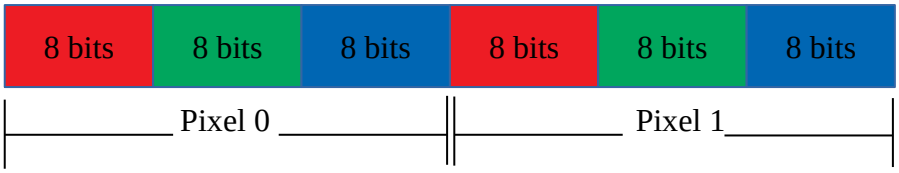
Results RGB to YUV using C++/Intrinsics

✦ Image formats required

- Input is RGB888 (RGB24):

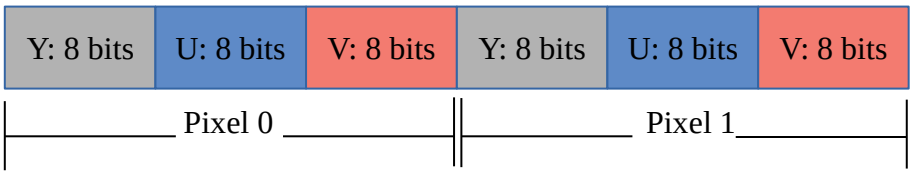
  - Pixels of 24 bits: 8 bits of red, 8 bits of green and 8 bits of blue.
  - Packed format.
  - Without alpha component.
  - Image size of 640x480 pixels.



RGB888 color format

- Output is YUV444:

  - Pixels of 24 bits: 8 bits of Y, 8 bits of U(C'b) and 8 bits of V(C'r).
  - Packed format.
  - Without alpha component.
  - Image size of 640x480 pixels.



YUV444 color format

✦ Algorithm description

The algorithm implemented first take all bits of the input file (The name of the file is stored in the variable input_image) and store all of them into an array of chars. Next, a casting is applied over that array to get the same values, but stored into an array of floats.

That array has the three components (RGB) of each pixel of the image.

The next step is to apply the conversion for each pixel from RGB to YUV, to do that, only is needed to apply the following matricial operation.

$$Y = (0.257 * R) + (0.504 * G) + (0.098 * B) + 16$$

$$U = Cb = -(0.148 * R) - (0.291 * G) + (0.439 * B) + 128$$

$$V = Cr = (0.439 * R) - (0.368 * G) - (0.071 * B) + 128$$

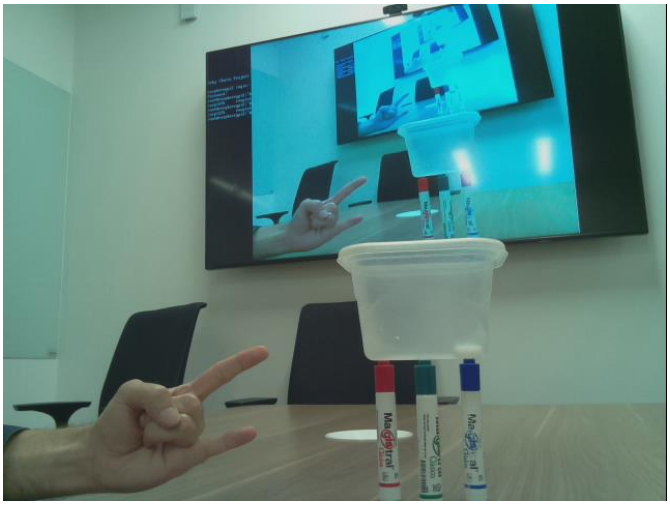
Taking advantage of the NEON intrinsics instructions, the mathematics were applied to vectors, so 4 pixels were processed at the same time. These are the intrinsics instructions used:

- vld1q_f32(): used to load values into the intrinsics data structures.
- vmulq_f32(): multiply intrinsics data structures.
- vaddq_f32(): add intrinsics data structures.
- vst1q_f32(): load intrinsics data structures into C++ data structures.

Each new YUV pixel is stored in a new array that then is written to the new image in YUV format.

✦ Sample image

- Input image (RGB888):



Select RAW data: Choose File sample_640x480.rgb width:640 height:480 offset:0 flip h: flip v: invert: zoom: 1

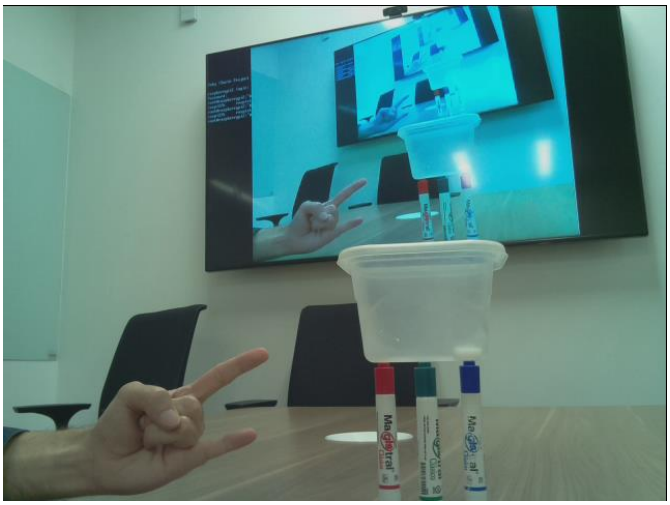
Predefined format: RGB24

Pixel Format: RGBA Ignore Alpha: Alpha First:

bpp1:8 bpp2:8 bpp3:8 bpp4:0 Litle Endian:

Pixel Plane: Packed alignment:1 subsamplig H:1 subsamplig V:1

- Output image (YUV444):



Select RAW data: Choose File sample_640x480.yuv width:640 height:480 offset:0 flip h: flip v: invert: zoom: 1

Predefined format: YUV444p

Pixel Format: YUV Ignore Alpha: Alpha First:

bpp1:8 bpp2:8 bpp3:8 bpp4:0 Litle Endian:

Pixel Plane: Packed alignment:1 subsamplig H:1 subsamplig V:1

✦ Processing time

Rgb2yuv-intrinsics was executed 5 times to get the average processing time. The following table summarize the results.

	Intrinsics	
1	51236	us
2	49841	us
3	49693	us
4	49427	us
5	49584	us
Average	49956,2	us