KWANGWOON UNIVERSITY

Color Conversion

2018.11.06

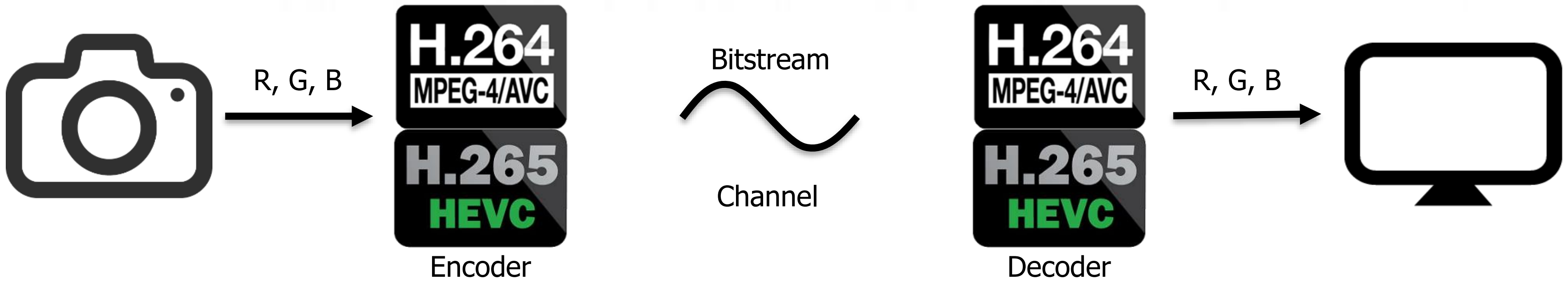
Seoungjun Oh(sjoh@kw.ac.kr)
Wooju Lee (krosea@kw.ac.kr)

Multimedia LAB

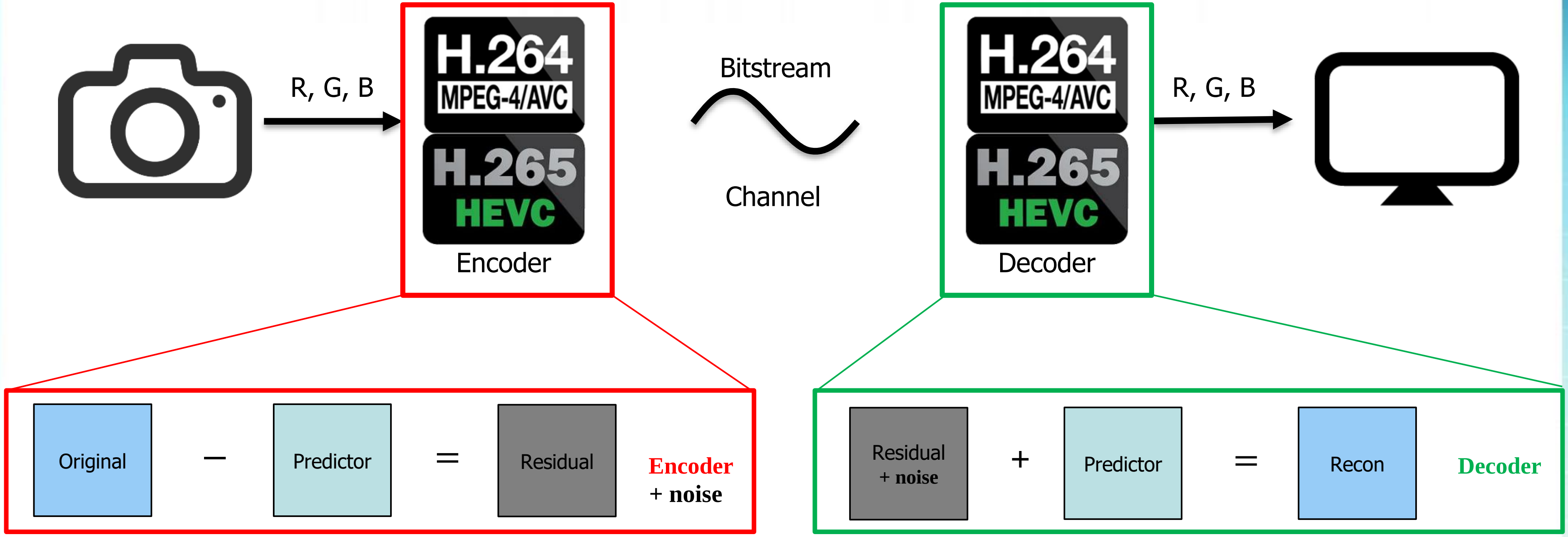
VIA-Multimedia Center, Kwangwoon University

Contents

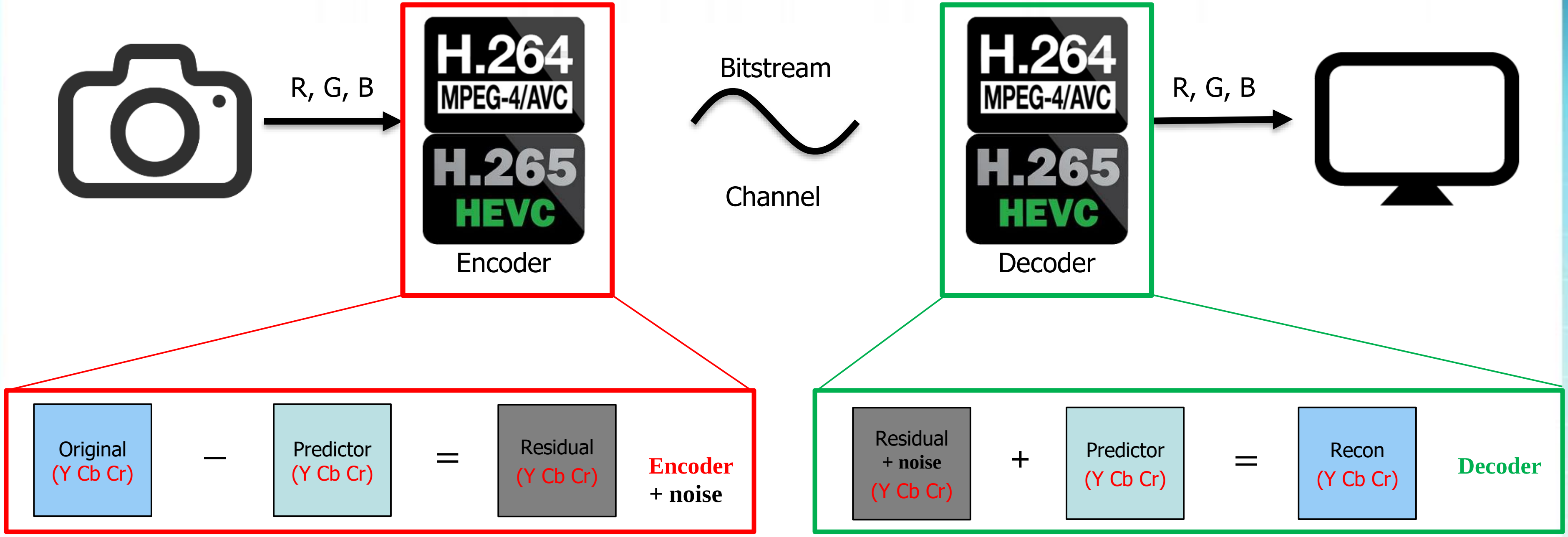
Color Conversion



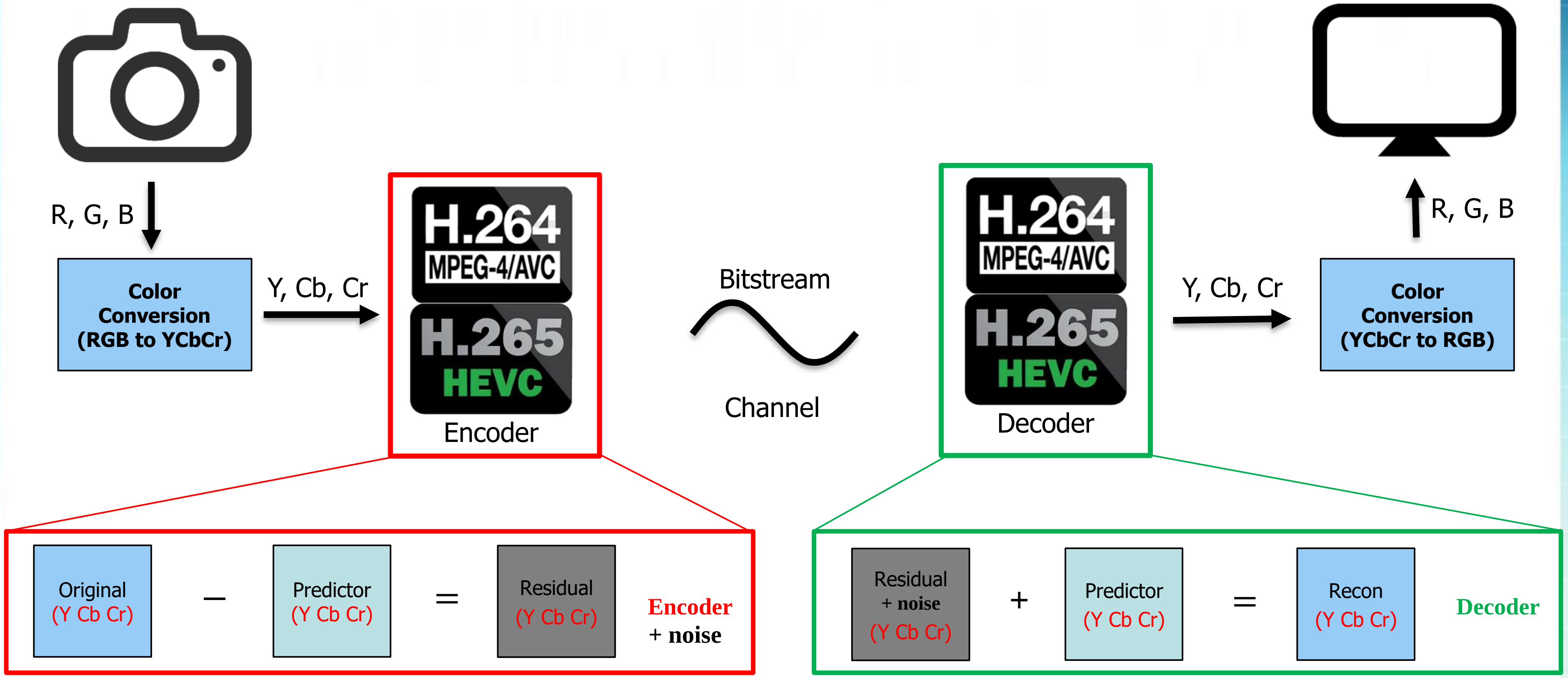
Color Conversion



Color Conversion



Color Conversion



RGB to YUV(YCbCr)

- Conversion from RGB:
  - Y=0.299(R-G) + G + 0.114(B-G)
  - Cb=0.564(B-Y)
  - Cr=0.713(R-Y)

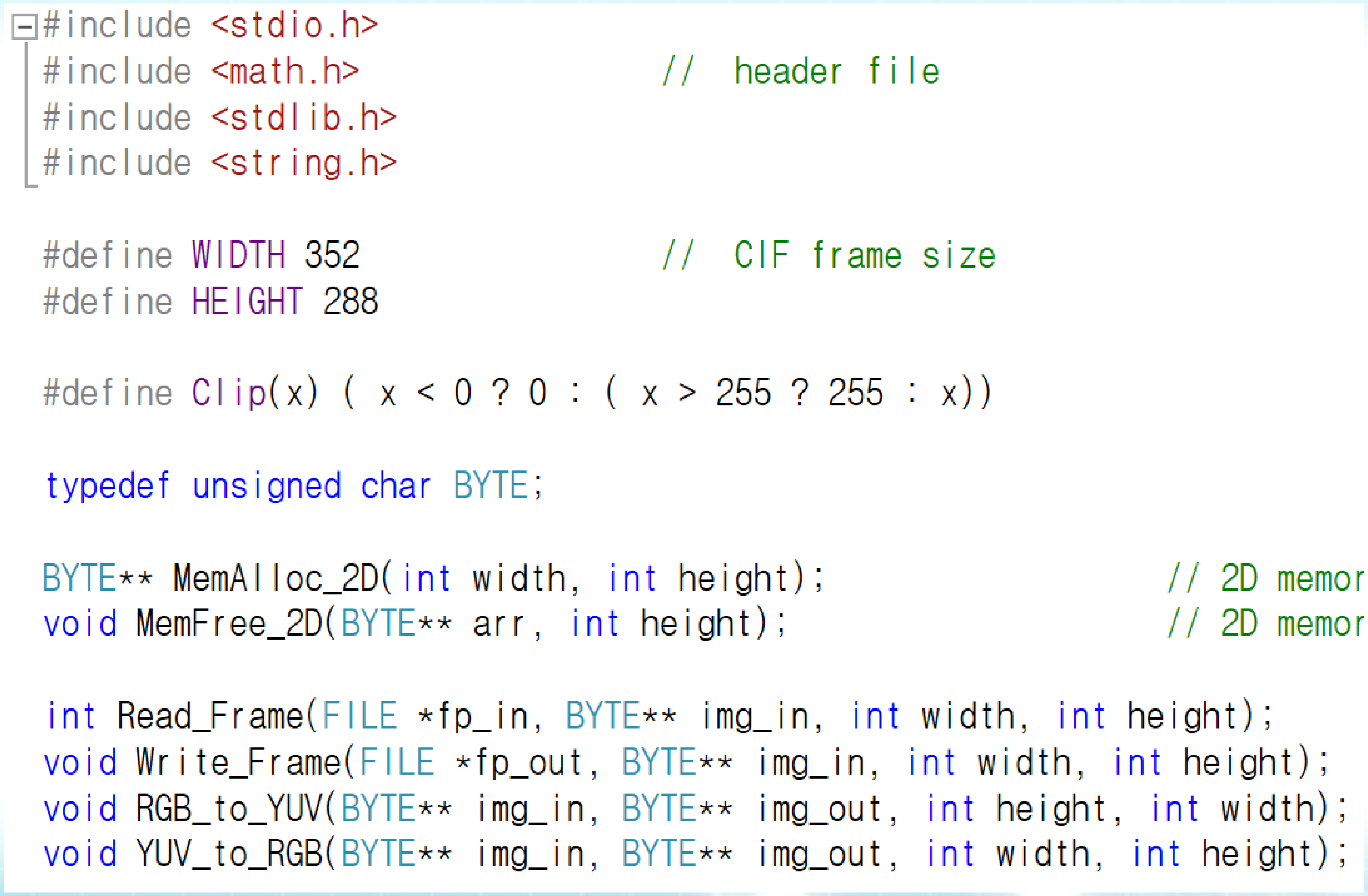
- The Matrix form:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168636 & 0.232932 & -0.064296 \\ 0.499813 & -0.418531 & -0.081282 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

RGB to YUV(YCbCr)

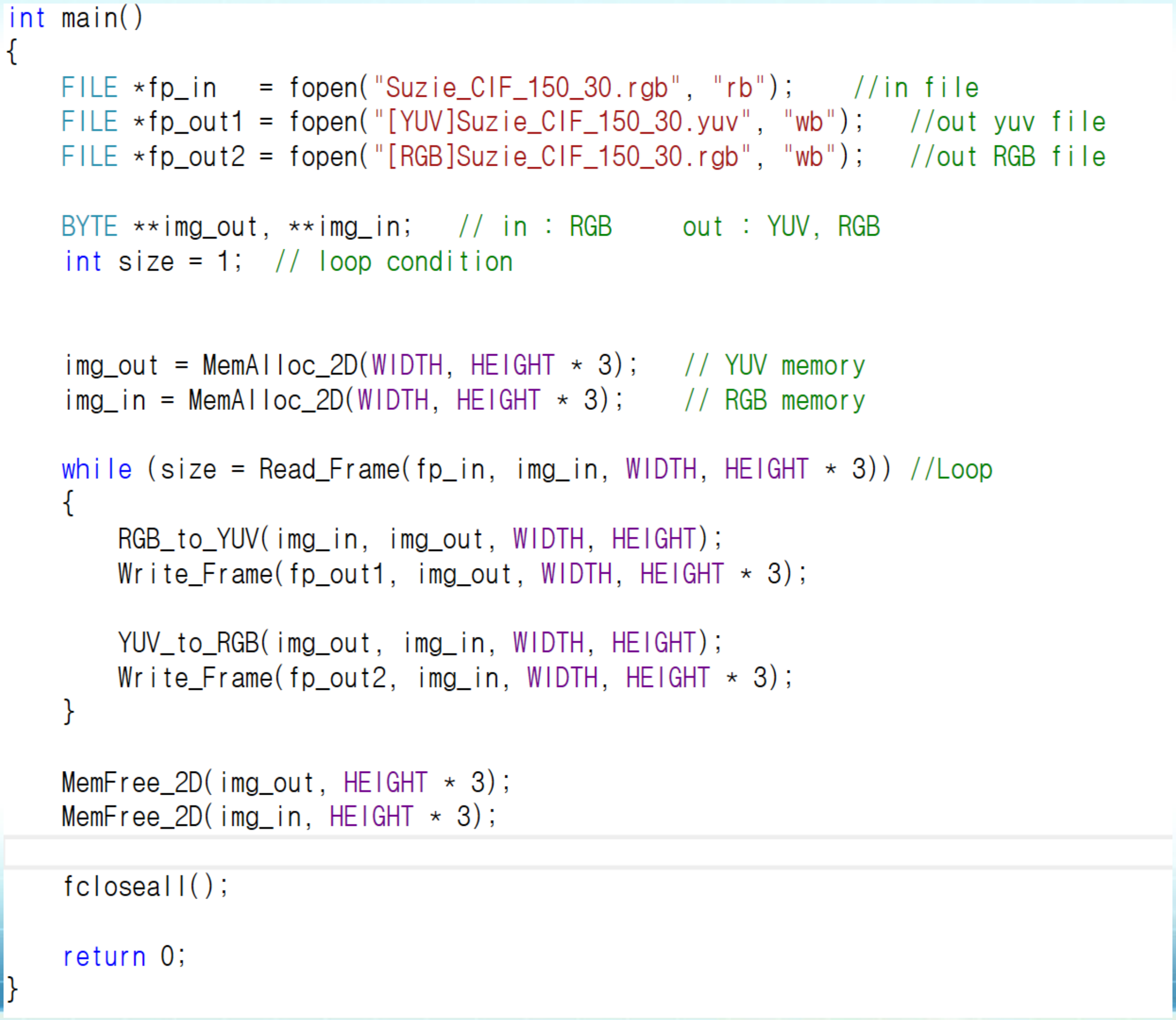
- RGB to YUV (integer)
  - $Y' = \left((66 \times R + 129 \times G + 25 \times B + 128) \gg 8\right) + 16$
  - $U = \left((-38 \times R - 74 \times G + 112 \times B + 128) \gg 8\right) + 128$
  - $V = \left((112 \times R - 94 \times G - 18 \times B + 128) \gg 8\right) + 128$
- YUV to RGB (integer)
  - $C = Y' - 16$
  - $D = U - 128$
  - $E = V - 128$
  - $R = clamp\left((298 \times C + 409 \times E + 128) \gg 8\right)$
  - $G = clamp\left((298 \times C - 100 \times D - 208 \times E + 128) \gg 8\right)$
  - $B = clamp\left((298 \times C + 516 \times D + 128) \gg 8\right)$

Example Code



```
#include <stdio.h>
#include <math.h>                 //  header file
#include <stdlib.h>
#include <string.h>

#define WIDTH 352                 //  CIF frame size
#define HEIGHT 288

#define Clip(x) ( x < 0 ? 0 : ( x > 255 ? 255 : x))

typedef unsigned char BYTE;

BYTE** MemAlloc_2D(int width, int height);                    // 2D memor
void MemFree_2D(BYTE** arr, int height);                      // 2D memor

int Read_Frame(FILE *fp_in, BYTE** img_in, int width, int height);
void Write_Frame(FILE *fp_out, BYTE** img_in, int width, int height);
void RGB_to_YUV(BYTE** img_in, BYTE** img_out, int height, int width);
void YUV_to_RGB(BYTE** img_in, BYTE** img_out, int width, int height);
```

Example Code

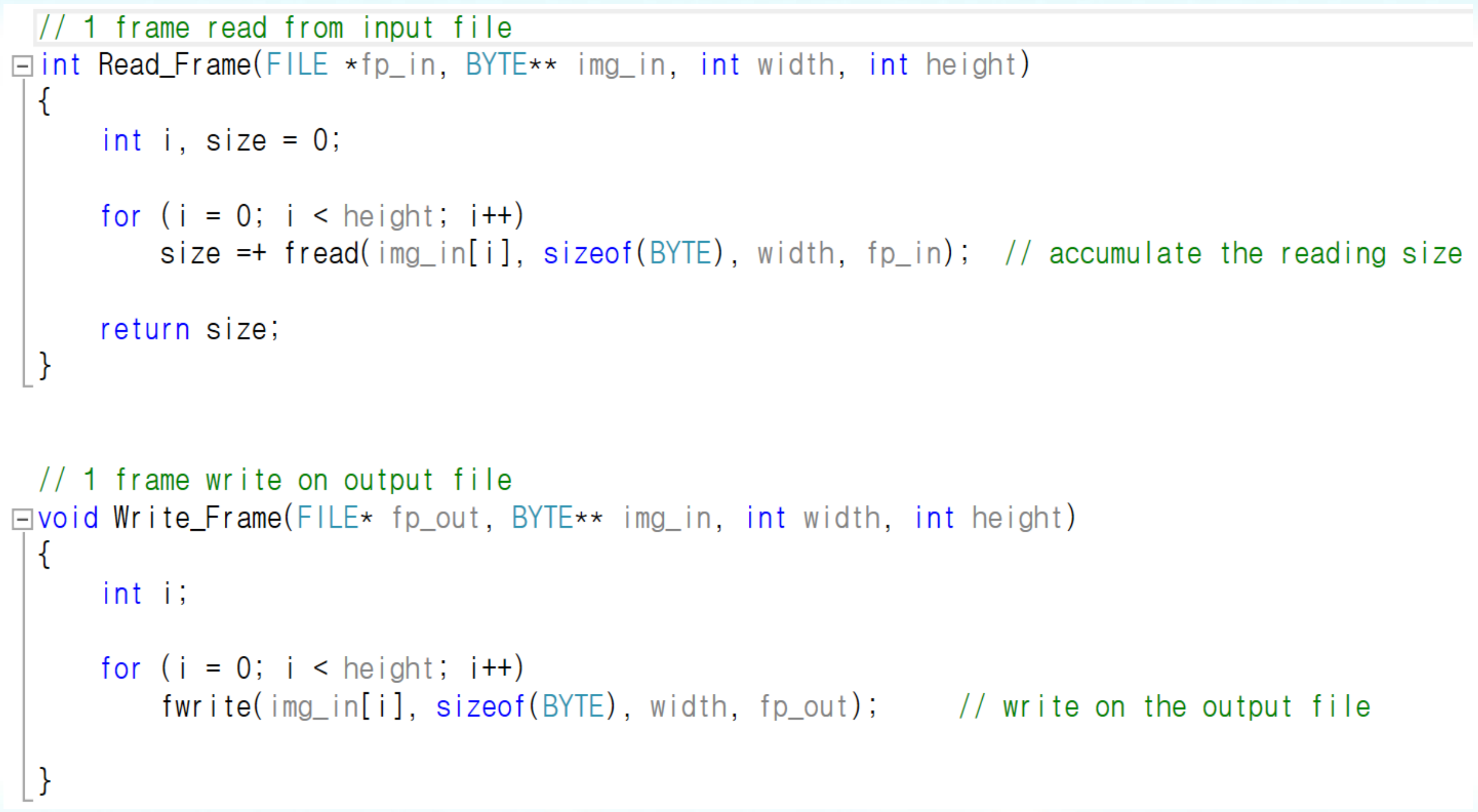


```c
int main()
{
    FILE *fp_in   = fopen("Suzie_CIF_150_30.rgb", "rb");    //in file
    FILE *fp_out1 = fopen("[YUV]Suzie_CIF_150_30.yuv", "wb");   //out yuv file
    FILE *fp_out2 = fopen("[RGB]Suzie_CIF_150_30.rgb", "wb");   //out RGB file

    BYTE **img_out, **img_in;   // in : RGB     out : YUV, RGB
    int size = 1;  // loop condition


    img_out = MemAlloc_2D(WIDTH, HEIGHT * 3);   // YUV memory
    img_in = MemAlloc_2D(WIDTH, HEIGHT * 3);    // RGB memory

    while (size = Read_Frame(fp_in, img_in, WIDTH, HEIGHT * 3)) //Loop
    {
        RGB_to_YUV(img_in, img_out, WIDTH, HEIGHT);
        Write_Frame(fp_out1, img_out, WIDTH, HEIGHT * 3);

        YUV_to_RGB(img_out, img_in, WIDTH, HEIGHT);
        Write_Frame(fp_out2, img_in, WIDTH, HEIGHT * 3);
    }

    MemFree_2D(img_out, HEIGHT * 3);
    MemFree_2D(img_in, HEIGHT * 3);

    fcloseall();

    return 0;
}
```

Example Code

```
BYTE** MemAlloc_2D(int width, int height)
{
    BYTE** arr;
    int i;

    arr = (BYTE**)malloc(sizeof(BYTE*)* height);
    for (i = 0; i < height; i++)
        arr[i] = (BYTE*)malloc(sizeof(BYTE)* width);

    return arr;
}

void MemFree_2D(BYTE** arr, int height)                    //  2D memory free
{
    int i;
    for (i = 0; i < height; i++){
        free(arr[i]);
    }
    free(arr);
}
```

Example Code



```
// 1 frame read from input file
int Read_Frame(FILE *fp_in, BYTE** img_in, int width, int height)
{
    int i, size = 0;

    for (i = 0; i < height; i++)
        size =+ fread(img_in[i], sizeof(BYTE), width, fp_in);  // accumulate the reading size

    return size;
}


// 1 frame write on output file
void Write_Frame(FILE* fp_out, BYTE** img_in, int width, int height)
{
    int i;

    for (i = 0; i < height; i++)
        fwrite(img_in[i], sizeof(BYTE), width, fp_out);     // write on the output file

}
```

Example Code

```
void RGB_to_YUV(BYTE** img_in, BYTE** img_out, int width, int height)
{

                              ?

}
void YUV_to_RGB(BYTE** img_in, BYTE** img_out, int width, int height)
{

                              ?

}
```