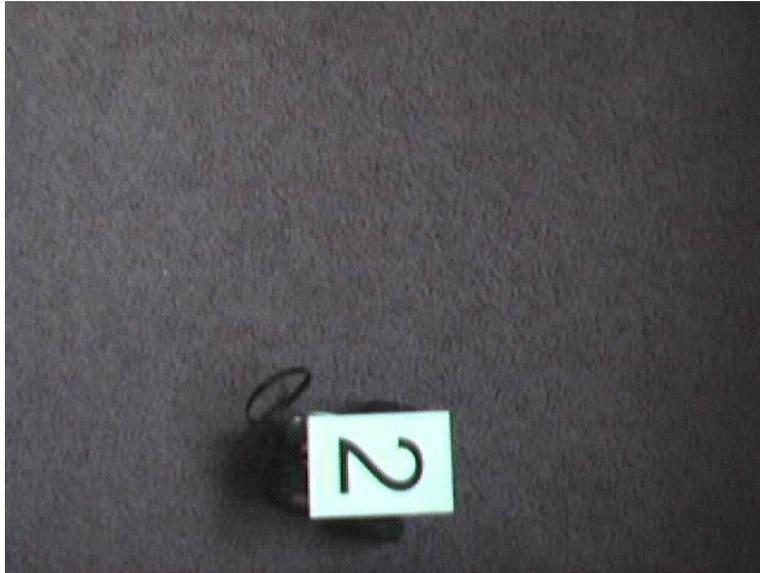# Using Vision Assistant for VI Generation

Author: Bob Sherbert

Keywords: LabVIEW, Vision Assistant, VI Export

This image shows successive steps from an image processing script created in NI's Vision Assistant. Image analysis and manipulation libraries are, in general, difficult and time consuming to learn and work with. As with many tasks, the addition of a graphical front end can aid those who wish to use the software without investing the time to develop complete mastery over it. Even with such a tool, it is still useful to have a quick guide or reference sheet on the topic. This document hopes to serve that purpose.

- Motivation and Audience
- Parts List and Sources
- Program Overview
- Export Operation
- Source Files
- Final Words

Motivation and Audience

National Instruments Vision Development Module provides a robust set of image processing algorithms for use with (and independent of) their LabVIEW software. By nature of what the module does, the algorithms included in it are all are fairly complicated, most requiring large configuration data structures as inputs along with very specific types of image data. Selection and configuration of these algorithms is made more difficult simply by their abundance. To help overcome this, NI has developed a program titled the 'Vision Assistant', which acts as a front-end to the bulk of the Vision VIs through a GUI interface and a simple scripting language. The program also has the ability to export its 'Vision Scripts' to LabVIEW VIs for usage with other user-written programs.

While working with the Vision VIs in LabVIEW without aid may be too complex an operation to be worthwhile, the task is greatly simplified by the Vision Assistant. This document will outline the software's features as well as describe the process of exporting a script to a VI.

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

- Familiarity w/LabVIEW

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 (critical functions disabled)
NI Vision Acquisition Software	NI	778413-01	$100	1	Y

Program Overview

Processing Function Categories

The Processing functions are divided based on the types of images they are designed to work with. These groups are represented on the toolbar pictured here and are described below in order from left to right. This document describes each group as a whole briefly, and then highlights some of the important functions from the group.

Image

Image functions are a generic class for dealing with all image types. They include such functions as Histogram, Brightness Adjustment, Masks, Storage and Retrieval, and Overlay. They are used mostly for script setup and management.

Color

Color functions are used for dealing with RGB(Red, Green, Blue) and HSL(Hue, Saturation, Luminance) images. Please note that color functions cannot be used with non-color images. The color functions include (among others):

- **Color Plane Extraction** - Used to remove one of the color planes (R, G, B, H, S, or L, depending on the image type you are working with) and return it as a resultant image. This effectively converts a color image into a Grayscale one.
- **Thresholding** - Based on RGB or HSL levels defined by the user, the function will separate each pixel into the image into a 'True' and 'False' set. The combination of these two sets constitutes a Binary Image which can be further processed by the appropriate functions.
- **Color Pattern Matching** - Searches for patterns based on predefined templates created by the user. Takes color levels and intensities into account.
- **Color Operators (arithmetic and logical)** - Numerically adds/subtracts/etc two images together. The operation is performed with each color channel.

Grayscale

- **Filters** - These are used to run different mathematical filters on the grayscale image. These include square, square root, and exponential modifications. They are useful (among other things) for operations such as increasing contrast, or softening edges.
- **Threshold** - Similar to the thresholding operations for color, these select all pixels within a certain range and return a binary image as a result.
- **Operators** - Similar to the color operators, these add/subtract/etc two different grayscale images together.

Binary

- **Inversion** - As expected, inverts the True and False parts of the binary image.
- **Shape Matching** - Searches through the image for a predefined pattern. The search is direction invariant, but the function only returns the location of the center of the identified pattern.

Machine Vision

- **Edge Detection** - Searches for
- **Pattern Matching** - This will attempt to match a pattern defined by the user based on image intensity levels in that pattern.
- **Geometric Matching** - This is similar to Pattern Matching, but performs matching by edge detection rather than intensity matching.
- **Caliper** - This is a tool that can measure distances (using a known reference) based on the outputs of other vision functions.

Identification

The Identification functions are mostly a collection of VIs used for identifying different types and varieties of bar codes.

Assembling a Script

The process of assembling a script is a very straightforward process. One simply needs to select the desired functions from the lower left hand menu and drag them into the Script window on the bottom of the screen. Most have options which can be configured by the user at the time they are selected. The functions can be arranged in the script windows in a linear fashion only. More information on the specifics of script assembly can be found in the tutorial Basic Pattern Recognition with NI Vision.
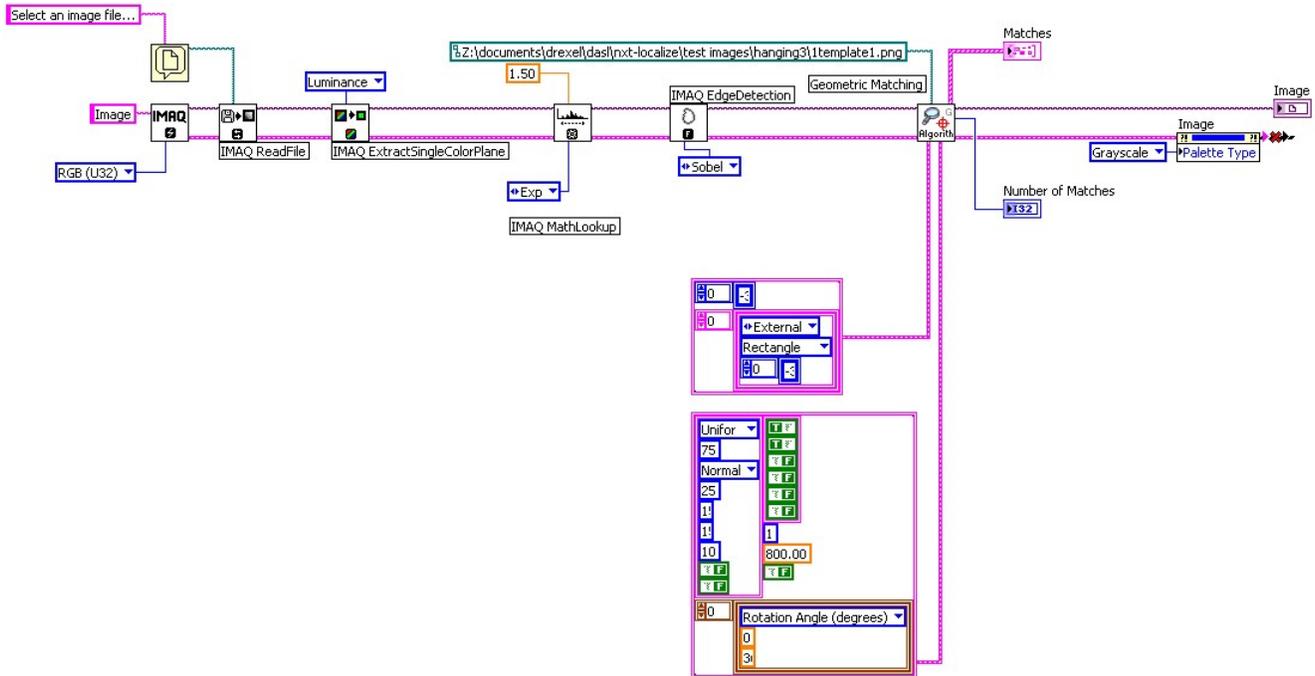
Exporting a Script

In Vision Assistant select Tools->Create LabVIEW VI. You will be prompted with a dialog box similar to this one:
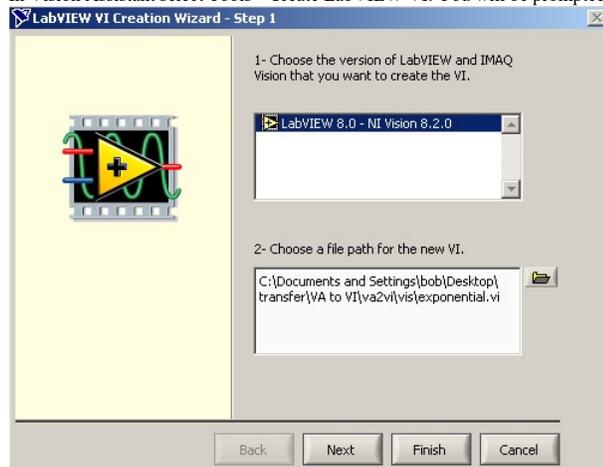
Within the box select your LabVIEW version and the path to the file to be exported and press 'Finish'. A dialog displaying the words "Creating LabVIEW VI" will be displayed while processing takes place.

Adapting to LabVIEW

When the export has finished you will be presented with a full block diagram that corrosponds to the script you created in the Vision Assistant (pictured above). However, the script has been exported as a stand alone VI, it has no connection terminals for input or output. To make this VI accessible to other external VIs follow this procedure:

1. Create Control	2. Wire Control	3. Place Control
Right click on the input you wish to replace and select Create->Control from the menu.	Delete the original input (file path in this case), and wire the new control into its former position.	Switch to the front panel and position the new control as desired.
4. Wire Terminal		
Switch the indicator icon on the front panel into 'connector mode' by performing a Right-Click->Show Connector. Pair the terminal of your choice and the new control by clicking first on the terminal and subsequently on the control. This will connect the control and terminal which may now be used by external VIs to pass data into yours.		

Many of the configurable options you had access to in the Vision Assistant, such as searchable regions, patterns to match against, color/intensity thresholds, etc, are now hard-coded into data clusters which are passed into the image processing VIs. (In the case of this example, theses are the large pink boxes wired into the 'Geometric

Matching' VI in the center of the diagram.) Part of the power of LabVIEW and the Vision Assistant is in their dynamic nature - we do not want to loose this configurability. You can reclaim the dynamic nature of these data structures and controls by following the above procedure for as many inputs as your VI requires.

Source Documents

LabVIEW and Vision Assistant
The source documents used in this tutorial are available in the archive exponential.zip

Final Words

In this tutorial we covered the basic components and divisions of NI's Vision Assistant. We also overviewed the process for exporting a Vision Script to a LabVIEW VI and integrating it for usage into larger programs.

The author can be reached by email