ECE418-Digital Video Processing

Elie Weintraub

Hillel Weintraub

**Final Project-Superresolution**

**Introduction:**

The superresolution problem involves taking several geometrically warped, blurred, noisy, and down sampled images, and producing an improved resolution restored image. In this project we implemented an algorithm proposed by K.V. Suresh, G. Mahesh Kumar, and A.N. Rajagopalan, in their paper, “Superresolution of License Plates in Real Traffic Videos” to perform superresolution. We also implement a simple extension of this algorithm to deal with the case of color images. In what follows we present our results.

**Results:**

We used the following image to represent our ideal high resolution image.

****

Figure 1: Ideal high resolution image

Next, we generate four low resolution, noisy, blurred and translated images from the ideal image. The following images were generated, using half pixel shifts, a 1-D 1x9 blur kernel , h, with all entries equal to .1111, a decimation factor of four. Zero mean Gaussian noise with variance=5 was also added to each image.

****

Figure 2: Low resolution images

We obtain an initial guess to feed into our algorithms, by realigning and up sampling the four images, and averaging them together.



Figure 3: Initial guess

We obtained the following improved quality image from our algorithm.



Figure 4: Superresolution Image

Clearly, our final result is a sharper image, and it is evident that much of the detail in the face as well as the camera has been restored. The ringing is an artifact of the motion blur and is not present when a Gaussian blur kernel is used instead.

We next present a quantitative analysis of our results. We use the PSNR metric to measure the quality of our results. Moreover, we calculated the PSNR over different segments of the superresolution image to see the variation in performance of our algorithm over the different image regions.

The following figure shows the different regions over which we calculated the PSNR. The first image in each row corresponds to a region from the original image, the second to that same region in our initial guess, and the third to the region in our final superresolution image.



Background

Grass

Camera Face

Figure 5: Regions of Interest Qualitative Analysis

The following table lists the PSNR in each region for the initial guess as well as for the final superresolution image.

**PSNR (dB) for Region of Interest**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Region | Background | Camera | Face | Grass | Whole Image |
| Initial Guess | 28.45 | 15.57 | 16.23 | 24.64 | 21.32 |
| SR Image | 36.14 | 18.72 | 20.25 | 26.18 | 25.02 |

Table 1: PSNR for various image regions of interest

The algorithm that we implemented was intended to be used on grayscale images; however, we decided to explore whether it could be used on color images as well. As a first attempt, we simply applied the algorithm separately to the R,G, and B components of the color image and combined the results. We found that the results were surprisingly fairly good.

We tested the algorithm on the following true color image.



Figure 6: Ideal high resolution true color image



Figure 7: R, G, B components of ideal true color image

We then generated noisy, blurred, and translated images from each component of the ideal image using half pixel shifts, a 5x5 Gaussian blur kernel with variance=3, and a decimation factor of four. Zero mean Gaussian noise with variance=5 was also added to each image. From these low resolution observations we obtained the following initial guesses for our algorithm.



Figure 8: Initial guesses



Figure 9: True color image obtained from initial guesses of R, G, B components

After running the algorithm on each component separately we obtained the following high resolution image.



Figure 10: High resolution image

The individual high resolution components obtained are shown below.



Figure 11: High resolution R, G, B components

The following table lists PSNR computed for the initial guess as well as for the SR image. PSNR was computed by treating the R, G, B components as one large composite image three times the size of the actual image.

**PSNR (dB) for Initial Guess and SR Color Images**

|  |  |
| --- | --- |
| Image | PSNR(dB) |
| Initial Guess | 27.59 |
| SR Image | 33.44 |

Table 2: PSNR for color images