MSX Article

MARMSX

Screen 2 image adaptation

Summary

This article aims at discussing on some techniques used to adapt an image to the MSX screen 2, which is based on 8x1 pixels with only two different colors on it.

1. Introduction

After using some technique to properly reduce an 24-bit RGB image to the MSX 1 default color set, we must fit that image to the MSX screen 2 color pattern. This means that for each 8x1 line, we can only have two different colors, called front color and background color.

For that, we will propose some methods to find out the best two colors that will represent the whole group. So, the following strategies will be analyzed:

- Most Frequent Colors
- Random colors
- Shortest Distance Colors
- Predefined pixels

All these strategies are based on the 8x1 pixels line information. Our target is to find out the two colors that best represent such region.

It is also interesting to analyze the neighboring pixels effect on the image adaptation. For that, we will select a MxN block instead of the 8x1 line.

2. Methods description

2.1. Most Frequent Colors

This first approach will select the most frequent colors on the 8x1 pixel line (or MxN block). So, we must calculate the histogram for the 8x1 line. The histogram counts the frequency of each MSX 16 colors set present in that line.

Let's take a look on the following example. Suppose the following 8x1 line extracted from an image:

Pixel	1	2	3	4	5	6	7	8
Color	14	14	13	13	13	13	13	0

The histogram for that line is:

Frequency	1	0	0	0	0	0	0	0	0	0	0	0	0	5	2	0
Color	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

This means that for the given 8x1 line we have 1 pixel with color 0, 2 pixels with color 14 and 5 pixels with color 13.

After sorting the 2x16 histogram matrix, based on the first line and using descend order, we get this result:

Frequency	5	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Color	13	14	0	1	2	3	4	5	6	7	8	9	10	11	12	15

According to the table above, we notice that the two most frequent colors are 13 and 14. So, these colors will be used to represent the 8x1 line.

2.2. Random colors

This strategy will select randomly two colors among the existing colors in the 8x1 line.

Let's take a look on the following example:

Pixel	1	2	3	4	5	6	7	8
Color	14	14	13	13	13	5	5	4

From the previous table, we extract the following set of unique colors:

$$C = \{4, 5, 13, 14\}$$

Our task is to select randomly two members from the *C* set.

2.3. Shortest Distance Colors

In this technique we will select the two colors based on those who produce the least error when compared to all pixels from the 8x1 line.

The first task is extract the unique set of colors from the 8x1 line, as done in the previous section. Then, we will calculate the squared error for each color in the set compared with all pixels in the 8x1 line, as described in the figure 2.1. At last, we sum the errors for each color set member.

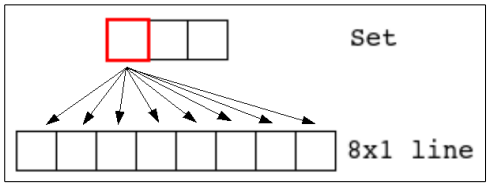


Figure 2.1. Error calculation for each member.

The squared error for two colors C1 and C2 is calculated as follows:

$$Error = (C1_r - C2_r)^2 + (C1_g - C2_g)^2 + (C1_b - C2_b)^2$$

Where *r*, *g* and *b* are the red, green and blue components, respectively.

Let's take a look on the following example:

Pixel	1	2	3	4	5	6	7	8
Color	14	14	13	13	13	5	5	4

We have the following set of unique colors: {4, 5, 13, 14}.
Let's calculate the squared error for each set member:

Memb / Pixel	1	2	3	4	5	6	7	8	Total
4	47961	47961	39749	39749	39749	6625	6625	0	228419
5	22758	22758	28014	28014	28014	0	0	6625	136183
13	13396	13396	0	0	0	28014	28014	39749	122569
14	0	0	13396	13396	13396	22758	22758	47961	133665

According to that table, the shortest errors belongs to the color index 13 and 14 (122569 and 133665, respectively).
In this approach, two similar colors may be stronger than one more frequent color.

2.4. Predefined pixels

This is the easiest technique. For color selection we only define two pixels for all 8x1 lines and take the corresponding colors. Figure 2.2 shows an example where pixels 1 and 8 are define as color reference.

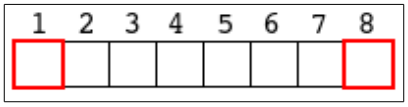


Figure 2.2. Predefined pixels 1 and 8.

Let's take a look on the following example:

Pixel	1	2	3	4	5	6	7	8
Color	14	14	13	13	13	5	5	4

If we define pixels 1 and 8, the following colors will be returned: 14 and 4.

3. Fitting original colors to the two selected colors

The next question is: how can I replace the 8x1 line using the two colors found? For each pixel in the line we calculate the Euclidean Distance to both colors using the RGB colors (not indexes). Take the color that has the shortest distance.

The Euclidean Distance for two colors c1 and c2 is calculated as follows:

$$d = \sqrt{(C1_r - C2_r)^2 + (C1_g - C2_g)^2 + (C1_b - C2_b)^2}$$

Where *r*, *g* and *b* are the red, green and blue components, respectively.

Figure 3.1 depicts how to replace each 8x1 pixel. The shortest Euclidean Distance from the current pixel determines if front color or background color is selected.

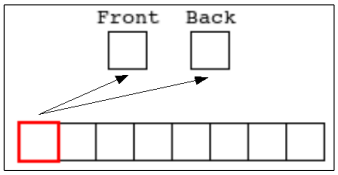


Figure 3.1. Final color replacement.

4. Experiments and results

Two images converted previously to the default MSX 1 palette color were used on the experiments. For a 8x1 line, we applied all strategies. For MxN block, we applied only the Most Frequent Color strategy.

Experiment 1: all strategies for 8x1 line.

Original	Most Frequent Colors	Random	Shortest Distance Colors	Predefined color
				
				

The Most Frequent Colors strategy presented far away the best results.

Experiment 2: Most Frequent Colors strategy for block sizes 10x3 and 16x8.

Each 8x1 line is centered at pixel 4. So, each block size will produce the neighboring pixels according to figure 4.1. In this picture, the gray color delimits the 8x1 line and the "x" marks the line center.

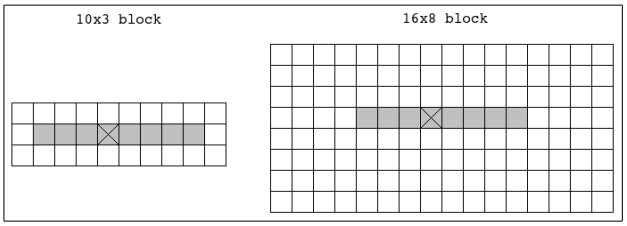
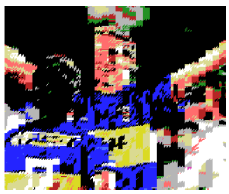
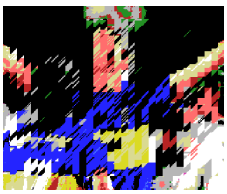
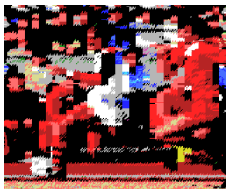
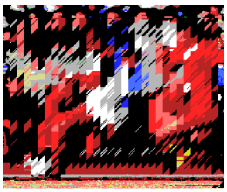


Figure 4.1. Blocks neighboring.

Each block is then used to calculate the colors frequence.

Original	8x1 line	10x3 block	16x8 block
			
			

We conclude that the greater the block, the worst the results.

5. Credits and References

This article was written by Marcelo Teixeira Silveira.

Date: March 2007.
Site: http://marmsx.msxall.com
E-mail: flamar98@hotmail.com

References:

MSX Viewer 5 project - http://marmsx.msxall.com

Note: this article is followed by the experiments source codes for Matlab / GNU-Octave and images. The codes are under GNU GPL version 3.x license.