

OpenCV Tutorial

Part 3

Image Correlation

Gavin S Page
gsp8334@cs.rit.edu

Tasks

After learning to work with images it is important to learn some of the accessory functions OpenCV has to offer. This tutorial will discuss a simple image correlation example.

Steps Performed

Load an Image (Explanation Skipped)

Convert to Gray

Extract Template Region

Apply Match Functions

At this point loading an image and converting it to grayscale should be a simple task and can be copied from past tutorials.

Extract Template Region

Specify Region

Determine the starting point and the size of the region and create the *CvRect*.

Use *cvGetSubRect* to copy the template from the region.

```
//define the starting point and size of rectangle
int xVal = 1145;
int yVal = 890;
int neighLength = 25;
CvRect rect = cvRect(xVal,yVal,neighLength,neighLength);
//create the template and extract it from the source image
CvMat* tplate = cvCreateMat(neighLength, neighLength, CV_8UC1);
cvGetSubRect(imG, tplate, rect );
```

Here the template region is specified and extracted.

Use Template Match

Create Result Images

The image targets for the result of the match function have to be of size **$W-w+1 \times H-h+1$** and of type 32-bit single channel floating point.

Apply each of the match techniques for the example.

```
//specify the size needed by the match function
int resultW = imG->width - tplate->width + 1;
int resultH = imG->height - tplate->height + 1;
//create each of the result images
IplImage* result0 = cvCreateImage(cvSize(resultW, resultH), IPL_DEPTH_32F, 1);
IplImage* result1 = cvCreateImage(cvSize(resultW, resultH), IPL_DEPTH_32F, 1);
IplImage* result2 = cvCreateImage(cvSize(resultW, resultH), IPL_DEPTH_32F, 1);
IplImage* result3 = cvCreateImage(cvSize(resultW, resultH), IPL_DEPTH_32F, 1);
IplImage* result4 = cvCreateImage(cvSize(resultW, resultH), IPL_DEPTH_32F, 1);
IplImage* result5 = cvCreateImage(cvSize(resultW, resultH), IPL_DEPTH_32F, 1);

//apply each of the matching techniques
cvMatchTemplate(imG, tplate, result0, CV_TM_SQDIFF);
cvMatchTemplate(imG, tplate, result1, CV_TM_SQDIFF_NORMED);
cvMatchTemplate(imG, tplate, result2, CV_TM_CCORR);
cvMatchTemplate(imG, tplate, result3, CV_TM_CCORR_NORMED);
cvMatchTemplate(imG, tplate, result4, CV_TM_CCOEFF);
cvMatchTemplate(imG, tplate, result5, CV_TM_CCOEFF_NORMED);
```

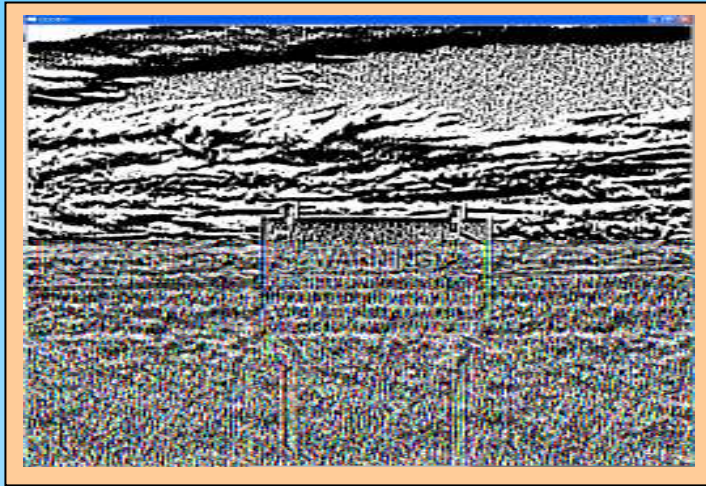
This slide documents the creation of the target images and the usage of the *cvMatchTemplate* function.

Original Image and Template



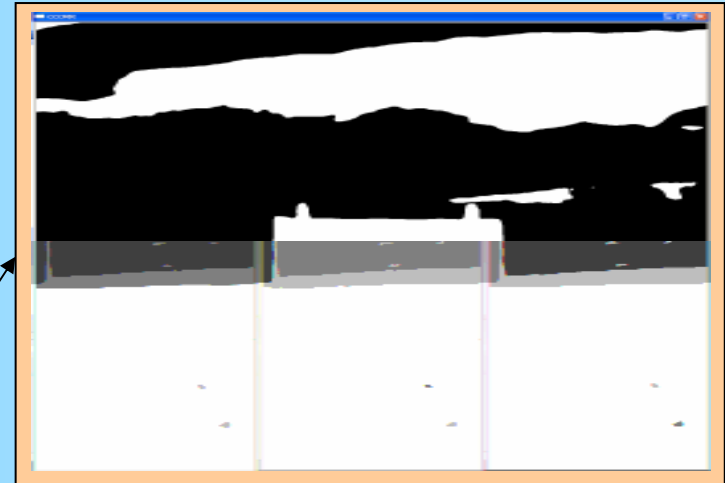
The original, grayscale image with template. Notice the region from which the template was extracted is labeled in the image.

Poor Results



CCOEFF

CCORR

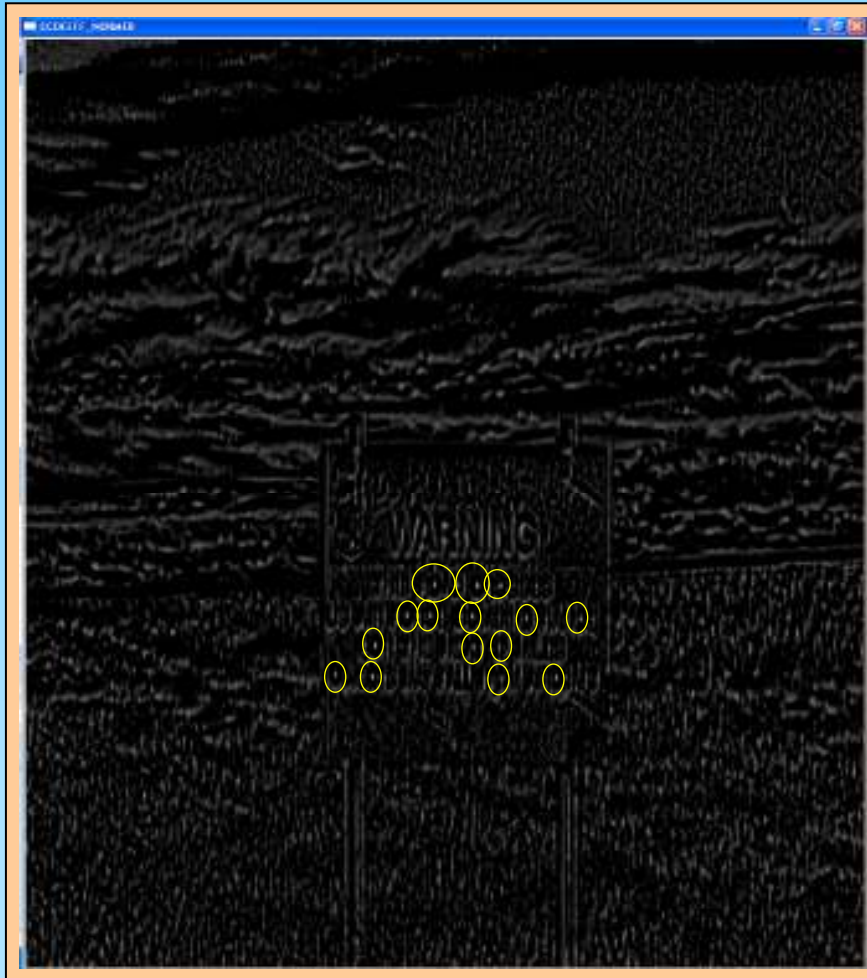


These particular methods did not demonstrate good results.

SQDIFF



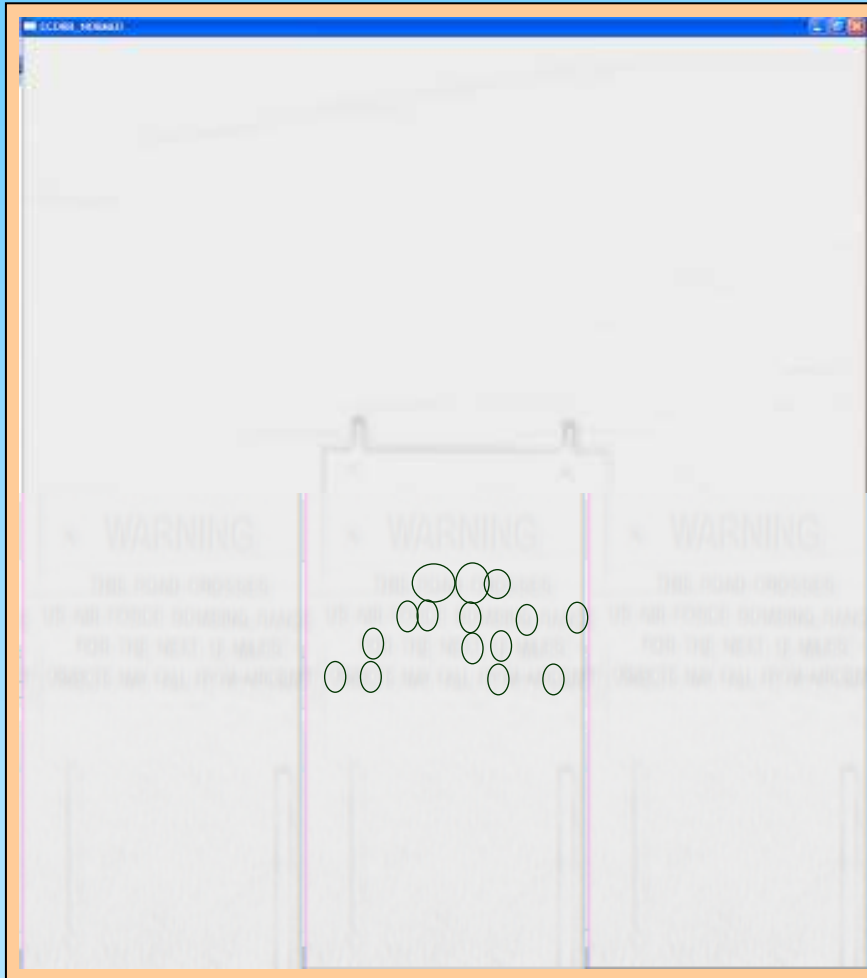
Good Results



CCOEFF_NORMED

Notice the high values on the circular letters on the sign

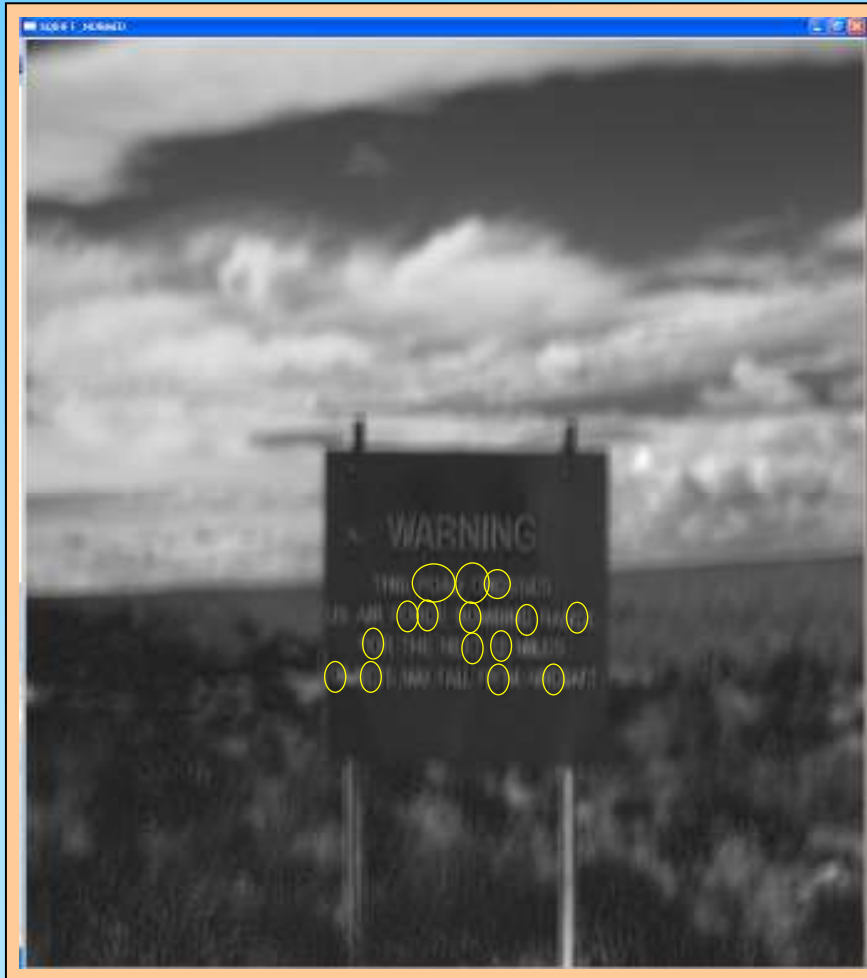
Good Results



CCORR_NORMED

Notice the high values on the circular letters on the sign

Good Results



SQDIFF_NORMED

Notice the low values on the circular letters on the sign

Final

This tutorial illustrated a simple example of image correlation. It showed that the normalized techniques exhibited better results