrial applications it is often necessary to coordinate the timing of an image capture to an industrial process. In applications using a resettable camera, the computer often generates the camera reset pulse. Also, some cameras can use a computer-generated pulse width to set their exposure time. In other applications, the electrical pulse that is sent to the camera to expose a new image can also be sent to the frame grabber to cause an image capture. If your frame grabber can handle these signals itself, you won't need to buy a separate digital I/O card, saving overall system cost.

A minimal digital I/O capability for a frame grabber might be a single output, often called the strobe, and a single digital input, sometimes called the trigger. Other frame grabbers may have eight or more general purpose digital I/Os to control other industrial devices.

Using software to control the width of an exposure pulse is imprecise, particularly in Windows-based systems which can interrupt the software during time-critical routines. Camera control is very convenient when the frame grabber is capable of generating a programmable width pulse.

Another useful digital output that the frame grabber should supply are horizontal and vertical drive outputs. These are used to synchronize some cameras in a process known as "genlocking." When multiple cameras are used with a single frame grabber, they are often genlocked so the frame grabber doesn't encounter different video timing when it switches between cameras. Many cameras have H & V drive inputs for this purpose. Unfortunately, the nature of these signals is not standardized. Some are simply TTL inputs, while others are 75 ohm inputs like the video signal. Also, the pulse polarity may be positive or negative. It is very convenient to have a frame grabber that can be programmed to generate these various camera synchronizing drive signal formats. Some cameras can be reset to expose the image precisely when an object is in position.

With other cameras, this can be accomplished by firing a strobe light to time the exposure. Unfortunately, there is a brief period during vertical blank in each video field when the camera is not amenable to exposure. During this blind period, which is called the transfer gate, the image in the camera's light sensitive elements is being unloaded. One service a frame grabber can supply is to generate a delayed trigger. If the object-in-position pulse arrives during the transfer gate, the frame grabber can delay it until the blind period has passed. As a result, the camera will always capture an image without failure. For example, the in-position pulse can be sent through the frame grabber's trigger input to one of its digital I/Os, except during a software selected period representing the transfer gate.

## Power Output

Some frame grabbers offer a 12 volt output for supplying power to a camera. The frame grabber takes the power from the computer's bus. This power output can save you the cost of a separate power supply, lowering the overall system cost. Make sure such power sources are fused in the frame grabber, ideally with a self-resetting fuse.

## Application Development Support

When it comes to time-to-market, the application development support available for the frame grabber might be the most important aspect of the entire decision. You'll want to consider several points in application development support:

- Support for high-level programming languages (C/C++ , Pascal, FORTRAN) and operating systems (DOS, Windows 3.x, Windows 95, Windows NT)
- Support for prototyping or rapid application development (Visual Basic, Delphi)
- Support for third-party libraries for image processing or other specialized applications
- Source code examples
- Concise, complete documentation covering the frame grabber hardware and software
- Knowledgeable technical support
- Software drivers which can write an image into the VGA display under Windows, ideally at a real time rate and with arbitrary image scaling.

The key is to make sure the frame grabber comes with everything you need to successfully develop your system. Make sure libraries are available for the programming languages and operating systems you plan to work with. In some cases, you can mix languages; for example, calling Windows DLL routines developed in C from an application program written in Visual Basic.

If your application involves image processing or other specialized calculations, you might be able to take advantage of software supplied by a third-party. If so, it's good to know that the vendors supplying the frame grabber software and the third-party software have worked together to make sure their products are compatible. Major vendors of image processing packages include Optimas, XCaliper, Image Pro, and Sentinel.

No matter which language, operating system, or third-party software you'll be using, you'll find it much easier to start your development with some good source code examples. Modifying and patching together vendor-supplied routines is the quickest and surest way to get your software development started. Vendor-supplied examples are written to use the frame grabber or other library in the most efficient and effective way, and the routines come already tested and debugged.

Good documentation is indispensable. The documentation should be complete, but concise. It should be well-organized, so you can quickly locate the information you need, when you need it. The documentation should include both information about the hardware features and detailed reference information for the software library functions and other supplied software.

Just as important as the software you get from the vendor is the support that backs it up. When you get stuck, think you've encountered a bug, or just need some advice, it's nice to know that help is just a phone call away. Can you quickly get in touch with someone really knowledgeable about the frame grabber and software? And, what if you're working late or on a weekend to meet a deadline—do they have a bulletin board or World Wide Web site you can check for recent software updates, application notes, lists of frequently-asked questions, and other materials that might help you solve problems?

Application development support is absolutely crucial to your success. No matter how many wonderful features and specifications the frame grabber has on paper, if you can't get it to work in your system, and do so on schedule, it's not the right choice.

## Reliability

Hardware reliability is important in any production system. Shutting down a production line due to an instrument failure can quickly cost you more than the instrument that caused the shutdown. Unfortunately, most vendors don't publish reliability specifications, such as mean time between failure (MTBF).

There are a couple of rule-of-thumb techniques you can use for estimating relative reliability between different boards: look at the parts count and the power consumption. Heat is a real enemy of electronic components. Try to specify a frame grabber that uses as little power as possible. All other things being equal, a more complicated board with many parts will generate more heat than a board with fewer parts. Good board designs use application-specific integrated circuits (ASICs) and programmable devices to provide high functionality with low parts count. You can also improve reliability by not purchasing a board with lots of features you don't need, avoiding unnecessary complexity.

Over-voltage protection is an important feature for reliability. Nearby lightning can induce very large surges in a video cable. Over-voltage protection on video inputs and outputs and digital I/O can protect the frame grabber circuits from voltage spikes generated by noisy industrial environments. An industrial installation should be designed to operate for years.

## Summary

Like buying any piece of equipment, buying a frame grabber is a matter of understanding your needs and evaluating the ability of various products to meet those needs. We hope this paper answers some of your questions about evaluating frame grabber designs and helps you find the right product for your application. The following check list can help you evaluate potential products.

# Checklist for Choosing a Frame Grabber

- \_\_\_ Low pixel jitter for precision digitizing accuracy:  $\pm 5$  ns or better.
- \_\_\_ Low gray-scale noise to eliminate false edges and handle low-contrast images: 0.7 LSB or less.
- \_\_\_ Gain and offset ( $\pm 100\%$ ) controls to adjust for amplitude problems in the incoming signal.
- \_\_\_ Stable sync timing with the ability to re-synchronize to the first field of incoming video for working with resettable cameras and other variable signal sources: look for a crystal-controlled digital pixel clock.
- \_\_\_ Input lookup tables for pixel gray-scale translations for gamma correction or thresholding, and output lookup tables for false color.
- \_\_\_ For international applications: support for both NTSC, PAL, and SECAM video signals.
- \_\_\_ The right bus design for your application: STD or PC104 bus for embedded system designs; PCI bus for high-performance.
- \_\_\_ On-board memory for data transfer-intensive applications and applications where a high-performance bus isn't available.
- \_\_\_ Single-monitor solution to minimize system cost.
- \_\_\_ Graphics overlay for adding annotations or alignment marks to an image.
- \_\_\_ Digital I/O for communicating with cameras and other devices: trigger, strobe, and any other required signals.
- \_\_\_ Power output for supplying a camera or other device without adding a separate power supply to the system.
- \_\_\_ Comprehensive software support: high-level programming languages, choice of operating systems, rapid application development tools, source code examples, and support for third-party libraries for image processing.
- \_\_\_ Complete, concise documentation on the hardware and software.
- \_\_\_ Responsive technical support via telephone, BBS or World Wide Web, and email.
- \_\_\_ Reliability: low parts count, low power consumption and a strong reputation in the industry.