

## Exercise 1: Compare Your Lossy Codec to JPEG

### Evaluate the Coding Efficiency of JPEG

- Choose a PPM image from our data base
- Encoding: Convert image to JPEG using different quality parameters (e.g., using ImageMagick)
- Decoding: Convert the JPEG file back to PPM format (e.g., using ImageMagick)
- Measure the RGB-PSNR between original and reconstructed image (tool available in KVV)
- Measure the bit rate (in bits per sample) based on size of the JPEG file

### Evaluate the Coding Efficiency of your Codec

- Encode and decode the PPM image (same as for JPEG) with varying quantization step sizes
- Measure the bit rate of the compressed file and the RGB-PSNR of the reconstructed image

### Compare Coding Efficiency of your Codec with that of JPEG

- Plot the RGB-PSNR over the bit rate for both your codec and JPEG (for multiple operation points)
- Compare your codec and JPEG by plotting the PSNR-rate curves into one diagram

## Exercise 2: Lossy Image Compression Challenge

### Improve your codec for lossy coding of PPM images

- Use any implementation of last weeks exercise as basis (see KVV)
- Try different simple techniques discussed in lectures and exercises

### The following might be worth trying

- Use YCoCg format for actual coding (see implementations for lossless coding)
- Add prediction between transform blocks:
  - Prediction of quantization index for DC coefficient (as in JPEG); or
  - Subtract mean of surrounding reconstructed samples before transform (should be better), add that mean after reconstruction (dequantization + inverse transform) of prediction error
- Improve entropy coding of quantization indexes:
  - Adaptive arithmetic coding (see implementations for lossless coding)
  - Adaptive binary arithmetic coding (see implementations for lossless coding)