Basic Vision with LabVIEW

Raul G. Longoria
Department of Mechanical Engineering
Fall 2006

Vision Lab Goals

- Learn how to **read/create** **image files** using LabVIEW, and how to manage those files
- Learn about built-in functions for **analyzing image files** (select areas, measure intensity, etc.)
- Learn how to **acquire images** using a USB camera
- Build a **vision-based measurement** system, particularly for object motion.

Options on Pre-Lab Exercises

- Concept of image data type, and handling images
- Intensity measurements and use of histograms
- Understanding basic camera hardware and limitations
- Basic functions for acquiring images with IMAQ USB VIs
- Understand basic analog meter circuit

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 lab, we'll primarily be concerned with low-level vision, and will utilize some functions of medium-level vision.

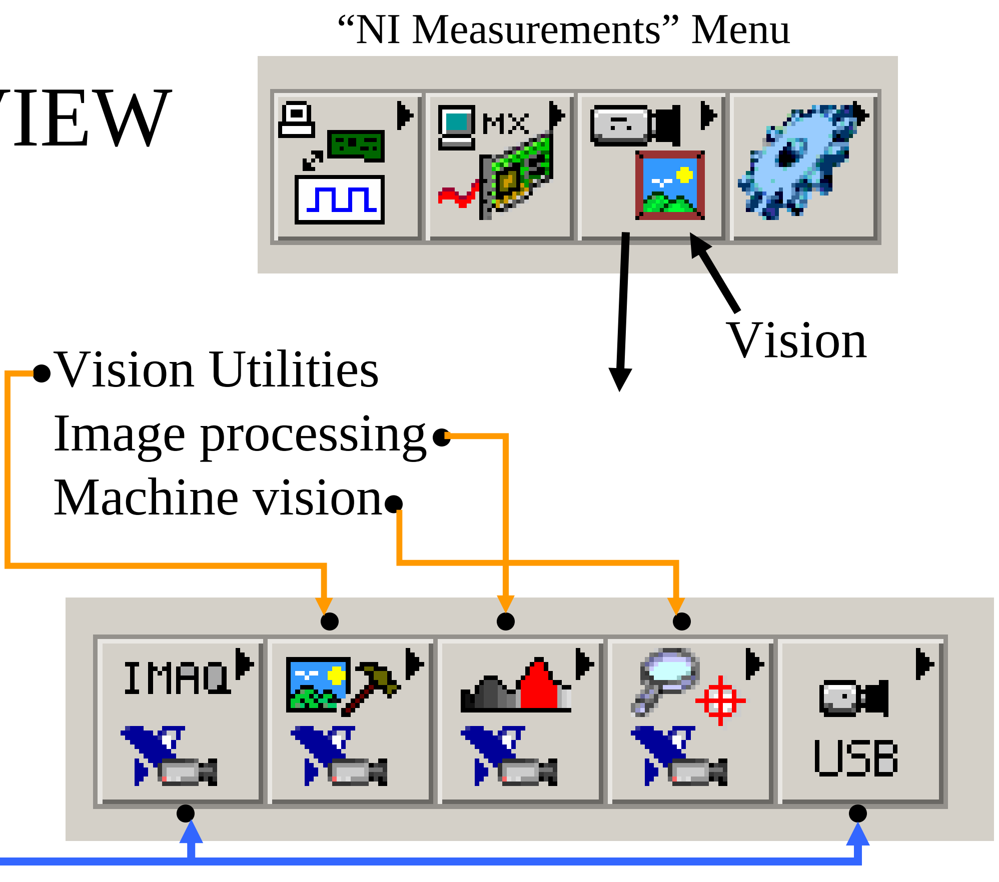
Machine Vision Concepts*

- 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 cognition.

*From Fu, Gonzalez, and Lee, Robotics: Control, Sensing, Vision, and Intelligence, McGraw-Hill, New York, 1987.

Overview of LV-Based Vision Tools

- Vision tools in LabVIEW
- Image data type
- Analyzing images
- Capturing images



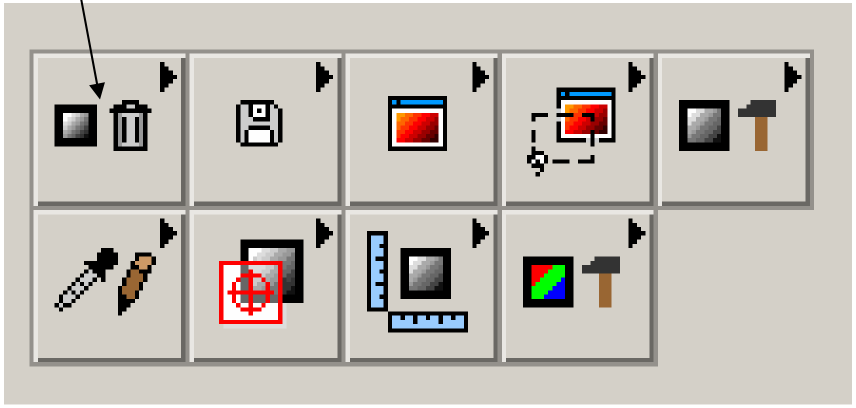
Analyzing Images

- 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

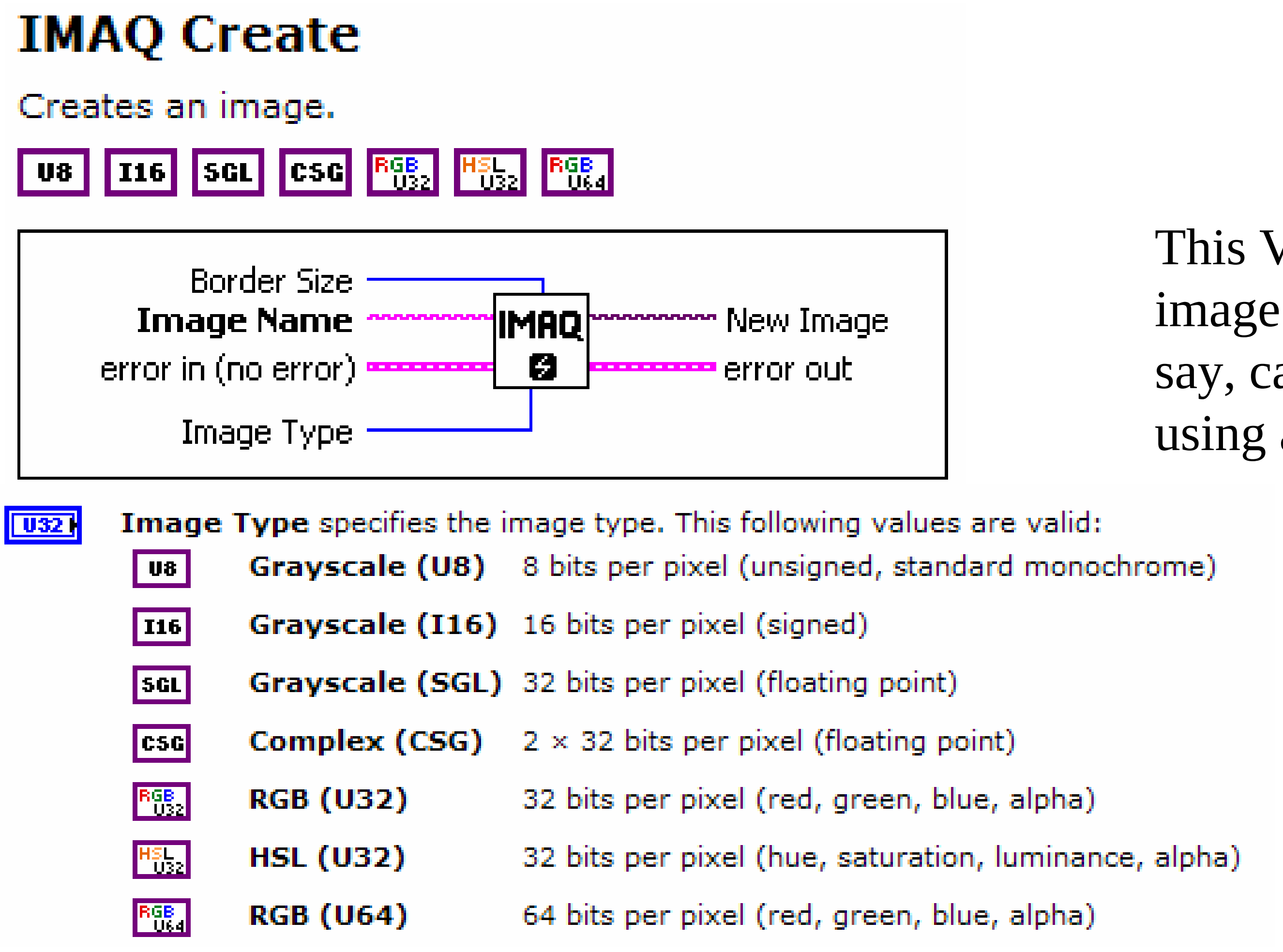
- Image management (create, dispose, etc.)
- File handling
- Image manipulation
- Pixel editing
- etc.



- Best to learn use through examples.

Image Data Type

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



IMAQ Create

Creates an image.

U8 I16 SGL CSG RGB U32 HSL U32 RGB U64

U32 **Image Type** specifies the image type. This following values are valid:

U8	**Grayscale (U8)**	8 bits per pixel (unsigned, standard monochrome)
I16	**Grayscale (I16)**	16 bits per pixel (signed)
SGL	**Grayscale (SGL)**	32 bits per pixel (floating point)
CSG	**Complex (CSG)**	2 × 32 bits per pixel (floating point)
RGB U32	**RGB (U32)**	32 bits per pixel (red, green, blue, alpha)
HSL U32	**HSL (U32)**	32 bits per pixel (hue, saturation, luminance, alpha)
RGB U64	**RGB (U64)**	64 bits per pixel (red, green, blue, alpha)

This VI is used to create an image. It is called prior to, say, capturing an image using a camera.

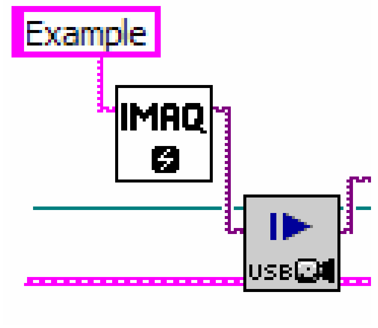
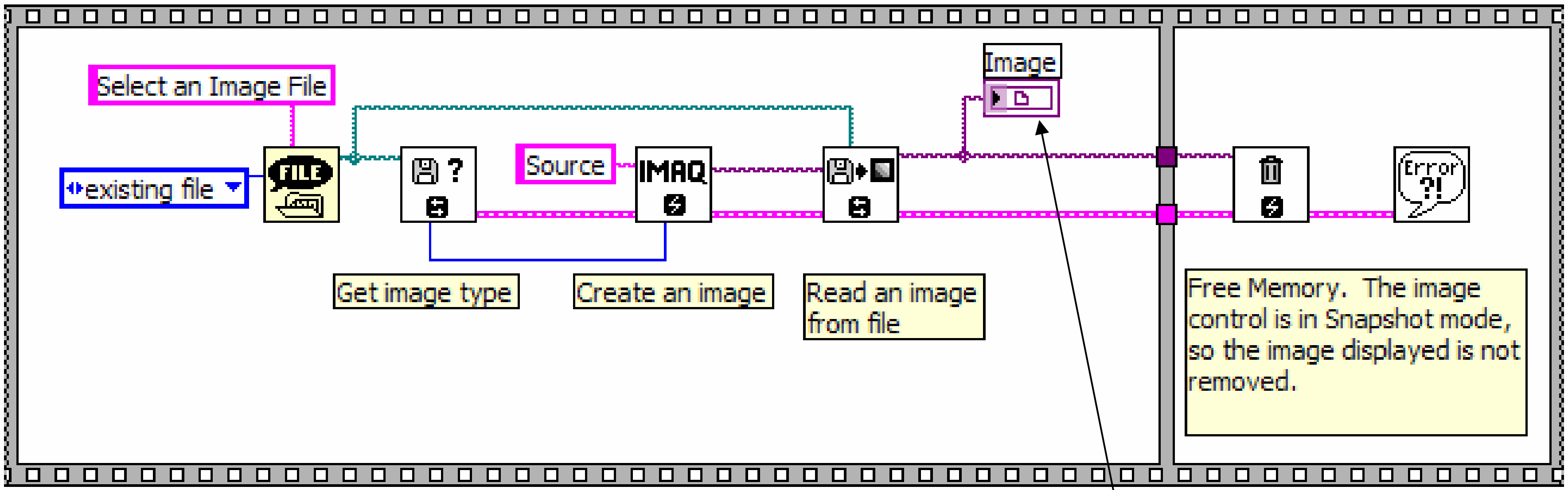


Image Type and Bit Depth

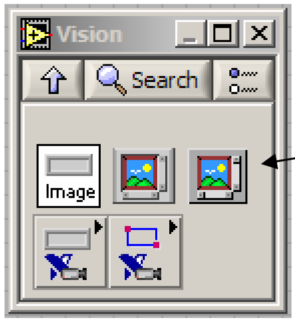
- We know digital images are formed by an array of pixels, and each pixel is **quantized** into a number of levels based on the **number of bits** available.
- Depending on whether pixels are black and white, grayscale, or color, pixels have different **bit depths**. Bit depth refers to the amount of 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.

Reading an Image File

This VI block diagram opens an existing image file (e.g., a bitmap), reads the file, and then displays it.



Free Memory. The image control is in Snapshot mode, so the image displayed is not removed.



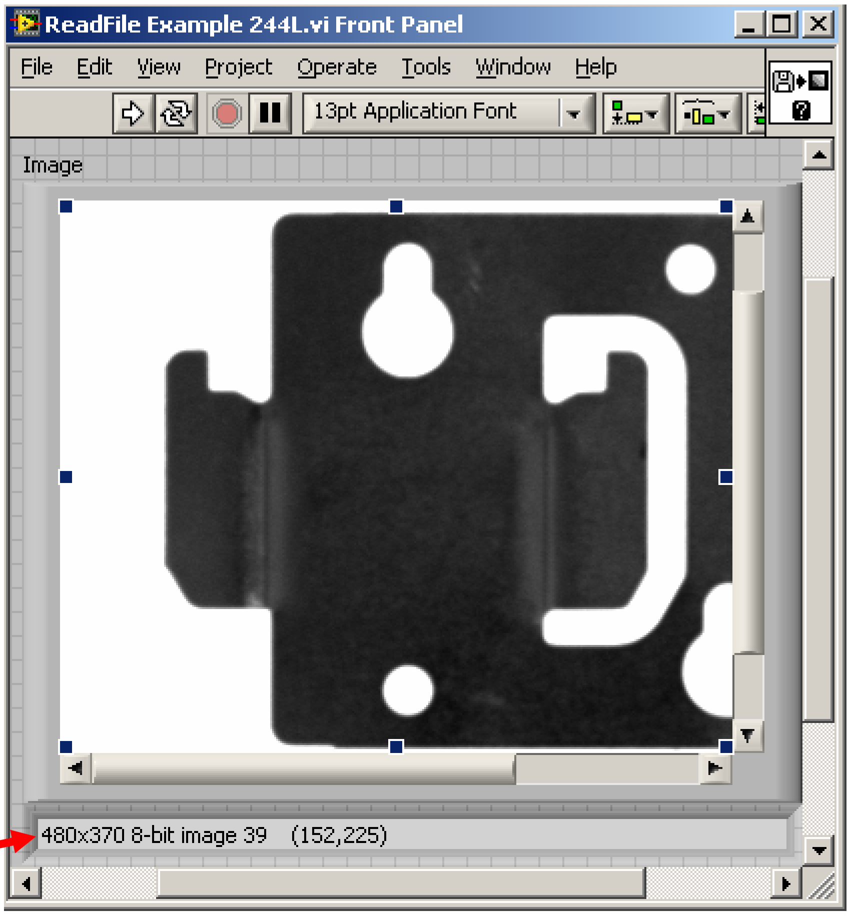
On the Front Panel, place an 'Image Display' to get this terminal; then wire image data.

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.



480x370 8-bit image 39 (152,225)

This is the 'clamp' example file provided in LabVIEW

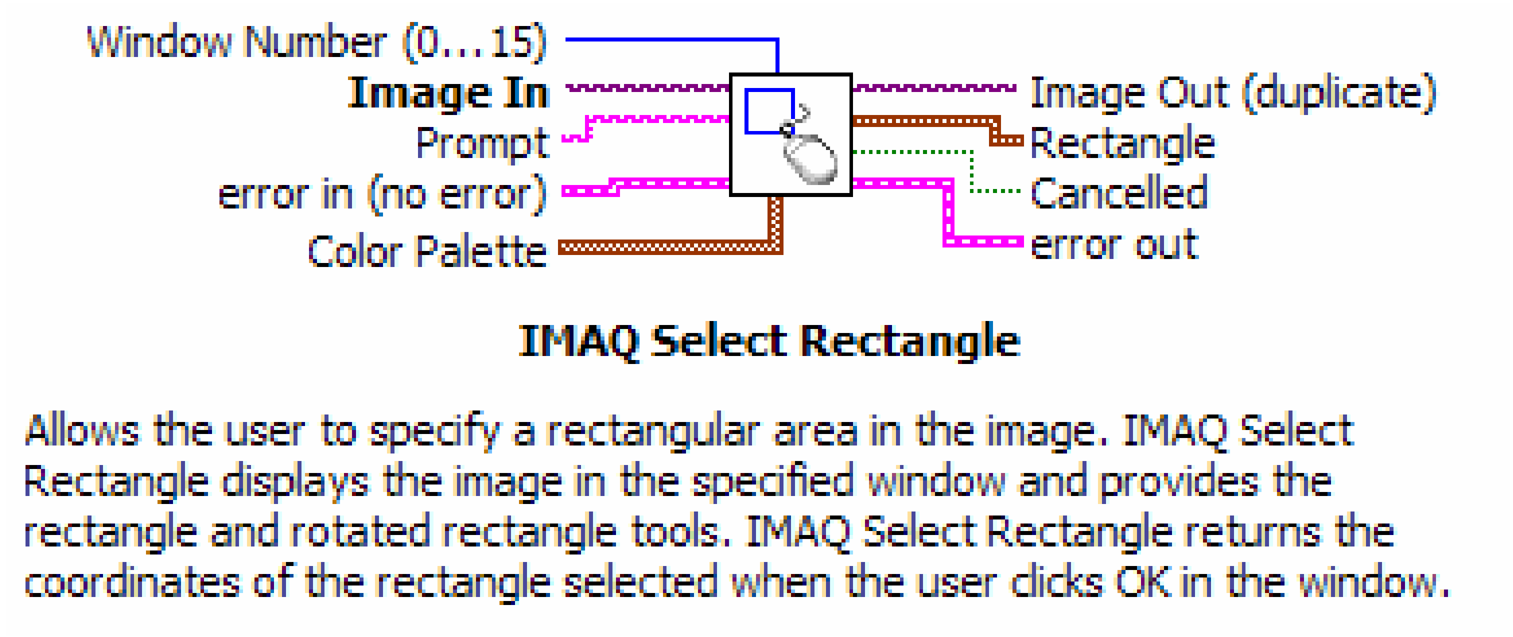
Some LV Machine Vision VIs

- 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

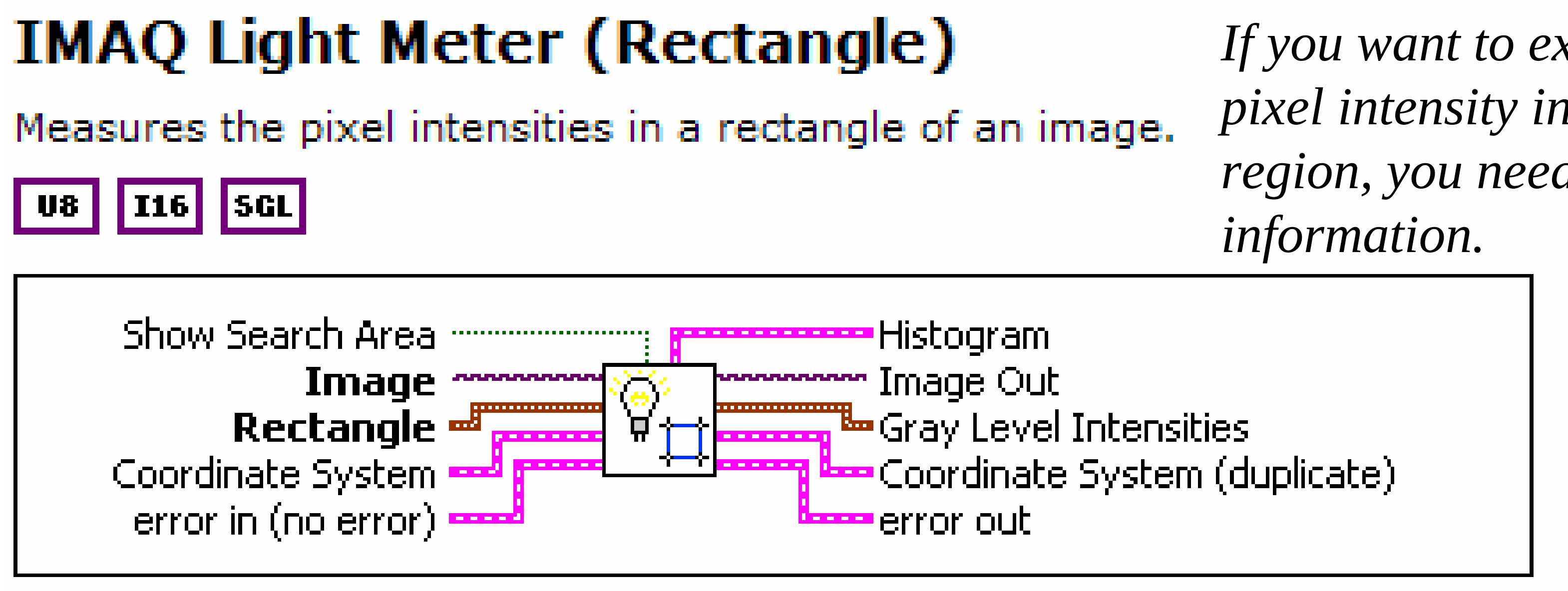


IMAQ Select Rectangle

Allows the user to specify a rectangular area in the image. IMAQ Select Rectangle displays the image in the specified window and provides the rectangle and rotated rectangle tools. IMAQ Select Rectangle returns the coordinates of the rectangle selected when the user clicks OK in the window.

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

IMAQ Light Meter

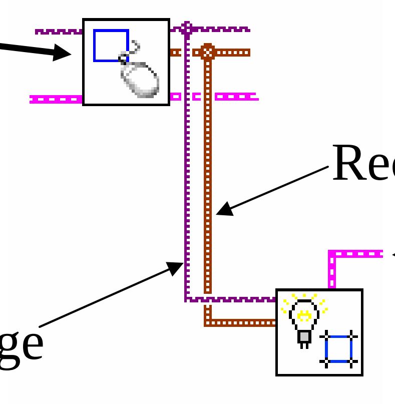


IMAQ Light Meter (Rectangle)

Measures the pixel intensities in a rectangle of an image.

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

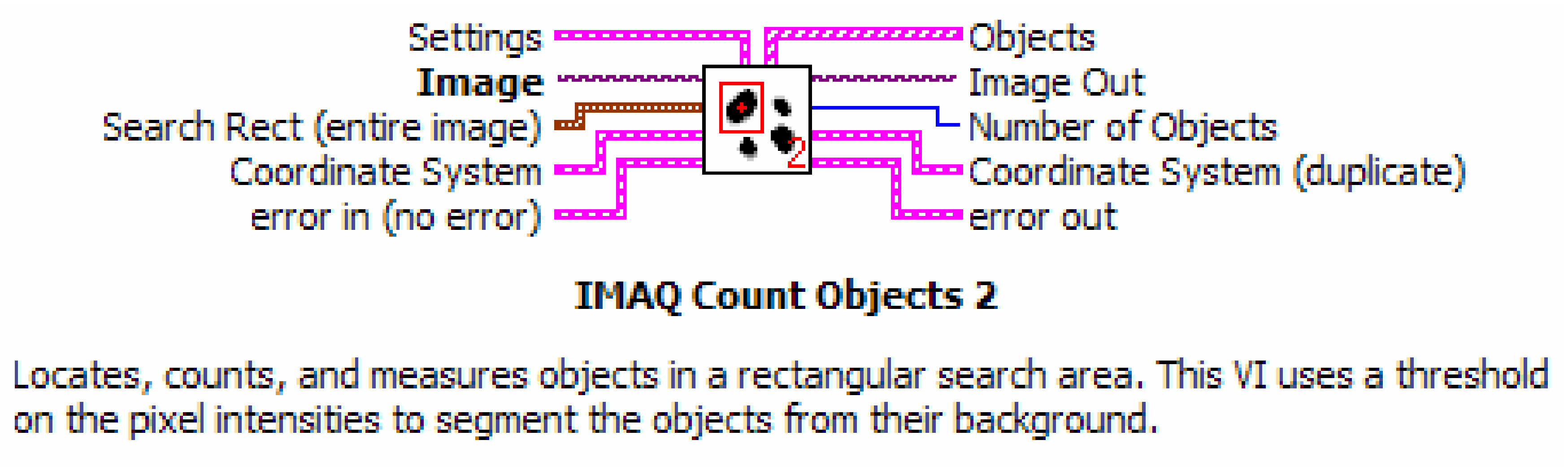
IMAQ Select Rectangle
Use to specify a rectangular area in the image. Rectangle coordinates are output and can be sent to next function.



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.

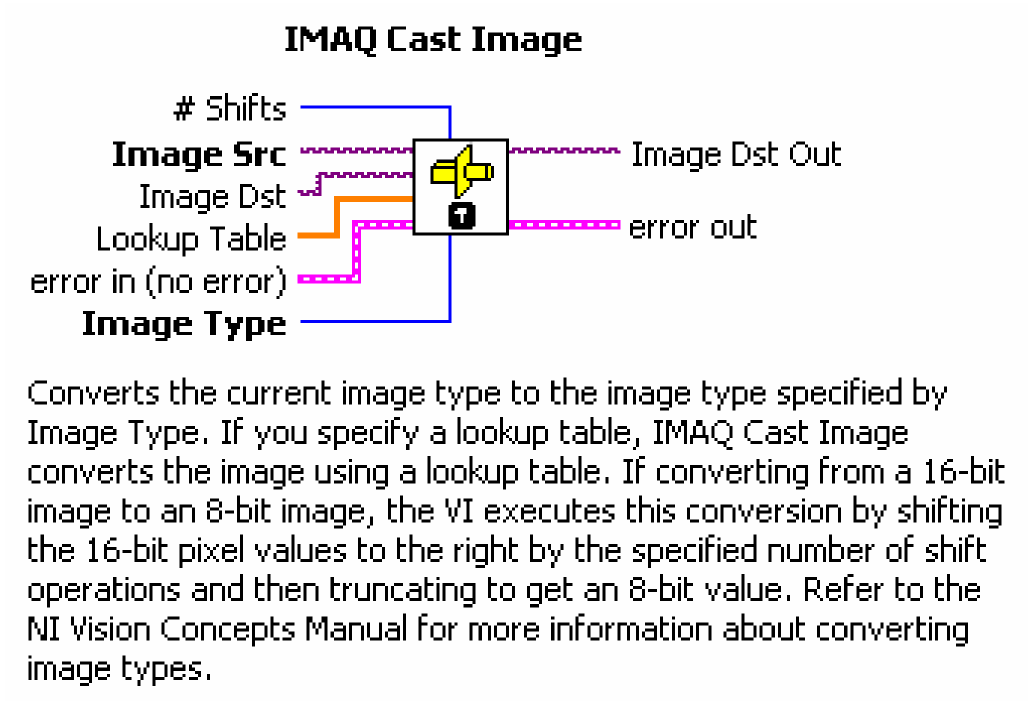
This VI needs several inputs, as shown below.



Locates, counts, and measures objects in a rectangular search area. This VI uses a threshold on the pixel intensities to segment the objects from their background.

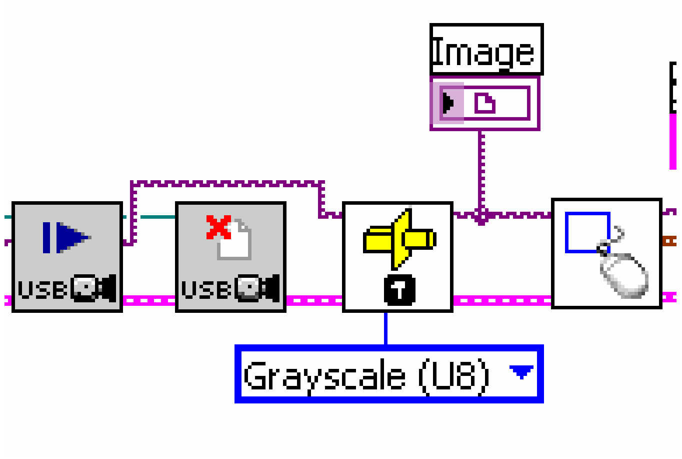
NOTE: This VI requires that you convert the image to grayscale.

IMAQ Cast Image

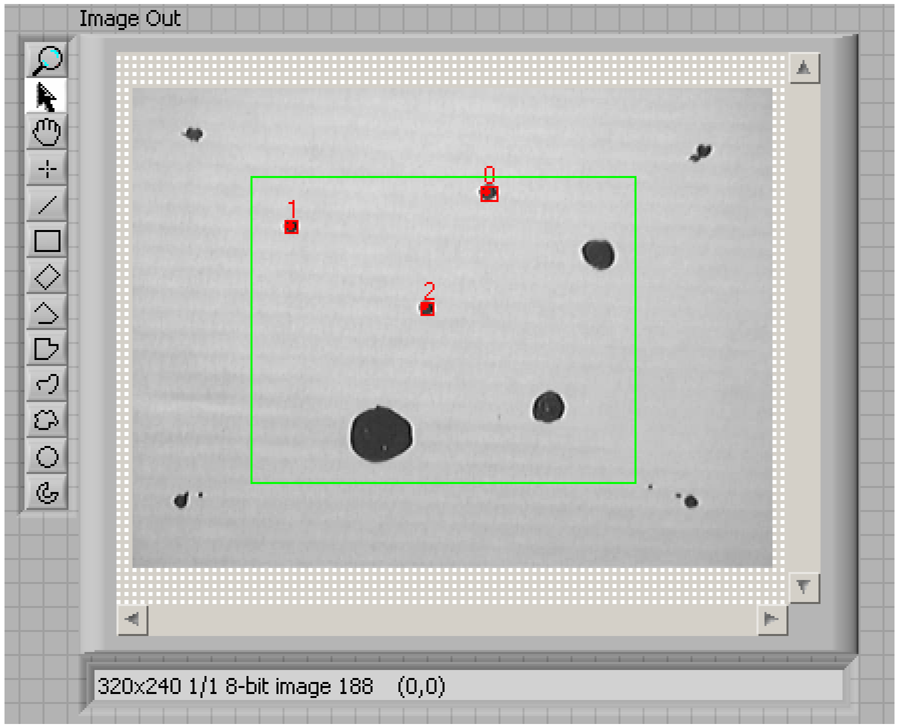


Converts the current image type to the image type specified by Image Type. If you specify a lookup table, IMAQ Cast Image converts the image using a lookup table. If converting from a 16-bit image to an 8-bit image, the VI executes this conversion by shifting the 16-bit pixel values to the right by the specified number of shift operations and then truncating to get an 8-bit value. Refer to the NI Vision Concepts Manual for more information about converting image types.

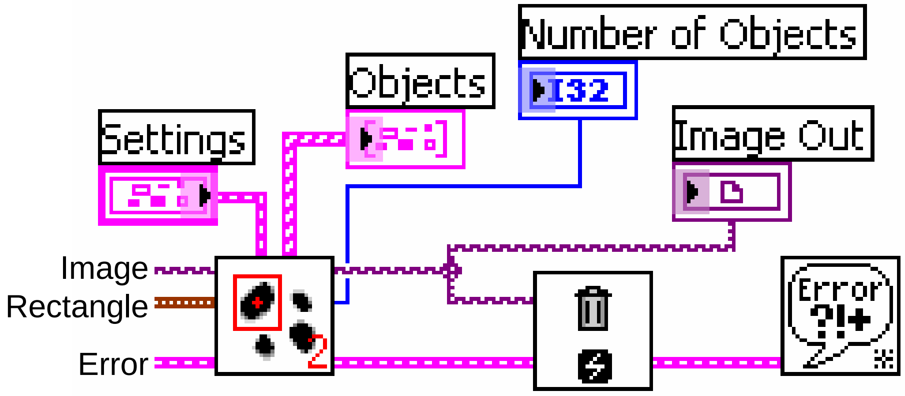
Example usage shown below:

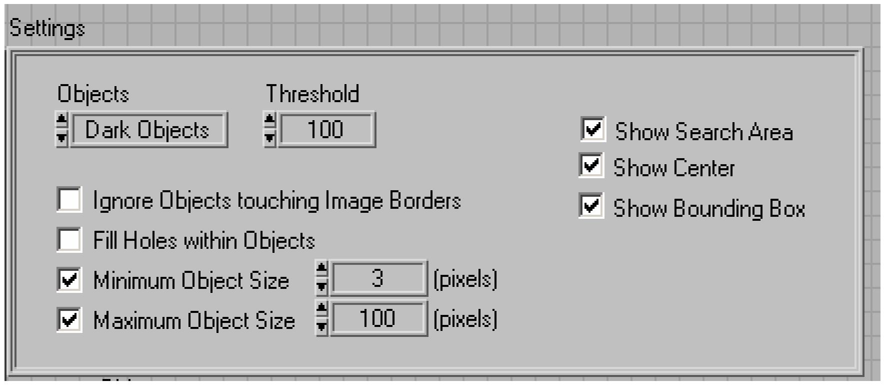


Example – Finding Objects and Intensities



3 objects detected





Settings

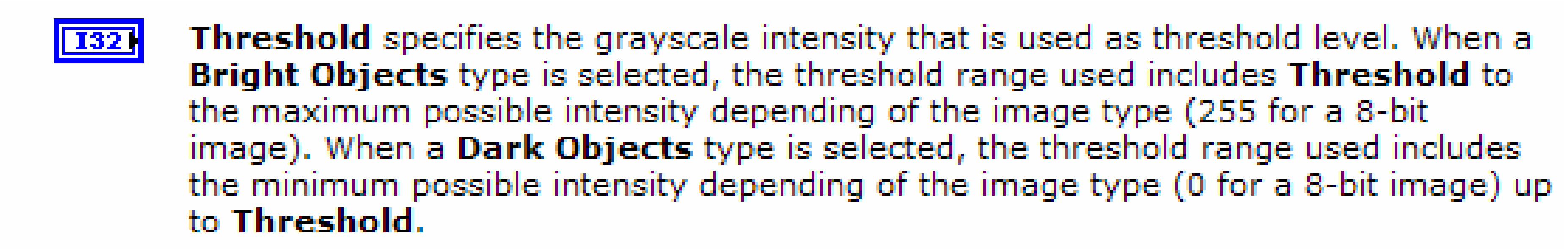
Objects: Dark Objects

Threshold: 100

- [ ] Ignore Objects touching Image Borders
- [ ] Fill Holes within Objects
- [x] Minimum Object Size: 3 (pixels)
- [x] Maximum Object Size: 100 (pixels)
- [x] Show Search Area
- [x] Show Center
- [x] Show Bounding Box

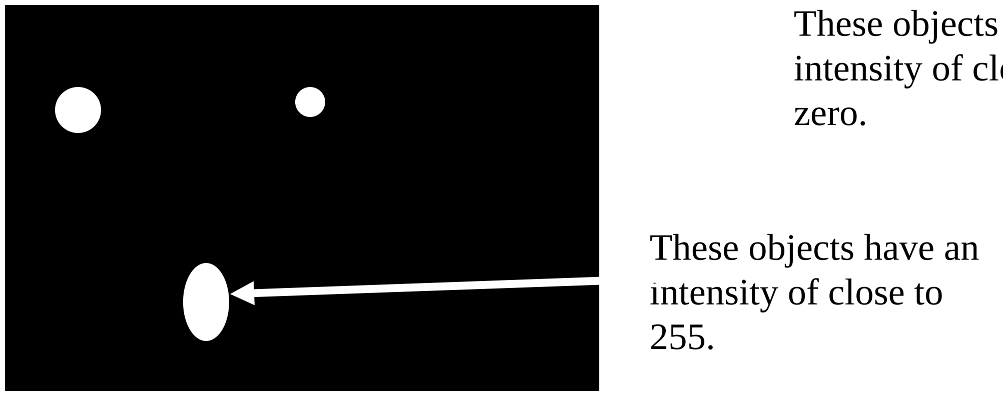
The limit on object size prevented the 3 larger objects in the ROI from being identified

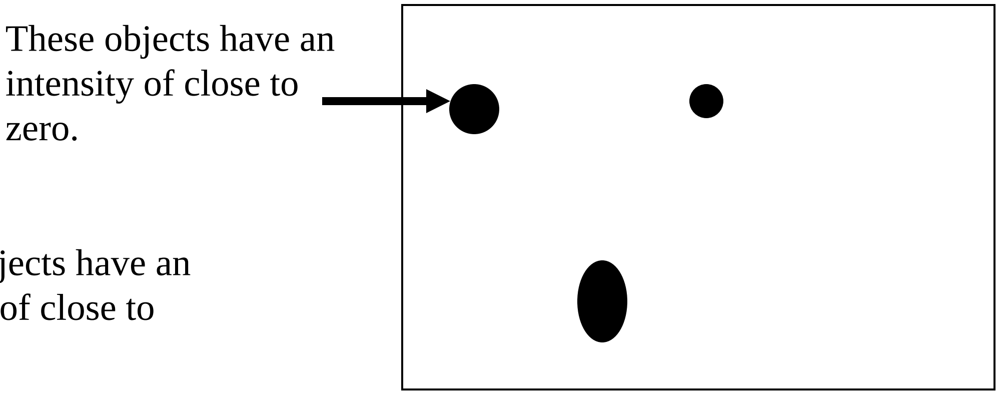
Define ‘Threshold’



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)

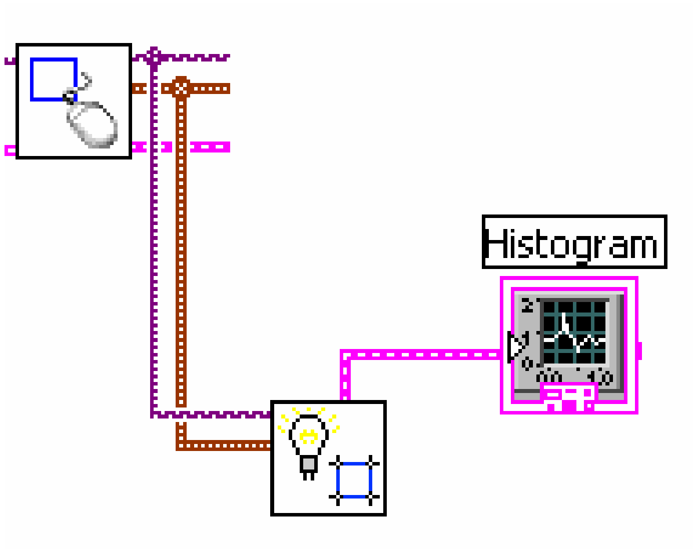




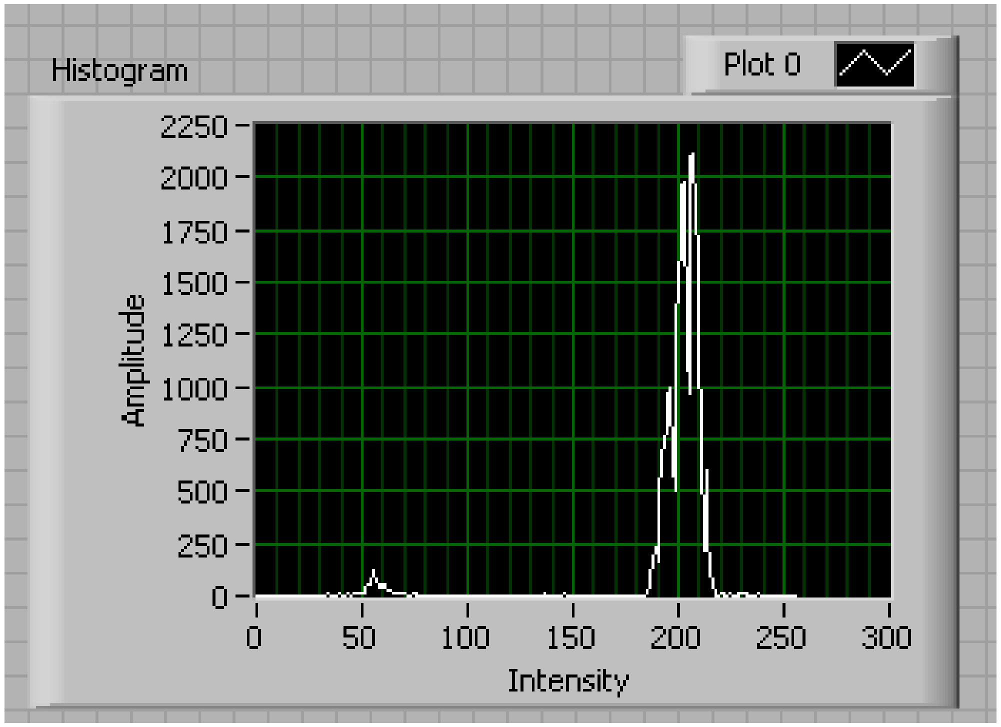
For 8-bit image

The ‘Threshold’ must often be specified as an input to some machine vision VIs.

Generating Intensity Histogram



Light Meter



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
- USB cameras* (web cams, etc.)
- Firewire cameras

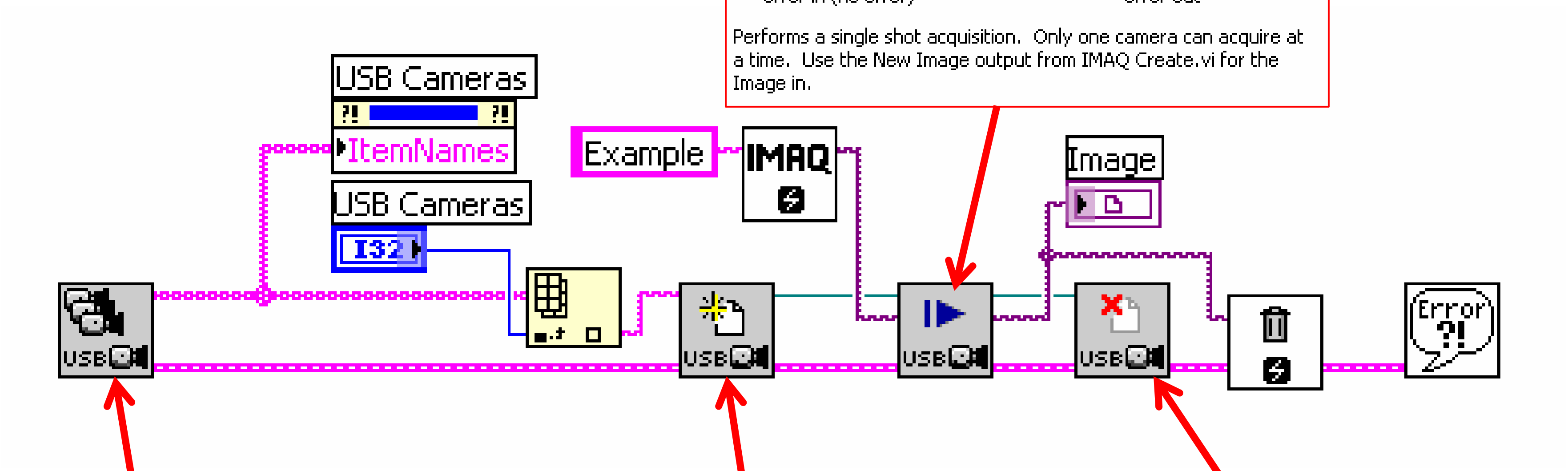
- USB is targeted for this course:
  - Low-cost
  - Easy to use
  - 'make it yours'

For DIYers, there are a lot of interesting websites on using 'webcams' for astrophotography, etc.

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



IMAQ USB Snap.vi

IMAQ USB Session In — IMAQ USB Session Out
Image in — Image out
error in (no error) — error out

Performs a single shot acquisition. Only one camera can acquire at a time. Use the New Image output from IMAQ Create.vi for the Image in.

IMAQ USB Enumerate Cameras.vi

USB Camera Names
error in (no error) — error out

Creates a list of all available USB cameras found on the system that can be initialized. Use the list with IMAQ USB Init to create an IMAQ USB session.

IMAQ USB Init.vi

USB Camera Name — IMAQ USB Session Out
Video Mode
error in (no error) — error out

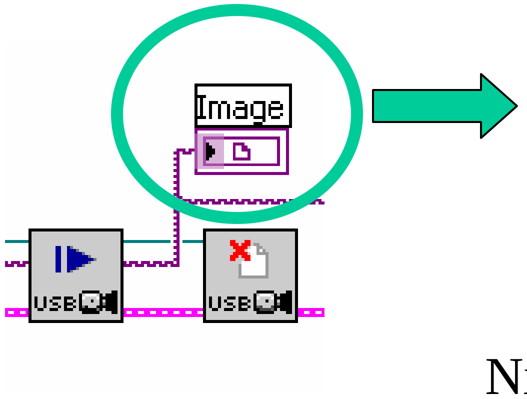
Creates an IMAQ USB session given the name of a USB camera. IMAQ USB Enumerate Cameras outputs a list of available cameras that can be initialized. The Video Mode paramater allows you to specify a particular acquisition mode of the camera. Use the IMAQ USB Property Page.vi with Video Mode to see the available Video Output formats the camera supports.

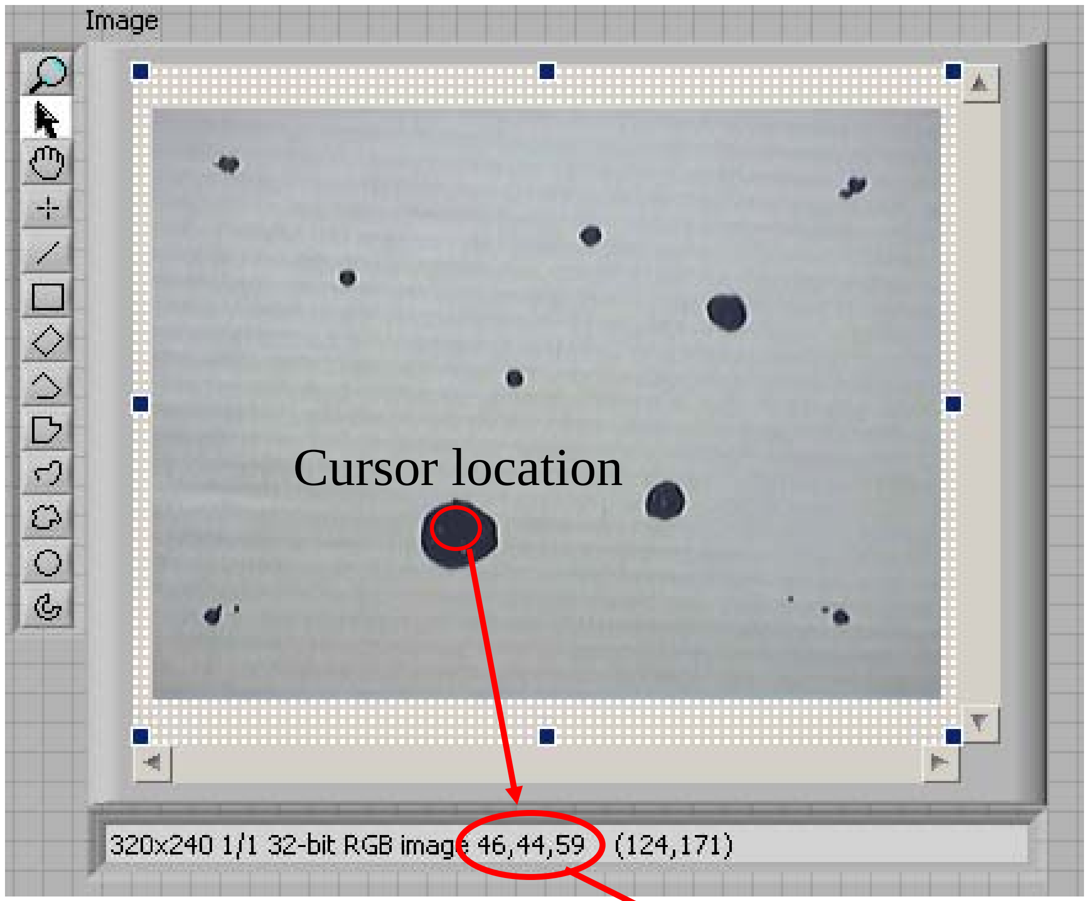
IMAQ USB Close.vi

IMAQ USB Session In
error in (no error) — error out

Closes a session to a USB camera that was opened with IMAQ USB Init.

Front Panel



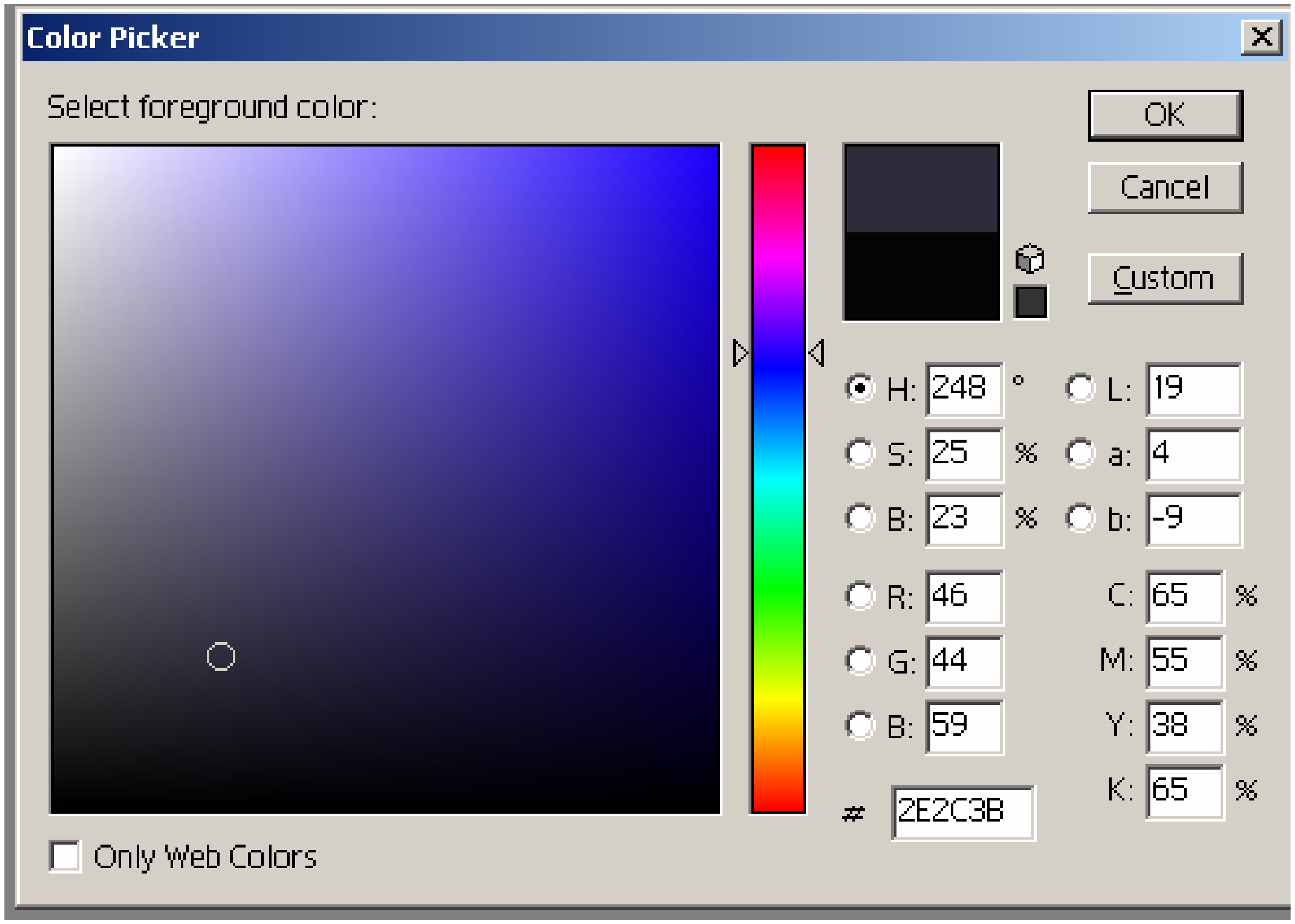


320x240 1/1 32-bit RGB image 46,44,59 (124,171)

NxM, 32-bit RGB image

RGB levels…

Just checking…in Photoshop


Color Picker

Select foreground color:

(● H: 248 °) (L: 19)
(S: 25 %) (a: 4)
(B: 23 %) (b: -9)
(R: 46) C: 65 %
(G: 44) M: 55 %
(B: 59) Y: 38 %
K: 65 %
2E2C3B

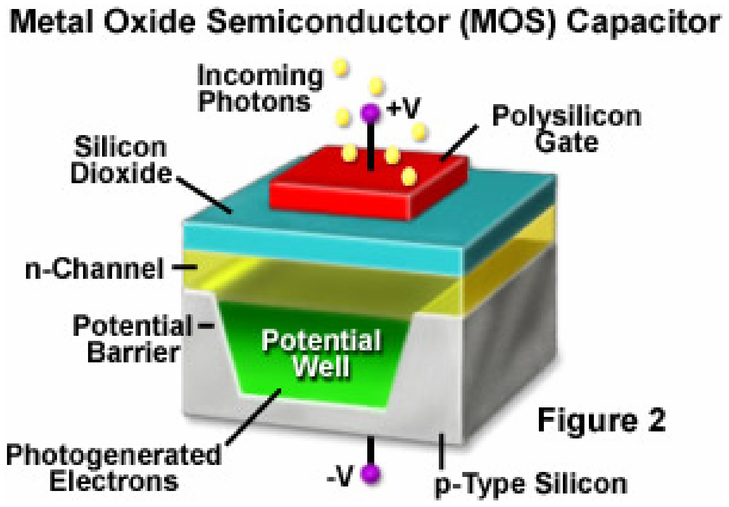
☐ Only Web Colors

About digital image capture…

- The following is a collection of slides on digital imaging technology collected from **various sources**, particularly using ‘webcams’**.**
- It is not necessary to understand *all* this information to use the webcams.
- **Do** make note of the difference between the two basic type of image ‘sensors’ used in modern webcams.
- If interested in ‘hacking’ webcams, check online for astrophotography with webcams.

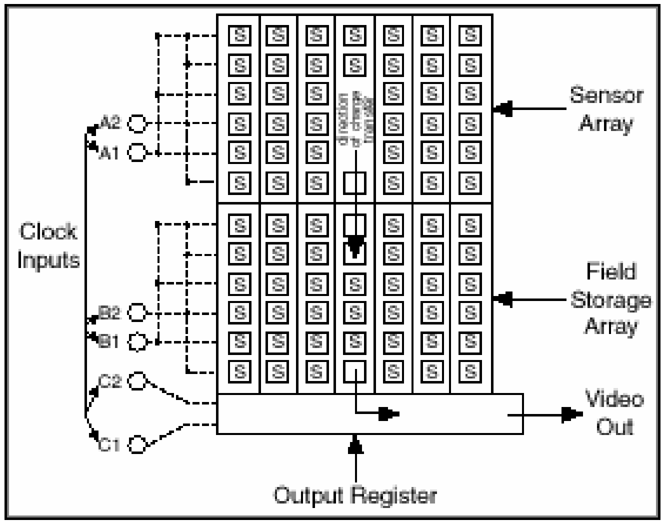
Solid-State Cameras

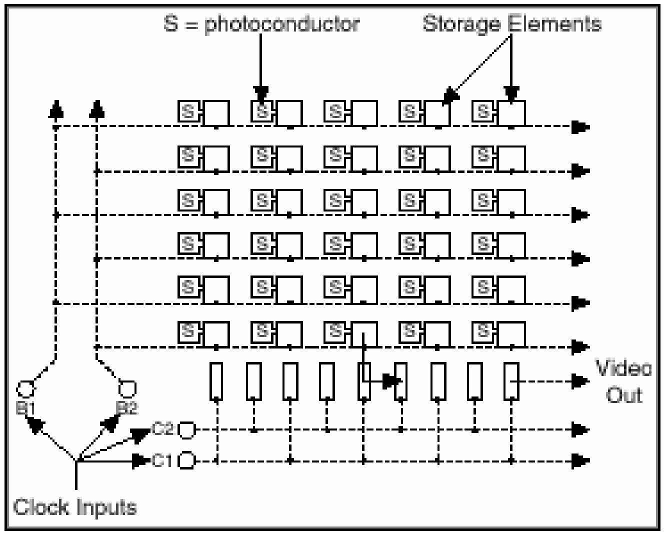
- A solid-state camera focuses incoming light onto a light-sensitive, solid-state sensor.
- These sensors are typically made up of a rectangular image array (imagers) or a single line (line-scan) of equi-spaced, discrete light-sensing elements, called photo-sites.
- Each photo-site acts as an optoelectric converter because it becomes electrically "charged" to a level directly proportional to the amount of light that strikes it during a given time period, called the integration time.



CCD – Charge-Coupled Device

- In a CCD, the charges that build up on all of the array photo-sites are linked or "coupled" together so that they can be transferred out of the array directly (digital output) or processed into a time-varying video signal (analog output).
- Two basic designs for transferring the accumulated charges out of a CCD array: frame-transfer (FT) shown right (top) and interline-transfer (IT) shown right (bottom).



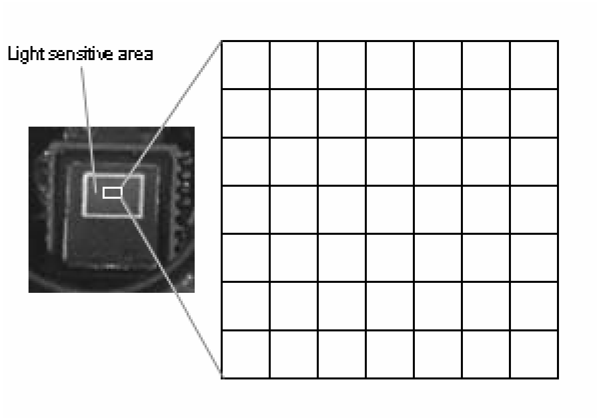


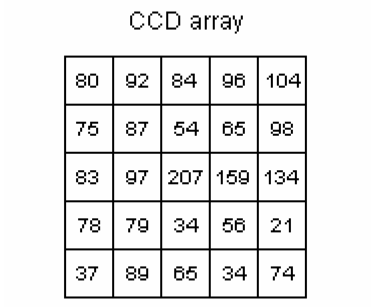
Some history…

In 1969 Willard S. Boyle and George E. Smith, while working at Bell Laboratories, designed the first Charge Coupled Device (CCD), a working version was produced just a year later.

A more basic description

A CCD (Charge Coupled Device) chip is a light sensitive device, made of silicon. It is an array of light sensitive pixels. When light falls on a pixel, it will be converted to a charge. This charge is captured in the pixel. It can’t go to other pixels. The more light falls on the pixel, the more charge will accumulate in the pixel. The amount of charge is a measure of the amount of light that felt on the pixel.




CCD array

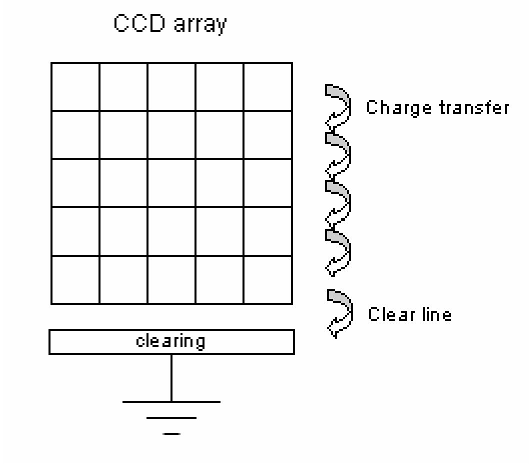
80	92	84	96	104
75	87	54	65	98
83	97	207	159	134
78	79	34	56	21
37	89	65	34	74

The amount of charge in every pixel makes the image. The computer can read this charge and convert it to an image.

CCD clearing phase

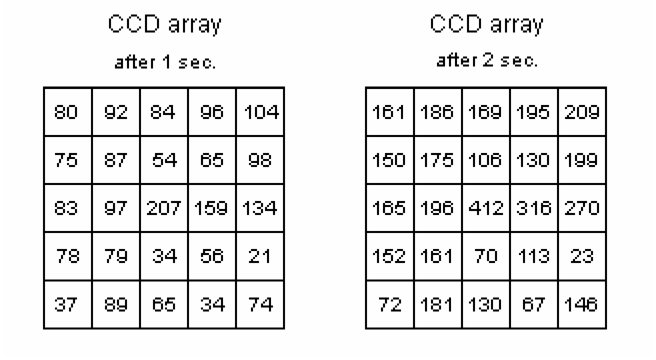
There are three main phases:

1. Clearing phase. There is no shutter in a digital camera, so there is always light falling on the CCD chip. An image is 'captured' by taking all existing charge from the pixels. The CCD chip can shift lines downward with electrical pulses. All charge in a line will move to the line below of it. This happens with all lines of the chip at the same time. The lowest line will move the charge to a 'clearing line', where it will be removed. When all lines are shifted away, there is no charge left anymore. This phase takes a few milliseconds.



CCD exposure phase

Exposure or integration phase
When exposed to an image, a CCD chip accumulates charge in the pixels. The longer you wait, the more light will be registered by the chip. This phase controls the exposure time, also called: integration time, because the chip integrates the amount of light in a certain period. With lunar and planetary photography it usually takes a few tenths of a second, but it can take minutes or even hours.



CCD array
after 1 sec.

80	92	84	96	104
75	87	54	65	98
83	97	207	159	134
78	79	34	56	21
37	89	65	34	74

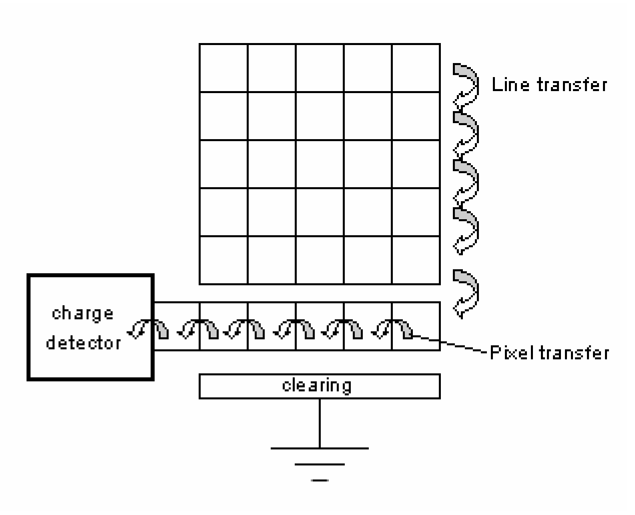
CCD array
after 2 sec.

161	186	169	195	209
150	175	106	130	199
165	196	412	316	270
152	161	70	113	23
72	181	130	67	146

The CCD array after 1 and 2 seconds exposure. The charge in the pixels accumulated with a factor of two.

CCD readout phase

Readout phase
To get the image in your computer, it will have to read the amount of charge of every pixel. There is a device, the charge detection node, in the CCD chip that is able to measure the charge of a pixel. Every pixel must be shifted in this device. The lowest line is able to shift pixels to the left in the direction of the device. So these pixels can be measured one by one.



When all pixels of the line are measured, all lines are shifted one line down. Then again the lowest line can be measured. This will be repeated until all lines are measured. The computer can read the measurements through the camera electronics. This phase can take a few seconds.

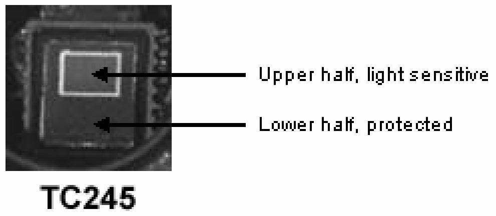
When the computer has collected all measurements, it can show an image of it.

Details on the CCD chips

The most used types of CCD chips:

1. Full frame device. This is the most simple device as described above. It is used in the CB211 camera. This type can't be used with lunar and planetary photography without a shutter. Making images of this kind of objects typically requires an integration time of a few tenths of a second. The problem is the readout time, which can take a few seconds. During readout, the chip is still receiving light, which will destroy the original image, before it's completely read by the computer.

2. Frame transfer device. The CCD chip in the CB245 camera has a TC245 CCD chip from Texas Instruments, this is a frame transfer device. With this kind of chip, the lowest half of the chip is protected against light. The image is only made on the upper half of the chip. When you are finished making your image, the image will be transferred from the upper half to the lower half, where it will be protected from light. This will take only a few ms. There is enough time, in this area, to read the image. A frame transfer device can be used with lunar and planetary photography.



The TC245 is divided in 2 parts.

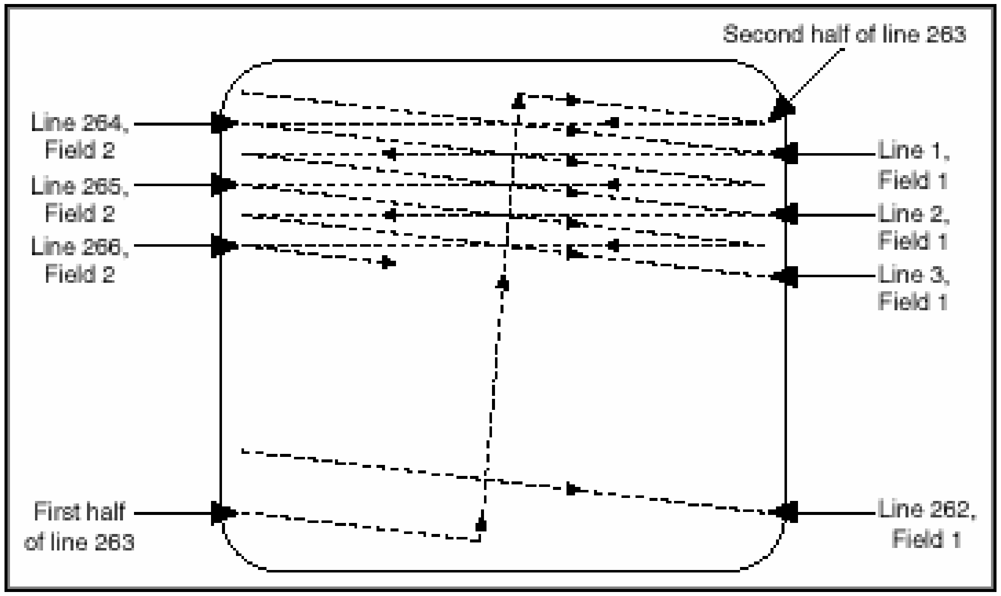
3. Interline transfer device. One line of every two lines is protected from light. During integration, only one line receives light. At the end, this line is shifted to the protected line, where it will be read by the computer. An interline transfer device can be used with planetary photography, Lunar photography will be more difficult, because of it's brightness. The bright light can influence the protected line. The protected area of a frame transfer device is safer.

CMOS Designs

- CMOS designs differ from CCD designs in the way the sensors are manufactured and how the charges are read out of the array.
- CMOS sensors are manufactured on standard semiconductor production lines, whereas CCDs are manufactured on dedicated production lines. This advantage allows CMOS sensor designers to leverage the constantly improving manufacturing capabilities of standard CMOS production lines (larger wafers, smaller feature sizes) to incorporate onboard signal processing functionality at a fraction of the cost of CCDs.
- This manufacturing advantage allows CMOS sensors to be more tightly integrated into entire camera assemblies on a single production line than CCDs. However, CMOS sensors traditionally suffer from poor dynamic response when compared to CCDs.
- A definite design advantage that CMOS sensors have over CCDs is that each photo-site in a CMOS sensor array can have its own amplifier and output circuitry. This allows each photo-site to be read out independently of the other photo-sites and eliminates the need for charge-coupling. It also gives CMOS sensors the ability to have much better onboard programmability for more efficient read-outs and easier integration with “off-the-shelf” digital signal processors for more powerful and smaller overall camera design.

Scanning Techniques

- Television takes advantage of a measurable threshold in people's ability to sense movement of images over time, or flicker.
- Flicker is minimized by interlacing or stitching two successive images read from a standard camera.
- 2:1 interlace is described as follows:
  - Read/display all even-numbered lines (even field, half-size)
  - Restart
  - Read/display all odd-numbered lines (odd field, half-size)
  - Stitch the even and odd fields together and form a single, full-size frame
  - Output the full-size frame



2:1 Interlaced scanning, shown here for the NTSC video format. PAL and SECAM interlacing are similar, with the difference being the number of lines in each field.

Interlace Scanning

- Solid-state camera manufacturers design **2:1 interlace** scanning into their cameras so that their cameras can be connected directly to standard televisions and monitors. There are two variations.
- Frame integration – only the even or odd field is repeatedly read from the image sensor, while the other field is discarded. This results in a full-size frame that contains only the even or odd field available on the image sensor.
- Field integration – pairs of consecutive frame lines are read from the image sensor simultaneously, using all frame lines in the overall scan. For example, lines 1 and 2 would be combined and read as line 1, lines 2 and 3 would be combined as line 2, and so on, until a full-size frame is constructed.
- Interlacing requires a finite amount of time to produce a final full-size frame, so if any movement occurs in a scene between the time of the scans, the final frame will appear blurry. New camera designs employ **progressive-scanning** techniques to offset this effect.

Progressive Scanning

- Progressive-scan cameras operate by immediately transferring an entire frame at once from the image sensor without any performing any line-interlacing.
- The obvious benefit of this design is in motion detection applications, but the down side is incompatibility with standard television systems.
- However, computer-based image processing applications do not require that raw camera images be displayed on a television monitor. Rather, the camera is used as the sensor and input device to a plug-in acquisition board inside a computer with a built-in display board that can handle the output of the acquisition board directly.
- Progressive-scan camera designs are gaining in popularity for these computer-based applications, because they incorporate direct digital output and eliminate many other time-consuming and unnecessary processing steps associated with television systems.

Digital vs. Analog Cameras

- Digital cameras have several technical advantages over analog cameras.
- By digitizing at the CCD camera rather than at the image acquisition board, the signal-to-noise ratio is typically higher, which results in better image resolution.
- Many digital cameras now come with 8- to 12-bit gray levels of resolution as a standard feature. This higher image resolution is often required in medical, machine vision, astronomy, and thermal imaging applications.

References

- “Anatomy of a Camera”, National Instruments, http://zone.ni.com/devzone/cda/tut/p/id/2703