Implementation of YUV420 to RGB Image

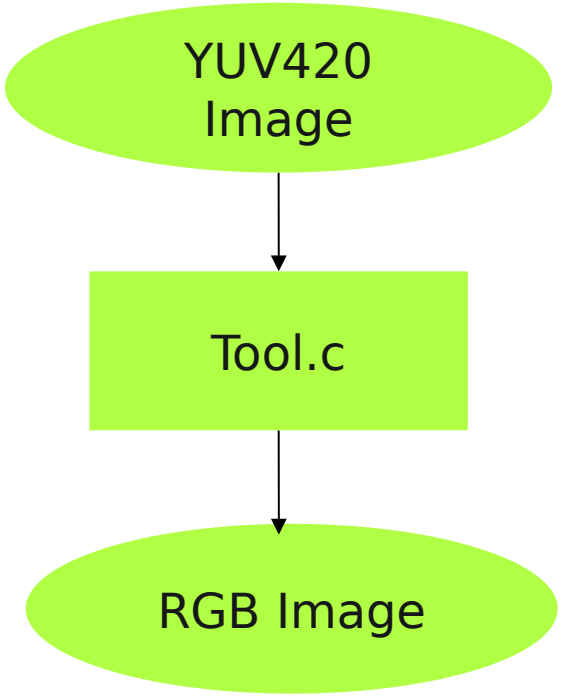
Mohd Wamique Husain
Emp Id:140655

Agenda

- Planning and Requirement Analysis
- Defining and Designing Product Architecture
- Implementation
- Testing
- Issues Faced

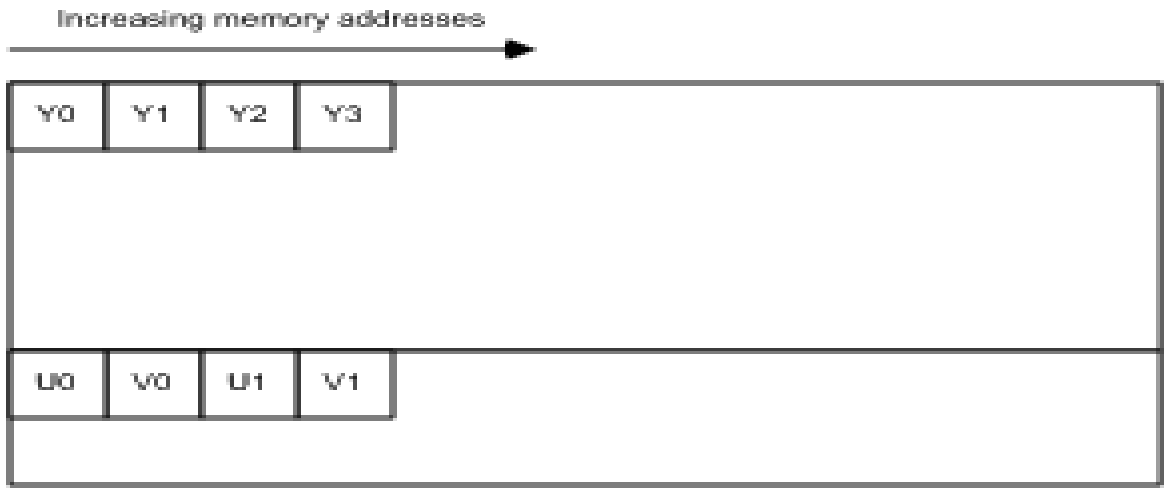
Planning and Requirement Analysis

- To implement a tool that converts YUV420 image to RGB image.
  - To Extract Data Information from Input i.e. YUV420
  - To process the input data
  - To generate RGB image



Defining and Designing Product Architecture

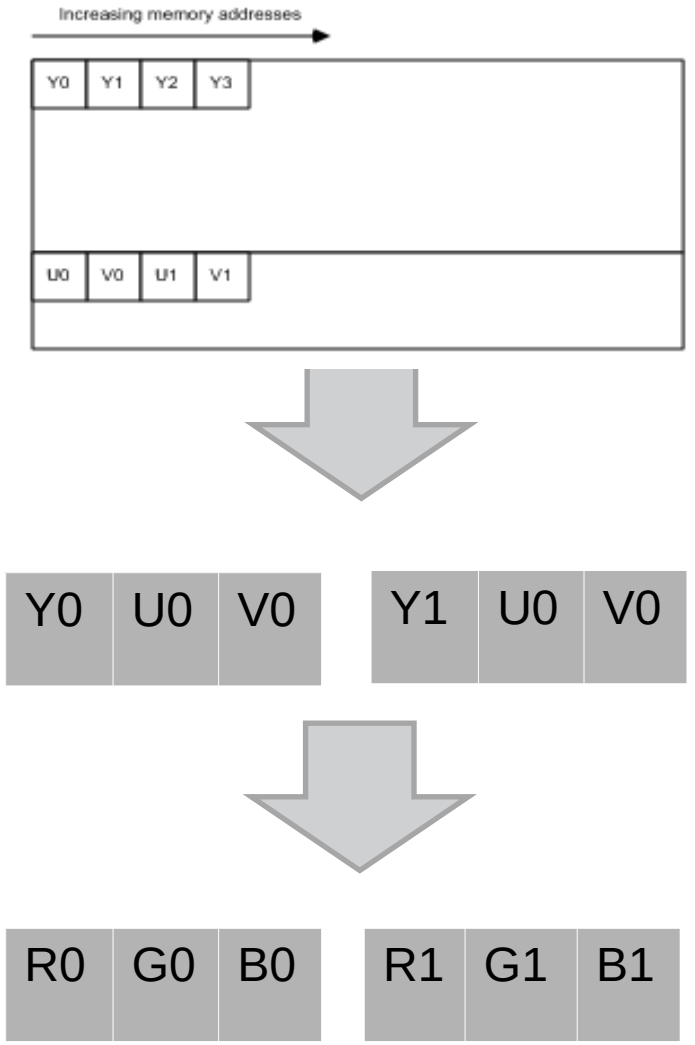
- Here as we know **Raw image** is just a binary file storing the image information in an **Array**.
- Provided Image is “NV12” type of yuv420.
- Here Image is stored in form of Planes or Semi-planer as shown below:
  - Y planes
  - U-V interleaved plane.



Basic Working of Tool

This diagram depicts the basic working of tool:

- As, tool first converts the provided Yuv420 Image to Yuv444 (**here one Chroma** samples are shared between **four pixels** while doing Up-sampling from Yuv420)
- Afterwards, Yuv444 is converted to RGB using the basic formula.



YUV 4:2:0:

- Four Pixel Information
- 6 Bytes
- Thus total size of image equals, (no of Pixel *1.5)

YUV 4:4:4:

- four Pixel Information
- 12 Bytes
- Thus total size of image equals, (no of Pixel *3)

RGB 8:8:8:

- Four Pixel Information
- 12 Bytes
- Thus total size of image equals, (no of Pixel *2)

Implementation

```
void Yuv420ToYuv444(uint8_t Yuv420Data[],uint8_t Yuv444Data[],int width,int height){

    int i=0,k=0,j=0,l=0;

    /*pointer to process Y data*/
    uint8_t *Luma=Yuv420Data;

    /*pointer to process UV data*/
    uint8_t *Chroma=&Yuv420Data[width*height];

    /*Total no.of bytes in width of image (1280 pixels)*/
    int Yuv444width=width*3;

    /*In this loop we are fillng "Y" data from yuv420 Image to yuv444 Image buffer(leaving spaces for U-V)*/
    for(i=0;i<width*height*3;i=i+3)
    {
        Yuv444Data[i]=Luma[i/3];

    }
    /*In this loop we are fillng "U-V" data from yuv420 Image to yuv444 Image buffer
     Here, i is used to traverse height of image , j is used to traverse width of image,
     l is used to traverse alternate couloumn of image
     */
    for(i=0;i<height;i=i+2){
        for(j=1;j<width*3;j=j+6){
            for(l=i;l<(i+2);l++){
                Yuv444Data[(Yuv444width*l)+j]=Chroma[k];
                Yuv444Data[(Yuv444width*l)+j+1]=Chroma[k+1];
                Yuv444Data[(Yuv444width*l)+j+3]=Chroma[k];
                Yuv444Data[(Yuv444width*l)+j+4]=Chroma[k+1];
            }
            k=k+2;
        }


    }
}
```

```
void Yuv444ToRGB(uint8_t RGBBuff[],int width,int height){

    uint8_t c=0,d=0,e=0,R=0,G=0,B=0;
    int i=0;
    for(i=0;i<(width*height*3);i=i+3){
        c=RGBBuff[i];
        d=RGBBuff[i+1];
        e=RGBBuff[i+2];
        /*formal Used for the conversion of YUV To RGB*/
        RGBBuff[i]=clamp(c+1.402*(e-128));
        RGBBuff[i+1]=clamp(c-0.344*(d-128)-0.714*(e-128));
        RGBBuff[i+2]=clamp(c+1.772*(d-128));
    }

}
```

```
uint8_t clamp(int16_t ch){
if(ch>255)
    return 255;
if(ch<0)
    return 0;

return ch;

}
```

Testing

- Tested with “vooya” with RGB setting’s only and It works as expected
- checked Pixel’s details and it only contains RGB details

Issues Faced

- I was getting some of the pixels of image either Full Blue or Green.
  - Then I degugged a little and found that negative value of RGB is rolling back to positive number as they are of unsigned int type. Example “-2” equals “254” in unsigned int.
  - To solve this I have used “int16_t” in the clamp function and picked “-2” as “-2” and done the further processing.


KPIT
Reimagining Mobility *with YOU*
Thank you