

MSX Article

MARMSX

Visão 3D no MSX

Resumo

O objetivo deste artigo é mostrar como é possível simular uma imagem 3D no MSX, através da técnica de anaglifo.

1- Introdução

O sistema visual humano possui um mecanismo de visão 3D, no qual é constituído de dois olhos, deslocados entre si de uma certa distância no sentido horizontal, em que se permite observar uma cena a partir de dois pontos de vista diferentes.

Ao compararmos cada imagem captada pelos olhos, os objetos situados mais perto do observador irão sofrer um deslocamento aparente maior do que aqueles situados mais distante. É dessa forma que o cérebro consegue construir um ambiente 3D virtual.

Podemos “imitar” essa técnica, em uma dada cena, da seguinte maneira:

1. Tirar uma fotografia de uma cena, correspondente ao ponto de vista do olho esquerdo.
2. Tirar uma fotografia de uma cena, correspondente ao ponto de vista do olho direito.
3. Aplicar algum tipo de filtro que permita que somente o olho esquerdo enxergue a imagem da esquerda, bloqueando-a para o olho direito.
4. Aplicar algum tipo de filtro que permita que somente o olho direito enxergue a imagem da direita, bloqueando-a para o olho esquerdo.



Figura 1. Simulação da visualização 3D humana. Adaptado de [2].

Há diversas técnicas para se conseguir o efeito de filtragem para os olhos como por exemplo:

- Separação espacial
- Anaglifo
- Polarização ativa (shutter glass)
- Polarização passiva

A técnica mais simples e mais comumente empregada é o anaglifo. O anaglifo consiste naqueles óculos vermelho e azul (ciano), onde cada lado do óculos possui um filtro que irá permitir a passagem de informação (luz) somente da imagem correspondente àquele olho.



Figura 2. Exemplos de óculos anaglifo.

Para se conseguir atingir esse objetivo, cada imagem deverá emitir luz somente na cor correspondente ao seu filtro. Observe na figura 2, que o olho esquerdo possui o filtro vermelho, enquanto que o olho direito possui o filtro ciano (verde + azul).

Uma imagem colorida possui 3 componentes de cor, a saber: vermelho, verde e azul. Se pegarmos a imagem da esquerda e “apagarmos” toda a informação dos componentes verde e azul, isto é, fazer como que todos os *pixels* da imagem possuam valor igual a 0 nesses componentes, essa imagem será visível apenas para o olho esquerdo, uma vez que os componentes verde e azul não emitirão luz. De forma análoga, se pegarmos a imagem da direita e “apagarmos” toda a informação do componente vermelho, somente o olho direito enxergará a imagem.

Podemos “fundir” as imagens, pois em cada imagem, um ou dois componentes inteiros de cor foram “desativados”. Além do mais, os componentes apagados são complementares nas duas imagens. Por exemplo, enquanto que na imagem da direita o vermelho foi desativado, na imagem da esquerda está ativo. O mesmo se dá com os canais verde e azul, só que de forma invertida. Assim, o truque consiste em colocar no(s) componente(s) vazio(s) de uma imagem, a informação da outra imagem.

A imagem em anaglifo será composta da seguinte maneira:

- Aplicar filtro vermelho na imagem da esquerda: $I_{EF} = I_E(R_E, 0, 0)$
- Aplicar filtro ciano na imagem da direita: $I_{DF} = I_D(0, G_D, B_D)$
- Fundir imagens: $I = I_{EF}(R_E, 0, 0) + I_{DF}(0, G_D, B_D) = I(R_E, G_D, B_D)$

2- Filtragem de imagens no MSX

Sabe-se que nas screens de 2 a 7, cada pixel (ponto na tela) é representado por um valor, que corresponde a um índice para uma tabela contendo um determinado código de cor R,G,B (vermelho, verde e azul). Esse sistema é a paleta de cores. Quando se altera os valores R,G,B da cor de um determinado índice dessa tabela, a alteração é também refletida para todos os pixels da imagem que possuem o valor desse índice, mesmo até para as imagens que estejam em outras páginas.

Já na screen 8, cada pixel possui o seu próprio código de cor. Assim, as operações nas cores podem ser individualizadas tanto para um pixel, como para uma região da imagem ou também para toda a imagem.

Conforme visto na seção anterior, é necessário modificar cada imagem original, de forma a atribuir o valor igual a zero ao canal de cor que se necessita “apagar”. Observa-se

que essa modificação é particular para cada imagem, ou seja, cada imagem será filtrada de uma maneira distinta da outra. Logo, a screen 8 é a única em que isso pode de ser feito.

O código de cor para cada pixel na screen 8 é um número de 8 bits, onde:

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

bb - Os dois primeiros bits (0-1) controlam a componente de cor azul.

rrr - Os bits de 2 a 4 controlam a componente de cor vermelha.

ggg - Os bits de 5 a 7 controlam a componente de cor verde.

Como seria a representação R,G,B de um pixel P onde $P=(7,5,2)$?

Lembrando-se que:

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

Temos:

Vermelho: 7 ou **111**

Verde: 5 ou **101**

Azul: 2 ou **10**

A composição final da cor para a screen 8 será: **10111110**.

Esse valor convertido para decimal fica 190.

Observe que a intensidade de vermelho nesse exemplo possui valor máximo, ou seja, o vermelho brilha forte. Entretanto, podemos filtrar o pixel P, de forma que a cor vermelha seja apagada. Dessa forma, o valor do pixel resultaria em: **10100010**.

Para modificar o pixel P, de forma a “apagar” o canal vermelho, será utilizado o operador lógico AND.

Na operação lógica “A AND B”, para cada bit igual a um de B, o bit correspondente de A é mantido, enquanto que para cada bit igual a zero de B, o valor do bit correspondente de A é zerado. Nesse caso, B é chamado de máscara, pois decide o destino dos bits de A.

A correspondência de A e B é feita com os bits de mesma ordem, ou seja, bit 0 de A com bit 0 de B, bit 1 de A com bit 1 de B etc.

Ex:

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

Agora ficou mais claro qual é a máscara a ser utilizada para apagar somente o canal vermelho. Os bits de 2 a 4 da máscara, deverão possuir valor igual a 0. O restante, valor igual a 1.

Assim, tem-se:

Cor	g	g	g	r	r	r	b	b
P	1	0	1	1	1	1	1	0
Máscara	1	1	1	0	0	0	1	1
P_f	1	0	1	0	0	0	1	0

A representação dessa máscara em Basic fica: &B11100011.

Quando elimina-se o canal vermelho de um pixel, resulta em um pixel com dois canais apenas: verde e azul. A combinação de verde com azul forma o ciano. Dessa forma, esse processo nada mais é do que um filtro que bloqueia o vermelho e permite a passagem do ciano. Assim, o filtro será denominado pelas componentes de cor em que permite passagem de luz. Nesse caso, é o filtro ciano.

A tabela a seguir mostra os principais filtros que podem ser aplicados sobre uma imagem:

Filtro	Máscara	Bloqueia	Permite
Ciano	&B11100011	Vermelho	Ciano
Magenta	&B00011111	Verde	Magenta
Amarelo	&B11111100	Azul	Amarelo
Vermelho	&B00011100	Ciano	Vermelho
Verde	&B11100000	Magenta	Verde
Azul	&B00000011	Amarelo	Azul

Um teste simples em Basic demonstra como funcionam os filtros

filtro.bas
<pre> 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 </pre>

Uma tela toda branca é desenhada e depois o filtro é aplicado. Para testar outros filtros, substitua na linha 20 o valor da máscara “MA” por outros da tabela.

3- O Anaglifo no MSX

Foi visto na seção anterior, como filtrar as imagens de forma a prepará-las para a visão 3D. Agora, serão apresentados os passos necessários para se atingir esse objetivo.

Passos para o anaglifo no MSX:

1. Iniciar a screen 8
2. Carregar a imagem do olho esquerdo na página 1
3. Aplicar o filtro vermelho
4. Carregar a imagem do olho direito na página 0
5. Aplicar o filtro ciano
6. Fundir as imagens

Os filtros utilizados deverão ser os mesmos dos óculos anaglifo, conforme indicado nos passos. Caso contrário, a visão estéreo (3D) não será possível.

Para fundir as imagens, é interessante observar a importância de se filtrar canais de cor, sem que haja a repetição de canais “liberados” nas duas imagens. Assim, se um canal de cor está visível em uma imagem, deverá estar bloqueado na outra e vice-versa.

Dessa maneira, o operador lógico OU pode ser aplicado para mesclar as imagens, sem que haja prejuízo nas informações de cor de cada imagem. No caso do anaglifo vermelho-ciano, a informação de vermelho da imagem da esquerda, entra no canal vermelho da imagem da direita, que está vazio. Como os canais verde e azul (ciano) da imagem da esquerda estão vazios, nenhum prejuízo é causado a estes canais da imagem da direita.

4- Um caso prático

Em fotogrametria aérea, o mecanismo de visão humana é reproduzido em tomadas aéreas de cidades, como se fosse um “gigante” olhando para baixo a cidade. Assim, é possível através desse mecanismo ver objetos em 3D.

A partir de [1], foi utilizada um par de imagens aéreas do bairro do Maracanã, Rio de Janeiro, onde foi destacado como objeto de interesse o estádio do Maracanã. Cada imagem foi convertida para o formato de screen 8 do MSX, utilizando-se o MSX Viewer 5.

O programa em Basic, juntamente com as imagens convertidas para o formato do MSX, encontram-se disponíveis para download, juntamente com esse artigo na opção da tabela “Extra”. O arquivo chama-se “maracana.zip”.



Imagem do olho esquerdo – MRCLEFT.S08

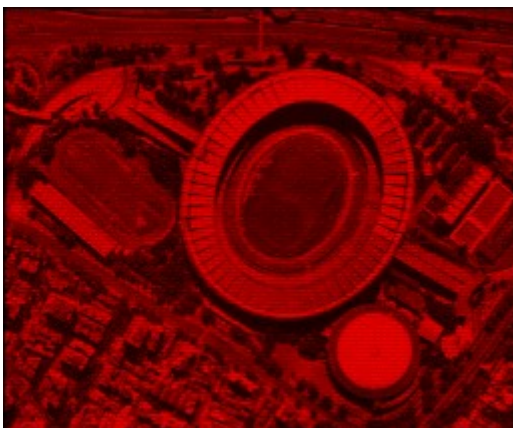


Imagem do olho direito – MRCRIGHT.S08

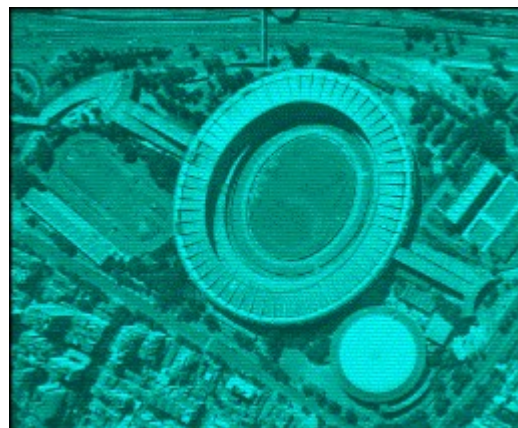
O programa a seguir irá aplicar o filtro em cada imagem e em seguida fundi-las.

```
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
```

O MSX 2 é muito versátil com operações lógicas em imagens. Um filtro é aplicado sobre cada imagem, utilizando-se uma máscara e o operador lógico AND para apagar os componentes de cor relacionados.



Filtro vermelho na imagem da esquerda



Filtro ciano na imagem da direita

Na fusão das imagens, a imagem da esquerda foi copiada sobre a imagem da direita utilizando-se o operador lógico OR.

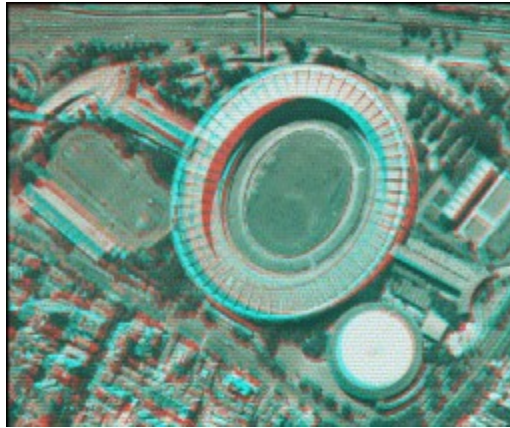


Imagem resultante em anaglifo.

Não se esqueça de colocar o óculos anaglifo para ver a essa imagem em 3D.

5- Créditos

Este artigo foi escrito por Marcelo Silveira, em Outubro de 2016.

E-mail: flamar98@hotmail.com

Referências:

[1] Projeto E-FOTO: <http://www.efoto.eng.uerj.br>

[2] Livro: Fotogrametria Digital, Brito e Coelho Filho, Editora EdUerj, 2007.