Experimental Class

Xin Jin

Image upsampling

The original color image is represented as a 64×64×3 matrix of intensities *I*. The image is provided by “lena64color.tiff”. The downsampled color image is represented as a 32×32×3 matrix of intensities *D*. The image is provided by “lena32color.tiff”. Your job is to upsample the image *D* to be a 64×64 color image, by guessing the missing pixels. Generally, for a upsampling application, the original high resolution image is not available. If the upsampled image will be represented by *U*, we can try to guarantee that *U* satisfies the condition: $U_{ij} == D_{mk}$, $m = i/2+1$, $k = j/2+1$. So:

1. Please design the convex optimization model to derive the upsampled image. You can think about minimizing ℓ_2, ℓ_1 and ℓ_∞ norm and compare the effect of each of them;
2. Please try to introduce some regularization items to your model for some smooth regions to guarantee the smoothness in the variation of intensities, such as ℓ_2 regularization:
   $$\sum_{i=2}^{512}\sum_{j=2}^{512}\left(\left(U_{ij}-U_{i-1,j}\right)^2+\left(U_{ij}-U_{i,j-1}\right)^2\right) ,$$
   and ℓ_1 regularization:
   $$\sum_{i=2}^{512}\sum_{j=2}^{512}\left(\left|U_{ij}-U_{i-1,j}\right|+\left|U_{ij}-U_{i,j-1}\right|\right).$$
3. Image *I* cannot be used in the model. Ｉ
4. Compare the upsampled image *U* with *I* using PSNR (Average PSNR of RGB). Let’s see who can win among you.
5. Please provide:
   (ア) The m file (including the code of calculating the PSNR performance) and the upsampled images;
   (イ) The report describes your model, your implementation, results and performance comparison in detail.