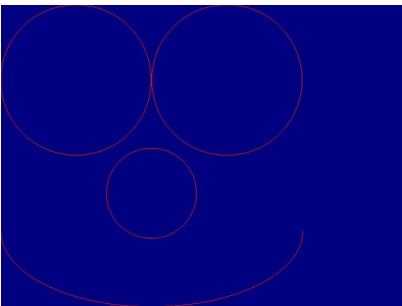
Image Overlays with NI Vision

Author: Bob Sherbert

Keywords: NI Vision, Image Overlay, LabVIEW



This image was generated in order to demonstrate the functionality of the NI vision overlay functions. The source was a plain blue image. The result is, as you can see, a giant smiley face. NI Vision functions can be used to overlay lines, arcs, points, text, and bitmaps in different combinations and arrangements onto any source image which can be opened in LabVIEW. These functions can be useful for real-time representation of complex data sets.

- · Motivation and Audience
- · Parts List and Sources
- Programming
- Final Words

Motivation and Audience

When working with any data in real time, it is usually useful to be able to interpret and examine it in real time. When such data is visual and mathematically intensive, it is in difficult for a user to interpret a simple numerical data dump if any more then 1 or 2 variables is present. Being able to display the data visually on top or along side of the source data is often useful. This tutorial covers methods which can be used to overlay simple shapes onto images. While it only covers an application to a single image, the techniques can easily be expanded upwards to moving video feeds, both live and recorded.

This tutorial assumes that the reader has the following skills/experience:

- Basic exposure to NI Vision tools (opening/closing files/streams)
- Exposure to LabVIEW programming

Parts List and Sources

TABLE 1: Parts required for construction

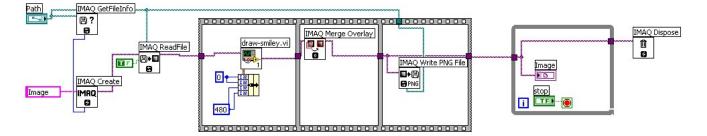
PART DESCRIPTION	VENDOR	PART	PRICE (2007)	QTY	Evaluation Available
LabVIEW 8	NI	776671-09	\$600	1	Y
NI Vision Development Module	NI	777859R-09	\$824	1	Y

Source Code

LabVIEW Code is available in the archive overlay.zip

The overlay functions are located in the menu 'Vision and Motion -> Vision Utilities -> Overlay'.

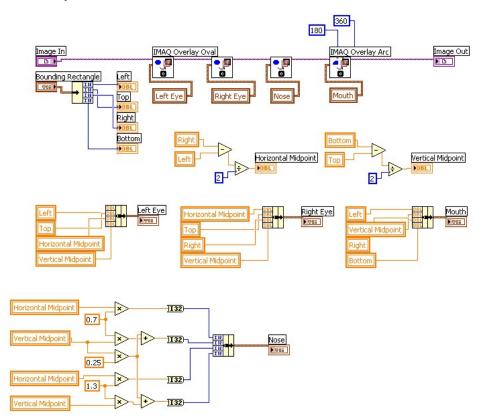
overlay.vi



The overlay VI is a simple shell to contain the actual commands which draw the smiley face onto the source image. Overlay.VI handles opening the source file, reading needed configuration information from it, writing the result, and then closing the file container. It also showcases a few simple design ideas for working with the overlay functions. Depending on the context of your program, it is often useful to use a sequence to properly order the processing. (For example: if you wanted to show a source and resultant images from your processing, LabVIEW may optimize your code in such a way that both the source and resultant images appear identical, unless you are explicit with the order that display and processing should occur in.)

The VI also shows a critical step in working with overlay functions. LabVIEW stores Image data (actual pixels) and overlay data as separate entities within a program. If you use write functions (save file, write to AVI, etc) the overlays will not be preserved along with the image data. To preserve them, you must merge the overlay data into the image data with the 'IMAQ Merge Overlay' VI.

draw-smiley.vi



The bulk of the program is contained in the file draw-smiley.vi. Do not be intimidated by the size and seeming complexity of the program, there is little being done here beyond simple math. There is only one line which serves as the overlay function demonstration. It is visible along the top of the file in a row of functions starting with the 'Image In' block, and ending with the 'Image Out' block. Two of the 12+ image overlay functions are showcased here, but all are very similar in their functionality and usage. We construct the smiley face out of three circles (two eyes and a nose) and an arc (mouth). Each of these two functions takes a 'bounding box' as its main input. The box (specified by the coordinates of its extremities at each side) is used to draw the shape. The box is represented as a cluster of 4 Int32 variables. Clusters of coordinates are the standard representation of input data for the overlay functions. For the sake of cleanliness in the program, each of the four needed boxes is computed from variable calls, the result is saved to a variable, and the result is only called when needed by the overlay VIs. The 'Overlay Arc' VI takes two additional inputs beyond those taken by 'Overlay Oval'. These are the starting and ending angle (on a standard coordinate system) that the part of the arc to be drawn is swept through. The numbers are given in degrees referenced off of the positive X-axis.

Building the Face

The face is constructed from some simple ratios based on the image size. The horizontal and vertical midpoints of the image are calculated and used to determine where the eyes should be divided (each occupies a quarter of the image), where the nose should be centered, and where to draw the arc for the mouth (around the lower arc of the ellipse formed in the bottom half of the image.)

Final Words

Working from the samples and explanations provided here, it should be possible for the reader to utilize the NI Vision Overlay functions. A small sampling of the functions were displayed in a sample program, and should provided insight into the remaining functions. Points, Lines, and Bitmap functions all require similar inputs to the ones used in this tutorial. These functions should, on a whole, be useful for visually debugging complex data sets.

The author can be reached by email