By Robert Cravotta, Technical Editor

MACHINE VISION CONTINUES TO FIND ITS WAY INTO MORE TYPES OF APPLICATIONS.

The eyes of the machine

Illustration by Dan Guidera

Any real-world data-recognition system is challenging to implement. Machine vision is an example of such a system, but such systems can emulate any of the senses. The goals of a real-world data-recognition system are to capture an energy profile, extract key features from the data, and autonomously or assistively perform some logic based on the extracted information. Designers are implementing real-world data-recognition systems for automated voice recognition, facial recognition, high-speed sorting and quality inspection, manufacturing robotics, automotive safety, video surveillance, traffic control and license-plate inspection, biometrics, medical imaging, and homeland-security applications.

The threshold for implementing real-world data-recognition systems is constantly shifting as their cost and the risk to integrate them into a design continue to drop. The range of turnkey and customizable systems for adding real-world data recognition to your application is expanding each day. The increasing processing power is allowing high-value, low-volume applications, such as homeland security and medical imaging, to more reliably perform more functions with more complex recognition. For manufacturing and inspection systems, these applications improve productivity with more consistency and fewer errors, especially in hazardous environments.

The set of logical components and functions that can make up a generic real-world data-recognition system includes a signal source, a sensor, data acquisition, a data processor, an executive controller, digital I/O, and network connectivity (**Figure 1**). The context of each component and function is application-specific and relevant to the type of energy the system is capturing and acting upon. For example, for a sound-capture application, such as speech recognition, the target generates and emits the signal source as compression waves. The system can capture compression waves as 1-D data by using a single microphone or as multidimensional data by using multiple microphones in conjunction with each other.

The signal source need not originate outside the system. For a radar, sonar, or laser application, the system could generate and emit the appropriate energy pulse and capture the reflected energy with the sensor in correlation to the pulse generation. The time delay in receiving the signal as well as how the returning signal strength varies can impart details such as distance, direction, and the materials in the area by measuring how those materials or any other objects in the environment absorb the pulse energy. For example, you could use a laser pulse tuned to an appropriate frequency to detect the presence of a gas in an area based on how the laser-pulse energy drops (when the gas absorbs it) when returning to the sensor.

AT A GLANCE

▷ Consider each component of a data-recognition design system together from the start to avoid unnecessary costs and risks.

▷ Using the appropriate illumination and configuration can reduce your image-processing requirements.

▷ Heavy data-processing applications, such as machine vision, can benefit from careful memory-use planning and using DMA (direct memory access) for the data movement.

For an optical machine-vision system, the illumination or signal source could be emitting from the target object; it could be light from a natural light source reflecting off of the target object; or it could be light from light sources specifically associated with the machine-vision-design configuration reflecting off the target object. The characteristics of the signal source—such as frequency, color, intensity, and angle of presentation—can play a significant part in a sensor's ability to capture usable data (see **sidebar** "Controlling image quality"). You need to choose the type of signal source that best highlights and contrasts

CONTROLLING IMAGE QUALITY

Delaying your choice of lighting component for your machine-vision application until the end of the project can be detrimental to the success of your design effort. As with all other systems, you should consider the strengths and weaknesses of all the components together; otherwise, you may find yourself in an intractable situation or one that requires what would otherwise have been avoidable costs and heroics. When you can control your lighting conditions, the choice you make affects your ability to reliably create contrast between the important features in your application. Choosing lighting that amplifies the contrast between the features that most interest you can reduce the performance requirements you need for the image processing your application requires; it can reduce not only your bill-of-materials cost, but also the time and cost of your engineering effort.

Lighting options abound, and common options include fluorescent, LED (light-emitting-diode), halogen, and xenon lamps (**Reference A**). Fluorescent lamps emit a white light similar to natural sunlight. LED lamps, which generally have the longest life, are available in a variety of colors and shapes that support a range of applications, and you can operate them at frequencies of 20 kHz to eliminate inconsistencies from flickering. Halogen lamps can provide high brightness that is appropriate for high-speed inspection lines and localized areas; you can adjust their illumination. Xenon lamps work well as strobes of intense light with high flash rates that are appropriate for high-speed line inspection, but they cannot continuously emit light.

To capture consistent images, you need stable illumination–without which you risk complicating your image-processing requirements. The shape of the illumination characterizes the types of lighting you can use. Fluorescent, LED, and halogen lamps are sources for ring illumination, which is a general, multipurpose technique that emits light in a ring shape. Fluorescent and LED lamps are sources for straight illumination, which is linear, can be as long as 2m, and is appropriate for wide areas, such as sheet material. Fluorescent and LED lamps are also sources for flat illumination, which is even backlighting over a square or rectangular area. Halogen lamps are sources for spot illumination, which provides intense illumination for a localized area and is suitable in image enlargement.

Whether your image target has a surface luster influences your illumination choices. Using oblique lighting, which relies on diffuse reflections, minimizes the effects of luster by directing the lighting so that it is not reflected into the image sensor. Typical oblique-lighting settings include mounting a ring lamp closer to the target, applying surface lighting from opposite angles, and using indirect lighting.

Using specular reflective lighting emphasizes the target's luster and directs light so that it reflects into the image sensor. You may consider using special illumination if the target lighting includes a mix of diffuse and specular reflections, because the bright areas of the image may result in glare and contribute to poor-quality images. Low-angle illumination uses an LED ring-type unit mounted close to the target, so that the image sensor sees reflections only from edge details. Coaxial, vertical illumination uses a one-way mirror to illuminate the target and to block any reflected light for a uniform specular reflection; it is effective for detecting alignment marks on pc boards. Dome illumination uses diffused reflections from the interior walls of a dome to provide shadowless illumination. Its setup is effective for lighting irregularly shaped targets with shiny, curved, or swelled surfaces.

It is also important to consider the output stability and maintenance of your illumination choice over time. For example, LEDs generally cost more upfront, but they provide 10 years of continuous use with stable output. The upfront cost of the LED may balance out when you consider the cost of shutting down a production machine at an unscheduled time due to lighting. Other quality considerations for your lighting sources are the fluctuations they exhibit from variations in voltage, temperature, vibration, and air quality. Choosing your lighting source, type, and configuration can significantly impact the quality of images you capture and the overall reliability of your image processing.

REFERENCE

A. *Keyence Guide to Correct Machine Vision Usage, How to Optimize Machine Vision Performance*, www.keyence.com.

the important features of the world objects or materials you are trying to sense. Optimally choosing and configuring your type of signal source can greatly ease your data-processing requirements if you can eliminate or reduce signal noise and variability and enhance the contrast between the features that interest you.

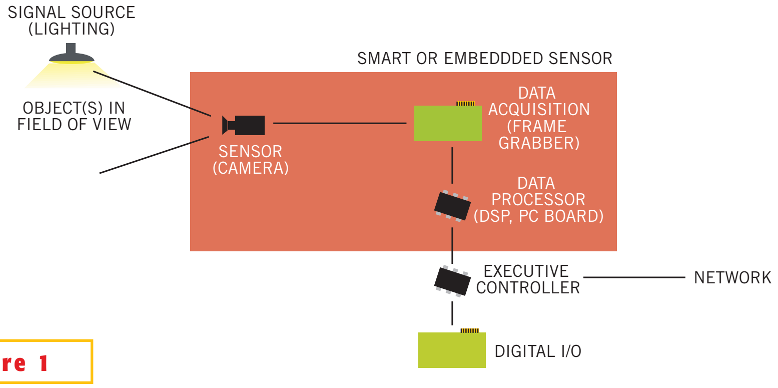

Figure 1

Below the generic descriptions of these basic components and functions of a generic real-world-data-recognition system are the component identifiers for a machine-vision system. Smart or embedded cameras integrate the highlighted components for cost savings and higher reliability.

A real-world data-recognition system needs at least one sensor to capture data, but systems may employ more than one sensor or even different types of sensors to accomplish the task at hand. An example of a system using multiple homogeneous sensors is a stereoscopic data-capture system in which the two sensors capture overlapping data to support the determination of depth information. Such systems are important in robotic assisted surgery, in which a surgeon observes an operation on a video monitor and depth perception is critical. An example of a heterogeneous-sensor configuration is a machine-vision system on an autonomously controlled vehicle. For example, in the recent DARPA Grand

MANAGING DATA MOVEMENT

Effectively managing the data flow and movement between memories is key to meeting the real-time constraints for many machine-vision applications. Efficient data movement leverages the processor architecture to avoid idling the processor core by masking the overhead of data movement so that the processor core is not waiting for data. The architecturally closer the memory is to the processor core, the faster memory accesses can be. The memory closest to the processor core is a limited resource that you must manage for applications in which the amount of data to process exceeds the limits of the internal memory resources. Because memory structures are architecturally farther from the processor core, they require more handshaking and control, which extends the time the processor core needs to access them.

The fastest and most expensive memory is the chip register, which holds temporary and intermediate data. Code compilers use and rely on the registers to schedule and optimize software performance. Cache or internal memory is the next fastest and next most costly type of memory; it supports memory accesses for instructions and data at or close to the processor core's operating speed. Some processor architectures employ multiple levels of cache or internal memory on-chip to trade off between access speed and cost. External memory is the memory farthest from the processor core and can take many forms that trade off access time, size of storage, and cost.

A DMA (direct-memory-access) controller can offload from the processor core the need to directly move data between the types of memory. It can also offload from the processor core the overhead of moving data between the peripherals and the memory. The processor core can issue a few instructions to the DMA controller to move data that it will soon need to access between slower and faster memory and then resume its activity while the DMA controller moves the data. You can indirectly benefit from the DMA controller without modifying your application code if it is sufficient to rely on the DMA implementation built into the operating-system and peripheral-device drivers.

However, for real-time, data-intensive applications, such as machine vision, it is often insufficient to rely solely on generic DMA control. In such cases, it is often beneficial to directly manage the data flow and movement structures to stage the data between the various levels of memory at the appropriate time. Several engineering-consulting companies have observed that it is not uncommon for designers new to real-time, data-intensive applications to improperly use the DMA controller the first time out. However, they realize large performance gains after learning how to directly employ it.

To be effective, the DMA controller should include enough channels to support the number of data streams, in addition to moving data between memories, in the application. In cases involving multiple bidirectional data streams, the DMA controller should provide priority or burst control so that a single data stream cannot consume the entire bus bandwidth, degrade the overall system performance, and possibly cause some data buffers to overflow. If the internal, on-chip memory uses banks, careful planning can allow the DMA controller to maximize data movement. A 2-D DMA capability can allow you to place data into memory in a more optimized order. An example is to interleave or deinterleave data, such as RGB data, so that data can sequentially come into the system, but the DMA controller automatically stores it in separate buffers.

It's usually ineffective to directly employ the DMA controller for all data movement. To squeeze more performance from the DMA controller, you need to instrument your code. Doing so increases the size and complexity of your code, and for legacy code, requires you to perform additional testing and validation. However, for areas of your application code that require high throughput, you may be able to justify the extra time, effort, and cost of planning and implementing a smart layout and data-management scheme using the DMA controller. Efficiently using the DMA controller can make the difference of using a lower clock rate, using a less costly bill of materials, or implementing a market-differentiating capability in your system.

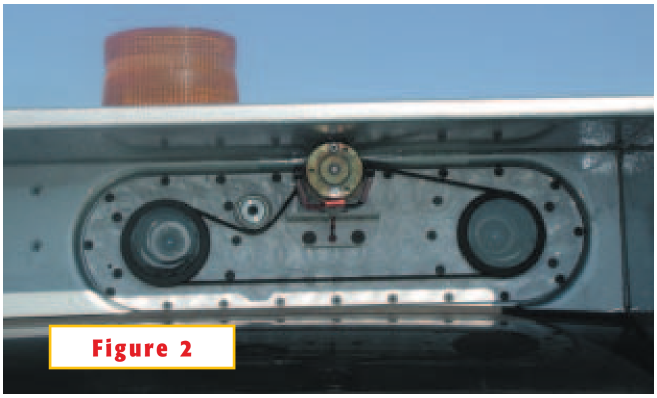

Figure 2

The stereoscopic vision system for this DARPA Grand Challenge entry uses two cameras for autonomous navigation and obstacle avoidance for the vehicle. The motor and belt assembly are part of a spinning disk mechanism used to keep the lenses clean during the event (courtesy Digital Auto Drive).

Challenge, several teams attempted to field vehicles that could autonomously travel between Los Angeles and Las Vegas (**Figure 2**). In this type of vision application, the vision system has to "boresight" the camera—correlating the vehicle-movement data from the vehicle's guidance, navigation, and control system with the apparent movement of road features as captured by the cameras. For simplicity, this article addresses the real-world data-recognition system as an optical machine-vision system, but many of the concepts apply analogously to the other types of real-world data.

CAPTURE THE SIGNAL

Analog cameras have been the dominant choice in machine-vision systems, because they have a huge installed base, use a mature technology, offer adequate performance for many applications, and are relatively inexpensive compared with digital cameras. Digital cameras can provide higher data rates, higher resolution, and higher bit depths, and they are less susceptible to noise than analog cameras. However, they have suffered from poor cable interchangeability between equipment providers and among different types of cameras and frame grabbers. The Camera Link standard has helped to eliminate many of these cabling and connectivity challenges. The proliferation of inexpensive consumer-grade digital cameras has exerted more downward price pressure on digital components that is evening the total cost between analog and digital systems.

To choose a camera system, you need to know what features you need to highlight during capture to yield an image that enables accurate and reliable feature extraction. The feature characteristics you need to consider include the target object's color, size, texture, quantity, orientation, markings, and contours. You also need to consider the lighting requirements; whether a light's brightness, frequency, or color enhance or obscure the features you need to extract; the amount of control you have over the environment; space constraints; how shadows and bright reflections affect the image quality; whether you can control the lighting conditions; and the smallest feature that you have to distinguish.

The target object's overall size and smallest discernible feature influence your choice of pixel resolution. As an example, for a 90-nm circuit-inspection system, the total pixel resolution can approach 1 million×1 million pixels. Choosing a system with higher resolution yields higher precision images but increases your data-bandwidth requirements and the amount of data to process between cycles. One way to minimize the impact of the exponential increase in bandwidth and processing requirements for higher resolution images is to lower the frame rate. The frame rate is the number of complete frames the camera sends to a data-acquisition system within a predefined time interval. The speed at which the target object may be moving can also affect your pixel-resolution choice, because you want to avoid blurring during image capture. You can control motion blurring by limiting the exposure time or using strobe lights. To avoid blurring in the image, you generally need to limit the exposure period to less time than it takes for the target object's smallest discernible feature to move more than 1 pixel.

The goals of the data-acquisition component are to acquire the image data from the camera, apply any data preprocessing, and minimize error. Preprocessing can consist of rotating the captured image, converting the analog output from the camera to digital data, and organizing the data to distribute it across multiple image-processing units. In PC-based machine-vision systems, the frame grabber or video-capture card bridges the camera and the host system in which the feature extraction will take place. The frame grabber is usually a plug-in card in a PC-based system.

IMAGE PROCESSING

In a PC-based vision system, the PC processor or another plug-in board populated with DSPs and FPGAs may per-

FOR MORE INFORMATION...

For more information on products such as those discussed in this article, contact any of the following manufacturers directly, and please let them know you read about their products in *EDN*.

Altera
1-408-544-7000
www.altera.com

Analog Devices
1-800-262-5643
www.analog.com

Atmel
1-408-441-0311
www.atmel.com

Celoxica
+44-0-1235-863656
www.celoxica.com

Cradle Technologies
1-650-210-3600
www.cradle.com

Data Translation
1-800-525-8528
www.datatranslation.com

Digital Auto Drive
1-408-465-2800
www.digitalautodrive.com

d3 engineering
1-585-429-1550
www.d3engineering.com

Keyence
1-888-539-3623
www.keyence.com

Mango DSP
1-408-437-2230
www.mangodsp.com

National Instruments
1-888-280-7645
www.ni.com

PolyCore Software
1-650-570-5942
www.polycoresoftware.com

Stretch
1-650-864-2700
www.stretchinc.com

Texas Instruments
1-800-336-5236
www.ti.com

3D-Computing
1-972-223-2904
www.3d-computing.com

Xilinx
1-408-559-7778
www.xilinx.com

form the image processing. When the PC processor performs the image processing, you can choose among a variety of third-party vision-software offerings. When a DSP- and FPGA-based system handles the processing, your choice of third-party vision software offerings may narrow. Non-PC-based vision systems often rely on DSPs and FPGAs to perform the image processing. The processing performance required for real-world data-recognition systems ranges from a single processor to many processors—in some cases, 1000 DSPs and FPGAs, operating in parallel and pipelined configurations. The optimal processing configuration for image processing depends on the application and is beyond the scope of this article.

A trend is to integrate more image-processing capacity into the camera assembly itself. These smart cameras integrate a camera and image-processing system that can be more cost-effective and reliable than a PC-based system, in part because its lack of fans and hard drives means fewer moving parts. Smart cameras can offer a smaller overall form factor than a PC system. Some smart cameras include the data-processing module in the camera assembly; others place it in a separate box that connects to the camera assembly, so that the camera assembly can remain small. Smart cameras are unlikely to replace PC-based systems; they are more complementary than competitive. For example, you can connect a smart camera to a PC to perform repetitive image-processing functions and pass only the results to the PC, which acts as the executive controller.

A strength of developing and using hardware-independent software is that you can operate it on a wider selection of hardware, which may enable you to choose your hardware across a variety of form factors and price and performance ranges with less chance that the software will become obsolete. However, efficiently managing the movement and data flow in a vision system can be critical to obtaining real-time performance with your design; hardware-independent software cannot use the data-movement structures, such as DMA controllers, as efficiently as hardware-dependent software (see **sidebar** "Managing data movement"). Hardware-dependent software differentiates itself by exploiting hardware-architectural features that provide highly optimized performance and can allow you to meet your image-processing needs with a lower cost processor configuration.

The executive controller is a logical component, and the data-processor component may physically include its tasks, such as in a PC-based system. It acts on the extracted feature information, which it identifies during image processing, and it acts through the digital I/O and network connection to communicate with other systems and databases. The executive controller may function autonomously, based on what the vision system captured, to directly command and control another aspect of the system, such as a motor. Or, it may function in an assistive fashion by flagging a condition and communicating that information to a person for further consideration.

As machine-vision systems apply greater processing performance to an increasing set of data, image processing will continue to evolve from coarse analysis to finer analysis. An expanding set of software-vision tools can help offset the increasing complexity of designing and implementing image-processing systems by abstracting common functions for image preprocessing, feature extraction and recognition, and analysis. Examples of software-vision tools include image-format conversion, image compression, space transformation, segmentation and contour following (edge detection), object classification, pattern matching, measurement of geometric shapes, and mark inspection. Depending on your application needs, you may build and use your own software-vision tool kit, but a growing body of third-party-software vision-tool offerings can effectively complement your custom code.□

AUTHOR'S BIOGRAPHY

Technical Editor Robert Cravotta in the early 1990s worked on detection systems using tunable lasers and vision systems for small autonomous spacecraft. You can reach him at 1-661-296-5096 and via e-mail at rcravotta@edn.com.

TALK TO US

Post comments via TalkBack at the online version of this article at www.edn.com.