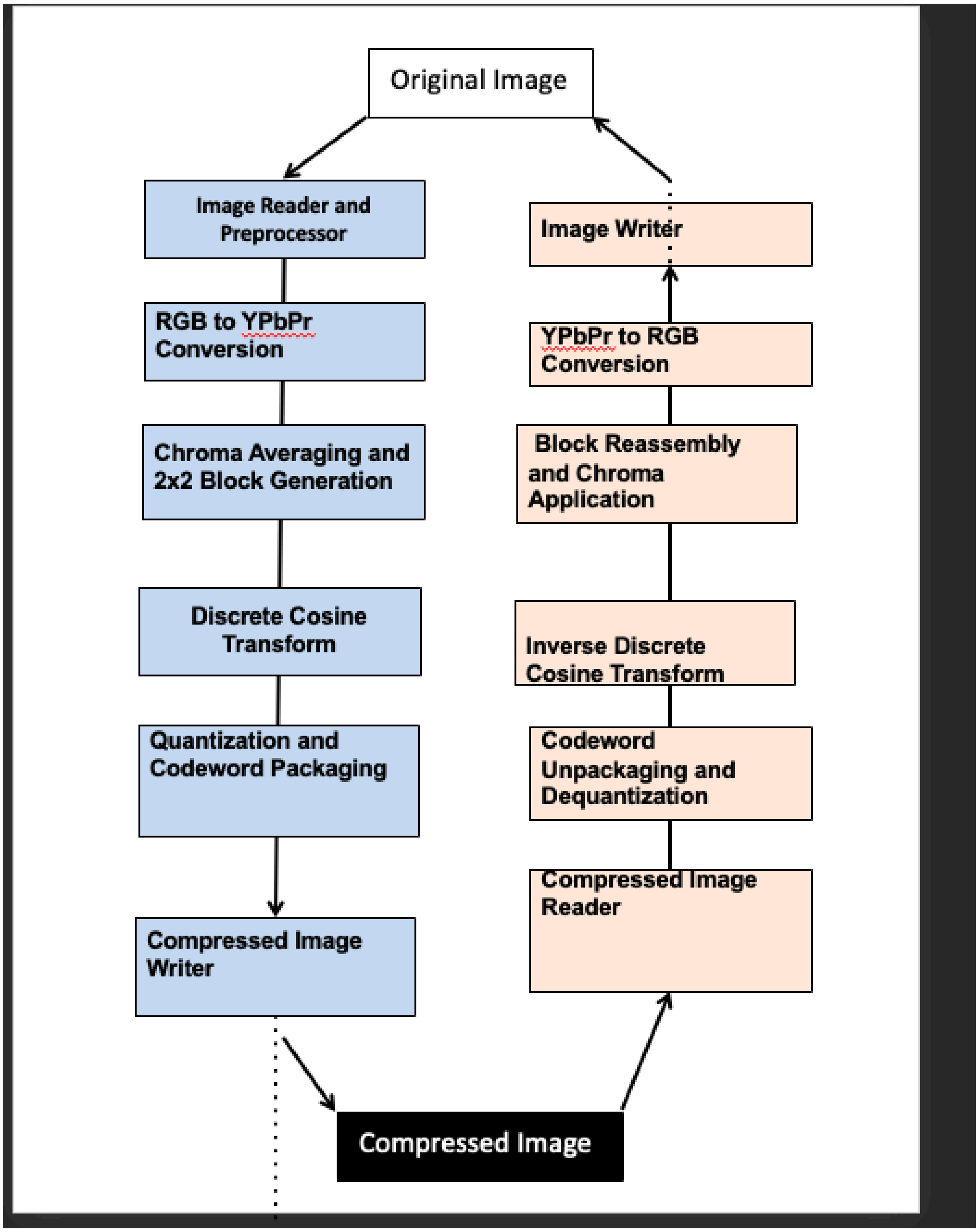

Compression Steps

1. Image Reader and Preprocessor

Reads the PPM image file and trims any odd-numbered rows or columns to make even dimensions.

- Input: PPM image file (from a file or standard input).
- Output: Preprocessed image data with even width and height.
- Information Loss: Yes. Possible loss of the last row or column if dimensions are odd, resulting in the removal of those pixels.

- Test Case 1: Odd Width and Even Height (and vice versa)
  Purpose: Verify trimming of the last column.
  Description: Use an image with odd width and even height; confirm the last column is removed.
- Test Case 2: Odd Width and Odd Height
  Purpose: Verify trimming of both last row and column.
  Description: Use an image with both odd width and height; ensure both the last row and column are removed.

2. RGB to YPbPr Conversion

Converts RGB values to Y (luminance), Pb, and Pr (chrominance) components.

- Input: Preprocessed image data in RGB format.
- Output: Image data in YPbPr color space (floats).
- Information Loss: Minimal. Possible loss due to floating-point precision errors during conversion.
  - Test Case 1: Primary Colors
    Purpose: Check chrominance accuracy.
    Description: Convert pure red, pure green, and pure blue pixels; verify Pb and Pr values correspond to expected chrominance.
  - Test Case 2: Black and White Pixels
    Purpose: Validate handling of extreme luminance values.
    Description: Convert RGB (0, 0, 0) and RGB (max, max, max); expect Y = 0 and Y = max respectively, with Pb and Pr being zero.

3. Chroma Averaging and 2x2 Block Generation

Groups the YPbPr pixels into 2x2 blocks and computes the average Pb and Pr values for each block (reducing chroma resolution).

- Input: Image data in YPbPr color space.
- Output: 2x2 blocks of Y values and average Pb/Pr values.
- Information Loss: Yes. Chroma resolution loss due to averaging.
  - Test Case 1: Uniform Chroma Blocks
    Purpose: Ensure accurate averaging with identical chroma.
    Description: Use a 2x2 block where all Pb and Pr values are the same; verify the averaged Pb and Pr remain unchanged.
  - Test Case 2: Maximum Chroma Variation
    Purpose: Test handling of extreme chroma differences.
    Description: Use a 2x2 block with Pb and Pr values at their maximum and minimum; verify averaging is accurate.

4. Discrete Cosine Transform

Applies the DCT to the Y values of each 2x2 block to obtain coefficients a, b, c, and d.

- Input: Y components from each 2x2 block.
- Output: DCT coefficients a, b, c, and d.
- Information Loss: Minimal. Floating-point errors may occur during the transform.

- Test Case 1: Uniform Y Values
  Purpose: Verify DCT coefficients for uniform brightness.
  Description: Use a 2x2 block where all Y values are identical; expect b, c, d coefficients to be zero.
- Test Case 2: Diagonal Contrast
  Purpose: Validate coefficient d for diagonal brightness differences.
  Description: Use a 2x2 block with diagonal Y values differing significantly; verify coefficient d captures the contrast.

5. Quantization and Codeword Packaging

Quantizes the DCT coefficients (a, b, c, d) and chroma values (Pb, Pr) and packs them into 32-bit codewords.
- Input: DCT coefficients and average Pb/Pr values.
- Output: 32-bit codewords representing compressed image data.
- Information Loss: Yes. Loss occurs during quantization.
  - Test Case 1: Boundary Quantization
    Purpose: Ensure correct handling of values at quantization limits.
    Description: Quantize coefficients and chroma values exactly at the maximum and minimum thresholds; verify proper clamping or indexing.
  - Test Case 2: Packing Consistency
    Purpose: Verify that packing followed by unpacking retains quantized values.
    Description: Pack known quantized coefficients and chroma indices into a codeword, then unpack; ensure retrieved values match the originals.

6. Compressed Image Writer

Writes the compressed image data (32-bit codewords) to disk in big-endian order.
- Input: Sequence of 32-bit codewords.
- Output: Compressed image file or data stream.
- Information Loss: None beyond prior quantization. Data is being formatted for output.
  - Test Case 1: Big-Endian Order
    Purpose: Validate byte ordering.
    Description: Write known codewords and check the byte order is big-endian by examining the output file in a hex editor.
  - Test Case 2: Large Image Handling
    Purpose: Verify writing performance and correctness for large images.
    Description: Write a compressed image with a large number of codewords; ensure all codewords are correctly written without truncation or corruption.

Decompression Steps

1. Compressed Image Reader

Reads the header and 32-bit codewords from the compressed image file or data stream.
- Input: Compressed image file or data stream.
- Output: Width, height, and sequence of codewords.
- Information Loss: None. Data is being read as-is.

- Test Case 1: Invalid Header Detection
  Purpose: Test robustness against malformed headers.
  Description: Use a compressed file with incorrect header format; ensure appropriate error handling or rejection.
- Test Case 2: Incomplete Codewords
  Purpose: Detect insufficient data for codewords.
  Description: Provide a compressed file with incomplete codewords (e.g., missing bytes); ensure the reader raises an error.

2. Codeword Unpackaging and Dequantization

Unpacks each 32-bit codeword to retrieve the quantized coefficients and chroma values, then applies inverse quantization to convert the quantized integers back to floats.
- Input: Sequence of codewords.
- Output: Dequantized coefficients and chroma values.
- Information Loss: No. Recover precision lost during compression.
  - Test Case 1: Known Codeword Unpacking
    Purpose: Validate accurate unpacking of codewords.
    Description: Use codewords with predefined quantized values; verify that unpacked coefficients and chroma indices match the expected values.
  - Test Case 2: Maximum Quantized Values
    Purpose: Ensure correct handling of extreme quantized values.
    Description: Use codewords with coefficients and chroma indices at their maximum and minimum; verify accurate dequantization.

3. Inverse Discrete Cosine Transform

Applies the inverse DCT to the dequantized coefficients to reconstruct Y values for each pixel.
- Input: Dequantized coefficients a, b, c, and d.
- Output: Y values for each pixel in the 2x2 block.
- Information Loss: Minimal. Loss due to prior quantization.
  - Test Case 1: Inverse Transformation Consistency
    Purpose: Ensure IDCT followed by DCT approximates original coefficients.
    Description: Apply IDCT to reconstructed Y values, then DCT again; verify that the coefficients remain consistent within acceptable error margins.
  - Test Case 2: Maximum Coefficients Impact
    Purpose: Test reconstruction with extreme coefficient values.
    Description: Use coefficients at their maximum and minimum; ensure Y values are within expected ranges.

4. Block Reassembly and Chroma Application

Combines the reconstructed Y values with the averaged Pb and Pr values to form full 2x2 pixel blocks.
- Input: Y values and averaged Pb/Pr.
- Output: 2x2 blocks of YPbPr values.
- Information Loss: None beyond previous steps.

- Test Case 1: Chroma Consistency Across Block
  Purpose: Validate consistent chroma application.
  Description: Use a block with averaged Pb/Pr; ensure all four pixels in the block share the same Pb and Pr values.
- Test Case 2: Multiple Block Assembly
  Purpose: Ensure proper handling of multiple blocks.
  Description: Assemble multiple 2x2 blocks into a larger image structure; verify correct placement and data integrity.

5. YPbPr to RGB Conversion

Converts the Y, Pb, and Pr values back to RGB values.
- Input: YPbPr values for each pixel.
- Output: RGB values for each pixel.
- Information Loss: Minimal, due to floating-point precision in prior steps.
  - Test Case 1: Clamping Verification
    Purpose: Ensure RGB values are within valid ranges.
    Description: Use YPbPr values that could potentially exceed RGB bounds; verify that resulting RGB values are correctly clamped to [0, max].
  - Test Case 2: Negative Chroma Handling
    Purpose: Test proper conversion with negative Pb and Pr.
    Description: Convert YPbPr values with negative Pb and Pr; ensure RGB values are correctly computed without underflow.

6. Image Writer

Writes the decompressed RGB image data back to a PPM file or outputs it to standard output.
- Input: RGB values for each pixel.
- Output: Decompressed PPM image file or data stream.
- Information Loss: None. Data is being formatted for output.
  - Test Case 1: PPM Format Compliance
    Purpose: Ensure correct PPM file structure.
    Description: Write a PPM file and verify its header and pixel data conform to the PPM specification.
  - Test Case 2: Image Viewer Compatibility
    Purpose: Validate that the written image can be opened by standard viewers.
    Description: Write a PPM file and open it using common image viewers to ensure it displays correctly.

Implementation Plan

1) Image Reader and Preprocessor (Compression)
2) Compressed Image Reader (Decompression)
3) RGB to YPbPr Conversion (Compression)
4) Codeword Unpackaging and Dequantization (Decompression)
5) Chroma Averaging and 2x2 Block Generation (Compression)
6) Inverse Discrete Cosine Transform (Decompression)

7) Discrete Cosine Transform (Compression)
8) Block Reassembly and Chroma Application (Decompression)
9) Quantization and Codeword Packaging (Compression)
10) YPbPr to RGB Conversion (Decompression)
11) Compressed Image Writer (Compression)
12) Image Writer (Decompression)