MSX Article

3D Vision on MSX

Summary

This article proposes to show how to generate a 3D image on MSX using the stereo technique called anaglyph.

1- Introduction

The human visual system is composed by a 3D vision mechanism, where two eyes horizontally displaced by a certain distance capture a scene from two different points of view.

If we compare each image captured by the eyes, the objects nearer from the observer present stronger displacement than the farer ones. These are the fundamentals used for the brain to build a 3D virtual environment.

We can simulate this technique for a given scene:

1. Take a shot from that scene, corresponding to the left eye's point of view.
2. Take a shot from that scene, corresponding to the right eye's point of view.
3. Apply some kind of filter on the left image that blocks the image to the right image, but allows the left eye to see it.
4. Apply some kind of filter on right image that blocks the image to the left image, but allows the right eye to see it.

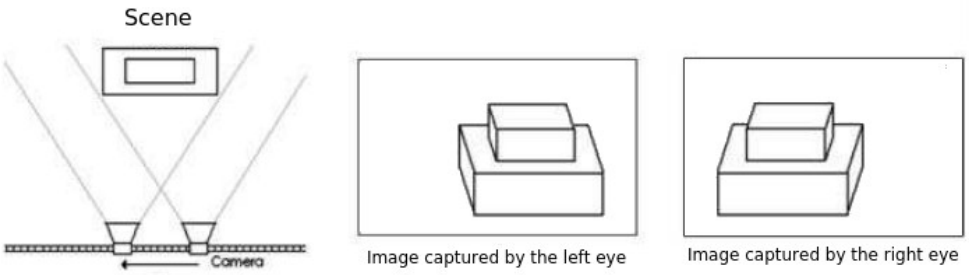


Figure 1. 3D human vision simulation. Adapted from [2].

There are several techniques for image filtering:

- Image separation (pocket stereoscopic, stereoscopic)
- Anaglyph
- Active polarization (shutter glass)
- Passive polarization

The most popular technique is the anaglyph. The anaglyph is based on the image's chromatic information and it is consisted by a pair of glasses with red and blue (cyan) lens, where each len is a filter that allows that only the correct information (color) reaches the corresponding eye.

Figure 2. Anaglyph glasses.

Each stereo image should emit only the colors corresponding to its filter. On figure 2, the left len has a red filter while the right len has a cyan (green + blue) filter.

A RGB image has three primary component colors (additive): red, green and blue. So, if we remove the green and blue components from an image, by setting all pixels' GB values to 0, that image would be visible only to the anaglyph's left eye. In the other hand, if we remove the red component, only the right eye would see that image.

Once the red-cyan filters have complementary colors, we can fuse the two stereo images into one image. The left image takes the red component, while the right image takes the green and blue components.

The anaglyph image is generated as follows:

- Apply red filter on the left image: $I_{LF} = I_L(R_L, 0, 0)$
- Apply cyan filter on the right image: $I_{RF} = I_R(0, G_R, B_R)$
- Compose images: $I = I_{LF}(R_L, 0, 0) + I_{RF}(0, G_R, B_R) = I(R_L, G_R, B_R)$

2- Image filtering on MSX

On MSX 2 screens from 2 to 7, the color of each pixel is controled by a value that represents and index to a table that contains the real color of that pixel. This is the palette system. So, whenever we change the RGB values of an index *i*, this change is reflected imediatelly to all pixels that are represented by the value *i*.

At screen 8, the color of each pixel is controlled by a value that represents the real color. According to that, we can apply some color operations on a single pixel, on areas or on the whole image.

Although it is possible to apply filters on the palette system, it is not possible to handle with individual pixels. In that case, we could not apply at the same time the red filter on the left image and the cyan filter on the right image. In addition, the screen 8 has 256 colors, while the other screens have only 16 colors. So, the screen 8 is more suitable for our experience.

Each pixel is represented by an 8-bit RGB code, where:

color	**g**	**g**	**g**	**r**	**r**	**r**	**b**	**b**
bit	7	6	5	4	3	2	1	0

bb - the two first bits (0-1) control the blue color component.
rrr - the bits from 2 to 4 control the red color component.
ggg - the bits from 5 to 7 control the green color component.

How do we represent a RGB pixel P, where P=(7,5,2)?

Remember that:

Binary	000	001	010	011	100	101	110	111
Decimal	0	1	2	3	4	5	6	7

Then, we have:

Red: 7 or 111
Green: 5 or 101
Blue: 2 or 10

The final binary value would be: 10111110.
Where is represented in decimal as 190.

The value 0 is the lowest intensity (dark), while the maximum value (111 or 11) is the highest intensity (bright).

We can filter the pixel P in order to remove the red component. In that case, the resulting pixel should be: 10100010. The operator AND is used to filter P.

The logic operation “A AND B” establishes that each bit in A is preserved if B is 1, while each bit is set to 0, if B is 0. In this case, A is the pixel, while B is a mask.

Let's take a look on this example:

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A	1	1	0	0	1	0	0	1
B	1	0	0	1	1	0	1	1
A AND B	1	0	0	0	1	0	0	1

It is clear now how to create a mask to remove the red component. So:

Color	**g**	**g**	**g**	**r**	**r**	**r**	**b**	**b**
P	1	0	1	1	1	1	1	0
Mask	1	1	1	0	0	0	1	1
$P_{filtered}$	1	0	1	0	0	0	1	0

The Basic binary notation for this mask is: &B11100011.

The green and blue components are left after removing the red component. The green and blue combination results on the cyan. In other words, this filter blocks the red color and allows the cyan pass. This is the cyan filter.

Although the most common filters used are the red-cyan, other filters can be used. The following table shows all the filters that can be applied on an image:

Filter	Mask	Blocks	Allows
Cyan	&B11100011	Red	Cyan
Magenta	&B00011111	Green	Magenta
Yellow	&B11111100	Blue	Yellow
Red	&B00011100	Cyan	Red
Green	&B11100000	Magenta	Green
Blue	&B00000011	Yellow	Blue

We can always use complementary color components on stereo images like “R and GB”, “R and B”, “RG and B” etc, but never use filters that have the same color components like “RG and GB” or “RB and B”, once they would be visible in both images.

The next program in Basic demonstrates how to apply a filter on an image.

filter.bas

```
10 SCREEN 8
20 MA=&B11100011
30 LINE(0,0)-(255,211),255,BF
40 LINE(0,0)-(255,211),MA,BF,AND
50 GOTO 50
```

This program draws a white screen and then the filter is applied on it. For other filters, replace the mask “MA” value (line 20) to another one from the table.

3- Anaglyph on MSX

The previous section discussed on how to filter images. Now, we present the steps necessary to generate a stereo anaglyph image on MSX.

Steps to generate anaglyph on MSX:

1. Start the screen 8
2. Load the left image on page 1
3. Apply the red filter on it
4. Load the right image on page 0
5. Apply the cyan filter on it
6. Compose the images

The logical operator OR will be used to compose the images.

4- An 3D example

In aerial photogrammetry, the human vision system is reproduced as if there was a “giant” looking down to the city.

From [1], we took a stereo pair images from the Maracana neighborhood, located in Rio de Janeiro, Brazil. The object detached on both images is the Brazilian famous soccer stadium Maracana. The images were converted to screen 8 format using MSX Viewer 5.

All the sources as well as the images can be found at MarMSXpage, section Articles, together with this article.

Left eye image – MRCLEFT.S08

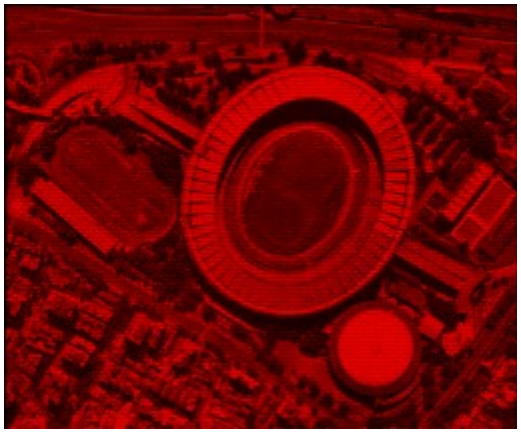
Right eye image – MRCRIGHT.S08

The next program applies on each image the proper filter and then fuses them into a single image.

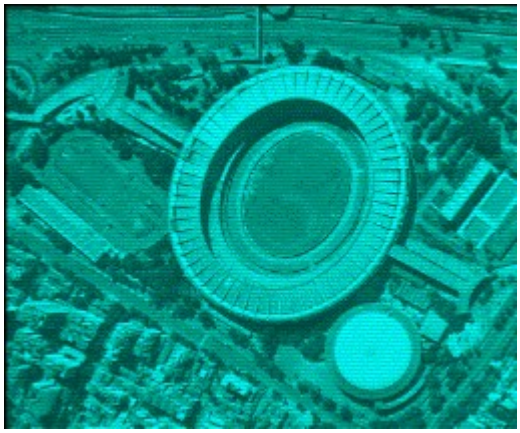
maracana.bas

```
10 SCREEN 8
20 SET PAGE 1,1
30 BLOAD"MRCLEFT.S08",S
40 LINE(0,0)-(255,211),&B00011100,BF,AND
50 SET PAGE 0,0
60 BLOAD"MRCRIGHT.S08",S
70 LINE(0,0)-(255,211),&B11100011,BF,AND
80 COPY(0,0)-(255,211),1 TO (0,0),0,OR
90 GOTO 90
```

MSX 2 is quite versatile with logical operations on images. So, a filter is applied on each image using the the logical operator AND combined with the proper mask in order to remove the color components not used.

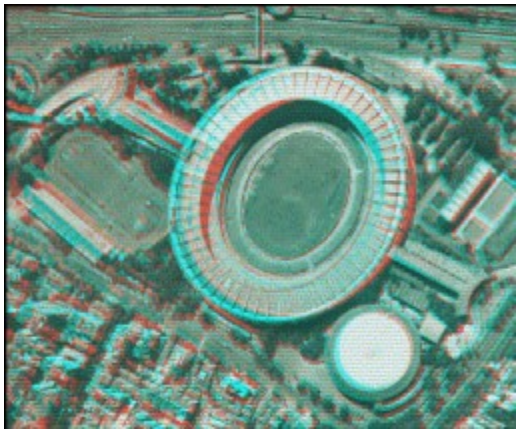


Red filter applied on the left image



Cyan filter applied on the right image

To compose the final image, the left image was combined with the right image using the logical filter OR.



Resulting anaglyph image.

Do not forget to wear your anaglyph glasses to see this image in 3D.

5- Credits and references

This article was written originally in Portuguese and translated into English by Marcelo Silveira on October 2016.

E-mail: flamar98@hotmail.com

References:

[1] Projeto E-FOTO: http://www.efoto.eng.uerj.br
[2] Book: Fotogrametria Digital, Brito e Coelho Filho, Editora EdUerj, 2007.