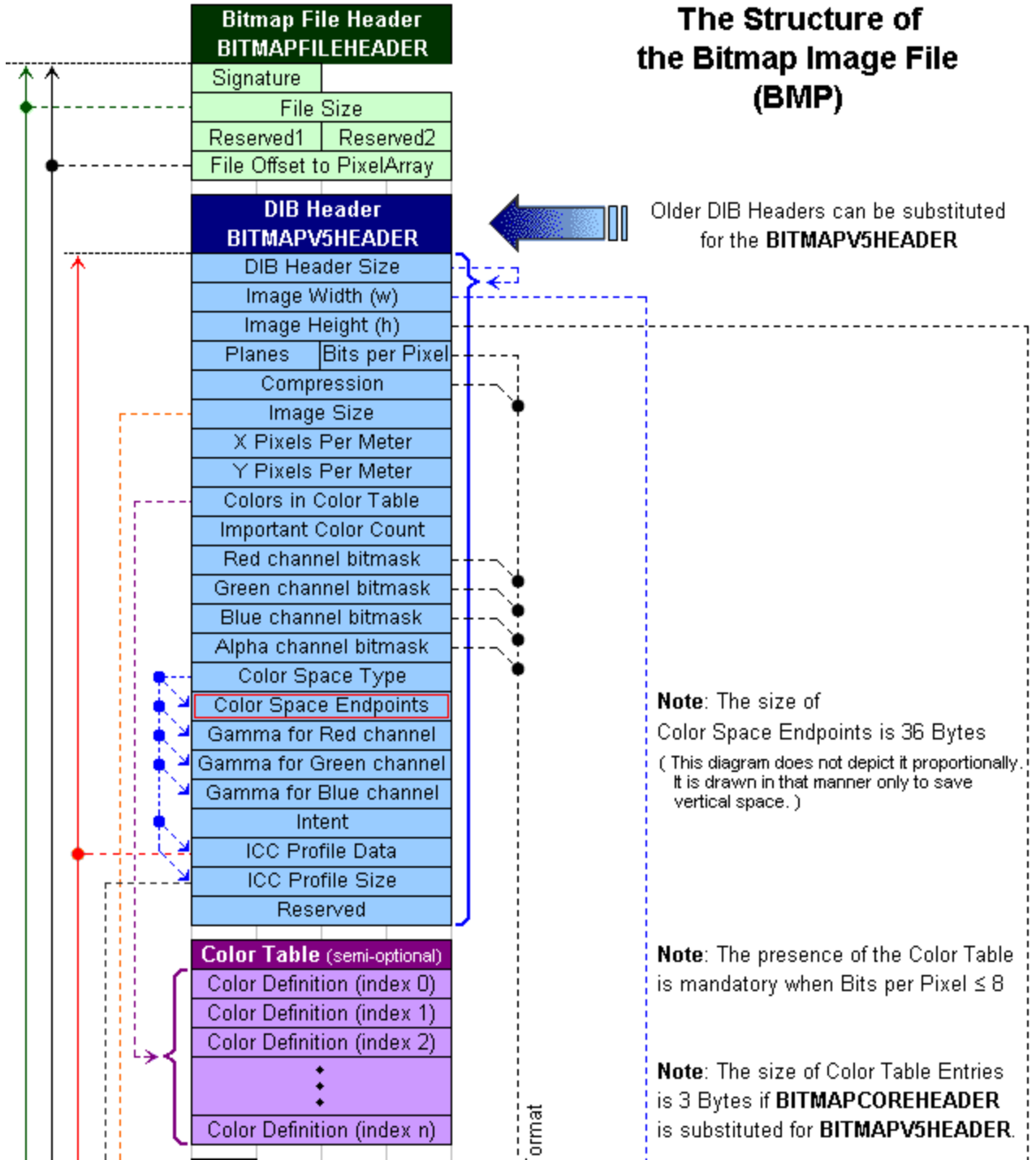
COMP 206 – Introduction to Software Systems

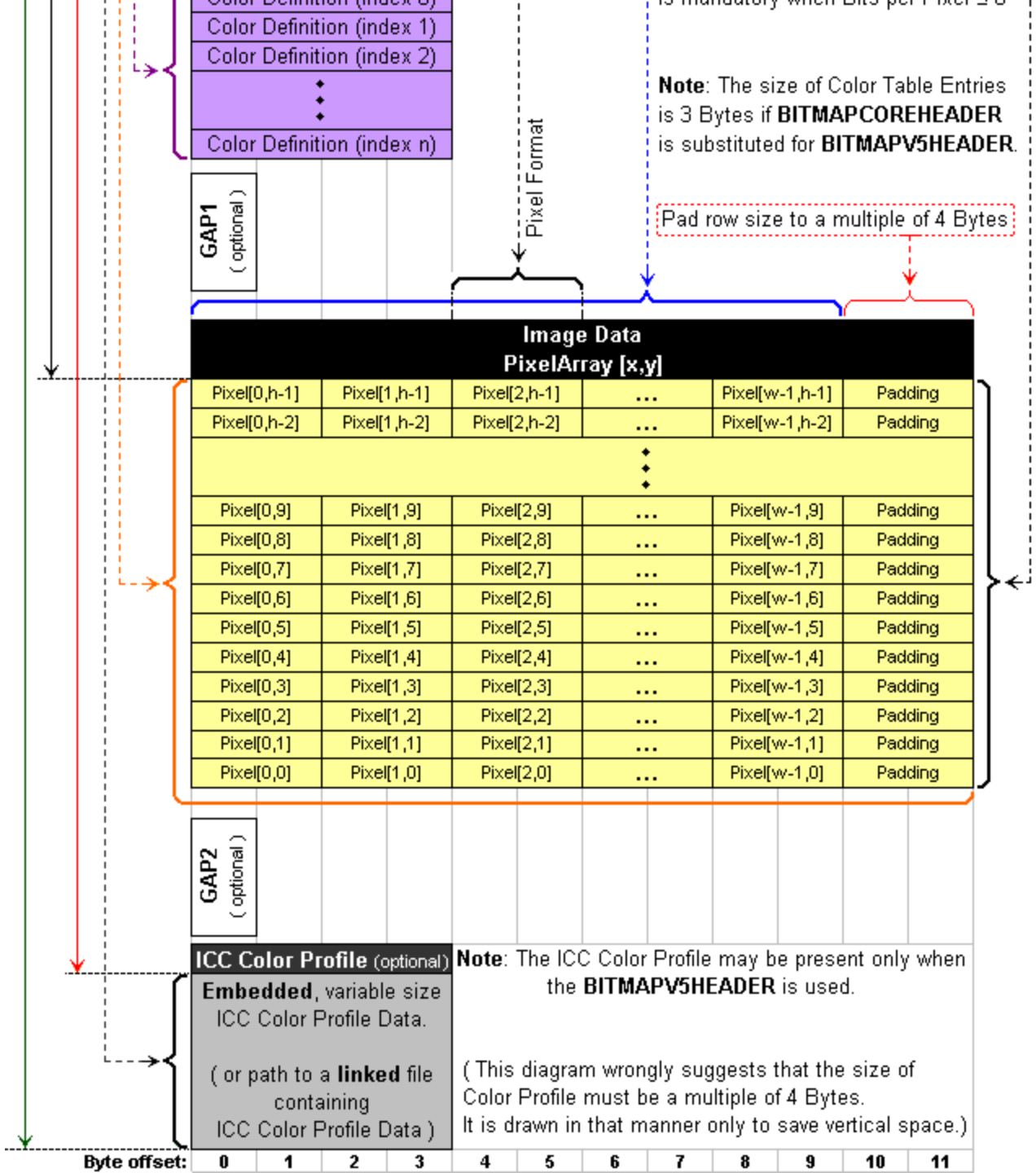
Lecture 13 – Working with BMP Images

October 12, 2018

Bitmap File (BMP) Example first half



Bitmap File (BMP) Example second half



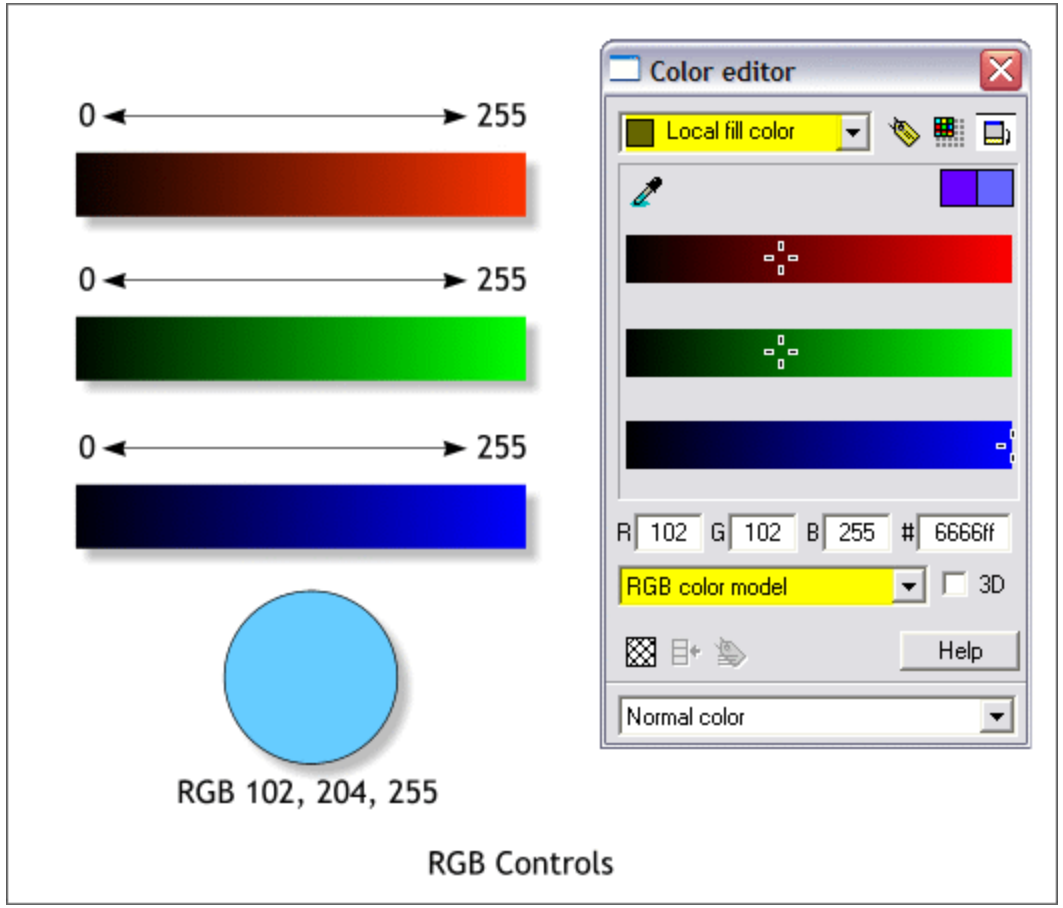
How can we read a BMP file using C?

- What works well:
  - Check the magic number:
    - If it matches very likely it follows the rules
  - File size field: makes it easy to access all of the data
  - Width and height, allows finding a specific pixel
  - Opening with code like “rb”
- What we must avoid:
  - Checking for AASCI code values: space, newline, etc
  - Attempting to use “atoi” “atof”, these are “aasci to …”
  - If we open with “r”, C will do some of this automatically
  - fgets, fscanf also typically bad choices

Now that we have the data…

- Each color of each pixel is stored as an integer between 0 and 255 (one byte)
- Easy to read these one at a time in C and store in unsigned char

- We want to use the least significant bit of this value to encode a bit of our secret message



ExampleCode: reader

```c
/* FILE: bmp_file_reader.c
 *
 * Uses the binary file reading toolkit such as fopen with "rb"
 * and fread rather than the string-based operations like fgets.
 * This is so that we dont have to worry about special characters
 * in the file and ensure we get all the header bytes.
 *
 * It should be a helpful start for Assignment 3 Question 1.
 */

#include <stdio.h>

int main(){

  // Open a binary bmp file
  FILE *bmpfile = fopen( "utah.bmp", "rb" );

  // Read the B and M characters into chars
  char b, m;
  fread (&b,1,1,bmpfile);
  fread (&m,1,1,bmpfile);

  // Print the B and M to terminal
  printf( "The first byte was: %c.\n", b );
  printf( "The second byte was: %c.\n", m );

  // Read the overall file size
  unsigned int overallFileSize;
  fread( &overallFileSize, 1, sizeof(unsigned int), bmpfile );
  printf( "The size was: %d.\n", overallFileSize );

  // Close the file, re-open it to be at the beginning, and read the entire contents
  fclose(bmpfile);
  bmpfile = fopen("utah.bmp", "rb" );

  char imageData[overallFileSize];
  fread( imageData, 1, overallFileSize, bmpfile );

  // Read the width size into unsigned int (hope = 500 since this is the width of utah.bmp)
        unsigned int* wp = (unsigned int*)(imageData+18);
  unsigned int width = *wp;

  // Print the width size to terminal
  printf( "The width is: %d.\n", width );

  return 0;
}
```