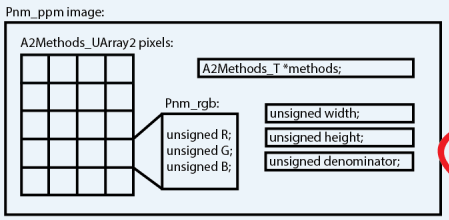
COMP40 HW4 Design Doc
Sitara Rao (srao03) and Era Iyer (eiyer01)
10/15/2019

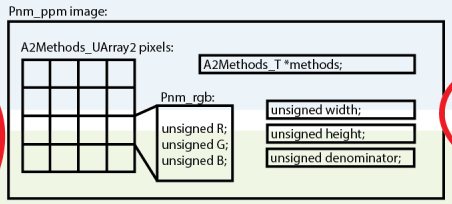
# Original Image

Module 1: ppm_mod.c

1.a) Read in ppm file
Input: ppm image from stdin/ command line filename
Output: Pnm_ppm object



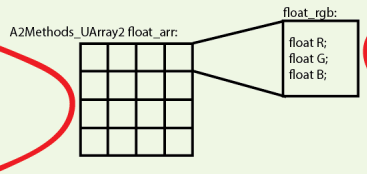

1.b) Trim to even width and height
Input: Pnm_ppm object
Output: Pnm_ppm object




8. Write Pnm_ppm to stdout
Input: Pnm_ppm object
Output: Ppm image in binary

Module 2: float_mod.c

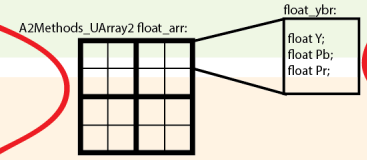
2.a) Convert RGB unsigned ints to RGB floats
Input: Pnm_ppm object
Output: A2Methods_UArray2 object, where each elem is a struct of 3 floats (struct float_rgb)




2.b) Convert floats R,G,B to floats Y, Pb, Pr
Input: A2Methods_UArray2 object, where each elem is a struct of 3 floats (struct float_rgb)
Output: A2Methods_UArray2 with blocksize = 2, where each elem is a struct of 3 floats (struct float_ybr)

7.b) Choose a denominator to scale R, G, B back to unsigned ints
Input: A2Methods_UArray2 object, where each elem is a struct float_rgb
Output: Pnm_ppm object

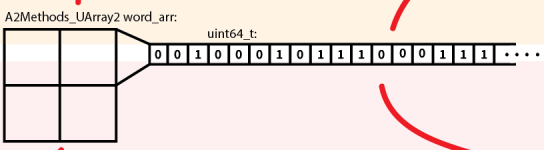
7.a) Convert floats Y, Pb, Pr to floats R, G, B
Input: A2Methods_UArray2 object, where each elem is a struct of 3 floats (struct float_ybr)
Output: A2Methods_UArray2 where each elem is a struct_rgb




Module 3: word_mod.c

3. Calculate a, b, c, d, avg(Pb), avg(Pr) for each block
Input: A2Methods_UArray2 with blocksize = 2, where each elem is a struct of 3 floats (struct float_ybr)
Output: A2Methods_UArray2 object, half the height and half the width of the input. Each elem is a uint64_t that represents a block of the input

6. Reconstruct Y1, Y2, Y3, Y4 (for each cell) and Pb, Pr (for the block)
Input: A2Methods_UArray2 with each elem a struct word32bit
Output: A2Methods_UArray2 object, twice the height and twice the width of the input. Each elem is a float_ybr; Y is unique for each cell, while Pb and Pr are the same across each 4-cell block




Module 4: bit_mod.c

4. Print compressed image to stdout
Input: A2Methods_UArray2 object containing uint64_t objects
Output: a header, followed by a row-major printing of 32 bit binary sequences (the contents of each uint64_t in the array) in big-endian order

5. Read 32-bit sequences in from stdin
Input: a 2 line header followed by a sequence of 32-bit words
Output: A2methods_Uarray2 with height/2 and width/2 dimensions (given in header), each elem is a uint64_t containing a word from input

# Compressed Image