Exercises: Implementing a Codec

The following tasks can be implemented in MATLAB.

1. Implement a codec.
   
   (a) Download the sample picture from the homepage
   
   \url{http://r0k.us/graphics/kodak/kodim08.html}
   
   Read the image with entries denoted here by
   
   \[ f(i, j) \quad \text{with } i = 1, 2, \ldots, \text{dim1}, \quad j = 1, 2, \ldots, \text{dim2}, \]
   
   where \text{dim1} and \text{dim2} are the image dimensions.
   
   (b) Convert the image into the YCbCr color space. Use only the Luma component. In MATLAB use: \texttt{rgb2ycbcr}.
   
   (c) Partition the image into blocks of size 8 × 8:
   
   • On each block, perform a 2D Discrete Cosine Transform. In MATLAB use: \texttt{dct2}.
   
   • Quantize the result for a uniform quantization parameter \( QP \in \{10, 15, 22, 32\} \):
   
   \[ \text{level}(i, j) = \text{round} \left( \frac{f(i, j)}{QP} \right) \]
   
   • Scale the level back
   
   \[ \tilde{f}(i, j) = \text{level}(i, j) \cdot QP \]
   
   and transform it back to spatial domain.
   
   (d) Put the image back together and display.

2. Create a Rate Distortion Plot.
   
   (a) For each \( QP \in \{10, 15, 22, 32\} \) approximate the rate \( R \) by calculating the number of samples times the entropy \( H \)
   
   \[ H = - \sum_{i=1}^{\text{dim1}} \sum_{j=1}^{\text{dim2}} p(i, j) \cdot \log_2(p(i, j)), \]
   
   where \( p(i, j) \) is the probability corresponding to \( \text{level}(i, j) \). To approximate the probability \( p \) of each level, count how often the value \( \text{level}(i, j) \) occurs in the image and denote this by \#\( \text{level}(i, j) \). Then it holds
   
   \[ p(i, j) = \frac{\#\text{level}(i, j)}{\text{dim1} \cdot \text{dim2}} \]
   
   **Hint:** In MATLAB you can use
   
   \texttt{hist(level,[min(level):1:max(level)])}.
   
   Here, make sure to take the zeros out before you calculate the entropy.
(b) For each $QP$ calculate the distortion $D$ using the mean squared error (MSE).

(c) Plot the pairs $(R, D)$ for each $QP$.

**Hint:** The Rate-Distortion Curve looks something like this:

![Rate-Distortion Curve](image)

**Food for Thought:**

- At which point of the algorithm do we lose information? How can we increase the compression rate?

- What is the structure of the transform coefficient matrix in step (1c) before we transform it back to spatial domain?