Basic Vision with LabVIEW

Raul G. Longoria

Updated Fall 2011
Machine Vision Concepts*

- Machine (or computer) vision has six principal areas
  1. Sensing – which yields a visual image
  2. Preprocessing – noise reduction, image enhancement, etc.
  3. Segmentation – partitioning an image into objects of interest
  4. Description – computation of features for differentiating among types of objects
  5. Recognition – identifying objects (e.g., bolt, wrench, etc.)
  6. Interpretation – assigning meaning to an ensemble of recognized objects

- Levels of processing are divided into low (1, 2), medium (3, 4, 5), and high (6)

- In this course, we’ll primarily be concerned with low-level vision, and will utilize some functions of medium-level vision.
Machine Vision Concepts*

- Low-level vision involves processes considered ‘primitive’ (or automatic) and requiring no ‘intelligence’ (1,2). This could be thought of as analogous to how a human eye senses and adapts.
- Medium-level vision extracts, characterizes, and labels components in an image.
- High-level vision refers to processes that attempt to emulate perception and cognition.

“To see is to think.”

— Salvador Dali

(and quoted by E. Dickmanns in his book Dynamic Vision for Perception and Control of Motion, Springer, 2007)
LabVIEW-based Vision

- LabVIEW Vision enables you to read/create image files and provides means for managing those files.
- There are built-in functions (VIs) for analyzing image files (select areas of interest, measure intensity, etc.).
- It is necessary to also have LabVIEW IMAQ software which enables you to acquire images from cameras.
- In this course, we want to demonstrate how you can use these software tools to develop a simple vision-based measurement system, particularly for object motion.
Overview of LV-Based Vision Tools

- Image data type
- Analyzing images
- Capturing images

Vision Utilities
Image processing
Machine vision
Analyzing Images

• **Vision Utilities** – VIs for creating and manipulating images, etc.

• **Image Processing** – provides ‘low level’ VIs for analyzing images.

• **Machine Vision** – groups many practical VIs for performing image analysis. For example, the “Count and Measure Objects” VI is found under this group.
Vision Utilities

To create and manipulate images

- Image management (create, dispose, etc.)
- File handling
- Image manipulation
- Pixel editing
- etc.

- Best to learn use through examples.
Reading an Image File

This VI block diagram opens an existing image file (e.g., a bitmap), reads the file, and then displays it. On the Front Panel, place an ‘Image Display’ to get this terminal; then wire image data.
Example – looking at an image

The simple VI in the previous slide can be used to open an image file.

This read-out indicates the size of the image (pixels).

When moving the cursor around the image, the readout shows cursor (x,y) coordinates and the ‘intensity’ value at that location.

This is the ‘clamp’ example file provided in LabVIEW.
Some LV Machine Vision VIs

- Select Region of Interest
- Light Meter
- Count and Measure Objects

- You’ll learn how to use some of these in the lab. There are many others you can skim through to get an idea of what is available.
IMAQ Select Region of Interest

You can focus on regions based on:
point, line, rectangle, annulus

The output from this VI can be sent to other VIs that require that bounding information.
IMAQ Light Meter

If you want to examine pixel intensity in a certain region, you need rectangle information.

Show Search Area
Image
Rectangle
Coordinate System
error in (no error)

IMAQ Select Rectangle
Use to specify a rectangular area in the image.
Rectangle coordinates are output and can be sent to next function.

Rectangle

Histogram data (send directly to a waveform graph)
Count and Measure Objects

Another VI that needs rectangle information, and which is very useful for basic segmentation is the ‘Count and Measure Objects’ VI.

This VI needs several inputs, as shown below.

NOTE: This VI requires that you convert the image to grayscale.
IMAQ Cast Image

Example usage shown below:
Example – Finding Objects and Intensities

3 objects detected

The limit on object size prevented the 3 larger objects in the ROI from being identified
**Define ‘Threshold’**

*Threshold* specifies the grayscale intensity that is used as threshold level. When a *Bright Objects* type is selected, the threshold range used includes *Threshold* to the maximum possible intensity depending of the image type (255 for a 8-bit image). When a *Dark Objects* type is selected, the threshold range used includes the minimum possible intensity depending of the image type (0 for a 8-bit image) up to *Threshold*.

Bright objects have ‘high’ intensity values (e.g., 255 for 8-bit)

Dark objects have ‘low’ intensity values (e.g., 0 for 8-bit)

For 8-bit image

The ‘Threshold’ must often be specified as an input to some machine vision VIs.
Within the ROI, a histogram is generated of the intensity values. Note that most of the image is made up of pixels with intensity greater than about 180. White is 255.
Capturing Images

- **PCI cards** (for capturing from streaming source)
- **USB cameras*** (web cams, etc.)
- **Firewire cameras**
- **Ethernet/wireless cameras**

**USB is the approach targeted for this course:**
- Low-cost
- Relatively easy to use

*IMAQdx refers to VIs that can be used with cameras that interface directly (‘direct show’)*
USB Cameras (Webcams)

- USB webcams are probably the slowest cameras available, especially the way they are to be used in this course.
- Our experience has shown that the maximum bandwidth we can achieve for image acquisition is about 10 frames/sec (within LabVIEW).
- Some online sources indicate that ‘hacked’ webcams can achieve 30 frames/sec.
- So, it is the software environment (Windows, LV, communications, etc.) that we’ve chosen that is placing the restrictions on the performance.
Acquire a USB Camera Image
Front Panel

NxM, 32-bit RGB image

RGB levels…

Cursor location

800x600 0.41X 32-bit RGB image 105,45,33 (481,414)
Summary

- Vision VIs in LabVIEW provide a way for us to include image acquisition and analysis to our existing set of tools (simulation, DAQ).
- The vision VIs alone allow you to use an image as a data type.
- This image can be loaded from a file or acquired using IMAQ routines.
- Once within LabVIEW, the image can be processed using some very sophisticated built-in programs.
- The laboratory work will introduce specific methods that will enable you to build a simple ‘motion capture’ system.
Image Data Type

Menu: NI Measurements->Vision->Vision Utilities->Image Management

**IMAQ Create**

Creates an image.

- **Image Name**
- **Image Type**
- **Border Size**
- **New Image**
- **error in (no error)**
- **error out**

This VI is used to create an image. It is called prior to, say, capturing an image using a camera.
Image Type and Bit Depth

- We know digital images are formed by an array of pixels, and each pixel is quantized into a number of levels based on the number of bits available.
- Depending on whether pixels are black and white, grayscale, or color, pixels have different bit depths. Bit depth refers to the amount of information allocated to each pixel.
- When pixels are either black or white, pixels need only two bits of information (black or white), and hence the pixel depth is 2.
- For grayscale, the number of levels used can vary but most systems have 256 shades of gray, 0 being black and 255 being white. When there are 256 shades of grey, each pixels has a bit depth of 8 bits (one byte). A 1024 x 1024 grayscale images would occupy 1MB of memory.
- In digital color images, the RGB (red green blue, for screen projection) or CMYK (printing color) schemes are used. Each color occupies 8 bits (one byte), ranging in value from 1-256. Hence in RGB each pixel occupies 8x3 = 24 (3 bytes) bits, in CMYK 8x4 = 32 bits (4 bytes).
- Note, LV uses an ‘alpha’ channel for RGB. The alpha channel stores transparency information--the higher the value, the more opaque that pixel is.