The Regression Approach to Searching for Image Changes

The archive system is designed to let each part do what it does best. The cameras collect images onto the server. Now, the cameras are capable of more sophisticated activities other than simply grabbing images and sending them to the server. However, those activities are better left to the server.

An example of this separation is the chore of searching for images. Typically, the task is to find changes in a particular scene and then have human eyes detect the target. The task of finding changes is better suited for the server since it is essentially a software task. Further, as will be shown below, the task makes good use of the server's processing power (speed, memory, etc.).

The task of finding changes boils down to taking image n and determining how similar it is to image n+1. As you may recall from a statistics class, similarity can be approached from the standpoint of correlation or regression. So, you may think of image n as having, say, 100000 points of data and image n+1 as also having 100000 point of data. Further, each of these points is at a fixed position on the image, e.g., point 1000 is always in the upper left corner of the image. Now, if we run a good old regression analysis on these two sets of 100000 points of data, the more similar the images, the closer our correlation (r or R) is to 1.0. And, the less similar the images, the lower the correlation. In other words, the correlation coefficient becomes a useful measure of the extent to which an image has changed.

Further, if we know something about the normal activity in the camera's field of view, we can specify, in terms of a correlation coefficient, the images we would like to examine. For example, let's assume we have a camera focused on a loading dock. And, we take all the images in a 30-minute period. Next we, step through these images and calculate the correlation between image n and n+1, then between n+1 and N+2, then N+2 and n+3, etc. Next we plot these correlations.

Now, let's pause a minute and think about what such a plot could tell us. What if 80% of the correlations are below .6? That would mean that the vast majority of the images are not similar to their preceding and following images. In other words, there is a lot of activity in the camera's field of view. So, if we look at only images with have a correlation of .60 and below, we are still going to have to look at a lot of images. In this case, whether we use correlation or any other technique, if there is a lot of activity, we will have to examine a lot of images. No technique is going to save us a great deal of time in this case.

On the other hand, suppose only 30% of the images are .60 and below. In this case, our technique is very helpful. If we decide to only look at images below .6, we eliminate 70% of the images.

In other words, there is not much happening in 70% of the images so we can ignore them. Of course, our target may actually be in that 70% in which case we would miss it. In such a case, we simply raise our criterion to, say, .75 and examine more images.

This statistical approach offers a wealth of possibilities. For example, if you are searching for changes in only a portion of the view field, that portion of the view field can be isolated by only using data from that part of the view field. Or, we might be interested in changes in one camera's field that correlate with a second camera's field. Such applications are accomplished by simply selecting the appropriate data to be analyzed.

Now, what about the details of the analysis? It is pretty straight forward but does take a bit of programming. First, the jpeg images must be converted to some sort of numeric form. One way to do this is to convert the jpeg to a ppm file. The ppm file is composed of numeric representations of each pixel in binary form. These representations can be converted to decimal and treated as input data for the regression equations.

In a nutshell, that is how the archive is searched. Initially, the plan was to use a multiple regression approach and separate the pixel information in a bit more detail. However, that has not seemed necessary since this approach appears to work just fine. Of course, since it is only code and statistics, more advanced analyses can always be added if necessary.