MSX Article

MARMSX

Otimização da Paleta de Cores

Resumo

O objetivo deste artigo é apresentar a técnica de otimização da paleta de cores utilizada no MSX Viewer 5 para encontrar a melhor paleta de cores do MSX 2 e do v9990, a partir de imagens de 24 bits do PC.

1- Introdução

O método de quantização de imagens utilizado no MSX Viewer 5 baseia-se em um conjunto de cores destino, no qual é um sub-conjunto do sistema de cores 24-bit do PC. O conjunto destino pode ser o universo de cores disponíveis para esse sistema ou um sub-conjunto desse universo. O primeiro caso é a screen 8 do MSX, que possui todas as 256 cores disponíveis para representar uma imagem. O segundo caso são as screen 5-7, no qual estão disponíveis apenas 16 cores do universo de 512. No primeiro caso, todas as cores necessárias para o quantizador estão disponíveis. Já no segundo, apenas algumas.

Uma imagem é normalmente quantizada para as screens 2-7 baseada na paleta de cores default do MSX 1. Dependendo da imagem utilizada, somente algumas cores dessa paleta podem ser correspondidas, resultando em uma imagem com poucas cores. Um exemplo disso pode ser visto na figura 1b, onde a quantização de uma imagem em tons de cinza para a paleta do MSX 1 resultará em uma imagem com três cores distintas: preto, cinza e branco.

Com uma oferta maior de cores, podemos encontrar as cores ideais para formar uma paleta e representar melhor a imagem a ser quantizada. Podemos ver o resultado disso na figura 1c, onde uma imagem foi quantizada no GIMP utilizando uma paleta otimizada com 256 cores de 16 milhões.

a) Imagem original   b) Quantização para as 16 cores do MSX 1   c) Quantização para 16 cores de 16 milhões

Figura 1. Quantização de imagem com paleta não-otimizada e otimizada.

Observa-se que para imagens que utilizam paleta de cores, é necessário encontrar o sub-conjunto de cores ideais para uma imagem antes de realizar a quantização.

Para se atingir esse objetivo, não seria simplesmente verificar as cores mais freqüentes na imagem? Não. Veja, por exemplo, o caso de uma imagem possuir grandes áreas com tonalidades de uma mesma cor (ex. céu azul). Essas cores poderiam prevalecer sobre as outras, resultando em uma paleta com diversos tons relativos a essa cor. Isso geraria um problema, onde os objetos de uma cor seriam representados por outra cor.

A proposta dessa pesquisa é utilizar o classificador K-Means para encontrar a paleta ideal no MSX para uma dada imagem.

2- O Classificador K-Means

K-Means [2] é um algorítimo de classificação iterativo, utilizado em *data mining*. A idéia básica é fazer o seguinte: para um dado conjunto $X=(x_1, x_2, \ldots, x_n)$, criar K sub-conjuntos de X. Cada sub-conjunto é representado por um centróide $C=(c_1, c_2, \ldots, c_k)$.

Em cada iteração do K-Means, ele irá:

1. Associar elementos aos centróides: para cada elemento de X, associe-o ao sub-conjunto do centróide mais próximo.
2. Recalcular os centróides: após as associações, calcular os novos valores de centróides, baseado na média dos elementos de cada sub-conjunto.
3. O algorítimo é interrompido quando todos os centróides não se alterarem entre uma iteração e outra.

O conjunto X é uma massa de dados qualquer. Dessa forma, os valores iniciais dos centróides são desconhecidos. Para a solucionar esse problema, pode-se atribuir valores aleatórios a esses centróides. Outro problema, é que o número K de sub-conjuntos é desconhecido. Assim, o usuário deverá fornecer esse número.

Não importando quantas dimensões tiver cada elemento, ele será associado ao centróide pelo valor da distância até ele. O cálculo da distância é através da Distância Euclidiana:

$$d(x_i, c_j) = \sqrt{\sum_{d=1}^{n} (x_{i,d} - c_{j,d})^2}$$

Onde:

i é o i-ésimo elemento de X.
j é o j-ésimo centróide de C.
d é a d-ésima dimensão de ambos.

O exemplo a seguir irá ilustrar o funcionamento do K-Means para um conjunto X bidimensional.

Seja X = { (7, 1), (5, 1), (4, 8), (7, 8), (7, 3), (3, 5), (6, 1), (2, 2), (4, 10), (10, 8) }.

O conjunto X pode ser visto graficamente na figura 2.

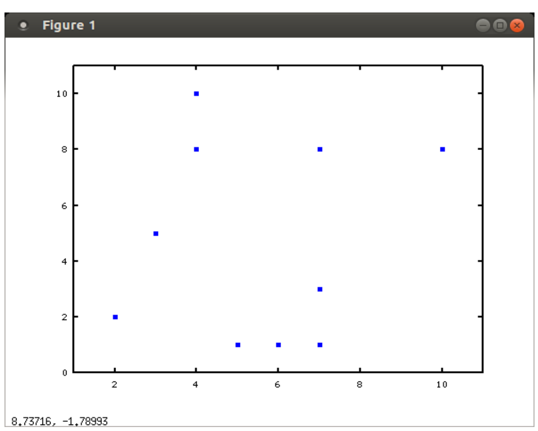


Figura 2. Conjunto X do exemplo.

Suponha que se deseje classificar esse conjunto em dois grupos. Assim, serão utilizados dois centróides para representar os grupos, com valores iniciais aleatórios.

Dado que os valores aleatórios encontrados para *C* sejam:

$C = \{ (2, 4), (9, 9) \}$

O conjunto X (em azul) e o centróides C (em vermelho) podem ser vistos na figura 3.

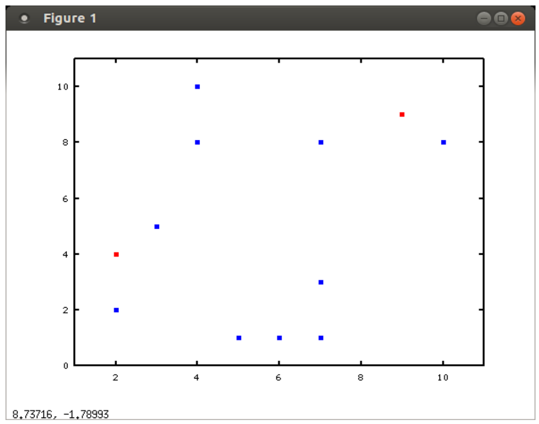


Figura 3. Conjunto *X* e os centróides *C* do exemplo.

A figura 4 apresenta o resultado da classificação dos dados utilizando-se o algorítimo de K-Means, bem como a trajetória dos centróides durante as iterações. O valor acima do centróide indica o número da iteração, onde o valor 0 é a posição inicial do centróide.

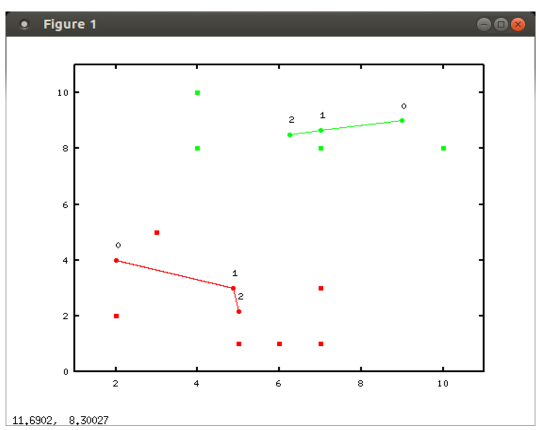


Figura 4. Resultado para a classificação por K-Means.

Foram necessárias duas iterações para que se encontrasse a convergência dos resultados. Os pontos ligados pelas linhas representam a trajetória dos centróides.

Ao final da classificação, dois sub-conjuntos foram encontrados. Eles estão assinalados pelos pontos em verde (grupo 1) e vermelho (grupo 2) na figura 4.

2.1- Os centróides vazios

O K-Means pode apresentar uma situação indesejável, onde a classificação é interrompida e o resultado é inadequado. Trata-se dos centróides vazios. Um centróide vazio é aquele que nenhum elemento do conjunto *X* está associado a ele.

A causa desse problema pode ser:

- Número *K* de grupos inadequado
- Má distribuição inicial dos centróides

A solução para o primeiro caso é modificar o número *K* de agrupamentos, até que se atinja um número adequado. Já no segundo, a quantidade *K* é o ideal ou o necessário, como o caso do MSX, que deve ter 16 ou 64 cores. Em tese, apenas reiniciando a classificação solucionaria o problema. Mas, nem sempre essa solução resolve o problema.

Para solucionar o problema de centróides vazios, a proposta é dividir os elementos dos centróides mais "gulosos" com os centróides "famintos". Assim, deve-se:

- Ordenar os centróides de forma decrescente, a partir do número de elementos de *X*.
- Modificar os valores dos centróides vazios para as coordenadas dos mais populosos, inserindo uma pequena "perturbação", para que os centróides não estejam exatamente na mesma posição (senão, ele continuará vazio).

Para exemplificar o problema dos centróides vazios, sejam os seguintes centróides *C* para o conjunto *X* do exemplo anterior:

$C = \{ (2, 4), (19, 19) \}$

Ao utilizar o classificador K-Means, todos os elementos serão associados a c_1, visto que c_2 está bem mais distante dos pontos.

Outro exemplo, agora com excesso de grupos:

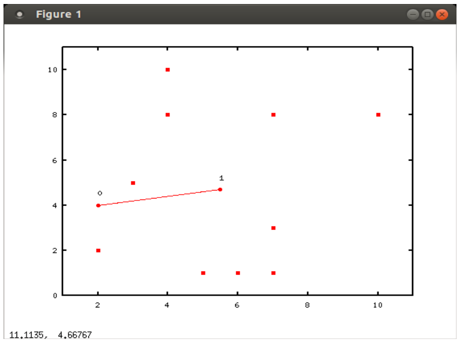
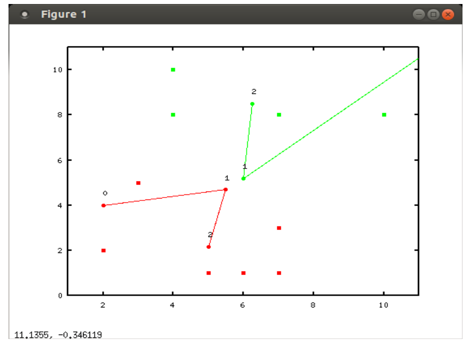
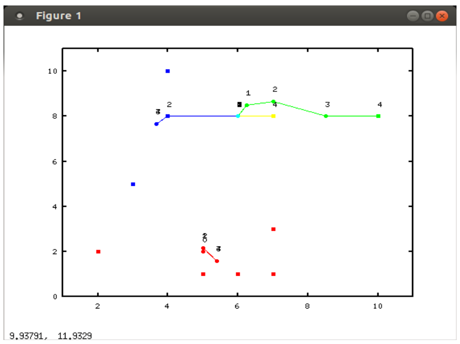
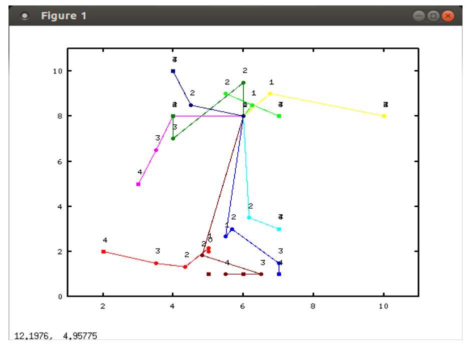
$C = \{ (5, 2), (6, 8), (6, 8), (6, 8), (6, 8), (6, 8), (6, 8), (6, 8), (6, 8) \}$

Ao classificar *X*, cada centróide terá os seguintes elementos:

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
x_1	x_{10}	x_3	x_4					
x_2		x_6						
x_5		x_9						
x_7								
x_8								

Observe que os centróides c_5 até c_9 ficaram vazios, ou seja, aquelas classes não tem qualquer representante. Isto representa um desperdício, principalmente quando se trata de cores em um sistema de cores limitado, como o do MSX.

Ao corrigir o problema do centróide vazio, tem-se graficamente para os dois exemplos:

<table>
<tr><th>Exemplo</th><th>Centróide vazio</th><th>Corrigido</th></tr>
<tr><td>1</td><td>
</td><td>
</td></tr>
<tr><td>2</td><td>
</td><td>
</td></tr>
</table>

Onde a nova classificação do conjunto *X* para o exemplo 2 seria:

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
x_8	x_4	x_1	x_{10}	x_6	x_5	x_2	x_3	x_9
						x_7		

Agora, todas as 9 classes no exemplo 2 possuem ao menos um elemento de *X*.

3- Otimização da paleta do MSX

O objetivo da aplicação do K-Means em imagens digitais de 24 bits é encontrar os agrupamentos de cor que melhor representem essa imagem nos sistemas de paleta do MSX 2 e do chip v9990.

A classificação da imagem original de 24 bits com o K-Means pode ser feita nesse espaço de cor ou no espaço de cor correspondente a uma das paletas do MSX. A paleta do MSX 2 possui 512 cores e a do chip v9990 possui 32768 cores. A diferença é que no maior espaço de cor, o ajuste é mais refinado, entretanto, leva mais iterações para convergir.

Em ambos os casos, o ajuste poderá levar a cores repetidas, ou seja, a centróides superpostos. Isto se deve ao processo de quantização, que arredonda os valores dos centróides. A solução para esse problema será discutida mais adiante.

O conjunto X será a imagem de 24 bits do qual será extraída a paleta otimizada. Os centróides de C correspondem às cores da paleta de cor do MSX. Entretanto, antes de começar a classificação, deve-se modificar a imagem de três dimensões para duas, onde cada componente de cor se torna um vetor, conforme mostra a figura 5.

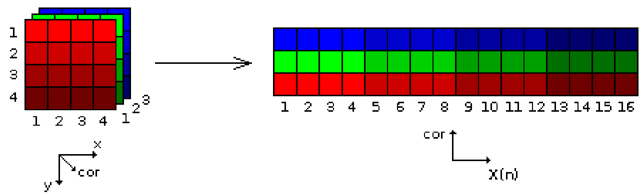


Figura 5: Conversão de matriz para vetor.

No caso da paleta do MSX 2, o número de sub-conjuntos de X são 16. Assim, K=16. No caso chip v9990, K=64. A paleta inicial é composta de valores aleatórios.

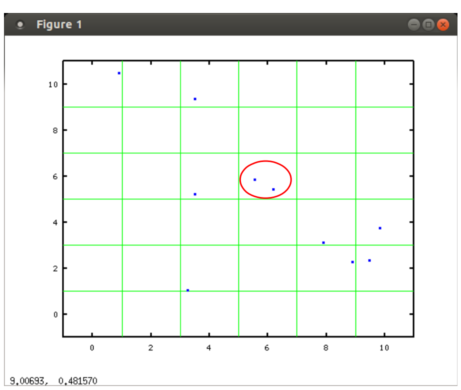
A Distância Euclidiana da cor de um pixel P à uma determinada cor da paleta C é calculada segundo a equação:

$$d(P,C)=\sqrt{(P_R-C_R)^2+(P_G-C_G)^2+(P_B-C_B)^2}$$

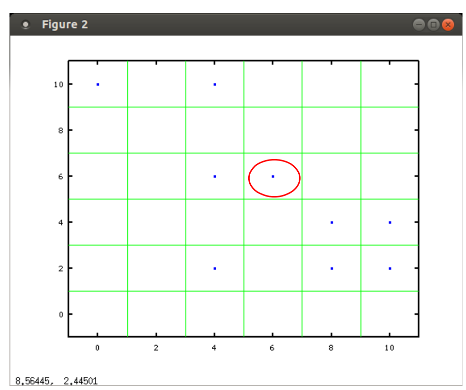
Onde R, G e B são os componentes de cor vermelho, verde e azul, respectivamente.

3.1- Centróides superpostos

A quantização e o arredondamento dos valores dos centróides podem levar a uma situação indesejada: a de cores repetidas. Isto acontece quando os centróides possuem valores distintos, mas quando quantizados, passam a ter valores iguais. A figura 6 ilustra esse problema.



a) valores originais



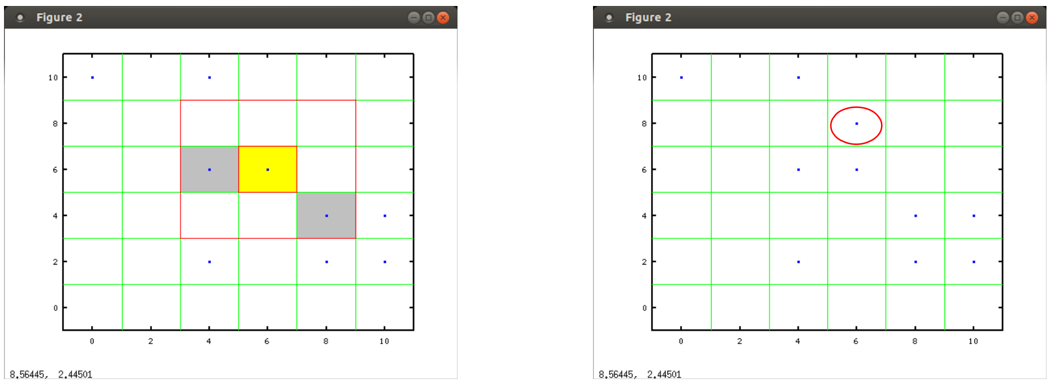
b) quantizado

Figura 6. Superposição de centróides.

Os pontos em azul representam os centróides, enquanto que as linhas verde correspondem à superfície de quantização.

Após a quantização, os dois pontos assinalados na figura 6a pelo círculo vermelho ficam superpostos, conforme ilustra o círculo vermelho na figura 6b. Então, o conjunto original de 10 centróides passa a ser de 9 centróides.

Uma possível solução para esse problema é deslocar um dos pontos superpostos para o espaço quantizado vizinho, desde que este não esteja ocupado. A figura 7 ilustra esse processo.



a) possíveis candidatos b) problema corrigido

Figura 7. Correção do problema de superposição de centróides.

Os centróides superpostos estão assinalados com o fundo amarelo na figura 7a. O retângulo vermelho representa a vizinhança de interesse dessa região. As regiões marcadas com o fundo cinza representam as regiões ocupadas e que não irão servir para deslocar o centróide superposto. Então, foi escolhida uma região livre aleatoriamente, assinalada pelo círculo vermelho, conforme mostra a figura 7b.

3.2- Resultados obtidos

De forma a testar a otimização da paleta de cores do MSX, foi utilizado o MSX Viewer 5. Nesse caso, foi aplicada a quantização de cores através da Distância Euclidiana tanto para a paleta do MSX 1, como para as paletas otimizadas do MSX 2 e V9990.

a) Imagem original b) Paleta do MSX 1 c) Paleta otimizada, *K*=16 d) Paleta otimizada, *K*=64

Figura 8. Resultados obtidos para o MSX, utilizando o MSX Viewer 5.

Quando foi aplicada a quantização de cores sobre a imagem a partir da paleta default do MSX 1, foram encontradas 4 cores: preto, verde, cinza e branco (figura 8b).

Após otimizar a paleta de cores do MSX 2 para a imagem de teste, foi aplicada a quantização sobre a imagem original utilizando-se essa paleta. Os resultados obtidos foram sensivelmente melhores do que a paleta do MSX 1 (figura 8c). Nesse caso, foram selecionadas as 16 cores que melhor representam a imagem de um universo de 512.

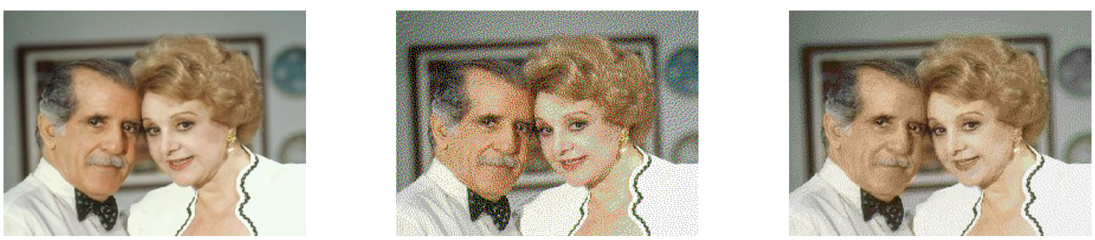
Entretanto, quando foi utilizada a paleta do processador de vídeo V9990 para esse mesmo fim, os resultados foram melhores ainda (figura 8d). Aqui, foram utilizadas 64 cores de um total de 32768.

Uma vez otimizada a paleta de cores, é possível aplicar o método de Error Diffusion sobre a imagem em vez da quantização. Nesse caso, os resultados são melhores ainda, conforme ilustra a figura 9. Para esse teste foi utilizada somente a paleta do MSX 2.

a) Imagem original　　b) Quantização sobre paleta otimizada　　c) Error Diffusion sobre paleta otimizada

Figura 9. Comparação entre a quantização e o Error Diffusion aplicados a uma paleta do MSX 2 otimizada.

No teste a seguir, foi comparado o resultado da aplicação do “Error Diffusion” sobre duas imagens: uma sem utilizar a paleta otimizada (paleta default do MSX1) e outra utilizando a paleta otimizada. Os resultados podem ser vistos na figura 10.



a) Imagem original　　b) Error Diffusion com paleta do MSX 1　　c) Error Diffusion com paleta otimizada

Figura 10. Combinação das técnicas de Error Diffusion e otimização de paleta.

A imagem da figura 10c corresponde à screen 5 do MSX 2, com 16 cores de 512.

4- Créditos e Bibliografia

Este artigo foi escrito por Marcelo Silveira, Engenheiro de Sistema e Computação, formado pela Universidade do Estado do Rio de Janeiro.

Escrito em: Julho de 2017, revisado em Fevereiro de 2018 e Maio de 2018.
E-mail: flamar98@hotmail.com
Homepage: http://marmsx.msxall.com

Referências:
[1] – Error Diffusion, artigo, Marcelo Silveira em MarMSX Development.
http://marmsx.msxall.com
[2] – K-Means Clustering, Wikipedia. http://en.wikipedia.org/wiki/K-means_clustering
[3] – MSX Viewer 5 Appendix, http://www.marmsx.msxall.com