Chapter 1

Visualization Tools

The main contribution of this project is automatic methods for visualizing how a webcam scene varies, and for understanding the most important variations. In most natural scenes, we notice changes in lighting, weather, and camera conditions which are interesting, but fail to describe the typical behavior of a scene. The goal of these tools is to learn these variations and to point out changes independent of them. PCA is a commonly used tool, and captures a linear model of consistent image variations. By analyzing the results of certain PCA setups, we can obtain important and interesting information.

1.1 Setup

The most obvious way to learn about scenes is to take the PCA decomposition of the entire webcam scene. Unfortunately, our data-set is too large for this to be feasible, so we must limit our inputs. Instead of taking a random subset of the images, if we can intelligently limit the images we give, we can get better inputs.

1.1.1 Temporal Narrowing

The AMOS data-set consists mostly of outdoor scenes. These outdoor scenes vary significantly over the course of a day, going from night to day and back. The change in lighting dominates all other changes across the scene, and causes many images to be completely dark. Instead of wasting

(a) (b)

Figure 1.1: Figure 1.1(a) shows the first PCA component of a webcam scene. By adding or subtracting this component, we control how dark or how light the sky is. Figure 1.1(b) shows that a simple thresholding of this image effectively segments the sky from the rest of the image.

1.1.2 Sky Mask

In many outdoor scenes, even when narrowed to a particular time of day, the most difficult image variation to characterize is the sky. PCA has a difficult time learning changes in sunlight, clouds, and other characteristics of the sky, even among a set of images of one time of day. The variance in the sky dominates the PCA reconstruction, causing it to ignore information that is more interesting in this context.

Fortunately, it is fairly easy to learn which regions of an image are most affected by this. In practice, a side effect of the rising and setting of the sun is that the first principal component of most scenes is the sky, as shown in 1.1(a). By thresholding the values of this vector, we can effectively segment the sky from the rest of the image, as shown in 1.1(b).

This simple mask allows PCA to focus on more interesting changes in the image, and most of the results that follow use a simple mask to ignore the sky regions of the images. By setting all pixels in the sky to 0, we can effectively remove this difficulty.

(a) (b)

Figure 1.2: Figure 1.2(a) shows a grayscale webcam image. Figure 1.2(b) shows the gradient magnitude image of that frame. Notice the noise is mostly removed and the edges are highlighted.

1.1.3 Gradient Image

Webcam images are very high-dimensional - a typical 320 x 240 gray scale image has 76,800 pixels, each of which is a value from 0 to 255. One way to simplify this space without changing the size of the image is to look at the gradient magnitude images of a scene.

The x and y derivatives of an image are defined as $I_x(x,y) = I(x,y) - I(x-1,y)$ and $I_y(x,y) = I(x,y) - I(x,y-1)$ where the function $I(x,y)$ describes the intensities of an image's pixels. Once we have calculated the derivative images, we define the gradient magnitude of an image to be $G(x,y) = I_x(x,y)^2 + I_y(x,y)^2$.

The gradient magnitude image highlights edges in an image, as they are image locations where pixel values differ greatly from their neighbors. By performing PCA on the gradient magnitude images, we tend to ignore the potentially noisy surfaces of objects, and instead focus on the locations of the objects. 1.2(a) and 1.2(b) show the edges of an interesting webcam image highlighted in a gradient magnitude image.

1.2 Visualizations

In this section, we present several tools to visualize the webcam scene based on criteria we will discuss in the next section.

1.2.1 Image Montage

The simplest way to visualize a webcam scene is to view a montage of several images from that scene. We can easily sort the images along some dimension, and then show the n images with the highest values in that dimension. 1.3(a) Is an example montage of several interesting images from a webcam scene.

1.2.2 Intelligent Image Montage

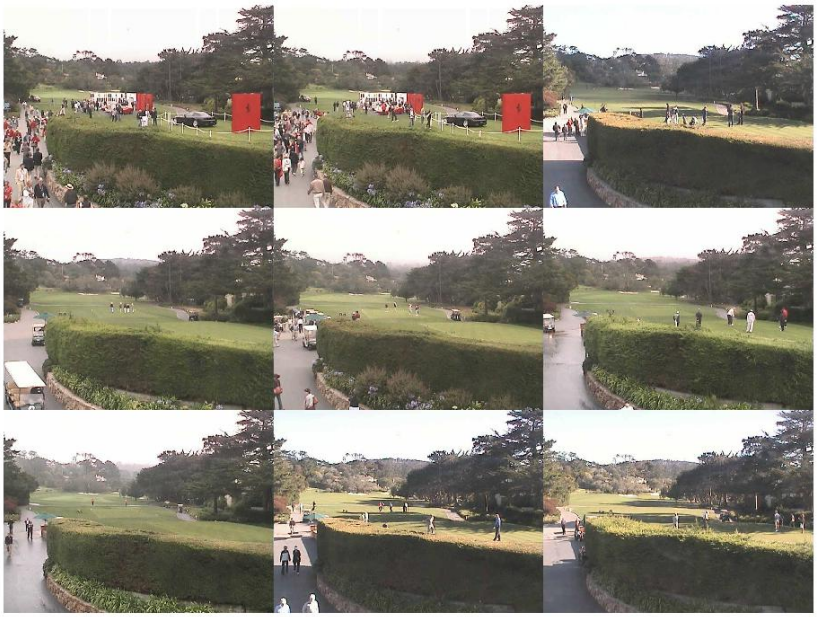
In several of these montages, we notice that many of the most unusual images are similar to each other. This presents the problem of finding unusual images that are different from the images we have already chosen.

A simple but effective algorithm for this problem uses the L2 norm in image space. If we consider an image as a vector of pixel intensities, such as $v = \{v_1, ..., v_m\}$, the L2 norm of two images v and u is

$$L2\_norm(v, u) = \sqrt{\sum_{i=1}^{m} (v_i - u_i)^2}$$

Given a large set of unusual images $\{x_1, ..., x_n\}$, we compute a distance matrix D where $D_{i,j} = L2\_norm(x_i, x_j)$. Now we iteratively find unusual images by choosing the image that has the largest distance from all of the images we have chosen so far. Specifically, for each remaining image calculate its distance to our set of exemplars, and choose the image whose distance is the smallest. We define the distance of an image x_0 to a set of exemplars $\{x_1, ..., x_n\}$ as

$$d = \min_i(D(x_0, x_i))$$



(a)

(b)

Figure 1.3: Figure 1.3(a) shows a montage of interesting images from a golf course webcam. Figure 1.3(b) shows a montage of interesting images of the same scene, but attempts to omit similar iamges.

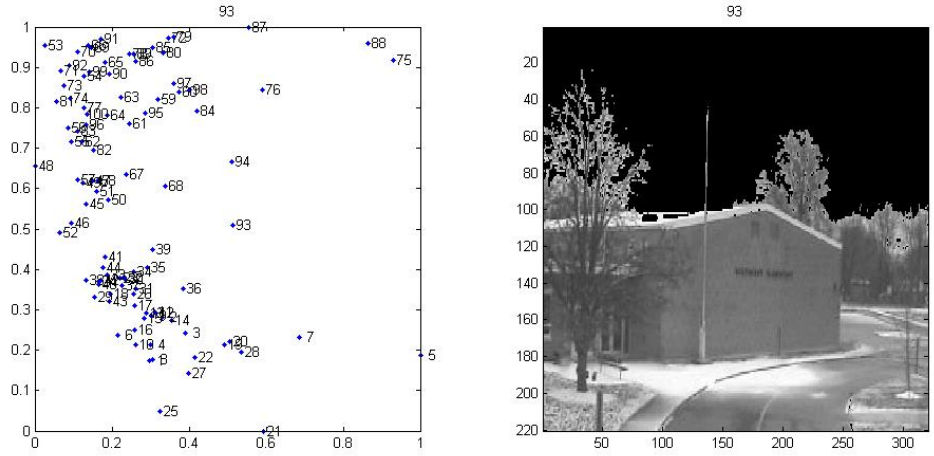


Figure 1.4: Figure 1.2.3 shows a webcam scene displayed using the 2D GUI.

In this way, on each iteration, we pick the image that is least likely to be similar to an image we have already selected. We have found that a good way to seed this process is choosing the image with the highest unusualness score as our first exemplar. 1.3(b) shows how similar images are omitted by viewing results in this way.

1.2.3 Two-dimensional Explorer

Using a simple GUI, we can explore a webcam scene in two dimensions. The GUI, shown in 1.2.3, displays a a plot of image scores for two different criteria, and displays the image corresponding to a mouseovered datapoint. Using this GUI, it is possible to see which criterias are related and which are not, as well as makes it easier to learn how the criteria effects the scene. [I don't really know what to say in this section].

1.3 Criteria

Once webcam scenes are projected onto a PCA basis, there are many different ways to analyze the results. In this section, we will present several different criteria for

evaluating this appearance model of a scene, the meaning behind each criteria, and results.

[I don't love the term criteria, but I think it encapsulates the idea of this section...]

1.3.1 PCA coefficient vector magnitude

For each image in a scene, PCA gives a vector of coefficients that correspond to the best linear combination of basis images to reconstruct that image. From this vector, we can assign each image a score equal to the magnitude of this vector. For a vector $v = (v_0, v_1, ..., v_n)$, the vector magnitude is

$$||v|| = \sqrt{\sum_{i=0}^{n} v_i^2}$$

This effectively gives us a measure for how far from the mean image in our basis space each image is. **??** shows the mean image of a webcam scene and **??** shows several images from that webcam scene that are especially far from the mean.

1.3.2 Residual Error

Given an image and a PCA basis, we can project the image onto the basis and see how well we can reconstruct the image.

[figure of subplot with image, reconstruction, and residual]

[figure of montage sorted be residualSSD, showing some noisy images and some with real foreground objects]

1.3.3 Variance Model

We can estimate the variance image of a webcam scene as the average of the square of each residual image.

Figure 1.5: Figure 1.2.3 show a webcam scene image, the reconstruction of that image from a PCA basis, and the residual image.

Figure 1.6: Figure 1.3.2 show a webcam scene image, the reconstruction of that image from a PCA basis, and the residual image.

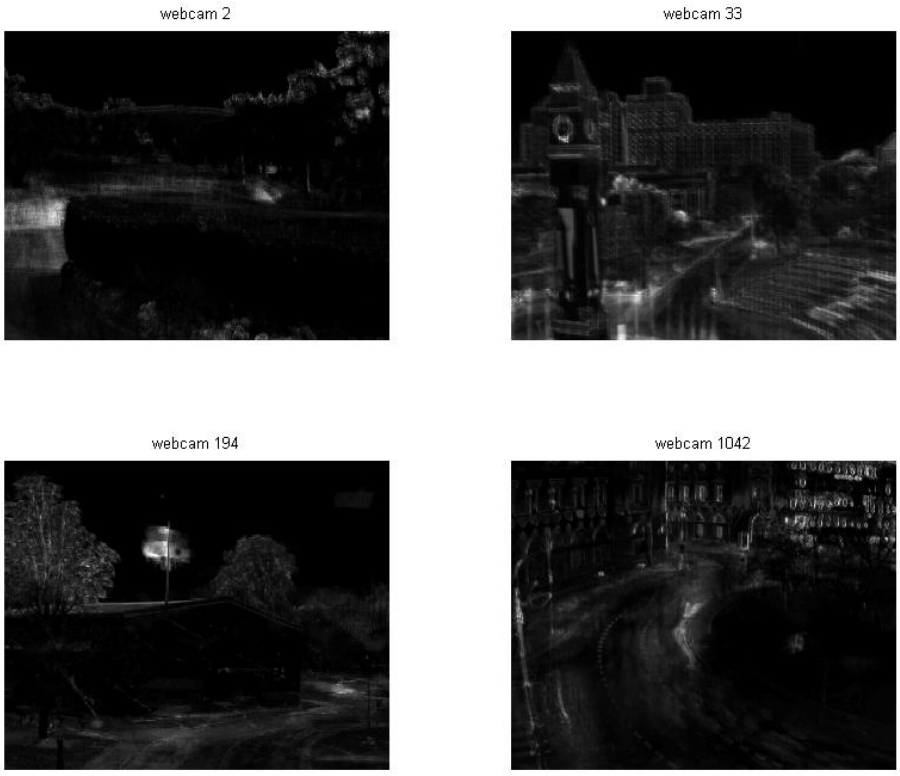


Figure 1.7: Figure 1.3.3 show a webcam scene image, the reconstruction of that image from a PCA basis, and the residual image.

One we have this variance image, we can attempt to isolate independent pixels in a z-score image. We define our z-score image as

$$Z(x, y) = \frac{R(x, y)}{V(x, y)}$$

where $R(x, y)$ is the residual value of the pixel and $V(x, y)$ is the variance value of the pixel.

[figure with variance image, residual image, and zScoreImage, highlighting how knowing the variance shows independent areas]

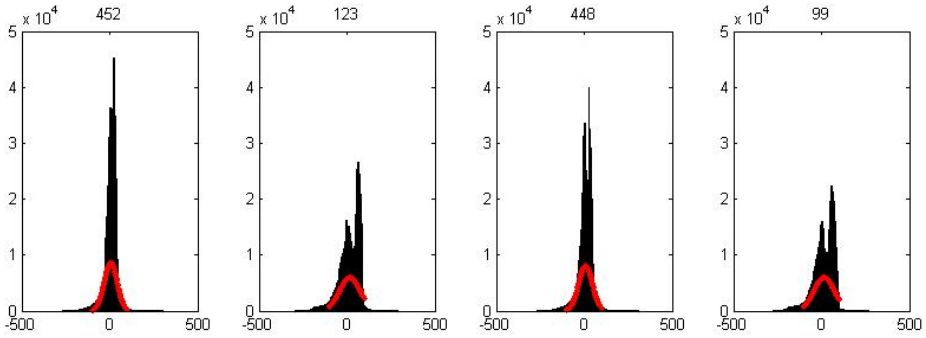


Figure 1.8: Several residual image histograms and the the best approximation of a normal distribution fitting the data.

1.3.4 Probabilistic

We can also try to learn about an image reconstruction by treating its residual image as samples from an underlying probability density function. If image deviations are due mostly to noise, we predict that these deviations will be normally distributed, but as shown in 1.3.4, this is not always the case. We have found that these distributions vary in the same way as webcam scenes, and analyzing them in the right way tells an interesting story.

[For each of these I'm going to explain what they should show, show a couple histograms with high values for them, and some example images]

1. **Normal Distribution Likelihood**

   The most obvious way to do this is to treat each residual image pixel as a sample from a normal distribution. We can easily estimate the mean and variance of this PDF and then

   $$f(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

2. **Laplacian Distribution Likelihood**

   Many of the residual images have a majority of pixels that are very close zero. This causes the histogram to look very similar to a laplacian distribution.

$$f(x|\mu,\beta) = \frac{1}{2\beta} e^{\frac{-|x-\mu|}{\beta}}$$

3. **Kurtosis**

   The kurtosis of a real-valued random variable is a measure of its peakedness. Larger values of kurtosis means more variance is due to less frequent extreme deviations rather than frequent less extreme deviations. It is defined as

   $$\gamma_2 = \frac{\mu_4}{\sigma^4}$$

   where μ_4 is the fourth moment about the mean and σ^4 is the estimated standard deviation to the fourth power. For a function $f(x)$, the k^{th} moment about the mean is defined as

   $$\mu_k = \int_{-\infty}^{\infty} (x-\mu)^k f(x) dx$$

   We can use this measure

4. **Skewness**

   The skewness of a random variable is a measure of asymmetry. Higher skewness values mean deviations on one side of the mean do not have corresponding deviations on the other side. It is defined as

   $$\gamma_1 = \frac{\mu_3}{\sigma^3}$$

   where μ_3 is the third moment about the mean and σ^3 is the cube of the standard deviation.

   This measure can be used