EXPERIMENT DSP3: IMAGE PROCESSING

Related course: KIE3007 (Digital Signal Processing)

OBJECTIVES:
To execute program of image processing

EQUIPMENT:
PC equipped with Labview Software

INSTRUCTIONS:
For each test, student is required to print screen final block diagram and final front panel, and save them in a file.

REFERENCE(S):
Refer to the main references of KIE3007

EXERCISE:
Exercise 1: Image Properties
Exercise 2: Bit-Depth
Exercise 3: Re-Sampling
Exercise 4: Basic Image Processing
Exercise 5: Scalar Arithmetic Image Processing

INTRODUCTION:

Digital Images

Images are a way of recording and presenting information in a visual form. Images can be thought of as pictures but an image can correspond to any kind of 2D data. Digital image processing refers to images being manipulated by digital means, commonly computers. The natural form of images is not suitable for processing by computers because computers cannot operate directly on pictorial data but require numerical data. Hence, images need to be converted into numerical data, called digital images, to enable computer manipulation.

A digital image corresponds to an array of real or complex numbers represented by a finite number of bits, showing visual information in a discrete form. In some cases, digital images might be directly synthesized in discrete form, but commonly they are translated from a physical image. The translation from a physical image into an appropriate digital form is accomplished by an analog to digital converter (ADC) that performs sampling and quantization.

Sampling

Sampling is the process of measuring the value of the physical image at discrete intervals in space. Most commonly, a rectangular sampling grid is employed. Each image sample corresponds to a small region of the physical image, called a *picture element* or *pixel*. The sampling process is shown in Figure 1. Thus, a digital image corresponds to a two-dimensional array of pixels.

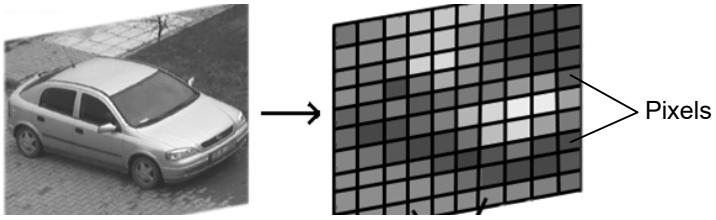

Figure 1: A physical image and the sampling process

Quantization

Quantization is the process of replacing the continuous values of the sampled image with a discrete set of quantization levels. The number of quantization levels employed governs the accuracy by which the image values are displayed. Conventionally, *L* quantization levels are defined by integers from 0 to *L*-1, where 0 corresponding to black and *L*-1 corresponding to white, with intermediate levels representing various shades of grey. All grey levels ranging from black to white are referred to as **grayscale**. Grayscale images do not convey any colour information. For convenience and efficiency, the number of quantization levels is usually an integral power of two $L = 2b$, so that that a total of *b* bits may use to represent *L* different quantization levels. The number of bits used to define a pixel value is called the **bit depth**. The greater the bit depth, the greater the number of quantization levels and thus the tones that can be represented in the digital image. An image represented by only 1 pixel with 0 representing black and 1 representing white is called a bi-tonal or binary image. Typically, a bit depth of 8 bits is employed in digital imaging so that 256 possible grey levels ranging from 0 (black) to 255 (white) are used as pixel values. This is shown in Figure 2.

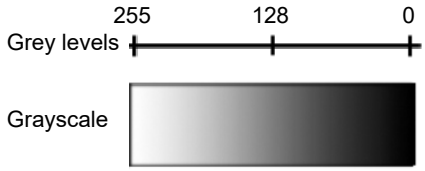

Figure 2: 256 different gray levels constructed for a bit depth of 8

Figure 3 shows an image at bit depths of 1 bit/pixel, 4 bits/pixel and 8 bits/pixel. 1 bit/pixel is insufficient to display image detail but at 4 bits/pixel, the visual appearance is reasonable.

Figure 3: Effect of various bit depths on visual appearance

Exercise 1: Image Properties

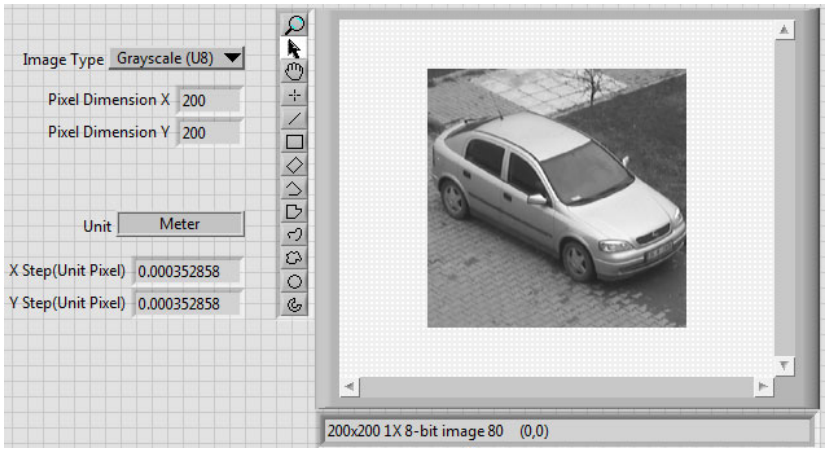

Figure 4: Front Panel of Exercise 1

1. Create a **Front Panel** as illustrated in Figure 4. First, from **Control>>Vision**, place an **Image Display.**
2. From **Control>>Modern>>Numeric**, place four **Numeric Indicators.** Name them as **Pixel Dimension X, Pixel Dimension Y, X Step(Unit/Pixel)** and **Y Step(Unit/Pixel).**
3. For the **Block Diagram,** wire all the controls and indicators as illustrated in Figure 5 by executing the following steps.

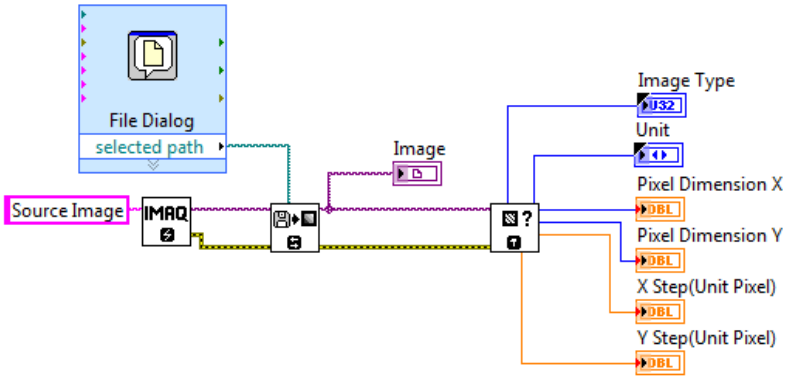

Figure 5: Block Diagram of Exercise 1

4. Place an **IMAQ Create** located on **Function>>Vision and Motion>>Vision Utilities>>Image Management.** This function palette will create a temporary image location for an image.
5. Move cursor to **Image NameNode** of the palette, right click and go to **Create>>Constant.** Name the **Constant** as **Source Image.** *(Check the nodes of the palette by enabling the **Context Help.** Go to **Help** and check **Show Context Help**).*
7. Place **IMAQ Read File** located on **Function>>Vision and Motion>>Vision Utilities>>Files.** This function palette will read a saved image file. Wire the **New Image** Node of **IMAQ Create** with **Image Node** of **IMAQ Read File.**
8. Place a **File Dialog Express VI** located on **Function>>Programming>>File I/O>>Advance File Function**. Leave the configuration as default and click **OK.** Wire the **Selected Path Node**of **File Dialog** with **File Path Node** of **IMAQ Read File.**
9. Place an **IMAQ Get Image Info** located on **Function>>Vision and Motion>>Vision Utilities>>Image Management.** Wire the **Image Out Node** of **Read File** with both **Image Node** of **Get Image Info**and **Image Display Indicator.**
10. Check the output nodes of **Get Image Info.** Wire the **Indicators** that you have dropped on **Front Panel** to the dedicated nodes (E.g: **Unit Nodes** to **Indicator** with **Unit** label). Move Cursor to **Image Type Node,** right click and select **Create>>Control.** Repeat the same step for **Unit Node.**
11. **Run** the program. The file dialog will appear, asking for an image file to be selected. LabVIEW accepts standard image files in BMP, TIFF, JPEG, PNG and AIPD formats.
12. Select an image file in one of these formats (due to program restriction, the file will be processed as 8-bit grayscale even if a colour image is selected).
13. The image will appear in the **Display Window**. The program panel will show the file format, pixel dimensions and image resolution (if specified). Note that the image resolution is not specified for all images. Standard image formats include an image header previous to the visual information. The image header gives side information about the image, i.e. the image format, image dimensions and image resolution.
14. Save the VI file as **Exercise 1**.

Exercise 2: Bit-Depth

1. Open **Exercise 1** and add a **Numeric Control** and name it as **Bit-depth.**Change the default value of **Bit-depth** to 8 as shown in Figure 6.

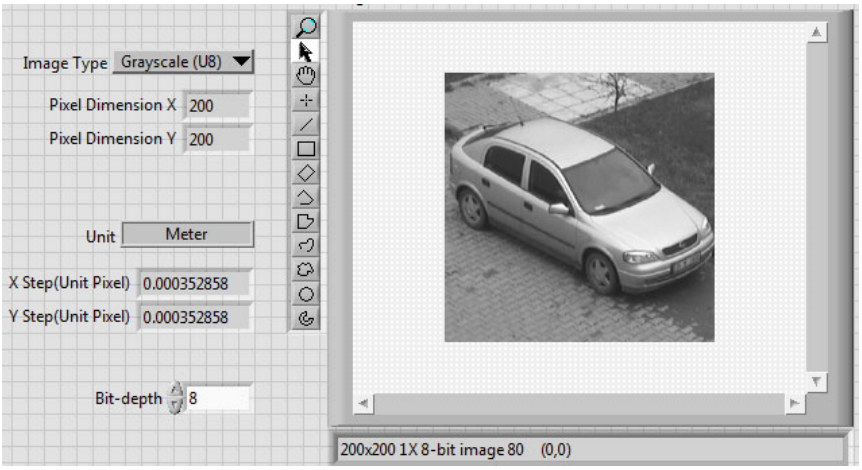

Figure 6: Front Panel of Exercise 2

2. Modify the **Block Diagram** as shown in Figure 7 by executing the following steps.

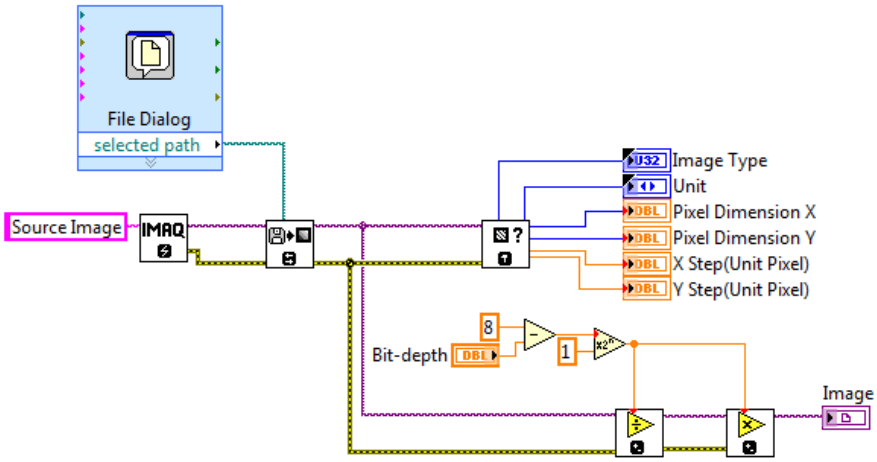

Figure 7: Block Diagram of Exercise 2

3. Place **Image Divide** and **Image Multiply** located on **Function>>Vision and Motion>>Image Processing>>Operators.** Wire the **Image Out Node** of **Read File** with **Image Node** of **Image Divide.** Then, wire the **Image Out Node** of **Image Divide** with **Image Node** of **Image Multiply.**
4. As shown in Figure 8, place **Subtract** and **Scale By Power of 2** located on **Function>> Mathematics>> Numeric**. Then, wire them.

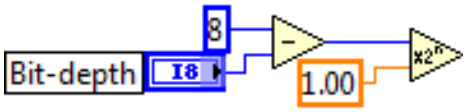

Figure 8: Wiring for Mathematical Function

5. Wire the **Output Node** of **Scale By Power of 2** with both **Constant Nodes** of **Image Divide and Multiply.**
6. Wire the **Image Out Node** of **Image Multiply** with **Image Display.**

7. Run the VI. A file dialog will appear, asking for an image file to be selected. Select an image file in one of the formats supported by LabVIEW. The image will appear in the display window.
8. Now, change the bit-depth of the image and examine changes in visual appearance. Observe that there is no obvious change between a bit-depth of 8 and a bit-depth of 7, while differences become visually significant when the bit-depth is further reduced.
9. Save the VI file as **Exercise 2**.

Exercise 3: Re-Sampling

1. Open **Exercise 2** and modify the **Front Panel** as shown in Figure 9 and Figure 10.
2. From **Control>>System>>Container**, add a **System Tab**.
3. Place controls of each exercise on different pages. To add new page, right click on the **Tab** and click **Add Page After.**
4. On the third page, add two Numeric Controls and name each as **New X Resolution** and **New Y Resolution**. Add a **Push Button** located on Control>>Modern>> Boolean. Name the button as **Resample**. Place a **Stop Button** outside the Tab Control.

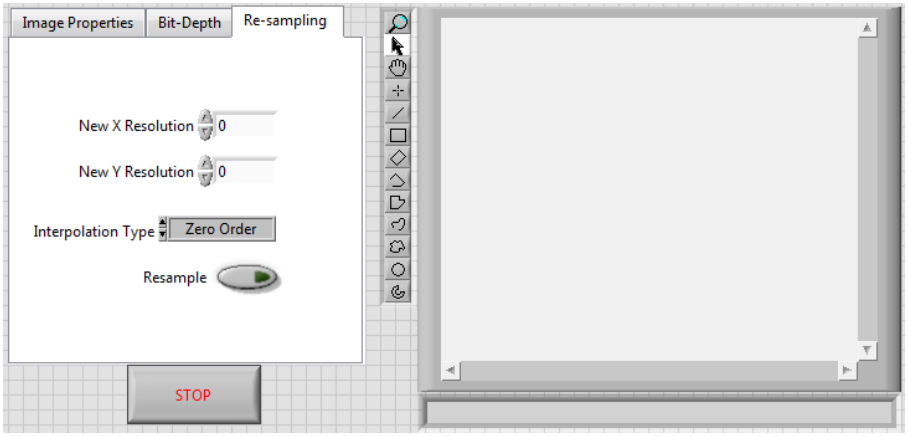

Figure 9: Front Panel of Exercise 3

5. Modify the **Block Diagram** as shown in Figure 10 by executing the following steps.

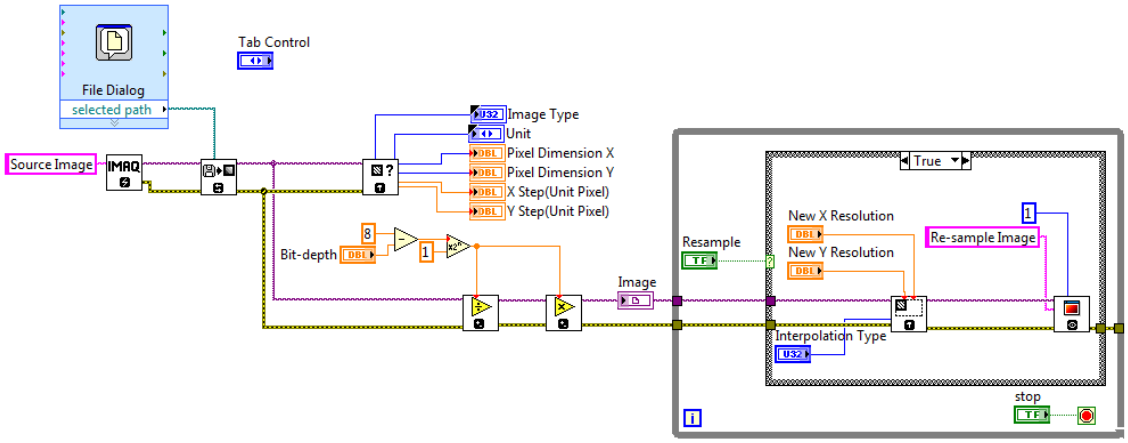

Figure 10: Block Diagram of Exercise 3

6. From **Function>>Programming>>Structures**, add a **While Loop.** Wire the **Stop Button** with the **Loop Condition.**
7. Place a **Case Structure** inside the **While Loop.** Wire the **Resample Controller** to the **Selector Terminal** of **Case Structure.**
8. Place **IMAQ Resample** located on **Function>>Vision and Motion>>Vision Utilities>>Image Manipulation**.
9. Place **IMAQ WindDraw** located on **Function>>Vision and Motion>>Vision Utilities>>External Display**.
10. Place both **IMAQ Resample** and **IMAQ WindDraw** in the 'True' **Case Structure.**
11. Move cursor to the **Interpolation Type Node** of the **IMAQ Resample.** Right click and go to **Create>>Control.** Move the **Control** into **Tab** on **Front Panel.**
12. Move cursor to the **Window Number Node** of the **IMAQ WindDraw.** Right click and go to **Create>>Constant.** Change the **Constant** to '1'.
13. Move cursor to the **Title Node** of the **IMAQ WindDraw.** Right click and go to **Create>>Constant.** Change the **Constant** to 'Re-sampled Image'.
14. Wire the rest as illustrated in Figure 11.

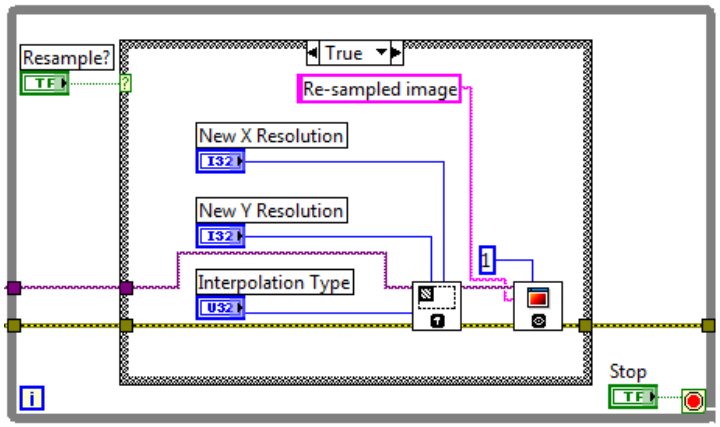

Figure 11: Block Diagram for "True" Case Structure

15. Refer to Figure 12 for "False" **Case Structure** wiring.

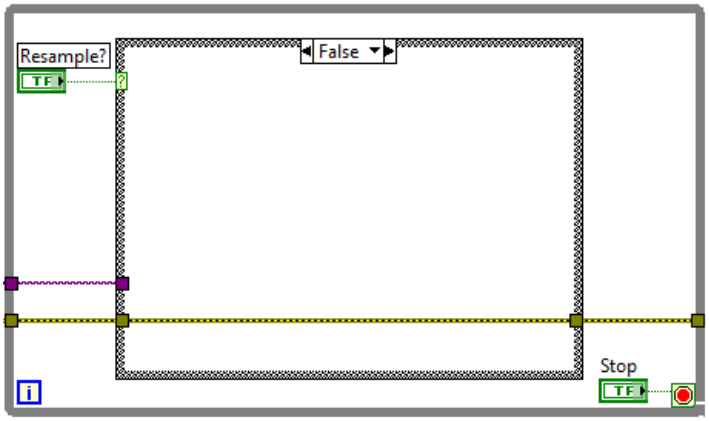

Figure 12: Block Diagram for "False" Case Structure

16. Wire the **Image Out Node** of **Image Multiply** to the **Image Node** of the **While Structure.** Wire the **Error Nodes** as well.

17. Run the VI. The file dialog will appear, asking for an image file to be selected. Select an image file in one of the formats supported by LabVIEW.
18. The image will appear in the display window and the original pixel dimensions of the image will be displayed. Change the pixel dimensions of the image and examine the visual appearance for various interpolation schemes.
19. Observe that blocking artefacts occur for zero-order interpolation if the pixel dimensions are significantly increased, while for other interpolation schemes new pixels are smoothly interpolated so that the image may seem blurry, however blocking is avoided.
20. Save the VI file.

Basic Image Processing

Image processing refers to the procedure of manipulating images. Commonly, image processing is carried out in digital domain by computers. Digital image processing covers many different techniques to change the properties of an image. On the easiest level, image processing can be accomplished by changing the physical location of the pixels of an image.

It is possible to take the symmetry of an image by reversing the pixels according to a symmetry location. Figure 13 shows the result of a vertical symmetry and horizontal symmetry. For vertical symmetry, the process can be thought of image pixels being upturned with respect to a vertical line. If image1[*x*][*y*] represents the pixel value of the original image at row *x* and column *y*, and the image dimensions are given as *width* × *height*, vertical symmetry can be formulated as image2[*x*][*y*]= image1[*width*-*x*][*y*] (0 ≤ *x* <*width* and 0 ≤ *y* <*height*). For horizontal symmetry, the resultant image is given by image2[*x*][*y*]= image1[*x*][*height*-*y*]. It is possible to increase the number of possible symmetries by combining vertical and horizontal symmetries and interchanging row and columns of pixels.

Figure 13: Image symmetry original

Arithmetic Image Processing

Basic image processing changes only the location of image pixels, i.e. takes the pixels of an image and moves them to another location. Another way of manipulating an image is to carry out arithmetic operations on image pixels. It is possible to add/subtract an integer to/from pixel values or multiply or divide image pixels by a constant value. Figure 14 shows the effect of adding 50 to the values of all pixels of an image. The resultant image appears brighter as pixel values are moved towards the white level. When 150 is added to the values of all pixels of the same image, the image appears very bright. This is due to the pixel values exceeding the representation range (255 for this 8-bit image) are truncated at the highest value, resulting in dominant white regions and loss of image detail.

Figure 14: Scalar arithmetic

Exercise 4: Basic Image Processing

1. Build a **Front Panel** as shown in Figure 15. First, add a **Tab Control** with 3 pages as **Symmetry, Shift and Rotate**.

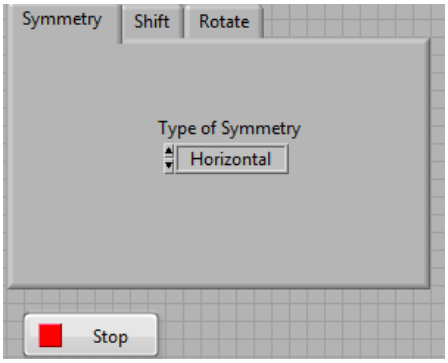

Figure 15: Front Panel of Example 4

2. On **Shift Page,** place two **Numeric Controls** and name them as **Horizontal Shift** and **Vertical Shift.**
3. On **Rotate Page**, place a **Numeric Control** and name it as **Angle(degrees).**
4. Place **Stop Button** below the **Tab Control.**
5. Build a **Block Diagram** as illustrated Figure 16 by executing the following steps.
6. From **Function>>Vision and Motion>>Vision Utilities >>Image Management**, place an **IMAQ Create**. This function will create a temporary image location for an image.
7. Move cursor to **Image NameNode** of the palette, right click and go to **Create>>Constant.** Name the **Constant** as **Source Image.** *(Check the nodes of the palette by enabling the **Context Help.** Go to **Help** and check **Show Context Help**).*
8. Place **IMAQ Read File** located on **Function>>Vision and Motion>>Vision Utilities>>Files.** This function palette will read a saved image file. Wire the **New Image** Node of **IMAQ Create** with **Image Node** of **IMAQ Read File.**

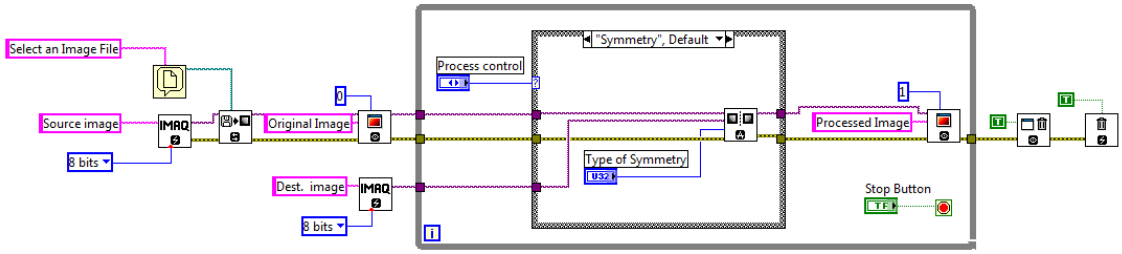

Figure 16: Block Diagram of Example 4

9. Place a **File Dialog Express VI** located on **Function>>Programming>>File I/O>>Advance File Function**. Leave the configuration as default and click **OK.** Wire the **Selected Path Node** of **File Dialog** with **File Path Node** of **IMAQ Read File.**
10. Place **IMAQ WindDraw** located on **Function>>Vision and Motion>>Vision Utilities>>External Display**.
11. Move cursor to the **Window Number Node** of the **IMAQ WindDraw.** Right click and go to **Create>>Constant.** Change the **Constant** to **0**.
12. Move cursor to the **Title Node** of the **IMAQ WindDraw.** Right click and go to **Create>>Constant.** Change the **Constant** to **Original Image**.
13. Add a **While Loop** and place a **Case Structure** inside the loop. Wire **Tab Control** with the **Selector Node.**
14. Place a **IMAQ Resample** in the **Symmetry Case,** located on **Function>>Vision and Motion>>Vision Utilities>>Image Manipulation.**
15. Move cursor to **Type of Symmetry Node** and create **Control.** Make sure the **Control** on the **Front Panel** is on the **Symmetry Page** of the **Tab Control**. Wire the rest of **Symmetry Case** as shown in Figure 17.

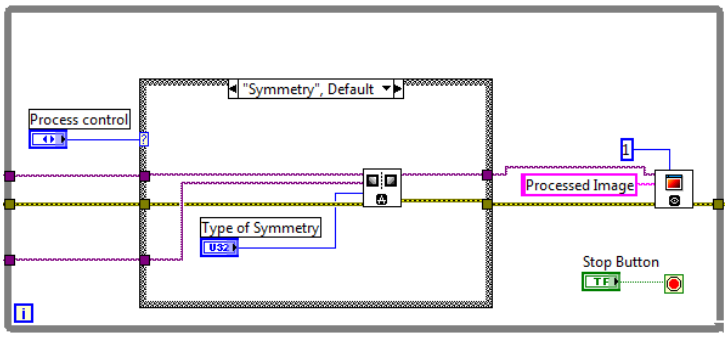

Figure 17: Wiring for Symmetry Case

16. From **Function>>Vision and Motion>>Vision Utilities>>Image Manipulation**, place a **IMAQ Shift** in the **Shift Case**. Wire the rest of **Shift Case** as shown in Figure 18.

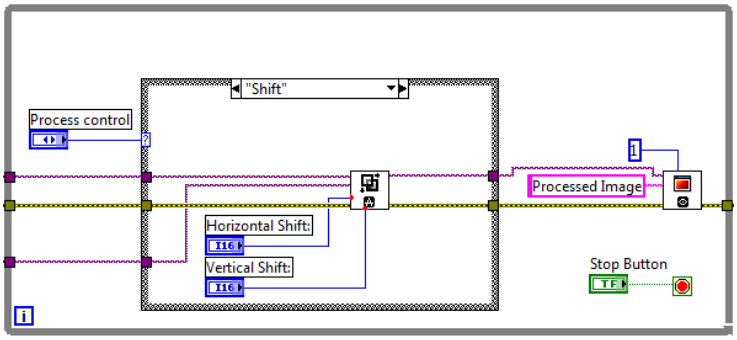

Figure 18: Wiring for Shift Case

17. Place **IMAQ Rotate** in the **Rotate Case,** located on **Function>>Vision and Motion>>Vision Utilities>>Image Manipulation.**
18. Wire the rest of **Rotate Case** as illustrated in Figure 19.
19. To complete the **Block Diagram,** add **IMAQ Dispose** located on **Function>>Vision and Motion>>Vision Utilities>>Image Management.** Additionally, add **IMAQ WindClose** located on **Function>>Vision and Motion>>Vision Utilities>>External Display.** Place both of them outside **While Loop.**

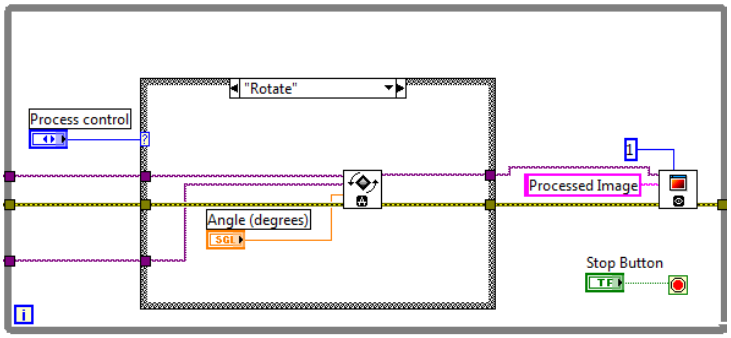

Figure 19: Wiring for Rotate Case

20. Complete the wiring as illustrated in Figure 16.
21. Run the program. It will ask for an image file. In the symmetry tab, choose between "horizontal", "vertical", "1st diagonal", "2nd diagonal" and "central" symmetry. Hit the "take symmetry" button. The image with the corresponding symmetry will appear.
22. You can choose to take the symmetry with respect to the original image or the already processed image so that you can conduct a sequence of different processes. In the **shift tab**, define the horizontal and vertical shift amount in pixels and hit the "shift" button. The processed image will appear. Then, choose the source image as either the original image or the already processed one. In the **rotate tab**, enter the angle in degrees and hit the "rotate" button. The rotated image appears. It is possible to save the processed image from within the save image tab.
23. Save the VI file.

Exercise 5: Scalar Arithmetic Image Processing

1. Build a **Front Panel** as shown in Figure 20. Add a two **Boolean Buttons** and name each as **Load Image** and **Process.** Add a **Numeric Control** and name it as **Scalar.**

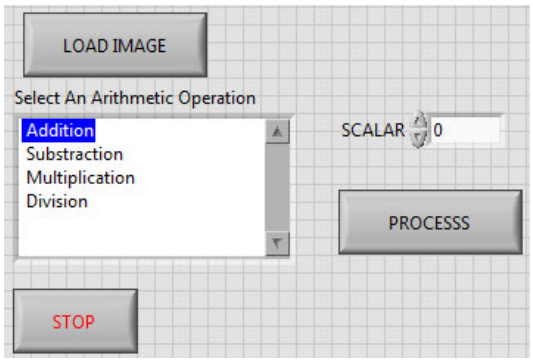

Figure 20: Front Panel of Exercise 5

2. From **Control>>Modern>>List, Table&Tree**, add a **List Box**. Add items by double click on rows. Add four items: **Addition, Subtraction, Multiplication** and **Division**.
3. Add a **Stop Button.**
4. Build a **Block Diagram** as shown in Figure 21 by executing the following steps.
5. Add a **Case Structure** and wire the **Load Image Button** to the **Selector Node.** Build the rest as shown in Figure 22. Refer to previous exercise if you do not remember the location of the palettes.
6. Add another **Case Structure** and wire the **Process Button** to the **Selector Node.**

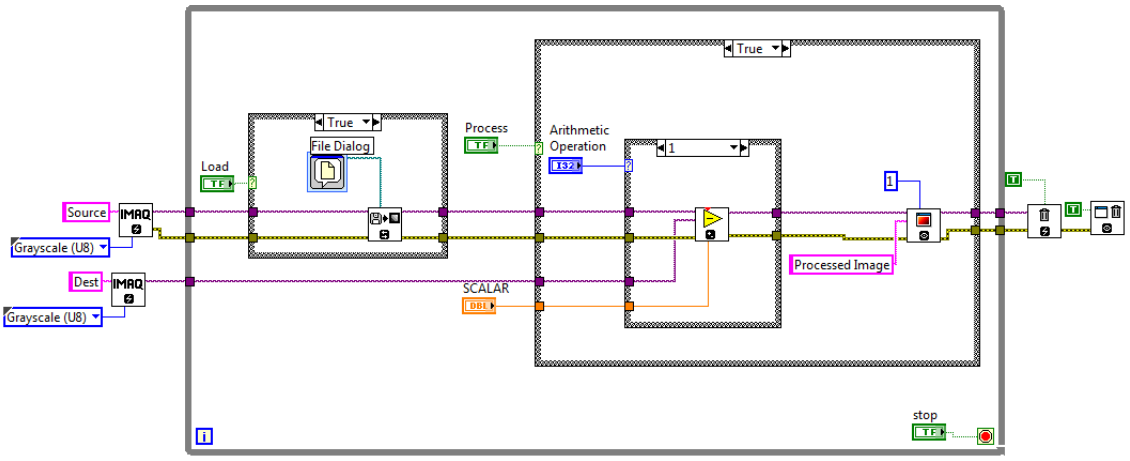

Figure 21: Block Diagram of Exercise 5

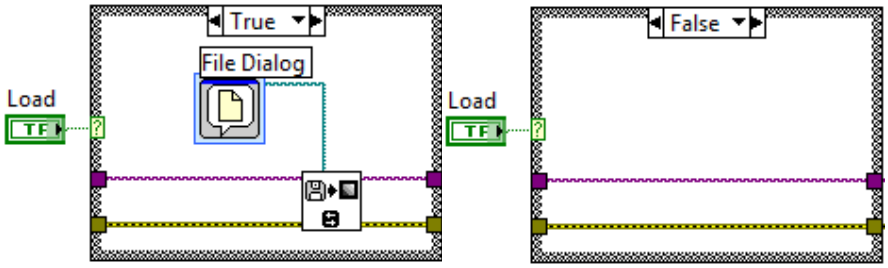

Figure 22: Left: "True" Case Block Diagram. Right: "False" Case Block Diagram

7. Place a **Case Structure** inside the previous **Case Structure** (Step 6). Wire the **List Box** with the **Selector Node.**
8. Place **IMAQ Add, Subtract, Multiply** and **Divide** on **Case 1, 2, 3** and **4** respectively. The palettes can be found on **Function>>Vision and Motion>>Image Processing>>Operators.**
9. Complete the structure as shown in Figure 23.

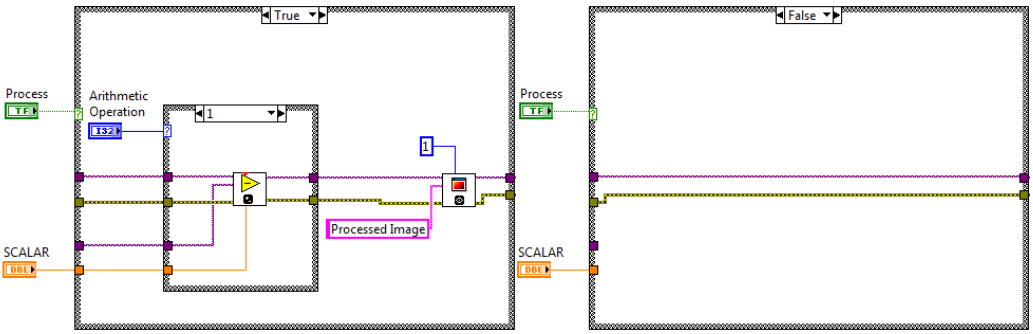

Figure 23: Left: "True" Case Block Diagram. Right: "False" Case Block Diagram

10. Finalize the **Block Diagram** as shown in Figure 21.
11. Run the VI. An image can be loaded using the **Load Image**.
12. Set the scalar value and after selecting the arithmetic operation as **Addition**, **Subtraction**, **Multiplication** or **Division**, hit the **ProcessButton** and the processed image will appear. Observe that pixel values might surpass the representation range.

END OF EXPERIMENT