



Integrator Column Vision and Sensors Magazine

Date: 2-5-2010

Author: Ned Lecky

Lecky Integration

ned@lecky.com

Work/Cell: 518-258-5874

GigE Camera Myths Debunked

A year and a half ago I penned an article in this publication touting the merits of Gigabit Ethernet (GigE) cameras for machine vision purposes. The low cost and raw data bandwidth of GigE was no news, but the applicability of the technology to real, high-value, mission-critical machine vision applications was not at all clear back then.

After eighteen months of experimentation and deployment, I'm thrilled to report that GigE cameras live up to all of my wildest expectations for use in machine vision. I don't want to use anything else ever again. The cameras are easy to setup, very full-featured, lower and lower cost (\$450 for a machine vision grade VGA camera), and easy to interface using supplied drivers and plug-ins available for all cameras and major machine vision applications.

But there are still some fears and concerns that persist in the vision community. I'll use this column to describe and debunk the more prevalent ones.

Network Congestion

The general networking rule of thumb is to limit total traffic to 80% of the link speed. As long as you stay there or well below there, system limitations will not be dictated by GigE- they will be dictated by camera speed, OS overhead, or CPU power.

As an example, take a 640x480 8b mono camera running at 60 frames per second (fps). This data stream requires $640 \times 480 \times 8 \times 60 = 147\text{Mbps}$, just 15% of a GigE link. You can see why handling four of these cameras per link is typically not much of an issue.

Higher-performance example? A 2k 3-sensor color linescan camera, like Basler's Runner, running at 5k lines/sec requires $2048 \times 3 \times 8 \times 5000 = 246\text{Mbps}$, or a quarter of a GigE link. We regularly run these in pairs through a switch on a single GigE port and have never experienced an imaging issue.

Lost Packets

It is possible to configure GigE cameras in ways that will guarantee network congestion and, therefore, flaky behavior such as lost packets or an overburdened operating system. But it is much harder than you think. In one round of testing, we set up a network with four 640x480 mono cameras running at 60 frames per second either using triggers or just free running. This is a typical high-performance machine vision setup- 240 images per second coming from four cameras continuously. Whether we routed these cameras through a switch to a single GigE port or separately connected them directly to individual GigE ports, we were incapable of inducing any lost images or even lost image packets. Ethernet is a very reliable data transfer technology and has much more R&D investment and real-world runtime than any other camera interface technology ever used in history. Really. Think about it.



Triggering and Strobes

Machine vision needs external triggers and strobes. External triggers are signals that come from sensors or other electronic systems which tell the system when to take the picture. Strobes signals are generated at the appropriate time to fire a very short light pulse (strobe) to freeze motion much more consistently than a simple camera shutter can.

In the GigE model, the trigger signal is routed directly to the camera so that there is minimal latency from the trigger generation to the image acquisition. Similarly, the strobe signals are generated directly from the cameras to insure that their timing is precisely correct for the image being acquired. After acquisition, the camera digitizes the image and transfers it at 1Gbps to the analysis computer. Multiple camera asynchronous applications, always the bane of machine vision existence, are actually no harder than anything else with GigE.

Operating System Woes

My company is an integrator working in real-time controls, embedded systems, Linux, and custom OSs. Many customers still specify Windows, though, due to other considerations in the integration process.

We've developed high-speed area- and linescan applications using GigE cameras in both Linux and Windows, and have had excellent results using both. While the Linux system has a higher theoretical throughput, we can still run four-camera 60 fps applications on Core 2 Duo processor running Windows XP and accomplish basic image processing and metrology without losing images or packets. So while Windows may not be ideal, it is certainly a serviceable platform for many vision applications and offers special advantages in many cases.

Power

Today, most GigE cameras still require a separate power cable to supply 12V or 24V to power the camera. The Power-Over-Ethernet (PoE) standard, however, is becoming ubiquitous and most camera vendors are planning their new camera lines to use PoE. The new Ace line from Basler, for example, is a complete machine vision camera line that boasts PoE capability and starts at list price of \$450. I expect other vendors to offer similar camera lines. With PoE, we can power the camera and communicate with it using 100 foot cables that cost in the tens of dollars. This is a huge step forward.

Support

All of the major machine vision companies and software products now offer support and advice for using their products with GigE cameras. The GigE Vision standards from the Automated Imaging Association (AIA) are widely supported and allow mixing and matching of different vendor's cameras and drivers within the same application framework. This is true mix-and-match, whether the vendors like it or not, and it is very good for machine vision consumers.

So, the obstacles to GigE vision and the fears of the past are falling by the wayside. Don't be afraid of GigE cameras, and don't fall for the old-school conventional wisdom on GigE camera limitations. Conventional wisdom is often just the sum total of all of the things that used to be true. I wonder what non-GigE cameras will even be available in another 18 months?