

Universidade Federal de Juiz de Fora

AMANDA BRUGIOLO SOUZA

**Abordagens Heurísticas para o Problema de  
Quantização de Imagem**

JUIZ DE FORA

2011

AMANDA BRUGIOLO SOUZA

# Abordagens Heurísticas para o Problema de Quantização de Imagem

Monografia submetida ao Departamento de  
Ciência da Computação da Universidade Fe-  
deral de Juiz de Fora como requisito par-  
cial para a obtenção do grau de Bacharel em  
Ciência da Computação.

Orientador:

Stênio Sã Rosário Furtado Soares

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

JUIZ DE FORA

2011

# Abordagens Heurísticas para o Problema de Quantização de Imagem

Amanda Brugiolo Souza

Monografia submetida ao Departamento de Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Aprovada por:

---

Prof. Stênio Sã Rosário Furtado Soares / DCC-UFJF  
(Orientador)

---

Profa. Luciana Brugiolo Gonçalves / DPI-UFV

---

Profa. Liamara Scortegagna / DCC-UFJF

Juiz de Fora, 09 de dezembro de 2011.

# Resumo

O processo de quantização de imagem consiste na redução da quantidade de cores distintas presentes em uma imagem digital. Este processo pode ser feito levando-se em conta ou não características individuais da imagem em questão. Na literatura são encontrados variados algoritmos que utilizam diferentes abordagens combinadas a diferentes estruturas de dados. Em geral, tendo em vista as características de problemas associados à análise de imagens no que se refere ao elevado volume de dados, estas abordagens procuram extrair informações das imagens de forma a auxiliar o processo de quantização. Trabalhos recentes demonstram que algoritmos de clusterização de dados podem ser adaptados para serem aplicados ao problema de quantização de imagem eficientemente. Neste trabalho foram implementadas tanto a versão clássica como variações do algoritmo k-means, adaptadas para o problema de quantização. Além das adaptações do algoritmo k-means, novas abordagens foram propostas. Testes computacionais mostraram que os resultados obtidos são equivalentes àqueles apresentados na literatura no que diz respeito à qualidade das imagens geradas após o processo de quantização mesmo sem a necessidade de manipulação de complexas estruturas de dados.

**Palavras-chave:** Quantização de imagem, Quantização de cor, Heurística, K-Means.

# Sumário

<b>Lista de Figuras</b>	<b>v</b>
<b>Lista de Tabelas</b>	<b>vi</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 O Problema de Quantização de Imagem</b>	<b>3</b>
2.1 Formulação do Problema . . . . .	4
2.1.1 Quantização e Quantização de Cor . . . . .	4
2.1.2 Célula e Nível de Quantização . . . . .	5
2.1.3 Quantização e Percepção . . . . .	6
2.1.4 Erro de Quantização . . . . .	8
2.1.5 Quantização Uniforme e Adaptativa . . . . .	9
2.2 Abordagens Consideradas . . . . .	13
2.2.1 O Algoritmo <i>K-Means</i> Convencional . . . . .	13
2.2.2 As Variações do Algoritmo <i>K-Means</i> . . . . .	14
<b>3 Abordagens Desenvolvidas</b>	<b>16</b>
3.1 A Primeira Abordagem . . . . .	17
3.2 A Segunda Abordagem . . . . .	19
<b>4 Resultados Computacionais</b>	<b>20</b>
4.1 Cenários de Teste . . . . .	20
4.1.1 Cenário 1 . . . . .	21

Sumário	iv
4.1.2 Cenário 2 . . . . .	23
4.1.3 Cenário 3 . . . . .	25
<b>5 Conclusões e Trabalhos Futuros</b>	<b>27</b>
<b>Referências</b>	<b>29</b>

# Lista de Figuras

2.1	Imagem original 24 <i>bits</i> e resultado da quantização em 4 <i>bits</i> (16 cores) . .	4
2.2	Imagem original em escala de cinza em 8 <i>bits</i> (à direita) e quantização em 1 <i>bit</i> (à esquerda) . . . . .	5
2.3	Diferentes níveis de quantização . . . . .	7
2.4	Efeito <i>Match Bands</i> . . . . .	8
2.5	Imagem "fish" original em 24 <i>bits</i> . . . . .	10
2.6	Quantização uniforme de "fish" em (a) 4 <i>bits</i> e (b) 8 <i>bits</i> . . . . .	10
2.7	Hierarquia dos métodos de quantização . . . . .	11
2.8	Quantização de "fish" pelo algoritmo de populosidade em (a) 4 e (b) 8 <i>bits</i>	12
2.9	Quantização de "fish" pelo algoritmo de corte mediano em (a) 4 e (b) 8 <i>bits</i>	13
4.1	Imagens originais "fish" e "parrots" em 24 bits . . . . .	20

# Lista de Tabelas

4.1	Comparação do uso das inicializações FGY, MMX, SSF e KPP da literatura com clusterização SEQ e RND para a quantização de "fish" 16 e 256 cores	21
4.2	Comparação do uso das inicializações FGY, MMX, SSF e KPP da literatura com clusterização SEQ e RND para a quantização de "parrots" 16 e 256 cores . . . . .	22
4.3	Comparação da combinação de diferentes valores dos parâmetros $\alpha$ e $\beta$ para a inicialização FRQ com clusterização RND para a quantização de "fish" 16 e 256 cores . . . . .	23
4.4	Comparação da combinação de diferentes valores dos parâmetros $\alpha$ e $\beta$ para a inicialização FRQ com clusterização RND para a quantização de "parrots" 16 e 256 cores . . . . .	24
4.5	Desempenho do algoritmo KMM inicializado com os esquemas FGY, MMX, SFF, KPP e FRQ, clusterização RND e diferentes valores de EXP para a quantização de "fish" 16 e 256 cores . . . . .	25
4.6	Desempenho do algoritmo KMM inicializado com os esquemas FGY, MMX, SFF, KPP e FRQ, clusterização RND e diferentes valores de EXP para a quantização de "parrots" 16 e 256 cores . . . . .	26

# Capítulo 1

## Introdução

Uma imagem digital pode conter milhões de cores, o que torna sua manipulação uma tarefa complexa.

A quantização de cor é um processo que tem sido largamente estudado nos últimos quinze anos [Braquelaire e Brun 1997], e consiste na redução do número de cores distintas presentes em uma imagem. Esse processo pode ser descrito basicamente em dois passos: o *design* da paleta de cores, em que um número reduzido de cores é especificado, e que pode ser feito levando-se ou não em conta características da imagem a ser quantizada (quantização uniforme e quantização não-uniforme, respectivamente); e o mapeamento dos *pixels*, em que para cada *pixel* da imagem é associada uma das cores da paleta; e costuma ser uma importante etapa de pré-processamento para várias aplicações de processamento de imagem.

Em termos teóricos, pode-se afirmar que o processo de quantização de imagem tem suas origens na transformação de tons contínuos de preto e branco em tonalidade discreta em escala de cinza. Ao se limitar o número de níveis de cinza que deve conter a imagem entre as cores preto e branco como extremos da escala, a questão chave é determinar quais os níveis de cinza devem ser usados para mapear a representação da imagem original.

No caso de imagens coloridas, o que torna complexo o processo é o fato de que cada cor presente na imagem é, em geral, definida por três componentes e o mapeamento deve se dar de um conjunto de centenas ou milhares de cores a partir de uma paleta com 256 cores ou menos.

Este trabalho tem como objetivo propor novas estratégias tanto para a fase de *design* da paleta de cores quanto para o mapeamento dos *pixels* que obtenham resultados comparáveis aos existentes atualmente na literatura, no que diz respeito à qualidade da

---

solução e ao tempo de computação necessário.

O restante deste trabalho está organizado como segue: no Capítulo 2 é apresentada a fundamentação do problema da quantização de cor e uma breve revisão da literatura. As abordagens utilizadas são apresentadas no Capítulo 3 e os resultados computacionais obtidos, no Capítulo 4. Finalizando, as conclusões e sugestões de trabalhos futuros são apresentados no Capítulo 5.

## Capítulo 2

# O Problema de Quantização de Imagem

Uma imagem *truecolor* geralmente contém milhões de cores, o que faz com que tarefas como exibição, processamento, armazenamento e transmissão se tornem processos árduos.

O *truecolor* é um método de representação e armazenamento de informações gráficas de uma imagem no espaço de cor de três componentes, vermelho, verde e azul (RGB), em que cada canal de cor de um *pixel* é representado por 8 *bits*, resultando em uma imagem 24 ( $3 * 8$ ) *bits*. Assim, o *truecolor* é definido pela combinação de 256 ( $2^8$ ) tons de vermelho, verde e azul, totalizando 16.777.216 ( $256^3$ ) possíveis cores distintas. Uma vez que acredita-se que o olho humano seja capaz de distinguir cerca de 10 milhões de cores [Judd e Wyszecki 1975], o *truecolor* é capaz de criar a ilusão de representar todas as cores do mundo real.

No que diz respeito à exibição de imagens, o gamute de cor da imagem (conjunto de cores distintas presentes na imagem) não pode ser maior do que o gamute de cor do dispositivo de exibição (quantidade de cores distintas que o dispositivo é capaz de representar) [Gomes e Velho 1997]. Assim, no passado, a quantização era uma necessidade, uma vez que os dispositivos não eram capazes de lidar com as 16 milhões de cores possíveis das imagens 24 *bits*.

Atualmente, dispositivos 24 *bits* são mais comuns, mas a quantização ainda mantém seu valor prático. Dela resulta, por exemplo, a diminuição do tamanho da imagem. Esse resultado não faz da quantização por si só uma técnica efetiva de compressão, mas a relaciona fortemente às técnicas de compressão de imagem. Assim, em aplicações mais modernas de compressão, segmentação, análise e recuperação de imagens, a quantização está presente como uma etapa importante de cada processo [Celebi 2009].

## 2.1 Formulação do Problema

A quantização de cor consiste na redução da resolução da cor (quantidade de *bits* que representam a cor), o que implica na redução da informação de cor associada a cada *pixel* da imagem. Em outras palavras, a quantização é o processo de redução da quantidade de cores da imagem [Braquelaire e Brun 1997], e que visa a menor distorção possível.

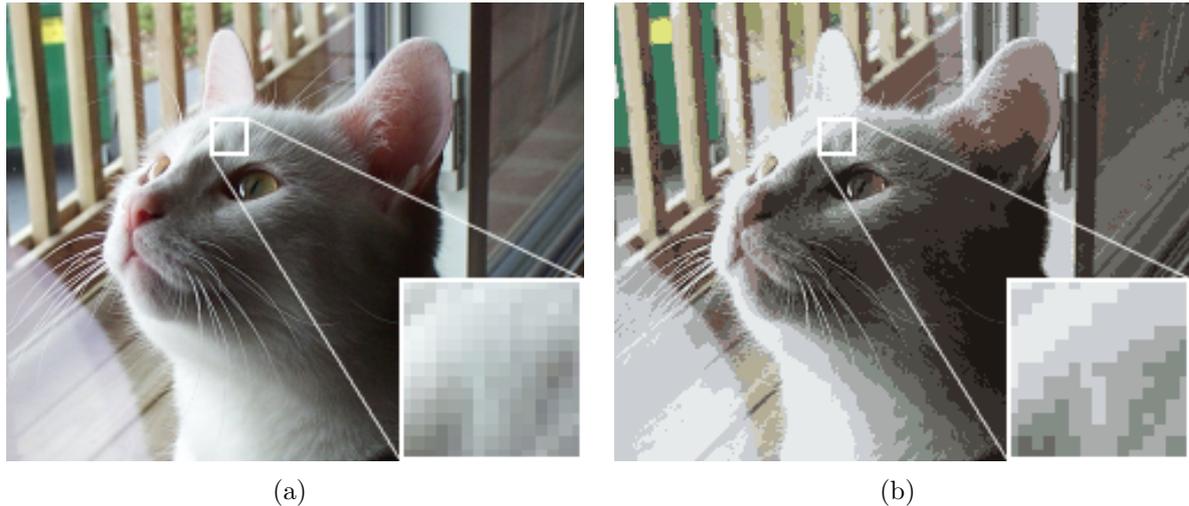


Figura 2.1: Imagem original 24 *bits* e resultado da quantização em 4 *bits* (16 cores)

A descrição formal do problema de quantização e suas características são descritas nas subseções a seguir.

### 2.1.1 Quantização e Quantização de Cor

A definição formal do problema de quantização é dada a seguir por [Gomes e Velho 1997].

Seja  $R_k = \{v_1, v_2, \dots, v_k\}$  um subconjunto de dimensão finita de  $\mathbb{R}^n$ . O caso geral da quantização de  $\mathbb{R}^n$  é o mapeamento  $q: \mathbb{R}^n \rightarrow R_k$ . Para o mapeamento de espaços de cor,  $\mathbb{R}^n$  é um espaço de cor de dimensão finita e  $R_k$  é um subconjunto finito do espaço de cores, e  $q$  é chamado mapa de quantização.

Frequentemente é preciso definir um mapa de quantização entre dois conjuntos finitos de cores. Isso ocorre, geralmente, quando é necessário definir um mapeamento  $q: C \rightarrow C'$ , de um sólido de cor  $C$ , em que as cores são representadas por  $M$  *bits*, em um sólido de cor  $C'$ , em que as cores são representadas por  $N$  *bits*, sendo  $N < M$ . Esse pode ser considerado um caso particular da definição anterior, uma vez que se inclui o espaço de cor  $C$  em  $\mathbb{R}^n$ , para algum  $n$ . Assim,  $q$  é chamado um mapa de quantização de  $N$ -*bits*.

Quando os espaços de cor  $C$  e  $C'$  possuem dimensão 1 (como no caso da escala de cinza, por exemplo), o processo de quantização é chamado quantização unidimensional. Quando os espaços de cor possuem dimensão  $n$  e a quantização de cada vetor de cor  $c = \{c_1, c_2, \dots, c_n\}$  é realizada pela quantização de cada componente  $c_i$  separadamente, tem-se a quantização escalar. Nesse caso, tem-se um mapa de quantização unidimensional  $q : C \rightarrow C'$  definido como  $q(c) = (q(c_1), q(c_2), \dots, q(c_n))$ . Quando a quantização não é escalar, é chamada de quantização vetorial.

Como exemplo, seja o problema de quantizar um espaço de cor monocromático com 256 níveis de cinza em somente dois níveis. Uma solução intuitiva é mapear para 0 todas as cores que estejam abaixo da metade do nível máximo de intensidade (ou seja, até 127), e as cores restantes para o valor máximo de 255. A Figura 2.2 mostra o efeito dessa quantização.

(a) Imagem original 8 *bits* (256 cores)(b) Quantização em um *bit* (2 cores)

Figura 2.2: Imagem original em escala de cinza em 8 *bits* (à direita) e quantização em 1 *bit* (à esquerda)

### 2.1.2 Célula e Nível de Quantização

Como pode ser observado na Figura 2.2, a quantização de espaços de cor em um *bit* (2 cores) divide o espaço em dois conjuntos, e em cada um a função de quantização assume um valor.

De forma mais geral, seja o mapeamento  $q : C \rightarrow C'$ . A cada cor quantizada  $c'_i \in C'$  corresponde um subconjunto composto por todas as cores em  $C_i \subset C$  que foram mapeadas

para  $c'_i$ . Ou seja:

$$C_i = q^{-1}(c'_i) = \{c \in C : q(c) = c'_i\}$$

e cada conjunto  $C_i$  é chamado célula de quantização. Para cada célula, o valor  $c'_i$  que a função de quantização assume é chamado nível de quantização.

Assim, o processo de quantização, ou seja, a determinação do mapa de quantização  $q$ , é composto basicamente por dois passos: a determinação das células de quantização e a determinação dos níveis de quantização. Uma vez conhecido o mapa  $q$  a quantização da imagem é feita como se segue: para cada cor  $c$  na imagem, substitui-se  $c$  pelo nível de quantização  $q(c)$  da célula que a contém. Os métodos de quantização podem tratar de diferentes maneiras estes dois passos:

- primeiro determinar as células de quantização e, em seguida, escolher os níveis de quantização para cada célula, ou
- primeiro determinar os níveis de quantização e então escolher quais cores serão mapeadas para cada nível.

Alguns métodos clássicos da literatura e seu comportamento serão descritos posteriormente.

### 2.1.3 Quantização e Percepção

Seja uma imagem monocromática em que o espaço de cor é quantizado em  $L$  níveis  $c_1, c_2, \dots, c_L$ . Essa quantização particiona o espaço de cor em  $L$  subconjuntos, cada um composto por *pixels* da imagem mapeados para um certo nível de quantização. Dependendo da diferença de valores entre os níveis de quantização entre regiões vizinhas, curvas de contorno podem se tornar perceptíveis. Tais curvas são chamadas de contorno da quantização. A Figura 2.3 a seguir ilustra esse fato.

Como pode ser observado, à medida em que o nível de quantização diminui, os contornos da quantização se tornam cada vez mais perceptíveis. Entretanto, esse fato deve-se não somente ao nível da quantização, mas também à imagem e ao método utilizado. Normalmente, para imagens em escala de cinza, 256 níveis (8 *bits*) são suficientes para evitar a aparição de contornos, independente do método de quantização utilizado. Para imagens coloridas, geralmente 24 *bits* são suficientes, 8 *bits* para cada componente do espaço RGB e, dependendo da imagem e do método utilizado, pode-se reduzir ainda mais o nível de quantização sem o aparecimento de contornos.



(a) Quantização em 256 níveis (8 bits)



(b) Quantização em 8 níveis (3 bits)



(c) Quantização em 4 níveis (2 bits)



(d) Quantização em 2 níveis (1 bit)

Figura 2.3: Diferentes níveis de quantização

A percepção do contorno da quantização torna-se mais acentuada pelo fenômeno conhecido como *Mach Bands*: o olho humano intensifica transições de intensidade da cor, fazendo com que a diferença entre cores similares seja mais perceptível se as cores estiverem imediatamente justapostas. A Figura 2.4 de [Gomes e Velho 1997] ilustra esse fenômeno.



Figura 2.4: Efeito *Match Bands*

Nota-se que a transição das listras é mais intensa na imagem (a) do que na imagem (b). Assim, regiões de uma imagem quantizadas em níveis diferentes, mas adjacentes, são facilmente detectadas pelo olho humano. Na quantização de dois níveis, uma vez que a imagem final só possui duas intensidades de cor, preto e branco, o contorno se torna extremamente marcado.

A conexão entre quantização e percepção de cor tem sido extensivamente explorada na busca por bons métodos de quantização.

#### 2.1.4 Erro de Quantização

A determinação ótima das células de quantização e do nível de quantização para cada célula, ou seja, a combinação desses fatores que cause a menor distorção na imagem, depende do critério utilizado para mensurar o erro da quantização e da distribuição de cores na imagem.

Assim, sejam  $q$  o mapeamento de quantização e  $c$  uma cor a ser quantizada, tem-se:

$$c = q(c) + e_q$$

em que  $e_q$  denota o erro da quantização.

A distorção na imagem é causada ao se substituir a cor  $c$  pelo valor de quantização  $q(c)$ . Para mensurar essa distorção, usa-se a medida  $d(c, q(c))$ . Há várias possibilidades de escolha para a medida  $d$  em  $C$ , e deve-se levar em consideração tanto o critério

de percepção quanto o custo computacional. É comum, nesse caso, fazer uso de uma pseudo-métrica ao invés de uma métrica, no sentido de obter alguma informação sobre "proximidade" no espaço de cor. Uma escolha possível, por exemplo, é o quadrado da distância Euclidiana, para três dimensões, dado por  $d(c, d) = (c_1 - d_1)^2 + (c_2 - d_2)^2 + (c_3 - d_3)^2$ .

Como a quantização consiste na alteração de cada *pixel* da imagem, a distorção é medida não só pela distorção das cores no espaço de cor, mas também pela frequência de cada cor na imagem. Assim, seja uma quantização de nível  $N$ , que particiona o espaço de cor em  $N$  células  $K_1, K_2, \dots, K_N$  com respectivos níveis de quantização  $q_1, q_2, \dots, q_N$ . Para estes parâmetros, a medida do erro é dada por:

$$E(c, q(c)) = \sum_{1 \leq i \leq N} \sum_{c \in K_i} p(c) d(c, q_i),$$

em que  $p(c)$  é a função de distribuição de probabilidade de cor em  $C$ . O uso dessa equação é bastante intuitivo: medir o erro da quantização, levando em conta a probabilidade de ocorrência de cada cor no espaço a ser quantizado.

### 2.1.5 Quantização Uniforme e Adaptativa

Os métodos de quantização de cor podem ser classificados em métodos uniformes e métodos não-uniformes.

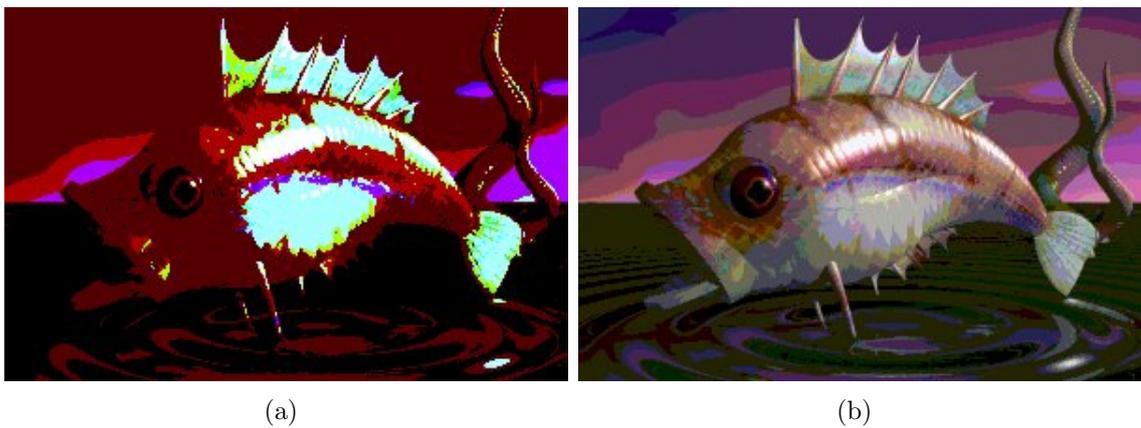
Os métodos uniformes determinam uma paleta de cor universal sem levar em consideração nenhuma imagem específica. Uma estratégia natural é dividir o espaço de cor em células congruentes e tomar o centro de cada célula como nível de quantização.

Assim, para uma quantização unidimensional de  $L$  níveis tem-se que as  $L$  células de quantização são dadas pelos intervalos  $(c_{i-1}, c_i]$  de tamanho constante  $c_i - c_{i-1}$ , e os níveis de quantização assumem o valor dado por  $q_i = \frac{c_{i-1} + c_i}{2}$ , com  $1 \leq i \leq L$ . Em outro cenário, se o espaço de cor considerado for o RGB e a quantização for feita em cada componente separadamente, as células de quantização serão pequenos cubos no espaço de cor, e o centro de cada cubo determinará o nível de quantização.

É intuitivo que estes métodos, apesar de serem de fácil computação e capazes de apresentar uma solução rapidamente, geralmente obtêm uma solução de baixa qualidade, uma vez que não levam em conta o conteúdo da imagem. Nos cenários descritos, por exemplo, algumas células de quantização podem não conter cor alguma do gamute de cor da imagem. Um exemplo da aplicação da quantização uniforme é mostrada na Figura 2.6.



Figura 2.5: Imagem "fish" original em 24 bits



(a)

(b)

Figura 2.6: Quantização uniforme de "fish" em (a) 4 bits e (b) 8 bits

Assim, a maior parte dos estudos da literatura se concentra nos métodos não-uniformes. Os métodos não-uniformes dividem o espaço de cor em células não-congruentes, e são chamados adaptativos quando a geometria das células de quantização são escolhidas de acordo com características específicas da distribuição de cores da imagem. O processo consiste em dois passos: estimar propriedades relevantes da imagem e aplicar essa informação no particionamento do espaço de cor.

Os métodos adaptativos são divididos em duas categorias: métodos pré-clusterização e métodos pós-clusterização. Os métodos pré-clusterização são divididos ainda em duas classes: métodos de aglomeração e métodos de subdivisão recursiva, como pode ser visto na Figura 2.7.

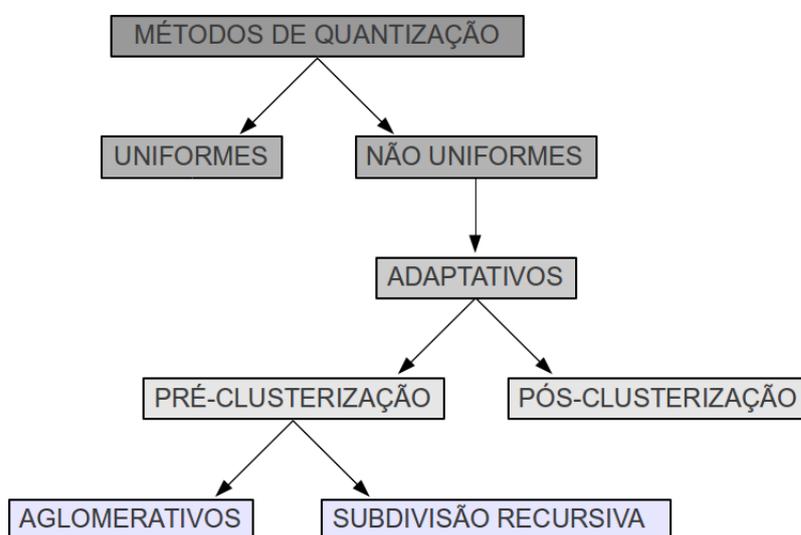


Figura 2.7: Hierarquia dos métodos de quantização

Os métodos de aglomeração trabalham determinando primeiro os níveis de quantização e depois as células de quantização. Assim, começam com  $L$  células de quantização, cada uma contendo um *pixel* da imagem, que são repetidamente agrupadas até que restem  $K$  células.

Um exemplo clássico de método adaptativo de aglomeração é o Algoritmo de Populosidade. Começando com a construção do histograma da imagem, o algoritmo seleciona em seguida os  $L$  níveis de quantização como sendo as  $L$  cores que ocorrem com maior frequência no gamute de cor da imagem. A função de quantização associa cada cor  $c$  do gamute de cor da imagem ao mais próximo nível de quantização  $q(c)$ , baseada, por exemplo, na distância Euclidiana. É fácil ver o que o algoritmo de populosidade ignora cores que não estão bem representadas no espaço de cor da imagem; entretanto, o algoritmo pode ser usado de forma satisfatória para imagens que possuam distribuição de cor

aproximadamente uniforme. A Figura 2.8 a seguir mostra o resultado da aplicação do algoritmo de populosidade para 4 e 8 *bits*.



Figura 2.8: Quantização de "fish" pelo algoritmo de populosidade em (a) 4 e (b) 8 *bits*

Já os métodos de subdivisão recursiva determinam em primeiro lugar as células de quantização e, em seguida, o nível de quantização em cada célula. O método começa com uma célula de quantização contendo as  $N$  cores da imagem que é recursivamente subdividida até que  $L$  células sejam obtidas. Um exemplo clássico de método adaptativo de subdivisão recursiva é o Algoritmo de Corte Mediano, descrito a seguir.

Seja  $L$  o número desejado de níveis de quantização. Pega-se o menor sólido  $V = \{[r_0, r_1] \times [g_0, g_1] \times [b_0, b_1]\}$  que contém todas as cores presentes na imagem de entrada. Pega-se o eixo do espaço de cor de maior tamanho, diga-se o eixo  $g$ . Ordena-se todas as cores de acordo com a sua componente  $g$  e computa-se a média  $m_g$  do conjunto baseado nessa ordenação. Isso divide  $V$  em duas sub-regiões:  $V_1 = (r, g, b) \in C : g \leq m_g$  e  $V_2 = (r, g, b) \in C : g > m_g$ . Recursivamente, subdivide-se os espaços  $V_1$  e  $V_2$  que contenham mais de uma cor até que se obtenha  $L$  células. Para cada célula de quantização, que é um sólido no espaço de cor, define-se o nível de quantização como a média dos valores das cores contidas na célula. A Figura 2.9 mostra o resultado da aplicação do método de corte mediano.

Outros métodos adaptativos de subdivisão recursiva encontrados na literatura incluem *octree* [Gervautz e Purgathofer 1988], *variance-based method* [Heckbert 1982], *binary splitting* [M. Orchard 1991], *greedy orthogonal bipartitioning* [Wu 1991], *center-cut* [Joy e Xiang 1993], *rwm-cut* [Yang e Lin 1996], e *optimal principal multilevel quantizer* [Wu 1992], .

Em contraste aos métodos pré-clusterização que computam a paleta de cores uma única vez, os métodos pós-clusterização determinam a paleta inicial e melhoram-na de

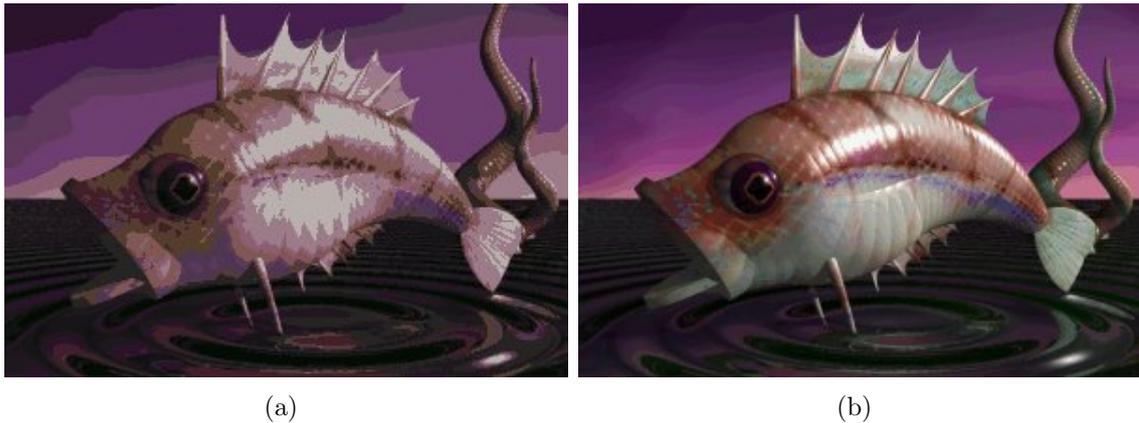


Figura 2.9: Quantização de "fish" pelo algoritmo de corte mediano em (a) 4 e (b) 8 bits

forma iterativa. Em geral, qualquer algoritmo de clusterização de dados pode ser utilizado para esse propósito, e, uma vez que estes algoritmos envolvem otimização iterativa, eles podem obter resultados superiores quando comparados aos métodos pré-clusterização, a menos do aumento do custo computacional. Algoritmos de clusterização adaptados ao problema de quantização de cor incluem o *k-means* [Kasuga et al. 2000], *fuzzy c-means* [Özdemir e Akarun 2002], *minmax* [Xiang 1997], *BIRCH* [Bing et al. 2004], *competitive learning* [Uchiyama e Arbib 1994], e *self-organizing maps* [Papamarkos et al. 2002].

Neste trabalho foi analisado o comportamento do algoritmo *k-means* e algumas de suas variações, adaptados para o problema de quantização de cor.

## 2.2 Abordagens Consideradas

O *k-means* é um algoritmo de clusterização de dados simples e de fácil implementação. A formulação geral do problema e o algoritmo *k-means* convencional são descritos a seguir.

### 2.2.1 O Algoritmo *K-Means* Convencional

Dado um conjunto de dados  $X = \{x_1, x_2, \dots, x_N\}$ , deseja-se particionar  $X$  em  $K$  clusters mutuamente exclusivos  $S = \{s_1, s_2, \dots, s_K\}$ , ou seja,  $\cup_{i=1}^K S_i = X$  e  $S_i \cap S_j = \emptyset$  para  $1 \leq i \neq j \leq K$ , minimizando a medida de erro:

$$E(x, c) = \sum_{i=1}^K \sum_{x_i \in S_k} \|x_i - c_k\|^2$$

em que  $\|\cdot\|$  denota a norma Euclidiana e  $c_k$  é o centróide do cluster  $S_k$ .

Esse problema pertence à classe de problemas NP-difícil. [Lloyd 1982] desenvolveu

uma solução simples que utiliza uma abordagem heurística. O algoritmo começa com um número arbitrário  $K$  de centróides, geralmente selecionados aleatoriamente de maneira uniforme no conjunto de dados. Em seguida, cada ponto é atribuído ao centróide mais próximo e os centróides são recalculados como o centro de massa de todos os pontos atribuídos a ele. Estes últimos dois passos são repetidos até que o algoritmo estabilize.

**Input:**  $X = \{x_1, x_2, \dots, x_N\} \in \mathbb{R}^D$  ( $N \times D$  input data set)

**Output:**  $C = \{c_1, c_2, \dots, c_K\} \in \mathbb{R}^D$  ( $K$  clusters centers)

```

1 Select a random subset  $C$  of  $X$  as the initial set of clusters centers;
2 while termination criteria is not met do
3   for  $i = 1; i \leq N; i = i + 1$  do
4     assign  $x_i$  to the nearest cluster;
5      $m[i] = \operatorname{argmin}_{k \in \{1, 2, \dots, K\}} \|x_i - c_k\|^2$ ;
6   end
7   recalculate the cluster centers;
8   for  $i = 1; i \leq N; i = i + 1$  do
9     cluster  $S_k$  contains the set of points  $x_i$  that are nearest to the center  $c_k$ ;
10     $S_k = \{x_i | m[i] = k\}$ ;
11    calculate the new center  $c_k$  as the mean of the points that belong to  $S_k$ ;
12     $c_k = \frac{1}{|S_k|} \sum_{x_i \in S_k} x_i$ ;
13  end
14 end

```

**Algoritmo 1:** Algoritmo *K-Means* convencional (Fonte: [Celebi 2009])

Diferentes técnicas podem ser aplicadas a fim de acelerar esse processo, como visto em [Celebi 2011] e [Elkan 2003], sem causar alterações na solução. Nenhuma das variantes implementadas, originais da literatura ou sugeridas neste trabalho, fez uso dessas técnicas.

### 2.2.2 As Variações do Algoritmo *K-Means*

Como visto anteriormente, o algoritmo *k-means* é composto por dois passos básicos: a seleção dos centróides iniciais e a atribuição dos pontos aos centróides mais próximos (processo chamado daqui em diante de clusterização).

É sabido da literatura que o *k-means* é bastante sensível à inicialização. Assim, diversas técnicas para a seleção dos centróides iniciais vêm sendo propostas ao longo dos anos, como pode ser visto em [Arthur e Vassilvitskii 2007] e [Celebi 2011]. Os esquemas

de inicialização implementados neste trabalho são brevemente descritos a seguir:

- Forgy (FGY) [Forgy 1965]: os centróides são selecionados aleatoriamente de forma uniforme do conjunto de entrada;
- Minmax (MMX) [Gonzalez 1985]: o primeiro centróide  $c_1$  é selecionado aleatoriamente e o  $i$ -ésimo centróide  $c_i$  é selecionado como sendo o ponto de maior menor distância aos centróides selecionados anteriormente;
- Subset Farthest First (SFF) [Turnbull e Elkan 2005]: é uma variação do minmax que seleciona os centróides em um subconjunto de  $2K \ln K$  pontos do conjunto de entrada, buscando evitar os *outliers* do conjunto de dados;
- K-Means++ (KPP) [Arthur e Vassilvitskii 2007]: o primeiro centróide  $c_1$  é selecionado aleatoriamente e o  $i$ -ésimo centróide  $c_i$  é selecionado para ser  $x'$  com probabilidade  $\frac{D(x')^2}{\sum_{i=1}^N D(x_i)^2}$ , em que  $D(x)$  denota da distância mínima de  $x$  aos centróides selecionados anteriormente;

Quanto à fase de clusterização, são propostas duas modificações:

- os pontos a serem atribuídos aos centróides são selecionados de forma aleatória (RND), diferentemente do algoritmo convencional em que os pontos são selecionados sequencialmente à leitura da imagem (SEQ); e
- o valor do centróide é atualizado toda vez que um novo ponto é atribuído a ele, e não ao final da clusterização como no algoritmo convencional.

# Capítulo 3

## Abordagens Desenvolvidas

Para analisar o comportamento do algoritmo *k-means* aplicado ao problema de quantização de imagem foram implementadas tanto a versão convencional do algoritmo bem como a sua combinação com os diferentes tipos de inicialização e modificações descritos no capítulo anterior. Também foram implementadas duas novas abordagens propostas, descritas nas próximas seções. Todas as abordagens seguem o esquema descrito a seguir.

```
Input:  $I$  = bmp 24 bits color image;  $k$  = quantization level  
Output:  $S$  = set of  $k$  clusters;  $I$  = bmp  $k$  colors quantized image  
1  $L = I.get\_colors()$ ;  
2  $C = initialization(initialization\_scheme)$ ;  
3  $S = clustering(clustering\_scheme)$ ;  
4 while (termination criteria is not met) do  
5   |  $C = S.get\_centroids()$ ;  
6   |  $S = clustering(clustering\_scheme)$ ;  
7 end  
8  $I.load(S)$ ;
```

**Algoritmo 2:** Esquema do algoritmo implementado

Basicamente, o algoritmo guarda na lista  $L$  as  $n$  cores distintas presentes na imagem, juntamente com a informação de frequência de cada uma. Em seguida, uma segunda lista  $C$  é preenchida com os centróides iniciais selecionados, mediante o esquema de inicialização escolhido; e, posteriormente, os pontos de  $L$  são distribuídos entre os clusters do conjunto  $S$ , também segundo o esquema escolhido. Os esquemas de inicialização e clusterização utilizados neste trabalho serão descritos detalhadamente no próximo capítulo. As linhas 4–6 definem a sequência de estabilização do algoritmo, em que a clusterização é executada repetidamente utilizando como centróides iniciais os centróides finais da iteração anterior, até que o critério de parada determinado seja atingido. Ao final, a imagem é atualizada com as informações do conjunto de clusters  $S$ , gerando a imagem quantizada.

## 3.1 A Primeira Abordagem

Inspirada no algoritmo de populosidade (descrito no Capítulo 2), essa abordagem é um esquema de inicialização que utiliza a informação de distribuição de cor na imagem para auxiliar o processo de seleção dos centróides iniciais. O uso de dois parâmetros ajustáveis faz com que o algoritmo possa se comportar como o próprio algoritmo de populosidade, tornando este método mais flexível do que o original.

A ideia central do algoritmo é, então, utilizar a informação da frequência das cores na imagem para guiar o processo de seleção dos centróides iniciais, porém, de forma mais sutil do que no algoritmo de populosidade. Assim, um parâmetro  $\alpha$  dita a porcentagem de cores a serem consideradas no momento de seleção do centróide, e um segundo parâmetro  $\beta$  trabalha um conceito de vizinhança, a fim de evitar que cores similares estejam presentes ao mesmo tempo no conjunto de centróides iniciais. O algoritmo, chamado de FRQ daqui em diante, e sua descrição são dados a seguir.

**Input:**  $L$  = color list;  $k$  = quantization level;  
**Output:**  $S$  = set of  $k$  clusters;  $L$  = color list;

```

1 /* inicializa variáveis */
2  $S = \emptyset$ ;
3  $d_{min} = \beta * d_{max}$ ;
4  $L.sort()$ ;
5 while ( $S.size() < k$ ) do
6      $i = random(\alpha * L.size())$ ;
7      $s = \emptyset$ ;
8      $s.add(L.get(i))$ ;
9      $L.remove(i)$ ;
10    for ( $j = 1$ ;  $j \leq L.size()$ ;  $j = j + 1$ ) do
11        if ( $(L.size() + S.size() + 1) = k$ ) then
12             $break$ ;
13        end
14        if ( $d(L.get(j), s.center()) \leq d_{min}$ ) then
15             $s.add(L.get(j))$ ;
16             $L.remove(j)$ ;
17             $j --$ ;
18        end
19    end
20     $S.add(s)$ ;
21 end

```

**Algoritmo 3:** Algoritmo FRQ proposto

Seja  $L$  o conjunto de  $n$  cores distintas presentes na imagem,  $S$  o conjunto de clusters e  $k$  o nível de quantização desejado. O algoritmo começa ordenando  $L$  com base na frequência das cores, de forma decrescente. Para a seleção dos centróides, escolhe-se aleatória e uniformemente um dentre os  $\alpha$  elementos mais frequentes em  $L$ , que é adicionado a um novo cluster  $s$  e excluído de  $L$ . Em seguida, percorre-se a lista e todo elemento  $l_i$  que estiver a uma distância mínima  $d_{min} = \beta * d_{max}$  do centróide selecionado é adicionado a  $s$  e excluído de  $L$ . Por fim,  $s$  é adicionado ao conjunto  $S$ . Ao final da seleção dos  $k$  centróides, a saída do algoritmo é o conjunto  $S$  com  $k$  clusters parcial ou totalmente preenchidos, e a lista  $L$ , que pode ou não conter algum elemento restante.

Neste trabalho o espaço de cor considerado foi o RGB, assim,  $d_{min} = \beta * d_{max} = \beta * 255\sqrt{3}$ . É fácil ver que tomando  $\alpha = 1.00$  e  $\beta = 0.00$  tem-se o algoritmo de populosoidade

tradicional.

## 3.2 A Segunda Abordagem

O algoritmo descrito a seguir, chamado de KMM, é uma variação do *k-means* (e não somente um esquema de inicialização, como a abordagem anterior) que, com o ajuste adequado dos parâmetros, descreve o algoritmo *k-means* convencional e pode fazer uso dos esquemas de inicialização apresentados anteriormente.

**Input:** *initialization\_scheme*; *clustering\_scheme*; *exp*;  $\alpha$ ;  $\beta$ ;

```

1  $k = k * 2^{exp}$ ;
2 while ( $k' \geq n$ ) do
3   |  $k = \frac{k}{2}$ ;
4   |  $exp --$ ;
5 end
6 initialization(initialization_scheme);
7 clustering(clustering_scheme);
8 while ( $-- exp > 0$ ) do
9   |  $k = \frac{k}{2}$ ;
10  | initialization();
11  | clustering(clustering_scheme);
12 end
13 initialization();
```

### Algoritmo 4: Algoritmo KMM proposto

A estratégia básica do algoritmo é iniciar o processo de quantização com um nível de quantização superior ao dado de entrada, e refinar o resultado até que se obtenha o nível  $k$  desejado.

Sejam  $k = 2^m$  e  $exp$  o parâmetro de ajuste do KMM. A seleção dos centróides iniciais é feita normalmente segundo o esquema de inicialização escolhido, porém, com nível de quantização  $k = 2^{m+exp}$ . A clusterização também é feita segundo o esquema escolhido. Em seguida, os dois passos básicos, seleção dos centróides e clusterização dos pontos, são executados repetidamente juntamente ao decremento de  $exp$  e  $k$ , até que se obtenha o nível de quantização  $k$  inicial desejado. A partir desse ponto, o algoritmo segue para a sequência independente de estabilização do processo.

# Capítulo 4

## Resultados Computacionais

Para comparar o desempenho dos algoritmos implementados foram definidos cenários de teste, visando focar determinadas características dos métodos, descritos a seguir.

### 4.1 Cenários de Teste

Em todos os cenários foram utilizadas as imagens "fish" (200 × 300, 28.170 cores, criada digitalmente) e "parrots" (200 × 300, 35.120 cores, natural), exibidas na Figura 4.1.

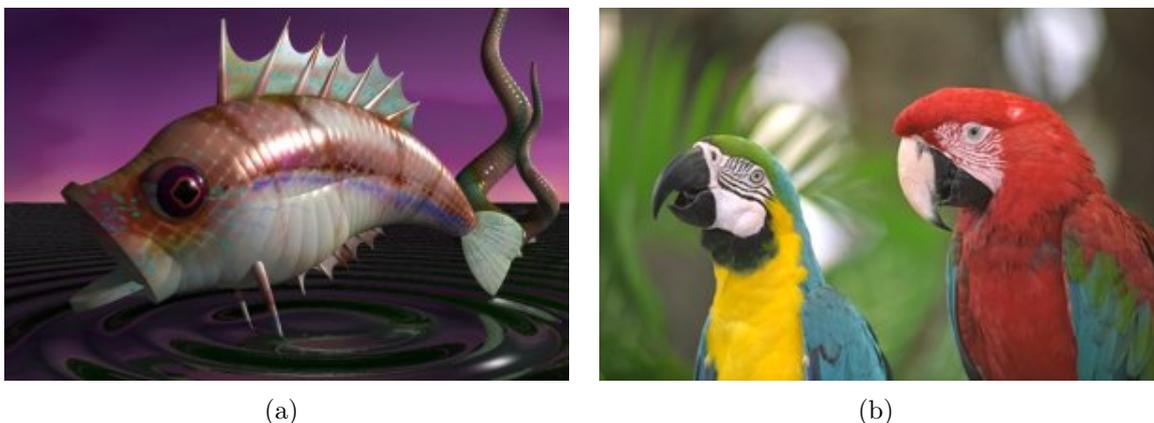


Figura 4.1: Imagens originais "fish" e "parrots" em 24 bits

Foram realizadas quantizações de 4 e 8 bits (16 e 256 cores, respectivamente) e os resultados apresentados foram obtidos através da média de 20 execuções para cada algoritmo. O desempenho dos algoritmos é dado pela distorção gerada na imagem quantizada, dada pela equação:

$$E(I, I') = \sum_{h=1}^H \sum_{w=1}^W (I(h, w) - I'(h, w))^2,$$

em que  $I$  e  $I'$  são as imagens original e quantizada, respectivamente, e  $H$  a altura e  $W$

a largura da imagem, em pixels; e pelo tempo de computação gasto, dado em segundos, também obtido através da média das execuções.

Os melhores e piores resultados aparecem em negrito nas Tabelas 4.1, 4.2, 4.3 e 4.4. Já nas últimas Tabelas, 4.5 e 4.6, somente os melhores resultados estão em evidência.

### 4.1.1 Cenário 1

Neste primeiro cenário foi investigado o comportamento dos algoritmos da literatura com relação à alteração do tipo de clusterização SEQ para RND.

Os resultados, que podem ser vistos nas Tabelas 4.1 e 4.2 mostram que, como esperado, a clusterização RND converge mais rapidamente; entretanto, pode afetar a distorção na imagem.

n° de cores		<i>fish</i>			
		16		256	
inicialização	clusterização	erro	tempo (s)	erro	tempo
FGY	SEQ	272.148307	4.520	<b>33.971963</b>	25.420
	RND	271.220740	2.520	35.739667	<b>10.208</b>
MMX	SEQ	<b>267.286817</b>	<b>10.394</b>	35.159383	<b>73.670</b>
	RND	279.020430	2.544	<b>36.096857</b>	40.334
SFF	SEQ	269.901337	7.346	34.510643	44.264
	RND	267.732903	<b>1.834</b>	35.847300	21.190
KPP	SEQ	<b>286.003980</b>	7.166	34.449163	48.550
	RND	279.458316	2.284	35.924637	32.436

Tabela 4.1: Comparação do uso das inicializações FGY, MMX, SSF e KPP da literatura com clusterização SEQ e RND para a quantização de "fish" 16 e 256 cores

		<i>parrots</i>			
n° de cores		16		256	
inicialização	clusterizacão	erro	tempo (s)	erro	tempo
FGY	SEQ	<b>430.238136</b>	<b>10.868</b>	48.414773	43.322
	RND	441.616757	2.810	<b>51.191487</b>	<b>14.236</b>
MMX	SEQ	439.641767	9.372	48.533037	<b>95.578</b>
	RND	435.497007	4.154	50.706057	47.818
SSF	SEQ	<b>444.835380</b>	8.546	<b>47.574297</b>	56.350
	RND	437.857397	<b>3.216</b>	49.599667	23.516
KPP	SEQ	442.695530	8.258	48.383917	69.396
	RND	434.623727	3.460	50.672430	43.556

Tabela 4.2: Comparação do uso das inicializações FGY, MMX, SSF e KPP da literatura com clusterização SEQ e RND para a quantização de "parrots" 16 e 256 cores

### 4.1.2 Cenário 2

Para a análise do algoritmo FRQ foram realizadas execuções combinando diferentes valores dos parâmetros  $\alpha$  e  $\beta$ , cujos resultados são mostrados nas Tabelas 4.3 e 4.4.

Os melhores resultados são, geralmente, obtidos pela combinação de valores dos parâmetros que restringe as escolhas do algoritmo para níveis de quantizações mais baixos. Já para níveis mais altos, a maior liberdade de escolha do algoritmo favorece os resultados.

n° de cores		<i>fish</i>			
		16		256	
$\alpha$	$\beta$	erro	tempo	erro	tempo
0.00	0.00	321.721547	3.954	<b>157.143023</b>	18.458
	0.01	300.053937	<b>5.112</b>	49.359857	12.528
	0.10	<b>266.322863</b>	2.150	61.955237	<b>61.010</b>
0.10	0.00	318.444254	4.620	68.670517	20.994
	0.01	<b>326.597717</b>	4.550	44.120773	14.822
	0.10	278.454997	<b>1.986</b>	40.112613	47.178
0.30	0.00	289.944433	2.078	41.404577	13.244
	0.01	302.073833	2.554	38.672820	16.068
	0.10	273.821160	2.044	43.009883	37.734
0.70	0.00	268.962657	2.212	35.882107	16.374
	0.01	270.880077	2.112	36.141177	16.622
	0.10	266.587273	2.404	37.815320	35.558
1.00	0.00	280.619910	1.512	<b>35.478403</b>	11.480
	0.01	278.764817	2.034	35.888707	<b>10.098</b>
	0.10	273.266877	2.214	38.260727	24.634

Tabela 4.3: Comparação da combinação de diferentes valores dos parâmetros  $\alpha$  e  $\beta$  para a inicialização FRQ com clusterização RND para a quantização de "fish" 16 e 256 cores

n° de cores		<i>parrots</i>			
		16		256	
$\alpha$	$\beta$	erro	tempo (s)	erro	tempo
0.00	0.00	448.175117	<b>5.688</b>	106.571983	<b>45.754</b>
	0.01	436.503627	5.568	63.008823	21.300
	0.10	<b>422.236567</b>	3.476	63.478083	42.818
0.10	0.00	435.850097	3.042	65.189707	20.080
	0.01	440.169630	3.772	61.215463	19.436
	0.10	440.647833	2.922	67.362213	37.178
0.30	0.00	<b>457.403517</b>	2.826	55.361190	22.818
	0.01	435.309770	2.810	55.152003	23.026
	0.10	438.182000	3.614	59.204133	30.992
0.70	0.00	436.460307	3.048	50.807597	22.036
	0.01	432.416837	3.956	51.456973	25.124
	0.10	430.904050	4.424	54.335590	32.412
1.00	0.00	435.205730	<b>2.686</b>	<b>50.448683</b>	<b>12.406</b>
	0.01	441.210903	4.156	50.460850	15.534
	0.10	439.813277	3.236	<b>145.691987</b>	22.788

Tabela 4.4: Comparação da combinação de diferentes valores dos parâmetros  $\alpha$  e  $\beta$  para a inicialização FRQ com clusterização RND para a quantização de "parrots" 16 e 256 cores

### 4.1.3 Cenário 3

O último cenário conta com a aplicação dos esquemas de inicialização (FGY, MMX, SFF, KPP e FRQ) ao algoritmo KMM, estudando seu comportamento mediante a variação do expoente.

A análise dos resultados mostra que, em geral, a aplicação dos esquemas ao algoritmo KMM e o aumento do expoente podem render bons resultados, embora isso não seja sempre verdade.

		<i>fish</i>			
n° de cores		16		256	
inicialização	expoente	erro	tempo (s)	erro	tempo
FGY	1	268.679747	3.296	36.150154	96.776
	2	268.552050	4.599	35.681857	317.890
MMX	1	269.062790	3.122	35.843867	98.630
	2	267.733592	4.815	36.168807	317.492
SFF	1	270.113257	2.594	35.718383	98.210
	2	271.603765	3.281	35.991584	302.306
KPP	1	268.133440	<b>2.668</b>	36.154250	<b>93.874</b>
	2	<b>266.059492</b>	3.905	35.954397	302.132
FRQ	1	267.003797	3.822	36.034297	92.880
	2	270.200550	3.810	<b>35.610617</b>	303.440

Tabela 4.5: Desempenho do algoritmo KMM inicializado com os esquemas FGY, MMX, SFF, KPP e FRQ, clusterização RND e diferentes valores de EXP para a quantização de "fish" 16 e 256 cores

n° de cores		<i>parrots</i>			
		16		256	
inicialização	expoente	erro	tempo (s)	erro	tempo
FGY	1	448.630077	3.614	50.677260	117.668
	2	439.775910	6.942	50.121433	389.922
MMX	1	455.314983	<b>3.258</b>	51.040777	115.914
	2	443.556247	6.396	<b>49.903333</b>	397.530
SFF	1	435.591667	3.720	50.830887	112.014
	2	433.690610	4.894	51.284467	370.070
KPP	1	<b>426.203553</b>	3.646	51.108707	<b>110.624</b>
	2	449.582030	4.154	50.626167	378.020
FRQ	1	438.995107	3.340	50.757223	111.432
	2	442.539327	5.030	50.366267	377.590

Tabela 4.6: Desempenho do algoritmo KMM inicializado com os esquemas FGY, MMX, SFF, KPP e FRQ, clusterização RND e diferentes valores de EXP para a quantização de "parrots" 16 e 256 cores

# Capítulo 5

## Conclusões e Trabalhos Futuros

A ideia inicial deste trabalho era propor um algoritmo baseado na metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) [Resende e Feo 1995] para o problema de quantização de imagem. Porém, devido ao grande volume de dados que deve ser manipulado nesse tipo de problema, a abordagem se mostrou inviável, no que diz respeito ao tempo computacional requerido por algoritmos de busca local.

Assim, foram apresentadas novas abordagens para a resolução do problema de quantização de imagem, baseadas em alterações nos algoritmos presentes na literatura, bem como estratégias propostas, tendo como base o algoritmo de clusterização de dados *K-Means*.

Embora a principal métrica de qualidade de uma solução para o problema seja a distorção causada pela quantização na imagem, o tempo de computação gasto para executar a tarefa se mostrou um importante atributo do processo. Geralmente, para processos mais longos tem-se como resultado uma imagem com menor distorção. Embora intuitivo, isso não se mostrou sempre verdade, e outros fatores, como o algoritmo utilizado e o ajuste de seus parâmetros, se houver, deva ser levado em consideração.

Assim, uma das sugestões para a continuação deste trabalho é a implementação nas novas abordagens dos métodos citados no Capítulo 2, que aceleram o processo de clusterização do *K-Means* sem causar alterações na solução. No caso deste trabalho, um outro atributo importante que deve ser lembrado é a natureza aleatória do algoritmo *K-Means*, que interfere diretamente no desempenho das abordagens citadas durante este texto.

Como citado nos capítulos anteriores, a distorção causada nas imagens e sua percepção pelo olho humano estão fortemente ligadas ao nível da quantização. Em todos os métodos estudados neste trabalho, o nível da quantização é um parâmetro fixo, e que deve

---

ser informado ao algoritmo. Fica, então, como forte sugestão, o estudo mais profundo da relação entre o nível de quantização e um algoritmo escolhido, assim como a possibilidade do ajuste automático desse parâmetro, levando-se em consideração não somente a distorção em termos do erro calculado, mas com ênfase na percepção da distorção.

# Referências

- [Arthur e Vassilvitskii 2007] Arthur, D. e Vassilvitskii, S. **K-Means++: The Advantages of Careful Seeding**. In: *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, : Society for Industrial and Applied Mathematics, p. 1027–1035, 2007.
- [Bing et al. 2004] Bing, Z.; Junyi, S. e Qinke, P. **An adjustable algorithm for color quantization**. *Pattern Recognition Letters*, v. 25, n. 16, p. 1787–1797, 2004.
- [Braquelaire e Brun 1997] Braquelaire, A. J.-P. e Brun, L. **Comparison and Optimization of Methods of Color Image Quantization**. *IEEE Transactions on Image Processing*, v. 6, n. 7, p. 1048–1052, 1997.
- [Celebi 2009] Celebi, M. E. **Effective Initialization of K-Means for Color Quantization**. In: *ICIP*, : IEEE, p. 1649–1652, 2009.
- [Celebi 2011] Celebi, M. E. **Improving the Performance of K-Means for Color Quantization**. *CoRR*, 2011.
- [Elkan 2003] Elkan, C. **Using the Triangle Inequality to Accelerate K-Means**. In: *ICML*, : AAAI Press, p. 147–153, 2003.
- [Forgy 1965] Forgy, E. **Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications**. *Biometrics*, v. 21, p. 768–780, 1965.
- [Gervautz e Purgathofer 1988] Gervautz, M. e Purgathofer, W. **A simple Method for Color Quantization: Octree Quantization**. In: *New Trends in Computer Graphics*. : Springer Verlag, Berlin, 1988.
- [Gomes e Velho 1997] Gomes, J. e Velho, L. **Image Processing for Computer Graphics**. In: \_\_\_\_\_. : Springer, 1997. cap. 11, p. 293–310.
- [Gonzalez 1985] Gonzalez, T. F. **Clustering to Minimize the Maximum Intercluster Distance**. *Theor. Comput. Sci.*, v. 38, p. 293–306, 1985.
- [Heckbert 1982] Heckbert, P. **Color image quantization for frame buffer display**. *SIGGRAPH Comput. Graph.*, ACM, v. 16, p. 297–307, July 1982. ISSN 0097-8930.
- [Joy e Xiang 1993] Joy, G. e Xiang, Z. **Short communication Center-cut for color-image quantization**. *The Visual Computer*, v. 10, n. 1, p. 62–66, 1993.
- [Judd e Wyszecki 1975] Judd, D. e Wyszecki, G. **Color in Business, Science, and Industry**. : John Wiley and Sons, 1975.

- [Kasuga et al. 2000] Kasuga, H.; Yamamoto, H. e Okamoto, M. **Color quantization using the fast K-means algorithm.** *Systems and Computers in Japan*, v. 31, n. 8, p. 33–40, 2000.
- [Lloyd 1982] Lloyd, S. P. **Least Squares Quantization in PCM.** *IEEE Transactions on Information Theory*, v. 28, n. 2, p. 129–136, 1982.
- [M. Orchard 1991] M. Orchard, C. B. **Color Quantization of Images.** *IEEE Trans. on Signal Processing*, v. 39, p. 2677–2690, 1991.
- [Papamarkos et al. 2002] Papamarkos, N.; Atsalakis, A. e Strouthopoulos, C. **Adaptive color reduction.** *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, v. 32, n. 1, p. 44–56, 2002.
- [Resende e Feo 1995] Resende, M. G. C. e Feo, T. A. **Greedy Randomized Adaptive Search Procedure.** *Journal of Global Optimization*, v. 6, p. 109–133, 1995.
- [Turnbull e Elkan 2005] Turnbull, D. e Elkan, C. **Fast Recognition of Musical Genres Using RBF Networks.** *IEEE Trans. Knowl. Data Eng.*, v. 17, n. 4, p. 580–584, 2005.
- [Uchiyama e Arbib 1994] Uchiyama, T. e Arbib, M. A. **An algorithm for competitive learning in clustering problems.** *Pattern Recognition*, v. 27, n. 10, p. 1415–1421, 1994.
- [Wu 1991] Wu, X. **Efficient Statistical Computations for Optimal Color Quantization.** In: *Graphics Gems Volume II*. : Academic Press, 1991. p. 126–133.
- [Wu 1992] Wu, X. **Color Quantization by Dynamic Programming and Principal Analysis.** *ACM Trans. Graph.*, v. 11, n. 4, p. 348–372, 1992.
- [Xiang 1997] Xiang, Z. **Color Image Quantization by Minimizing the Maximum Intercluster Distance.** *ACM Trans. Graph.*, v. 16, n. 3, p. 260–276, 1997.
- [Yang e Lin 1996] Yang, C.-Y. e Lin, J.-C. **RWM-cut for color image quantization.** *Computers and Graphics*, v. 20, n. 4, p. 577–588, 1996.
- [Özdemir e Akarun 2002] Özdemir, D. e Akarun, L. **A fuzzy algorithm for color quantization of images.** *Pattern Recognition*, v. 35, n. 8, p. 1785–1791, 2002.