

# Basic Pattern Recognition with NI Vision

**Author: Bob Sherbert**

**Keywords:** National Instruments, vision, LabVIEW, fiducial, pattern recognition



This tutorial aims to instruct the reader on the method used to go from a source image (left) containing a patten of interest (square/rectangle combination) and identify the pattern's position and orientation relative to the camera frame (right).

- [Motivation and Audience](#)
- [Parts List and Sources](#)
- [Hardware Setup](#)
- [Programming](#)
- [Acquire Images](#)
- [Run the Script](#)
- [Final Words](#)

## Motivation and Audience

The first step towards a computer control system of an object in the real world is localization of the object within its environment. In order to make intelligent decisions, the computer must be able to tell where its controlled entity is. One method for performing this localization is through computer vision. A camera (or series thereof) in combination with a distinct printed pattern, or fiducial, can be used to determine the distinct position of the controlled object. This tutorial aims to instruct the reader on the first step in this process; how to perform basic pattern recognition on within an image using NI's Vision module.

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

- Basic experience with LabVIEW (construct VI's, sub-vi's, etc.)
- Installed LabVIEW Vision tools and USB camera extension

## Parts List and Sources

TABLE 1: Parts required for construction

PART DESCRIPTION	VENDOR	PART	PRICE (2007)	QTY
------------------	--------	------	--------------	-----

Logitech QuickCam Chat (or comparable)	<a href="#">Logitech</a>	961462-0403	\$30	1
--	--------------------------	-------------	------	---

## Hardware Setup

### Step 1

Suspend a camera about 2 feet over a smooth, uniformly colored surface.

### Step 2

Print out one or more of the fiducial contained in the file: [fiducial.doc](#). Cut out this symbol and set it on the surface under the camera.

### Step 3

Surround the area with bright lights, but do not point any of them directly at the surface.

## Programming

---

### NI Vision Assistant Script

Code and sample images are available in the archive: [pattern-recog-code.zip](#)

---

### **rotate test.scr**

This file describes the NI vision script used to scan the input image for the template you will provide. The components of this file are detailed below, you can either rebuild the script to suit your needs or modify the one provided. NI Vision scripts are made up of "Processing Functions" represented as blocks. They are performed in a linear sequence.

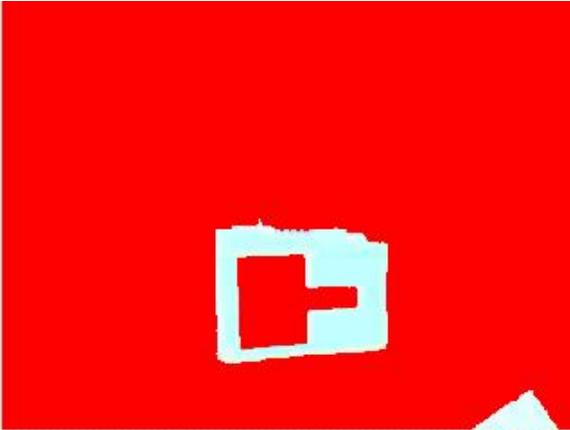
### **Block 1 - Original Image**

This is the image which processing will be carried out on. (See [Acquire Images](#) below) You can load an image by going to File->Open Image. Locate and Select an image you saved. If asked to remove previously acquired images say yes.

### **Block 2 - Color Threshold**

This function serves to divide the image into a binary selection of pixels which fall within the threshold, and those that fall outside of it. If working with the given fiducials you are working with black on white and can afford a very wide range. Working in RGB mode select values from 0 to above 240 for each channel. This will depend somewhat on your lighting conditions - play around a little until you can isolate the pattern from its

background as in the image.



### Block 3 - Lookup Table: Equalize

For our purposes Equalize serves to transform the binary image returned by the threshold block into a true grey-scale image which can be used the pattern matching block. Select "Equalize" from the menu in the lower left to obtain the desired function.

### Block 4 - Geometric Pattern Matching

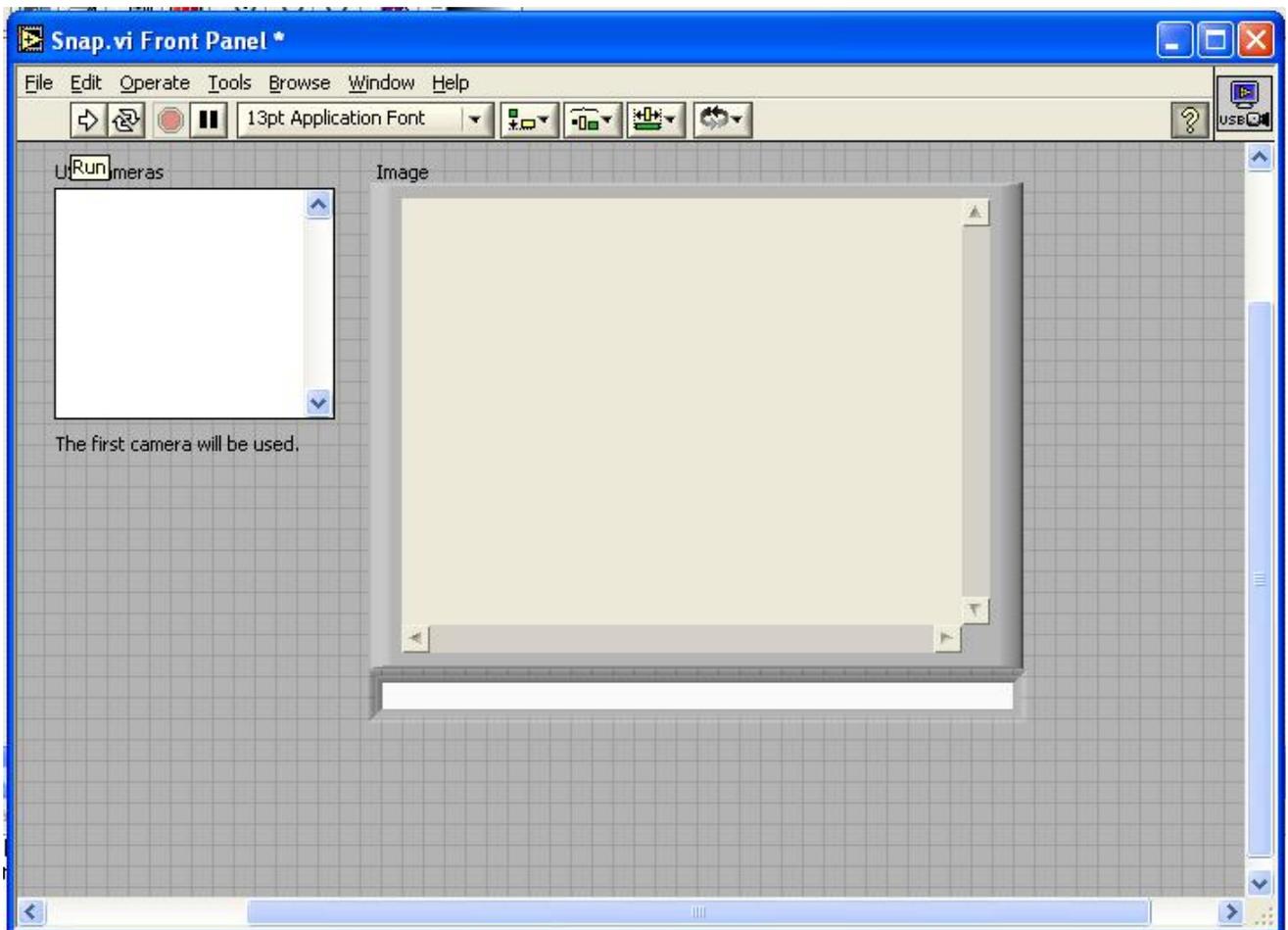
This is the true heart of the project. The pattern matching block will take a template provided by you (see [Acquire Images](#) for directions on creating the template) and search for it within the source image. It can identify the position of the template and its relative orientation. Once your template is created, switch to the "settings" tab in the lower left hand menu. The following settings were found to work with our setup (camera ~2ft above table): Select the check boxes next to "Rotated", "Scaled", and "Occluded" and input the following settings:

Setting	Min	Max
Rotated	0	360
Scaled	10	600
Occluded	0	50

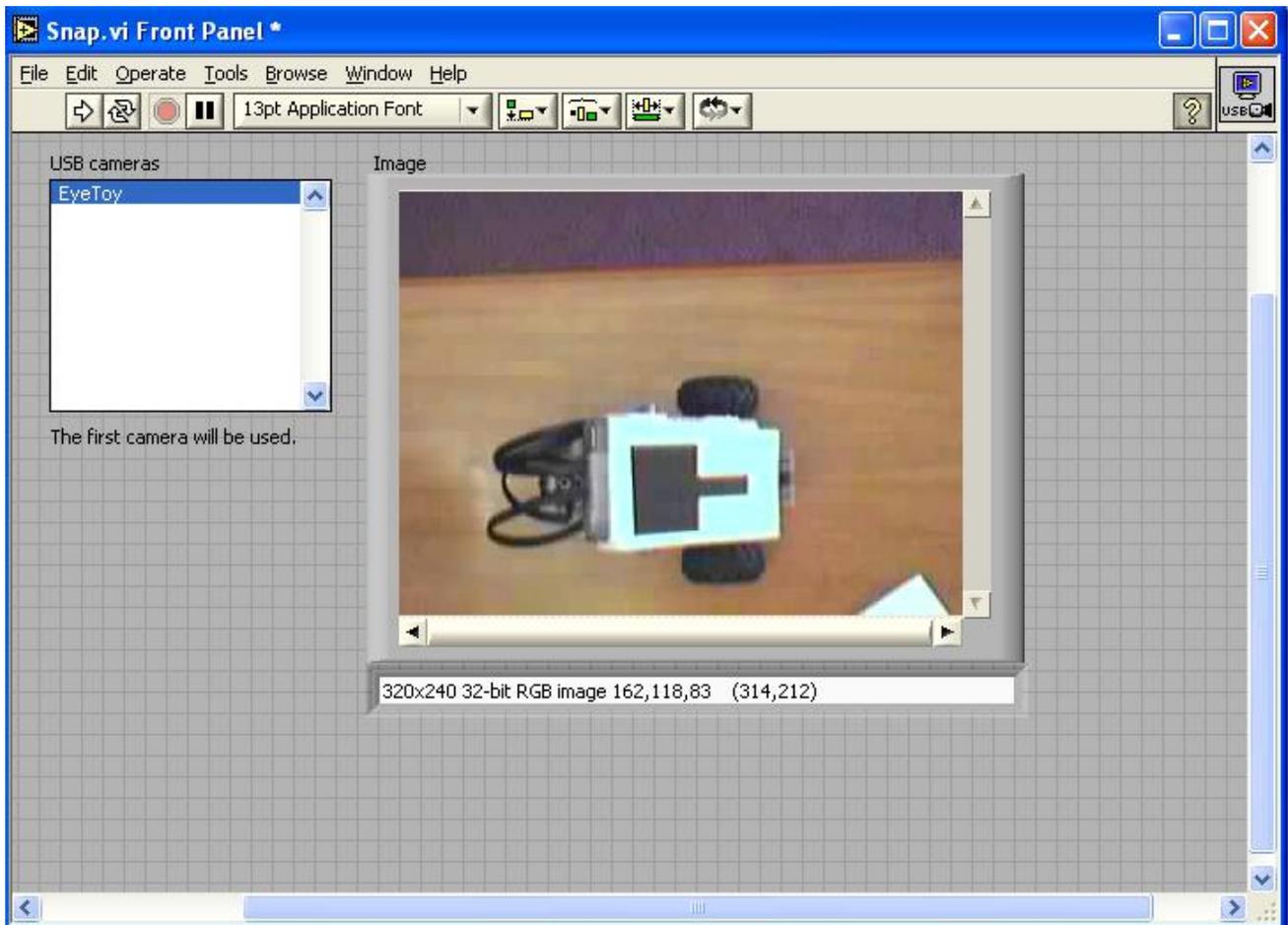
Click OK and the region should be drawn on your template.

### Acquiring Images

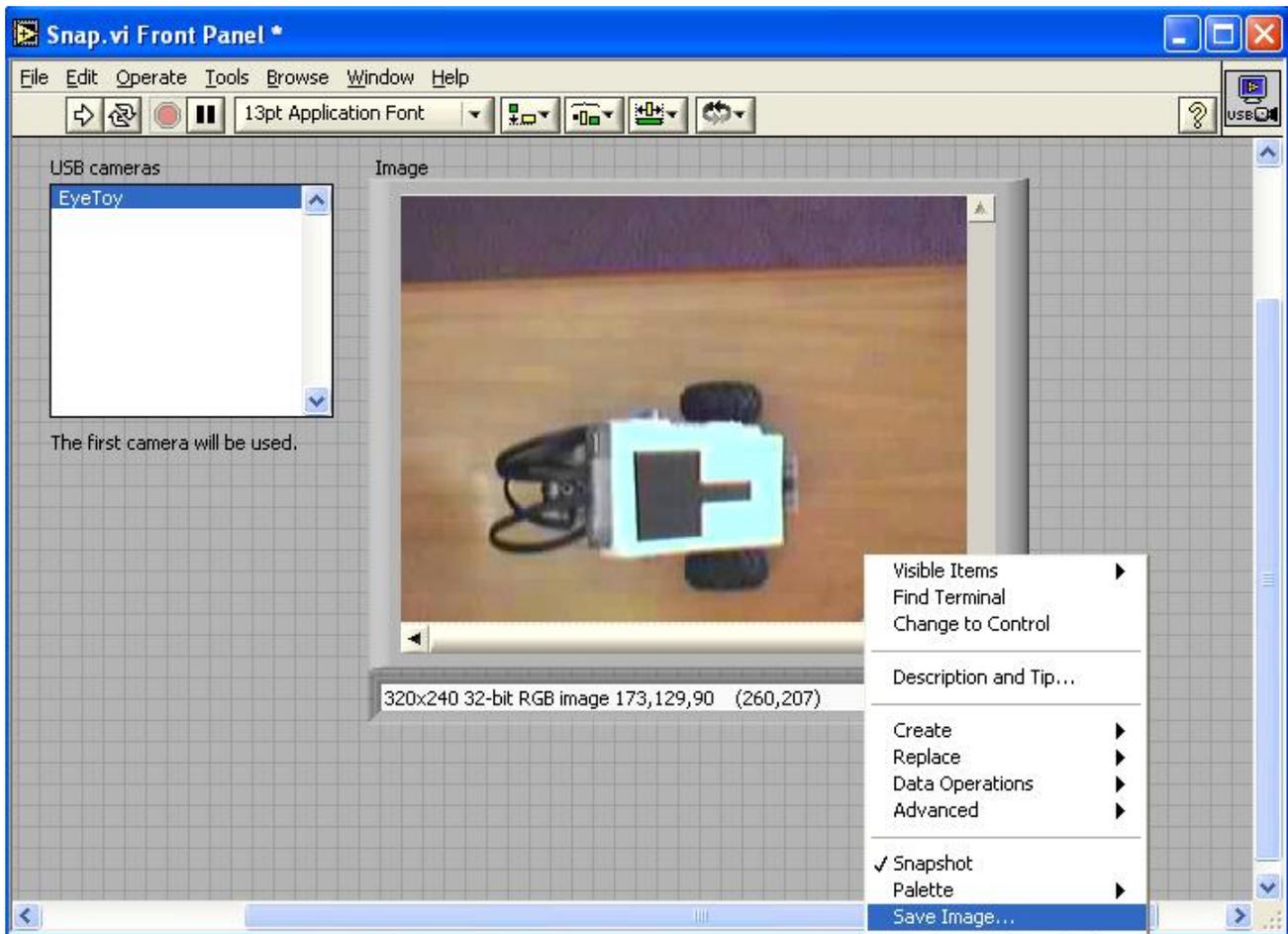
Open the LabVIEW USB Snap demo VI: LABVIEW/examples/IMAQ/imaqUSB examples.llb -> Snap.vi This VI is a demo for the USB module, but also allows you to take pictures using the same interface we will return to in the future for real-time pattern recognition.



Run the VI repeatedly. Align your subject matter with the frame of the image so that it is square with one of the sides. Continue taking snapshots until you have an image with adequate alignment.



Save the image:



Now repeat this process at a few different angles so you will be able to test later to test your template.

## Creating Templates

In the script sequence window select the Geometric Matching Step. Double Click on the block representing the step. Select "New Template" from the lower left hand menu. You will be presented with a pop-up containing the image as it is currently processed by the algorithm you designed. Select the region of interest using the zoom and selection tools. Select the finish button and select a location to save the template when prompted. If needed, adjust the center of the image to reflect the center of your object.

## Running the Script

Once the programming is done, the template is created, and the source image is loaded you can run the script. Click the "Run Once" button located directly above the script area to run the script. The scrip will run through and (provided that your template can be found in the images) highlight its location. You will see a "results" section similar to the one pictured below displayed where the script body usually appears giving numeric details on the identified region. It is advisable to load multiple images of your target patter and different orientations to be sure the recognition is working properly.

Results ...	1
X Position	152.01
Y Position	168.04
Angle	0.09
Scale	99.95
Score	1000.00

## Final Words

After completing this tutorial you should be able to create scripts capable of identifying arbitrary patterns within an image. This is the first step towards using a computer vision approach to localizing objects within a camera frame. From this beginning it should be possible to create real-time processing code by taking advantage of NI Vision Assistant's "Export to VI" feature.

The author can be reached by [email](#)