

# Resumo

A Computação Evolucionária tem sido bem aplicada na resolução de diversos problemas. Notadamente, a programação evolucionária apresenta bons resultados na minimização de funções multimodais. Tradicionalmente, a classificação de dados (*clustering*) aplicada à segmentação de imagens é realizada com os algoritmos k-médias e fuzzy c-médias, onde os elementos da imagem (*pixels*) são associados a uma quantidade pré-determinada de *clusters*. Esta classificação tem índices de qualidade associados: erro de quantização, distância máxima entre os pontos e os respectivos *clusters* e distância mínima entre os *clusters*. Combinando-se os conceitos do *clustering* de imagens com a Programação Evolucionária, é possível realizar uma otimização dos índices de qualidade da classificação.

# Abstract

The Evolutionary Computing has been successfully applied in resolution of many problems. Notably, the evolutionary programming gives good results in the minimization of multimodal functions. Traditionally, the classification of data (clustering) applied to the segmentation of images is performed with the algorithms k-means and fuzzy c-means, where the image elements (pixels) are associated with a predetermined amount of clusters. This classification is associated with indexes of quality: the quantization error, maximum distance between points and the clusters and distance between clusters. Matching up the concepts of image clustering and Evolutionary Programming, is possible to perform an optimization of that indexes.

# Sumário

<b>RESUMO .....</b>	<b>1</b>
<b>ABSTRACT .....</b>	<b>2</b>
<b>SUMÁRIO .....</b>	<b>3</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>5</b>
<b>ÍNDICE DE TABELAS.....</b>	<b>6</b>
<b>TABELA DE SÍMBOLOS E SIGLAS.....</b>	<b>7</b>
<b>TABELA DE SÍMBOLOS E SIGLAS.....</b>	<b>7</b>
<b>CAPÍTULO 1 INTRODUÇÃO .....</b>	<b>8</b>
1.1 ORGANIZAÇÃO DO TRABALHO.....	8
<b>CAPÍTULO 2 PROGRAMAÇÃO EVOLUCIONÁRIA .....</b>	<b>10</b>
2.1 INTRODUÇÃO .....	10
2.2 ESTRATÉGIAS DE MUTAÇÃO E ABORDAGENS DE EP .....	11
2.3 CONCLUSÃO .....	15
<b>CAPÍTULO 3 SEGMENTAÇÃO DE IMAGENS.....</b>	<b>16</b>
3.1 INTRODUÇÃO .....	16
3.2 SEGMENTAÇÃO DE IMAGENS ATRAVÉS DE <i>CLUSTERING</i> .....	16
3.3 ALGORITMOS DE <i>CLUSTERING</i> .....	18
3.4 CONCLUSÃO .....	22
<b>CAPÍTULO 4 UM NOVO ALGORITMO PARA SEGMENTAÇÃO DE IMAGENS .....</b>	<b>23</b>
4.1 INTRODUÇÃO .....	23
4.2 O ALGORITMO .....	23
4.3 IMAGENS E ÍNDICES ALCANÇADOS .....	26
4.4 ANÁLISE DOS RESULTADOS .....	42

4.5 CONCLUSÃO.....	43
<b>CAPÍTULO 5 .....</b>	<b>44</b>
<b>CONSIDERAÇÕES FINAIS E TRABALHOS.....</b>	<b>44</b>
<b>FUTUROS .....</b>	<b>44</b>
5.1 TRABALHOS FUTUROS.....	45
<b>BIBLIOGRAFIA.....</b>	<b>46</b>

# Índice de Figuras

<b>FIGURA 1.</b>	Algoritmo de EP .....	11
<b>FIGURA 2.</b>	Função Densidade Gaussiana e de Cauchy .....	13
<b>FIGURA 3.</b>	Classificação da imagem “cérebro” .....	26
<b>FIGURA 4.</b>	Classificação da imagem “ctps” .....	27
<b>FIGURA 5.</b>	Classificação da imagem “esqui” .....	28
<b>FIGURA 6.</b>	Classificação da imagem “peixe” .....	29
<b>FIGURA 7.</b>	Classificação da imagem “flor” .....	30
<b>FIGURA 8.</b>	Classificação da imagem “mandril” .....	31
<b>FIGURA 9.</b>	Classificação da imagem “pepper” .....	32
<b>FIGURA 10.</b>	Classificação da imagem “laika” .....	33
<b>FIGURA 11.</b>	Classificação da imagem “queixa” .....	34
<b>FIGURA 12.</b>	Classificação da imagem “rino” .....	35
<b>FIGURA 13.</b>	Classificação da imagem “sapatos” .....	36
<b>FIGURA 14.</b>	Classificação da imagem “soldado” .....	37
<b>FIGURA 15.</b>	Classificação da imagem “starfish” .....	38
<b>FIGURA 16.</b>	Classificação da imagem “tigre” .....	39
<b>FIGURA 17.</b>	Classificação da imagem “mulher” .....	40

# Índice de Tabelas

TABELA 1. Resultados das classificações .....	41
TABELA 2. Resultados das classificações (continuação).....	42

# Tabela de Símbolos e Siglas

(Dispostos por ordem de aparição no texto)

EP – Evolutionary Programming (Programação Evolucionária)

CEP – Classical Evolutionary Programming

FEP – Fast Evolutionary Programming

LEP – Lévy-type Evolutionary Programming

SPMEP - Single Point Mutation Evolutionary Programming

MSEP - Mixed Strategy Mutation Evolutionary Programming

# Capítulo 1

## Introdução

Segmentar uma imagem consiste em extrair desta um conjunto de *pixels* com alguma característica em comum. A automação da segmentação, de acordo com certo critério, não é uma tarefa trivial e diversas abordagens têm sido propostas, tais como k-médias e fuzzy c-médias. Contudo, não há uma unanimidade quanto ao método ideal.

A Computação Evolucionária pode ser definida como uma subárea da Inteligência Artificial, representada por um conjunto de cientistas, idéias e aplicações em constante evolução, que vem despertando o interesse de biólogos, cientistas da computação e engenheiros; a programação evolucionária (ou EP, do inglês *Evolutionary Programming*), uma vertente da Computação Evolucionária, vem sendo aplicada, especialmente, na resolução de diversas tarefas de automação. Na EP, a idéia central é a aplicação dos princípios de Darwin em algoritmos de automação. Recriando a pressão exercida pelo ambiente, que gera um método de seleção natural, ocorreria um aumento gradativo da aptidão (*fitness*) dos elementos; dessa forma o elemento mais apto para solução do problema seria o selecionado. De acordo com o problema proposto, pode-se escolher o algoritmo de EP mais adequado.

O presente trabalho propõe-se a examinar os resultados obtidos a partir da aplicação dos conceitos da EP na segmentação de imagens, comparando os resultados obtidos com um modelo tradicional de classificação de imagens, o k-médias. Adicionalmente, um novo algoritmo para clustering de imagens é apresentado.

### 1.1 Organização do Trabalho

Este trabalho está dividido em cinco capítulos, sendo este primeiro capítulo o de introdução e exposição dos objetivos.

No capítulo 2 os principais conceitos da programação evolucionária, como os algoritmos, estratégias de mutação e fundamentação matemática, pertinentes ao trabalho estão expostos.

No capítulo 3 são abordados os conceitos referentes a classificação de imagens, incluindo o algoritmo utilizado neste trabalho, além da definição dos conceitos de erro de quantização e distâncias intra e inter *cluster*.

O capítulo 4 traz um novo algoritmo para segmentação de imagens. Os resultados obtidos com a aplicação deste algoritmo estão ilustrados.

O capítulo 5 apresenta as conclusões formadas após observação do novo método de classificação de imagens com EP. Também são apresentadas sugestões para trabalhos futuros.

# Capítulo 2

## Programação Evolucionária

### 2.1 Introdução

A Computação Evolucionária pode ser definida como uma subárea da Inteligência Artificial, representada por um conjunto de cientistas, idéias e aplicações em constante evolução, que vem despertando o interesse de biólogos, cientistas da computação e engenheiros. Em comum, esses grupos têm o interesse de compreender melhor os processos evolucionários e de como aplicar os novos conceitos na prática [1].

Considerando a forma como as espécies evoluíram, contornando as adversidades do meio, a idéia de que a seleção natural é um robusto mecanismo de otimização é reforçada. Ainda segundo Fogel [2], o nível de complexidade comportamental atingido pelas espécies, está evidenciado na natureza em vários níveis: as células, os órgãos, os indivíduos, e a população.

A origem da programação evolucionária (*Evolutionary Programming* - EP, do inglês), data da década de 30 [1], quando Sewall Wright lançou a idéia de um sistema evolucionário como sendo uma espécie de explorador, num terreno com aspecto montanhoso, que busca as regiões com altitudes mais elevadas para formar nichos.

Na EP, a idéia central é a aplicação dos princípios de Darwin em algoritmos usados para automação de resolução de problemas complexos para os quais não existem soluções heurísticas ou, quando existem, não apresentam resultados satisfatórios [2]. Recriando a pressão exercida pelo ambiente, que gera um método de seleção natural, ocorreria um aumento gradativo da aptidão (*fitness*) dos elementos [3]. Para reprodução da seleção natural, é necessário ainda o uso de uma função objetivo como referência para avaliação da aptidão do conjunto de soluções candidatas; as melhores soluções encontradas serão usadas para geração de novas soluções. A figura 1 ilustra de forma resumida as etapas de um algoritmo de EP.

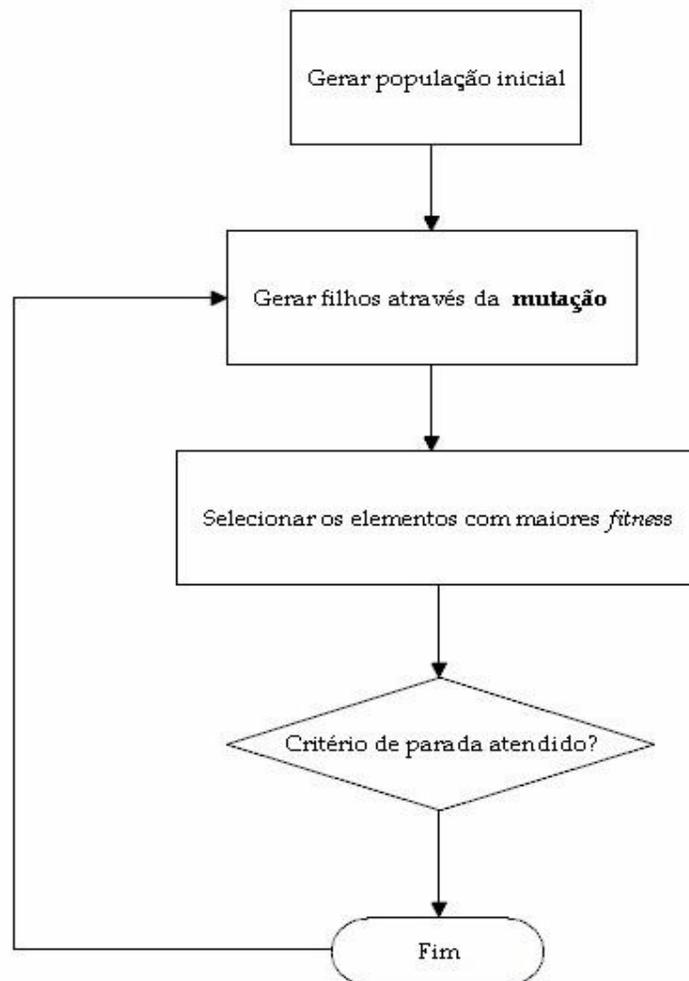


Figura 1. Algoritmo de EP

## 2.2 Estratégias de Mutação e Abordagens de EP

Dentre as etapas de um algoritmo de EP a mutação representa, notadamente, um ponto crítico, pois é ela que guia a busca por novas soluções [4]. Dessa forma, diferentes abordagens têm sido criadas usando diferentes estratégias, a saber: mutação com operador Gaussiano (CEP), mutação com operador de Cauchy (FEP), mutação com operador de Lévy (LEP), mutação *single-point* (SPMEP), mutação com estratégia mista (MSEP), entre outras.

Seguindo o raciocínio desenvolvido por Yao et al (1999) [5], a seleção dos elementos da nova geração é feita através da comparação entre os indivíduos da geração atual e  $q$  indivíduos recém-gerados. Assim, os indivíduos com os mais altos graus de aptidão serão, naturalmente, os sobreviventes, sem, contudo excluir-se a possibilidade de indivíduos com menor aptidão também serem selecionados, dada natureza estocástica do processo de mutação [3].

Como demonstrado por Yao, na abordagem clássica **CEP**, cada indivíduo é representado por um par de vetores  $(x_i, \eta_i)$ , onde  $x_i$ 's são pontos no espaço de busca e  $\eta_i$ 's são desvios-padrão para mutações Gaussianas, ambos com  $n$  dimensões. Primeiramente, deve ser criada a população inicial com  $\mu$  elementos e ajustar  $k = 1$ . Em seguida, a avaliação da aptidão desses indivíduos é feita através da função objetivo,  $f(x_i)$ . Posteriormente, inicia-se a mutação. Cada par de vetores  $(x_i, \eta_i)$  produzirá apenas um descendente  $(x_i', \eta_i')$ , calculado da seguinte forma, para  $j = 1, \dots, n$ :

$$x_i'(j) = x_i(j) + \eta_i(j) N_j(0,1) \quad (2.1)$$

$$\eta_i'(j) = \eta_i(j) \exp( -\tau' N(0,1) + \tau N_j(0,1) ) \quad (2.2)$$

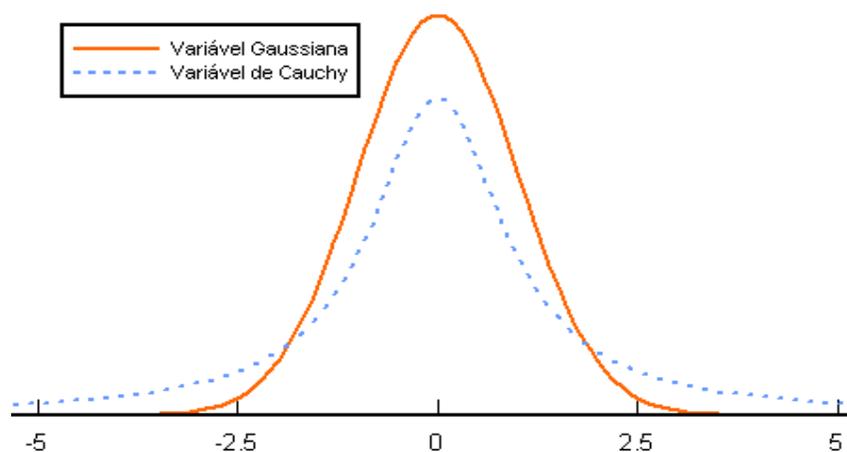
Onde  $x_i(j)$ ,  $x_i'(j)$ ,  $\eta_i(j)$  e  $\eta_i'(j)$  representam o  $j$ -ésimo componente dos vetores  $x_i$ ,  $x_i'$ ,  $\eta$ , e  $\eta_i'$ , respectivamente.  $N(0,1)$  representa uma distribuição aleatória normalmente distribuída com média 0 e desvio padrão 1.  $N_j(0,1)$  representa um novo número aleatório gerado para cada valor de  $j$ . Os fatores  $\tau$  e  $\tau'$  são iguais a  $(\sqrt{2\sqrt{n}})^{-1}$  e  $(\sqrt{2n})^{-1}$ , respectivamente. Após calcular a aptidão de cada descendente, os pais deverão ser comparados com seus filhos. Para cada indivíduo,  $q$  elementos são aleatoriamente escolhidos para esta comparação. Do conjunto atual, os  $\mu$  elementos com os melhores *fitness* serão escolhidos para a próxima geração. Nesse momento, deve-se consultar o critério de parada. Caso este tenha sido atendido, o algoritmo é encerrado; caso contrário, reiniciar o algoritmo fazendo  $k = 2$ .

O algoritmo **FEP**, apresentado por Yao et al [5], faz uso da função densidade de Cauchy, definida como  $f(x) = \frac{1}{\pi} \frac{t}{t^2 + x^2}$ , onde  $t$  é o parâmetro escalar. O algoritmo CEP é modificado, unicamente, em (2.1) onde haverá:

$$x_i'(j) = x_i(j) + \eta_i(j)\delta_j \quad (2.3)$$

Onde  $\delta$  é uma variável aleatória de Cauchy, com parâmetro escalar  $t = 1$  e é gerada para cada valor de  $j$ .

Assim como exposto por Yao [5], dada as características da função densidade de Cauchy, seu uso na mutação de algoritmos evolucionários propicia uma busca mais eficiente em termos de abrangência, ou seja, é menos provável (considerando-se o uso de um variável do tipo gaussiana) que o resultado final do algoritmo seja um mínimo local. O formato da curva da função de Cauchy está ilustrado na figura 2. As extremidades, mais alongadas que a da curva gaussiana, indica uma maior habilidade em buscas por mínimos de funções mais complexas; enquanto que o formato da curva gaussiana corrobora o melhor comportamento do algoritmo CEP em buscas mais detalhadas ao redor de uma vizinhança. Outros fatores são determinantes no desempenho dos algoritmos, como o tamanho do passo no espaço de busca e a distância entre o ponto atual e o ponto ótimo.



**Figura 2.** Função densidade Gaussiana e de Cauchy

No algoritmo **LEP**, apresentado por Ywamatsu (2002) [6], o operador usado na mutação é a função densidade de Lévy, definida como  $f(x; c) = \sqrt{\frac{c}{2\pi}} e^{\frac{-c}{2x}} x^{\frac{-3}{2}}$ , onde  $c$  é o parâmetro escalar. Assim como no FEP, o único ponto alterado em relação ao CEP será em (2.1), onde haverá:

$$x_i'(j) = x_i(j) + \eta_i(j)L_j(\beta) \quad (2.4)$$

Quando o parâmetro escalar  $\beta$  for igual a 1, o algoritmo torna-se equivalente ao CEP; quando for igual a 2, torna-se equivalente ao FEP. Contudo, na prática é difícil determinar qual o valor ideal de  $\beta$  para um dado problema [4].

Para o algoritmo *single-point mutation evolutionary programming*, **SPMEP**, apresentado por Ji et al (2004) [7], as equações (2.1) e (2.2), serão substituídas, respectivamente, por:

$$x_i'(j) = x_i(j) + \eta_i N_i(0,1) \quad (2.5)$$

$$\eta_i' = \eta_i \exp(-\alpha) \quad (2.6)$$

Onde  $j$  será escolhida aleatoriamente a partir do conjunto  $\{1, \dots, n\}$ . Os outros componentes de  $x_i'$  serão iguais aos respectivos  $x_i$ 's. Ou seja, a cada iteração, apenas um componente de cada solução sofre mutação, e o desvio-padrão, dado por (VI) é fixo [7]. O SPMEP é superior ao CEP e FEP para funções com alta dimensionalidade e muitos picos, porém não é tão efetivo quanto o CEP em funções com baixa dimensionalidade e poucos picos [4].

Baseando-se na teoria do jogo evolucionário, o **MSEP** introduz uma nova estratégia de mutação [4]. Os indivíduos passam a ter um vetor de estratégias, tais como CEP, FEP e LEP, e um vetor de probabilidades associados a estas estratégias, que será usado como referência pelo indivíduo, quando ocorrer a mutação, para decidir qual a estratégia tem maior probabilidade de sucesso.

## 2.3 Conclusão

A Computação Evolucionária aplica um modelo consagrado durante a evolução das espécies. Um ambiente externo que força a evolução (aumento no valor dos *fitness* dos elementos) é recriado, e as soluções candidatas para um dado problema geram descendentes. A partir desse processo o elemento mais apto será a solução para o problema em questão.

A forma como ocorre a geração de novos elementos é um ponto crítico para o desempenho do processo. As abordagens CEP, FEP, LEP e MSEP, possuem particularidades nas suas implementações que se traduzem em diferentes resultados obtidos para diferentes situações. Contudo, não pode-se apontar melhor ou pior, e sim indicar qual abordagem melhor se adequa à determinada situação.

# Capítulo 3

## Segmentação de Imagens

### 3.1 Introdução

De acordo com Gonzalez [8], a segmentação de uma imagem consiste na subdivisão de uma imagem em suas partes constituintes e o nível dessa subdivisão depende da aplicação. Diversos sistemas fazem uso da segmentação de imagens, tais como sistemas de processamento de imagens médicas que buscam o embasamento de diagnósticos através do detalhamento de regiões de tomografias ou radiografias, ou de imagens aéreas capturadas por aviões para análise da área sobrevoada, ou ainda de imagens de um ambiente qualquer capturadas pela câmera de um robô, para a devida interpretação do sistema que coordena seus movimentos. Estes exemplos ilustram a importância do papel que a segmentação representa nas aplicações atuais.

Dessa forma, existem diversas abordagens novas que vêm sendo propostas para realização da segmentação de imagens, como o uso Evolução Diferencial, feito por Das e Chakraborty (2008) [9] e Das e Konar (2009) [10] e de Otimização por Enxame de Partículas, realizado por Omran et al (2005) [11].

Neste capítulo serão abordados os conceitos relativos a segmentação de imagens, assim como a fundamentação matemática da teoria de *clustering*; também serão apresentados os índices de avaliação de *clustering* usados na fase de experimentos.

### 3.2 Segmentação de Imagens Através de *Clustering*

Segmentar uma imagem consiste em extrair desta um conjunto de pixels com alguma característica em comum. Uma segmentação bem feita é fundamental para determinados sistemas na área da visão computacional [9]; ademais, considera-se que a segmentação de imagens não triviais é uma das mais difíceis tarefas do

Processamento Digital de Imagens [10]. A automação da segmentação, de acordo com certo critério, não é uma tarefa trivial e diversas abordagens têm sido propostas, como o uso dos algoritmos k-médias e fuzzy c-médias; contudo, não há uma unanimidade quanto ao método ideal.

A partir da aplicação de um algoritmo de clustering, a imagem recebida como entrada para o processo de segmentação terá cada um dos seus pixels associados a algum dos *clusters* utilizados, de modo que os *pixels* identificados sob o mesmo cluster, devem ser o mais similar possível, enquanto os *pixels* identificados sob diferentes *clusters*, devem ser o mais diferente possível. Esta associação entre pixels e clusters é feita de acordo com a distância observada entre as bandas do pixel (R,G,B) e os centróides dos clusters: o pixel será associado ao centróide mais próximo.

Os algoritmos citados anteriormente, são exemplos da aplicação da teoria da classificação de dados (*clustering*) na segmentação de imagens. Tal processo consiste em organizar um conjunto de dados em grupos (*clusters*), onde os elementos pertencentes a um mesmo grupo devem ser o mais similar possível, e os elementos de grupos distintos são o mais diferente possível. Os algoritmos de classificação podem ser divididos entre supervisionados e não-supervisionado [11]. No primeiro grupo estão os algoritmos onde um supervisor fornece dados (média ou variância) que serão usados no processo de *clustering*; no segundo grupo, os algoritmos têm como entrada apenas o conjunto de dados que será classificado. Os algoritmos de *clustering* podem ainda ser classificados como claro (*crisp*) e não-claro (*fuzzy*). Em um algoritmo do tipo *crisp*, cada elemento pertence somente a um grupo; num algoritmo *fuzzy*, os elementos podem pertencer a mais de um grupo.

Os algoritmos de *clustering* não-supervisionados podem ser classificados em hierárquicos ou particionários; nos hierárquicos a saída será uma árvore onde cada cluster é uma partição do conjunto de dados; já os algoritmos particionários procuram dividir o conjunto de dados diretamente em *clusters* disjuntos, procurando minimizar algum critério (função do erro quadrático, por exemplo). Segundo Omran [11], o algoritmo não-supervisionado mais usado é o k-médias. Nesse algoritmo, no princípio, têm-se k *clusters* (determinados aleatoriamente ou obtidos a partir de alguma informação prévia) e cada *pixel* da imagem é associado ao centróide do

cluster mais próximo. Os centróides são então atualizados de acordo com os pixels e o processo é repetido até que um critério de parada seja satisfeito. As desvantagens para o uso do k-médias são:

- O usuário deve informar a quantidade de centróides;
- A abordagem usada no algoritmo tende a soluções não-ideais.

Um outro algoritmo bastante usado na segmentação de imagens é o fuzzy c-médias, que é considerado uma evolução do k-médias. Nesse algoritmo, cada pixel pode pertencer a mais de um cluster. Essa classificação é feita seguindo a lógica Fuzzy, de modo que os pixels têm um grau de inclusão para um dado *cluster*, ao contrário da abordagem rígida adotada no k-médias .

### 3.3 Algoritmos de *Clustering*

Nesta seção serão apresentados os algoritmos de segmentação usados neste trabalho. Assim como exposto por Das [10], um padrão é uma estrutura física ou abstrata de objetos, que diferencia-se de outros padrões por uma coleção de atributos, definida como características. Considerando  $P = \{P_1, P_2, \dots, P_n\}$  como sendo um conjunto de padrões, cada um com  $d$  características, pode-se definir uma matriz  $X_{n \times d}$  com  $n$  vetores de dimensão  $d$  para representar tal conjunto, onde cada linha  $X_i$  é um vetor (ou objeto do conjunto  $P$ ), composto por elementos  $x_{i,j}$  que representam a  $j$ -ésima característica do objeto  $X_i$ .

Então, um algoritmo de *clustering* particionário deverá identificar uma partição  $C = \{C_1, C_2, \dots, C_c\}$  onde a similaridade entre padrões de um mesmo cluster  $C_i$  seja máxima, e entre padrões de diferentes clusters seja mínima.

De acordo com Omran [11], as etapas de um algoritmo de *clustering* devem ser:

1. Inicializar aleatoriamente as médias dos  $N_c$  *clusters*;
2. **Repetir**

a) para cada pixel  $z_p$  da imagem, calcular seu grau de inclusão (será definido adiante)  $u(m_j | z_p)$  para cada centróide  $m_j$ , e seu peso  $w(z_p)$ ;

b) recalculer a média dos  $N_c$  clusters com:

$$m_j = \frac{\sum_{\forall z_p} u(m_j | z_p) w(z_p) z_p}{\sum_{\forall z_p} u(m_j | z_p) w(z_p)}, \text{ para } j = 1, \dots, N_c \quad (3.1)$$

**Enquanto** um critério de parada não for satisfeito.

O cálculo grau de inclusão e do peso dos elementos é definido de acordo com o algoritmo usado. Para o k-médias, tem-se:

$$u(m_j | z_p) = \begin{cases} 1 & \text{se } \arg \min_j \|z_p - m_j\|^2 \\ 0 & \text{caso contrário} \end{cases} \quad (3.2)$$

e

$$w(z_p) = 1 \quad (3.3)$$

Para o algoritmo fuzzy c-médias, tem-se:

$$u(m_j | z_p) = \frac{\|z_p - m_j\|^{-2/(q-1)}}{\sum_{j=1}^{N_c} \|z_p - m_j\|^{-2/(q-1)}} \quad (3.4)$$

onde  $q$  é um expoente não-fuzzy maior ou igual a 1; além de:

$$w(z_p) = 1 \quad (3.5)$$

Para avaliar a qualidade da partição realizada por um dado algoritmo de *clustering*, é necessário o uso de índices matemáticos. Estes índices foram usados nos testes apresentados neste trabalho.

Seguindo a abordagem de Omran [11], uma opção é o erro de quantização, definido como:

$$J_e = \frac{\sum_{j=1}^{N_c} \left| \sum_{\forall z_p \in C_j} d(z_p, m_j) \right| |C_j|}{N_c} \quad (3.6)$$

onde  $N_b$  é a quantidade de bandas espectrais da imagem;  $N_p$  é quantidade de pixels da imagem;  $N_c$  é quantidade de classes espectrais (*clusters*);  $z_p$  representa os componentes de  $N_b$  no pixel  $p$  e  $m_j$  representa a média do cluster  $j$ . O valor de  $d(z_p, m_j)$  é definido por:

$$d(z_p, m_j) = \sqrt{\sum_{k=1}^{N_b} (z_{pk} - m_{jk})^2} \quad (3.7)$$

Outro ponto importante no trabalho de Omran [11], também utilizado neste trabalho, diz respeito à função objetivo usada na otimização multiobjetiva, adaptada para a abordagem com programação evolucionária, utilizada no presente trabalho. Dessa forma, pode-se assumir que, no contexto da programação evolucionária, cada indivíduo  $x_i$  é composto por  $x_i = (m_{i1}, \dots, m_{ij}, \dots, m_{iN_c})$ , onde  $m_{ij}$  representa o  $j$ -ésimo vetor do centróide do cluster do  $i$ -ésimo indivíduo no algoritmo evolucionário; sendo assim, a qualidade de cada indivíduo pode ser medida através da seguinte função de *fitness*:

$$f(x_i, Z_i) = \omega_1 \bar{d}_{\max}(Z_i, x_i) + \omega_2 (z_{\max} - d_{\min}(x_i)) \quad (3.8)$$

onde:

- $z_{\max}$  é o pixel com valor mais alto da imagem (para uma imagem com  $b$  bits,  $z_{\max} = 2^b - 1$ );

- $Z_i$  é uma matriz com os valores de pertinência dos pixels em relação aos clusters do elemento  $i$ . Cada elemento  $z_{ijp}$  indica se o pixel  $z_p$  pertence ao cluster  $C_{ij}$  do elemento evolucionário  $i$ ;
- As constantes  $\omega_1$  e  $\omega_2$  são definidas pelo usuário;
- $\bar{d}_{\max}(Z_i, x_i) = \max_{j=1, \dots, N_c} \left\{ \sum_{\forall z_p \in C_{ij}} d(z_p, m_{ij}) / |C_{ij}| \right\}$  representa o valor máximo da média das distâncias euclidianas aos seus respectivos *clusters*.  $|C_{ij}|$  representa a cardinalidade do conjunto  $C_{ij}$ ;
- $d_{\min}(x_i) = \min_{\forall j_1, j_2, j_1 \neq j_2} \{d(m_{ij_1}, m_{ij_2})\}$  representa o valor mínimo das distâncias entre quaisquer par de *clusters*.

A partir desses preceitos, Omran [11] propôs uma nova função *fitness*, que será adaptada no presente trabalho:

$$f(x_i, Z_i) = \omega_1 \bar{d}_{\max}(Z_i, x_i) + \omega_2 (z_{\max} - d_{\min}(x_i)) + \omega_3 J_{e,i} \quad (3.9)$$

que é a equação VIII somando-se o erro de quantização apresentado na equação VI.

Outro índice difundido na literatura, usado por Das [9,10] é o Xie-Beni, definido como:

$$XB = \frac{\sum_{i=1}^C \sum_{j=1}^n u_{i,j}^2 \|\vec{X}_j - \vec{V}_i\|^2}{n \times \min_{i \neq j} \|\vec{V}_i - \vec{V}_j\|^2} \quad (3.10)$$

O número ideal de clusters é obtido minimizando-se este índice.

## 3.4 Conclusão

Neste capítulo foram introduzidos os conceitos referentes à segmentação de imagens que servirão como base nos experimentos práticos ilustrados no capítulo seguinte. A segmentação de imagens realizada através de algoritmos de *clustering*, tem se mostrado bastante eficiente para aplicações que necessitam da classificação não supervisionada.

# Capítulo 4

## Um Novo Algoritmo Para Segmentação de Imagens

### 4.1 Introdução

Neste capítulo serão apresentados os resultados obtidos a partir da aplicação da teoria revisada nos capítulos 2 e 3 sobre o *clustering* de imagens e a programação evolucionária respectivamente. Primeiramente é demonstrado o algoritmo utilizado; em seguida, os resultados obtidos estão ilustrados com as imagens resultantes e com os índices referentes das respectivas classificações. Ao final do capítulo, encontra-se a análise dos resultados apresentados.

### 4.2 O Algoritmo

De acordo com o trabalho de Saha e Bandyopadhyay (2008) [12], uma inicialização direcionada dos centróides é mais proveitosa em comparação com uma inicialização aleatória. Dessa forma, após criar a população inicial com 10 indivíduos, os centróides associados a cada um deles foram usados como entrada para 5 interações com o k-médias. A resposta do algoritmo k-médias passa a ser, então, os novos centróides do indivíduo.

A população será representada por uma matriz  $N_i \times (N_c \times 3) + 3$ , onde  $N_i$  representa a quantidade de indivíduos da população e  $N_c$  a quantidade de clusters. Cada linha  $i$  da matriz equivale a um indivíduo e suas respectivas  $N_c$  coordenadas de centróides. Adicionalmente, os indivíduos têm mais três valores associados: seu desvio-padrão, seu *fitness* e a quantidade de vitórias em comparações (quantidade de vezes em que comparando com outro indivíduo teve um *fitness* maior).

A estrutura do algoritmo utilizado é a seguinte:

1. Inicializar todos os parâmetros utilizados: pesos  $(\omega_1, \omega_2, \omega_3)$ , quantidade de comparações entre indivíduos ( $N_{\text{comp}}$ ),  $\tau$ ,  $\tau'$ , desvio inicial dos indivíduos, a população com indivíduos gerados aleatoriamente e a leitura da imagem.
2. Para cada indivíduo, fazer 5 iterações com o algoritmo k-médias; associar os centróides gerados pelo k-médias, como as novas variáveis-objetivo do indivíduo; calcular o *fitness* do indivíduo.
3. Enquanto o critério de parada não for satisfeito, faça:
  - a. Gerar novos indivíduos e calcular os respectivos *fitness*.
  - b. Comparar os indivíduos segundo seus valores de *fitness*, de forma aleatória (entre pais e filhos, entre filhos ou entre pais), e incrementar o número de vitórias do indivíduo “vencedor”.
  - c. Selecionar os indivíduos com maior número de vitórias para serem os pais da próxima geração e zerar número de vitórias dos indivíduos.

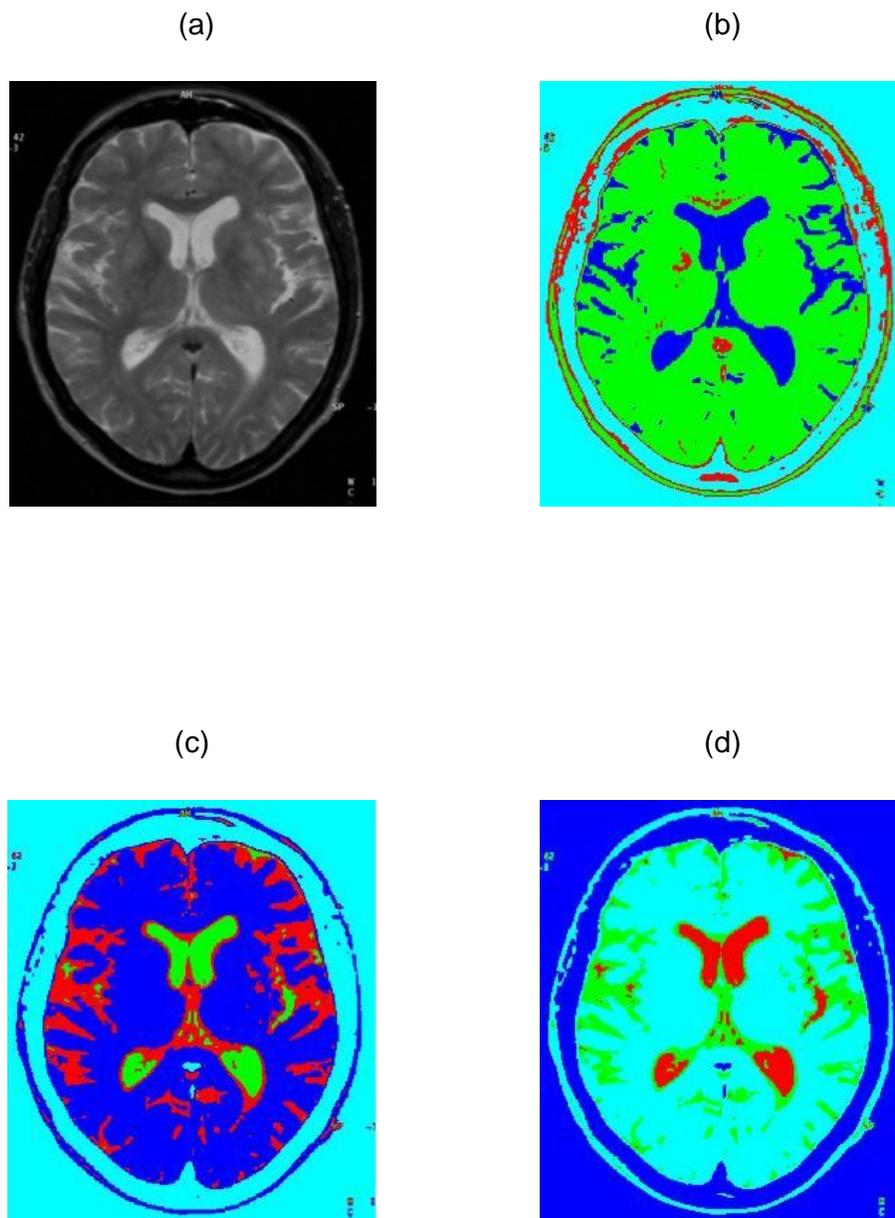
Este algoritmo desenvolvido foi testado com um conjunto de 15 imagens. Os pesos  $\omega_1$ ,  $\omega_2$  e  $\omega_3$  foram ajustados para 0,8, -0,5 e -0,5, respectivamente, para garantir a melhor qualidade possível das imagens segmentadas (estes valores foram obtidos a partir de sucessivas observações); os desvios padrão dos indivíduos foram ajustados, inicialmente, para 3, seguindo os experimentos de Yao [5]. Como critério de parada foi usada uma condição tripla:

- o Variação menor que 0,005 no erro de quantização;
- o Mínimo de 10 gerações (para evitar um falso mínimo local observado com uma quantidade de execuções inferior a este número);
- o A média dos valores dos indivíduos não ter sido alterada, assim o algoritmo se encerrará quando houver a convergência.

O tamanho da população é de 10 indivíduos em todas as segmentações e o limite de iterações é 50. Os dois primeiros critérios citados acima também foram

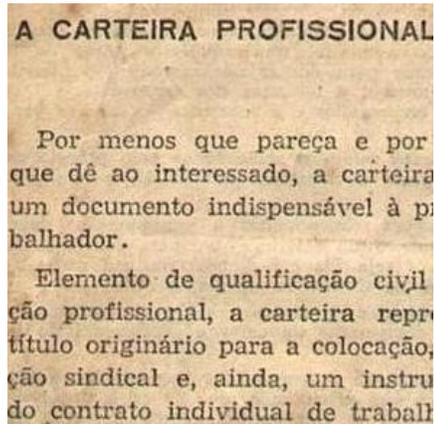
usados na segmentação feita com o k-médias. As imagens resultantes das classificações estão exibidas a seguir.

## 4.3 Imagens e Índices Alcançados

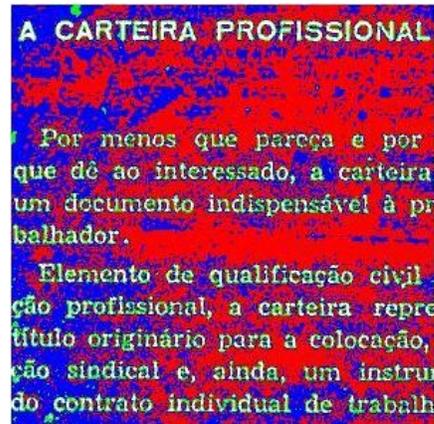


**Figura 3.** (a) Imagem “cérebro” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.

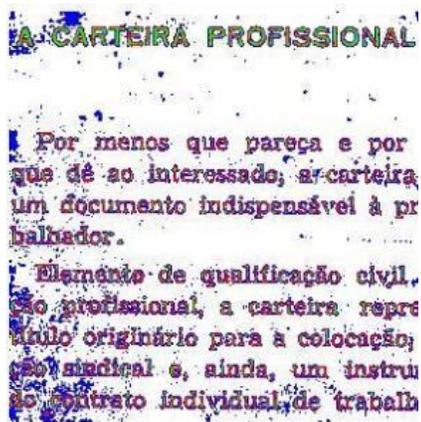
(a)



(b)



(c)



(d)

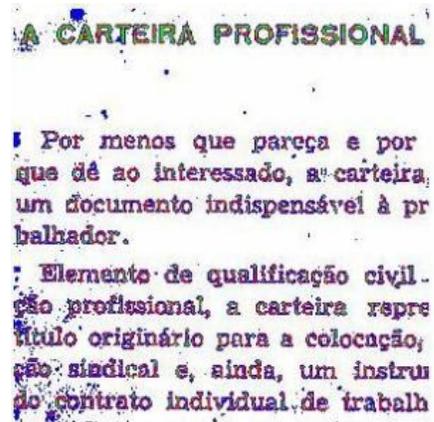
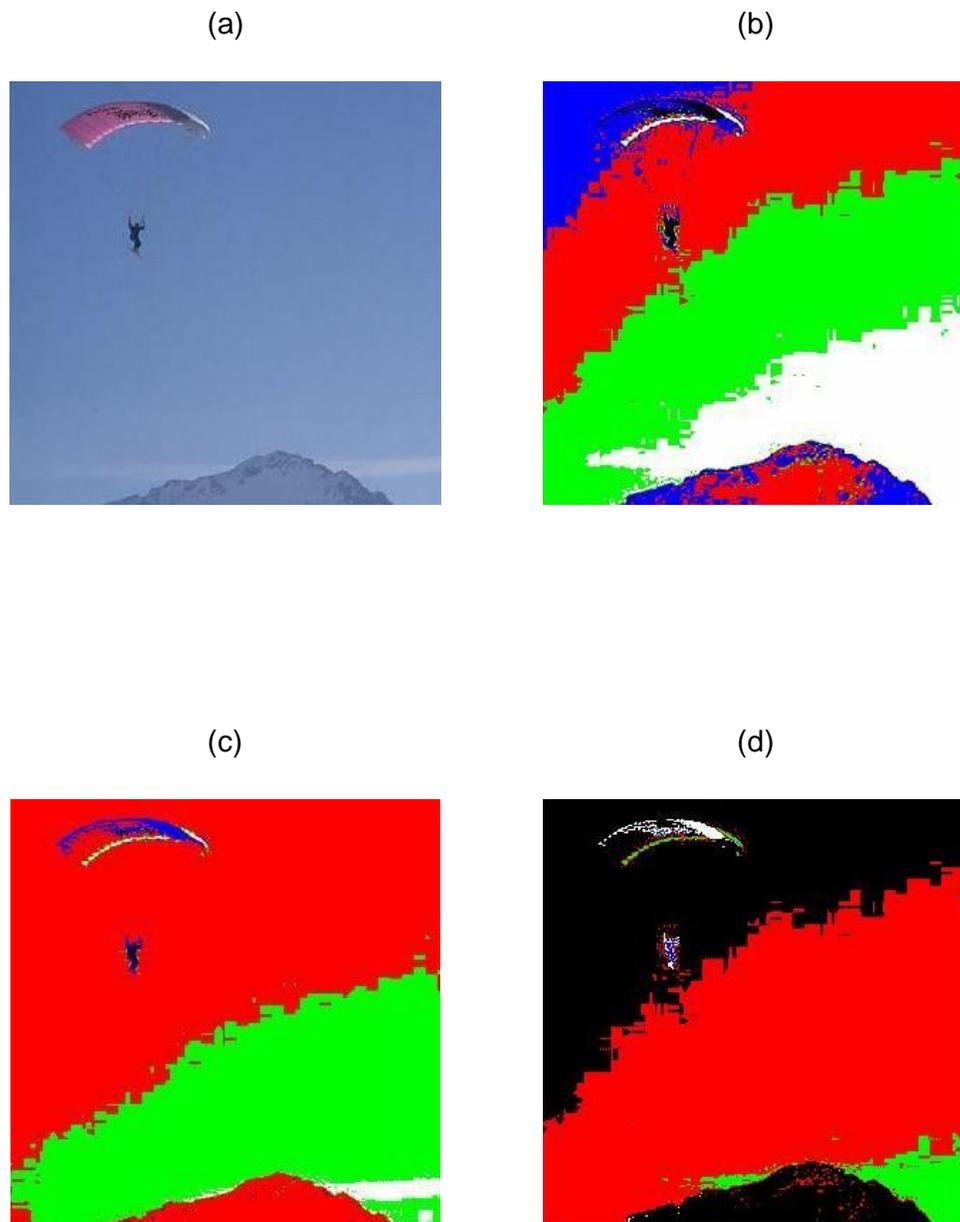
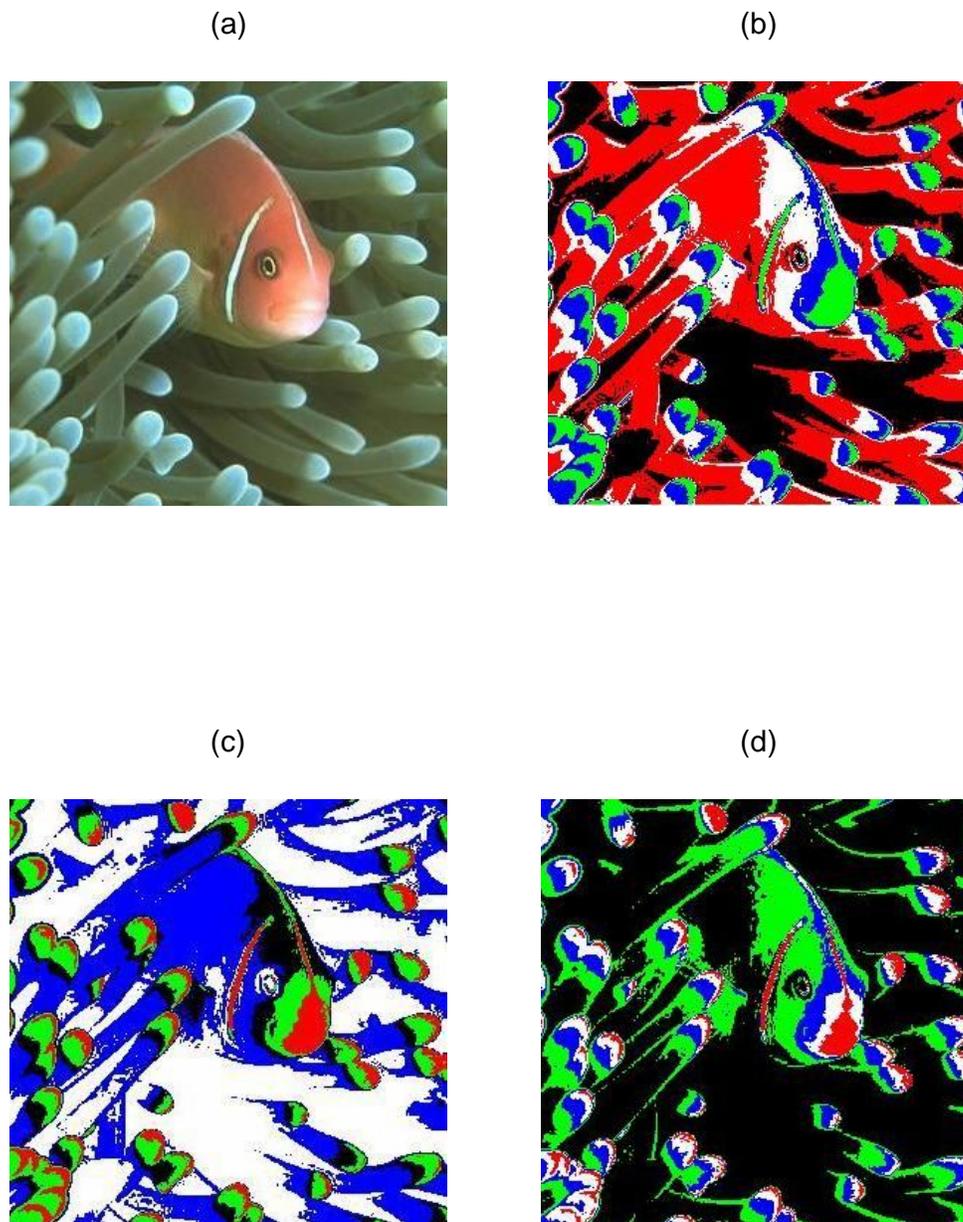


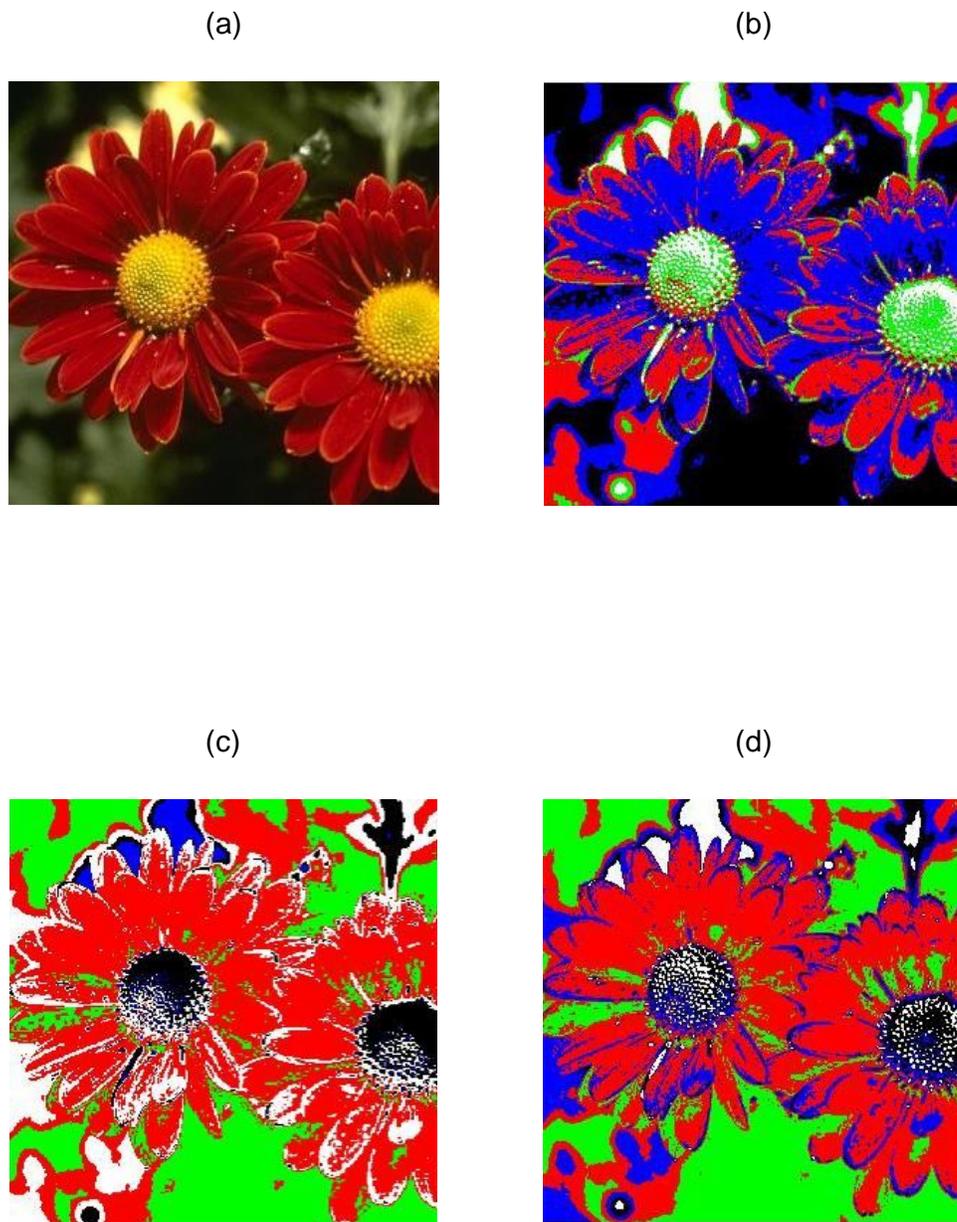
Figura 4. (a) Imagem “ctps” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.



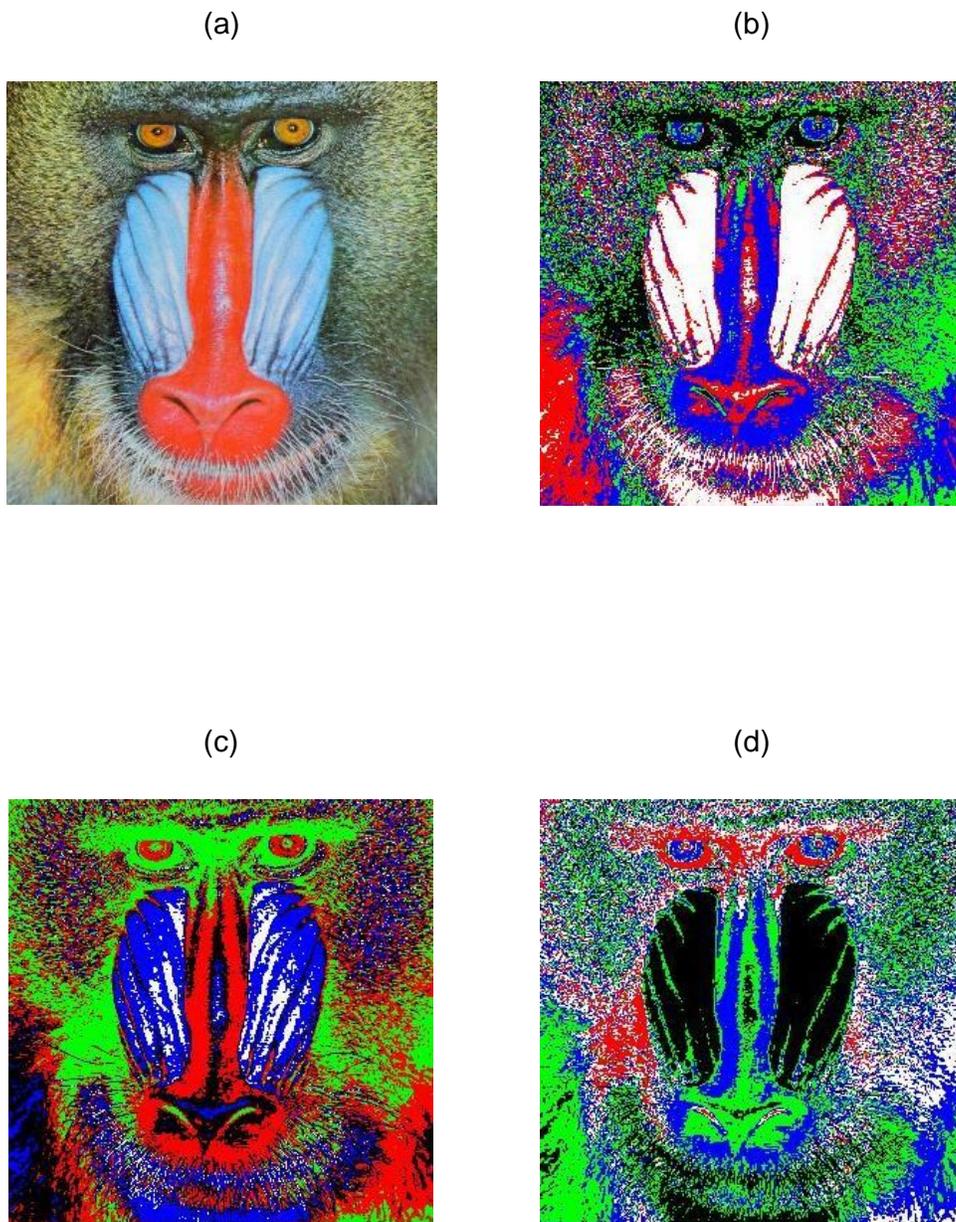
**Figura 5.** (a) Imagem “esqui” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.



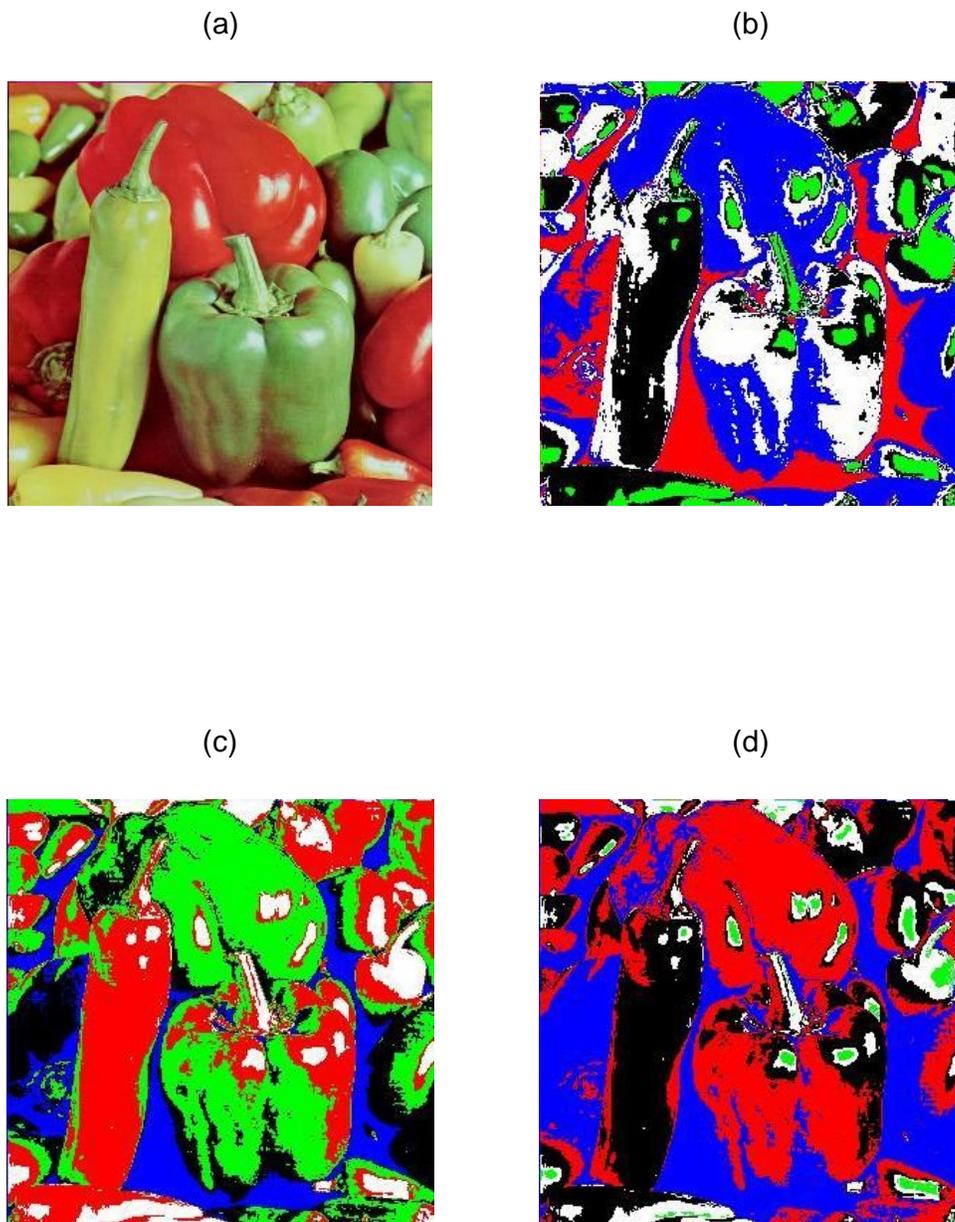
**Figura 6.** (a) Imagem “peixe” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.



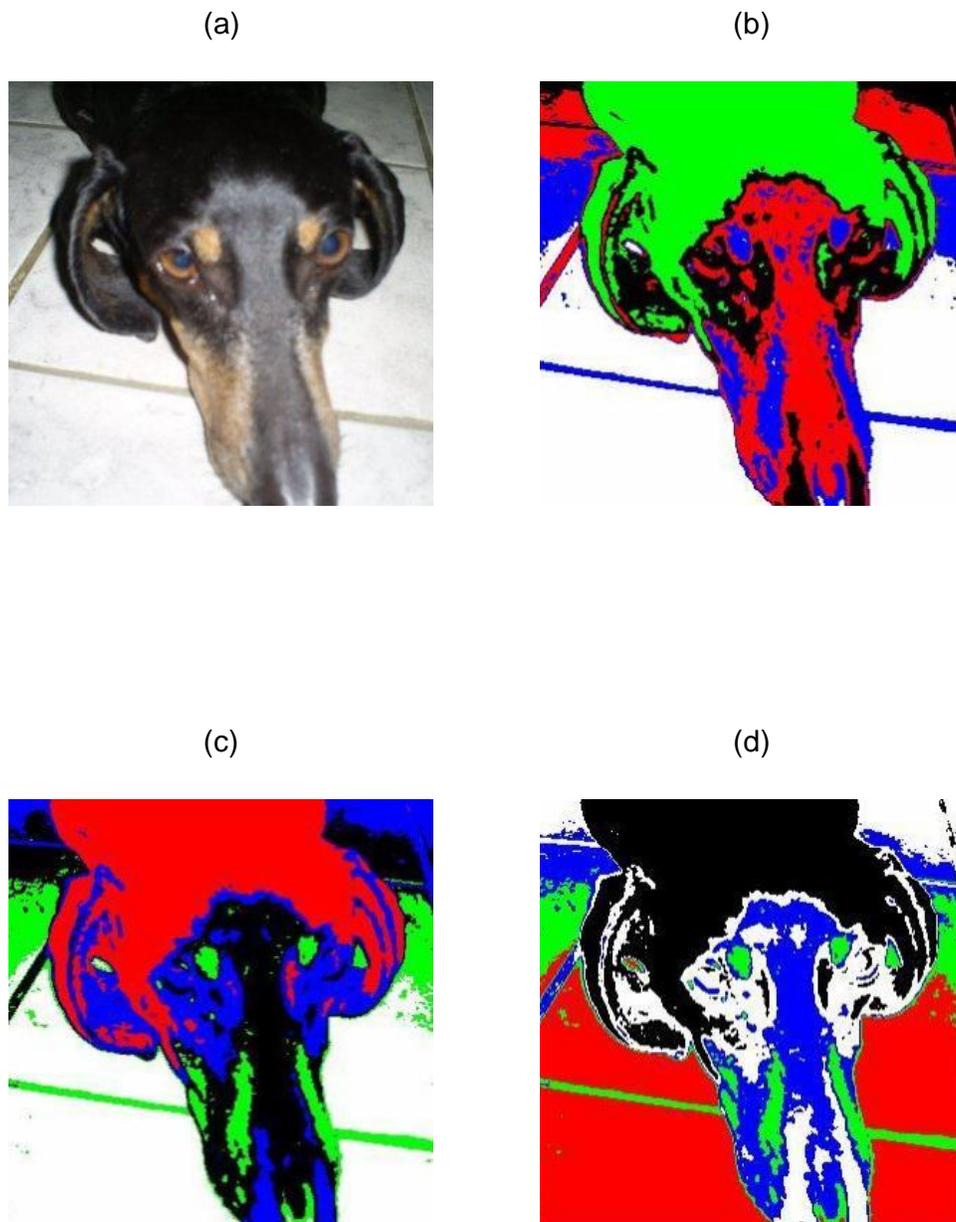
**Figura 7.** (a) Imagem “flor” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.



**Figura 8.** (a) Imagem “mandril” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.

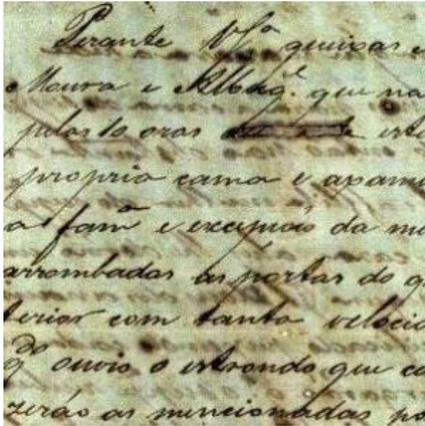


**Figura 9.** (a) Imagem “pepper” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.

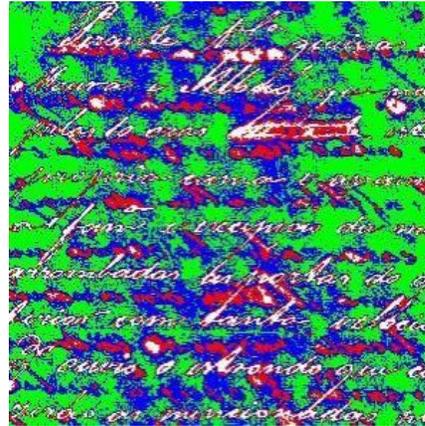


**Figura 10.** (a) Imagem “laika” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.

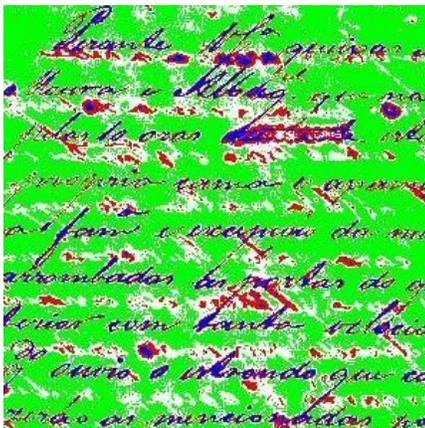
(a)



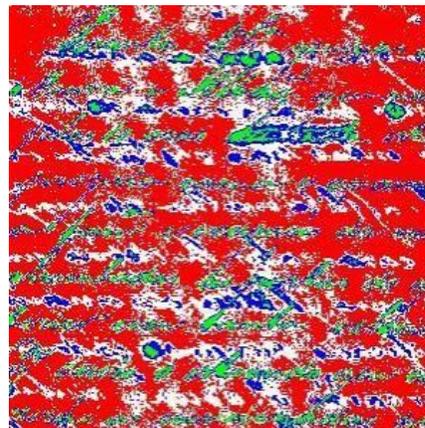
(b)



(c)



(d)



**Figura 11.** (a) Imagem “queixa” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.

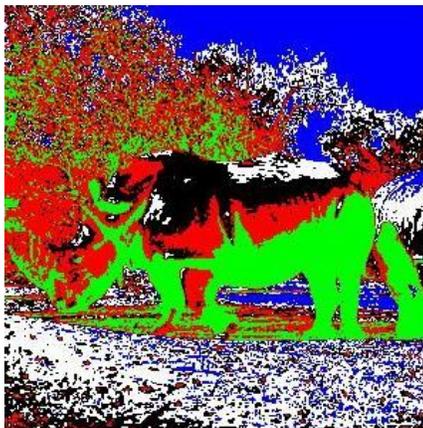
(a)



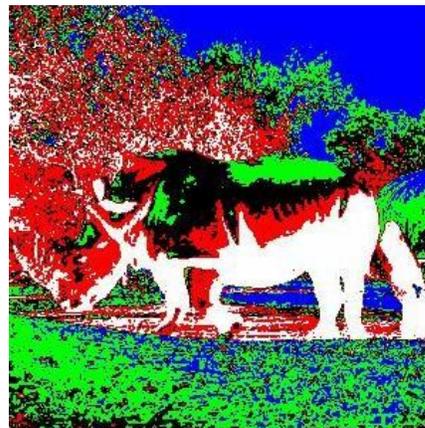
(b)



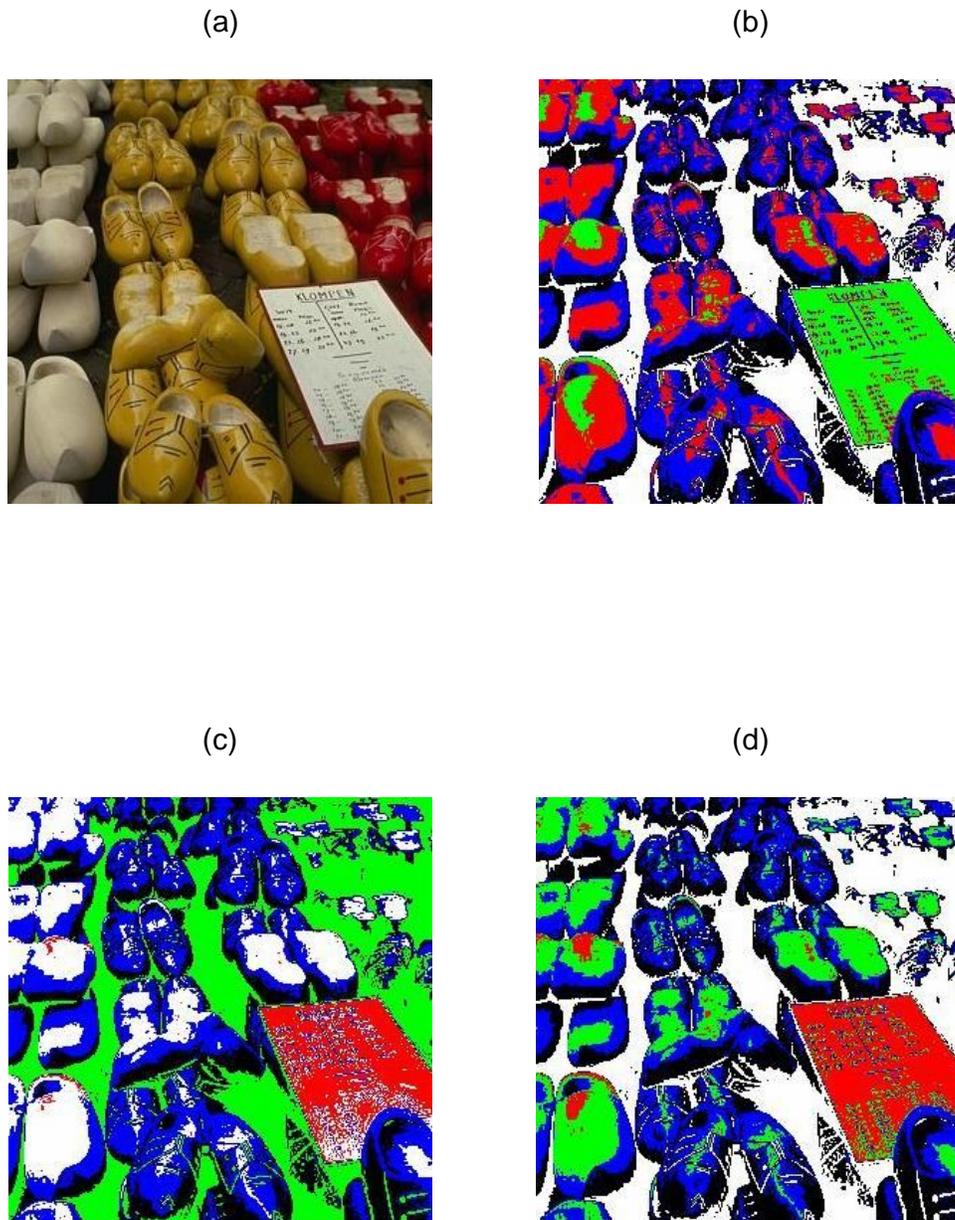
(c)



(d)



**Figura 12.** (a) Imagem “rino” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.

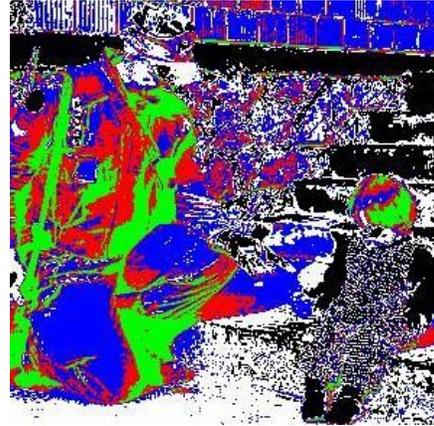


**Figura 13.** (a) Imagem “sapatos” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.

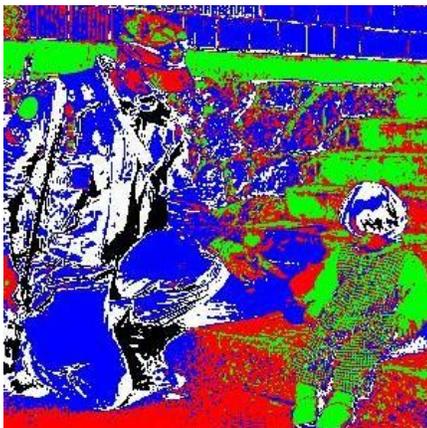
(a)



(b)



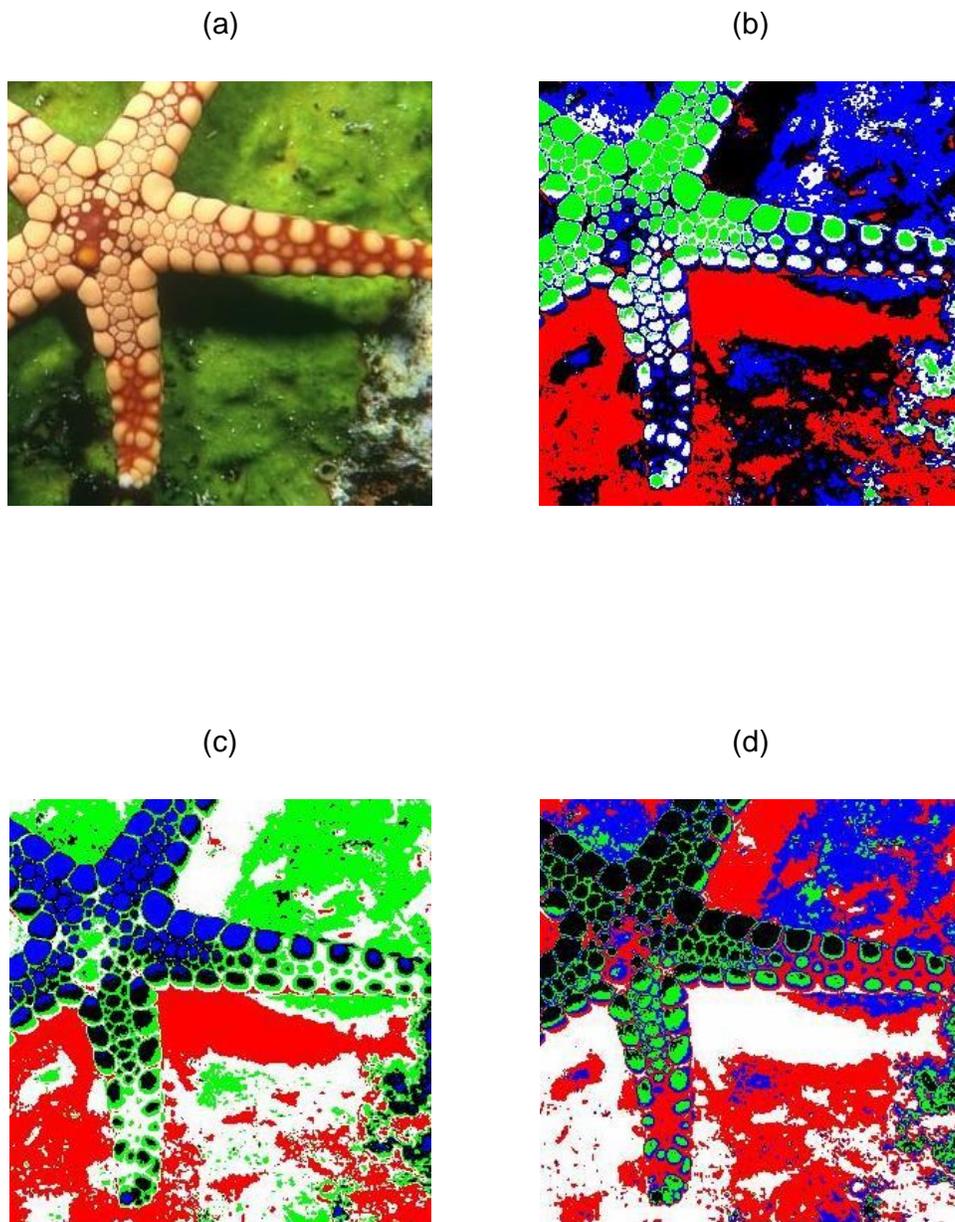
(c)



(d)



**Figura 14.** (a) Imagem “soldado” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.

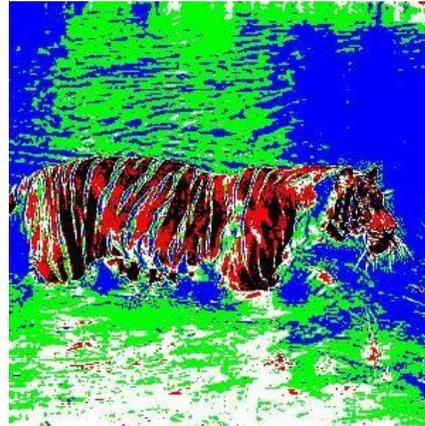


**Figura 15.** (a) Imagem “starfish” original. (b) Classificação com k-médias.  
(c) Classificação com CEP. (d) Classificação com FEP.

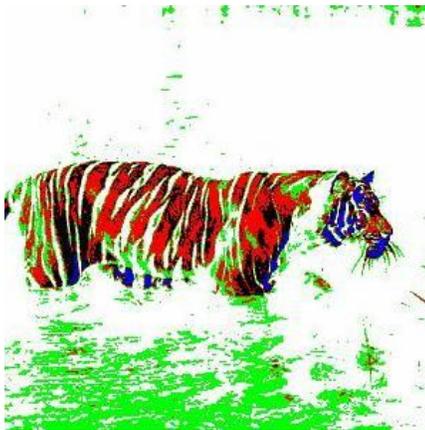
(a)



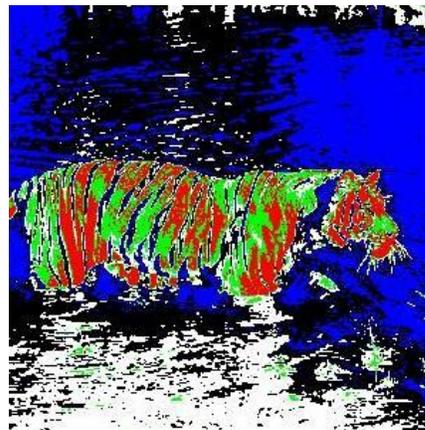
(b)



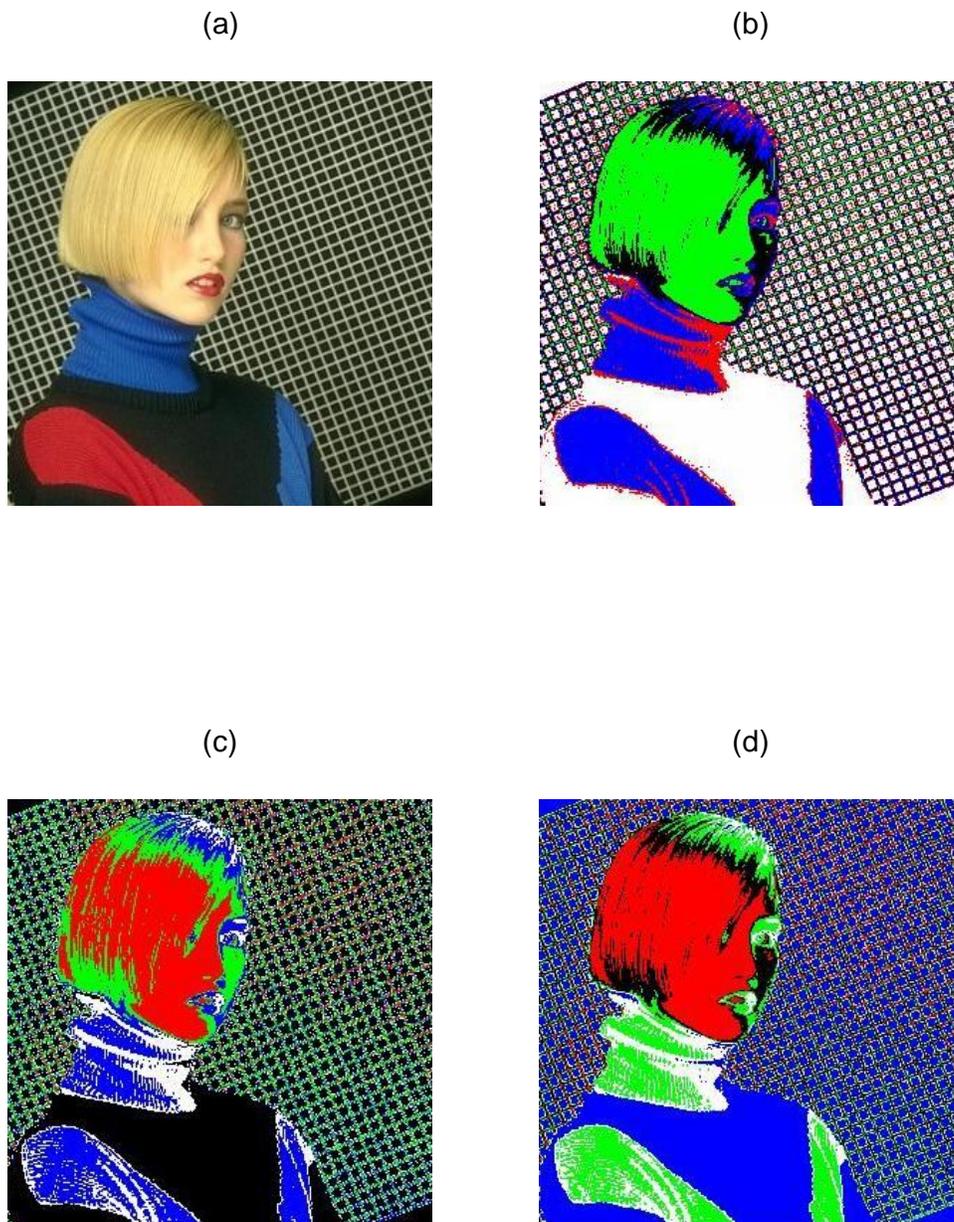
(c)



(d)



**Figura 16.** (a) Imagem “tigre” original. (b) Classificação com k-médias. (c) Classificação com CEP. (d) Classificação com FEP.



**Figura 17.** (a) Imagem “mulher” original. (b) Classificação com k-médias.  
(c) Classificação com CEP. (d) Classificação com FEP.

As tabelas 1 e 2 mostram o erro de quantização, a distância mínima entre os clusters e a distância máxima entre os pontos seus respectivos clusters, obtidos nas segmentações das imagens:

**Tabela 1.** Resultados das classificações

ALGORITMO	QTD CLUSTERS	IMAGEM	ERRO QUANT	DIST MAX	DIST MIN	QTD INTERAÇÕES
cep	4	<b>cérebro</b>	18,1731	23,8689	86,6025	10
k-médias	4	<b>cérebro</b>	17,7627	32,2244	56,0043	14
fep	4	<b>cérebro</b>	17,6415	26,589	71,0141	10
cep	4	<b>ctps</b>	44,587	46,9738	76,2102	22
k-médias	4	<b>ctps</b>	45,4382	47,4589	36,6338	29
fep	4	<b>ctps</b>	43,9846	47,1609	71,0141	10
cep	5	<b>esqui</b>	47,9027	58,0922	32,909	10
k-médias	5	<b>esqui</b>	52,56	57,9602	19,4638	33
fep	5	<b>esqui</b>	48,1051	58,338	38,1051	10
cep	5	<b>peixe</b>	41,2101	51,5251	53,6936	17
k-médias	5	<b>peixe</b>	39,7771	52,0612	43,6027	52
fep	5	<b>peixe</b>	43,4401	50,4526	50,2295	10
cep	5	<b>flor</b>	86,7896	128,1722	50,2295	15
k-médias	5	<b>flor</b>	88,8432	130,2059	42,9195	22
fep	5	<b>flor</b>	90,5647	121,7957	50,2295	11
cep	5	<b>mandril</b>	49,0795	59,0141	51,9615	10
k-médias	5	<b>mandril</b>	46,2288	65,6874	52,5729	18
fep	5	<b>mandril</b>	44,8451	58,4706	51,9615	15
cep	5	<b>pepper</b>	72,6223	90,4829	57,1577	11
k-médias	5	<b>pepper</b>	70,5762	97,9601	53,4896	12
fep	5	<b>pepper</b>	66,7629	90,1271	53,6936	10
cep	4	<b>laika</b>	23,7411	28,9875	83,1384	32
k-médias	4	<b>laika</b>	22,2921	30,977	83,7456	17
fep	4	<b>laika</b>	23,709	28,4515	72,7461	10
cep	4	<b>queixa</b>	38,1241	40,5601	76,2102	10
k-médias	4	<b>queixa</b>	38,036	42,9102	59,0429	20
fep	4	<b>queixa</b>	38,9412	40,1071	76,2102	11
cep	5	<b>rino</b>	43,6071	75,2424	55,4256	10
k-médias	5	<b>rino</b>	39,7033	72,0983	61,9213	21
fep	5	<b>rino</b>	38,5298	71,1557	60,6218	25

**Tabela 2.** Resultados das classificações (continuação)

ALGORITMO	QTD CLUSTERS	IMAGEM	ERRO QUANT	DIST MAX	DIST MIN	QTD INTERAÇÕES
cep	5	sapatos	50,6418	68,5111	57,1577	15
k-médias	5	sapatos	51,1381	70,0351	61,5582	23
fep	5	sapatos	49,8998	69,7634	65,8179	10
cep	5	soldado	65,2358	79,8616	58,8897	50
k-médias	5	soldado	67,2386	79,1141	60,5565	13
fep	5	soldado	65,4647	77,5524	67,55	10
cep	5	starfish	65,8416	86,5579	64,0859	27
k-médias	5	starfish	66,9425	85,0198	64,3066	10
fep	5	starfish	67,9723	86,3951	64,0859	14
cep	5	tigre	56,4045	82,5048	58,8897	50
k-médias	5	tigre	50,8882	84,3897	40,6102	68
fep	5	tigre	50,4996	83,8371	39,8372	10
cep	5	mulher	48,2314	64,9905	57,1577	50
k-médias	5	mulher	44,4552	68,9544	49,227	29
fep	5	mulher	48,5551	63,3465	51,9615	10

## 4.4 Análise dos resultados

Pela observação das imagens resultantes das classificações, conclui-se que os algoritmos FEP e CEP apresentaram melhores índices em comparação com os obtidos através do k-médias. O CEP apresentou resultados particularmente bons na segmentação de documentos antigos (“ctps” e “queixa-crime”), filtrando satisfatoriamente o texto dos ruídos do papel e da escrita no verso da página.

Em comparação com o FEP, o CEP mostrou-se pouco detalhista nas áreas das imagens que tem muitas bordas, como: no reflexo da água da imagem “tigre”; na área iluminada próxima a boca do peixe, na imagem “peixe”; na área acima dos olhos do cão, na imagem “laika”; na luminosidade presente nos cabelos da imagem “mulher”; na área dos olhos e nas saliências do focinho na imagem “mandril. Para uma classificação mais apurada destas áreas das imagens, o FEP mostrou-se mais eficiente. Se por outro lado o que procura-se é uma classificação que privilegie uma

visualização das formas em detrimento dos detalhes da imagem, o CEP parece mais indicado.

Com relação ao k-médias, tanto o CEP quanto o FEP mostraram-se mais eficientes, tendo em vista os resultados apresentados na tabela 4.1. Observando-se a parte do céu na imagem “esqui” percebe-se que o k-médias não classificou os padrões da forma mais satisfatória. O mesmo ocorre na área central da imagem “cérebro”, onde tanto CEP quanto FEP detalha uma variação de padrões na extremidade inferior direita do padrão central da imagem, assim como na gola da camisa na imagem “mulher”.

Com relação a convergência, percebe-se que o FEP converge, na maioria das vezes, mais cedo que o CEP, obtendo índices muito parecidos. Porém, este último apresenta índices um pouco melhores que aquele, principalmente no que se refere a distância mínima entre os clusters. Tanto CEP quanto FEP precisaram, na maior parte dos casos, de menos interações que o k-médias para convergir.

## **4.5 Conclusão**

Neste capítulo foi apresentado um algoritmo de clustering de imagens que utiliza os conceitos da programação evolucionária. Para examinar-se a eficiência do algoritmo foram classificadas 15 imagens e apresentados os índices alcançados com as respectivas classificações.

Os resultados obtidos apontam que os algoritmos CEP e FEP apresentaram resultados mais apurados que os do k-médias, considerando-se as imagens geradas a partir da classificação e os índices resultantes.

# Capítulo 5

## Considerações Finais e Trabalhos

### Futuros

Após observação dos resultados indicados no capítulo anterior, pode-se concluir que a capacidade exploratória dos mínimos de uma função presentes nos algoritmos FEP e CEP, que são muito bem aplicadas na busca dos mínimos de funções multimodais, otimiza consideravelmente a classificação de imagens, devido ao pequeno (em comparação com problema de minimização de funções) espaço de busca, já que numa imagem, as cores variam apenas de 0 a 255. Disto resulta que a quantidade de gerações necessárias para o algoritmo convergir é pequena, comparando-se com a quantidade de iterações usadas na abordagem com PSO [11]. Um fator que contribuiu para esta rápida convergência foi a aplicação do conceito introduzido por Saha e Bandyopadhyay [12].

Para algumas imagens do conjunto de teste, os resultados obtidos a partir do algoritmo proposto, comparados com os resultados do método tradicional (k-médias), não foram significativamente superiores. Contudo, pode-se considerar que as abordagens CEP e FEP representam uma boa alternativa para a segmentação de imagens, visto que alia bons resultados (comparando-se com os obtidos a partir do k-médias) à eficiência computacional, e a um o baixo número de iterações necessárias para se atingir a convergência. Em quase todas as imagens usadas para os testes, os índices alcançados com a nova abordagem foi melhor: ou apresenta um erro de quantização menor ou apresenta uma maior separação entre os *clusters*.

De modo geral, ficou constatado que para imagens que apresentam áreas com uma sutil variação de cor como na imagem da tomografia cerebral, ou no fundo da imagem “esqui”, ou ainda no papel, nas imagens de documentos antigos, a

segmentação feita com CEP e FEP apresenta resultados qualitativamente superiores que os obtidos através do k-médias.

## 5.1 Trabalhos Futuros

A partir da conclusão deste trabalho, novas possibilidades de trabalhos relacionados futuros tornam-se promissoras. Além das abordagens CEP e FEP da programação evolucionária presentes neste trabalho, outros algoritmos podem ser experimentados na segmentação de imagens, como por exemplo o LEP e o MSEP.

Além das medidas usadas neste TCC (erro de quantização e as distâncias inter e intra *cluster*), outras avaliações podem ser feitas, como o desempenho (velocidade) na execução das classificações.

# Bibliografia

- [1] K. A. De Jong. **Evolutionary Computation : A Unified Approach.** Massachusetts: MIT Press, 2006.
- [2] D. B. Fogel. **Evolutionary Algorithms in Engineering And Computer Science Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications (Hardcover).** Chichester: Wiley, 1999, p. 23.
- [3] A. E. Eiben, M. Schoenauer, Evolutionary Computing, **Information Processing Letters.** 82 (1) (2002) 1–6.
- [4] H. Dong, J. He, H. Huang, W. Hou. Evolutionary Programming Using a Mixed Mutation strategy, **Information Sciences.** 177 (2007) 312–327
- [5] X. Yao, Y. Liu, G. Lin, Evolutionary Programming Made Faster, **IEEE Transactions Evolutionary Comput.** 3 (2) (1999) 82–102.
- [6] M. Iwamatsu, Generalized Evolutionary Programming with Lévy-type Mutation, **Computer Phys. Commun.** 147 (2002) 729–732.
- [7] M. Ji, H. Tang, J. Guo, A Single-Point Mutation Evolutionary Programming, **Information Processing Letters.** 90 (2) (2004) 293–299.
- [8] R. C. Gonzales e E. R. Woods, *Processamento de Imagens Digitais.* São Paulo: Edgard Blucher, 2000.
- [9] S. Das, U. K. Chakraborty, Kernel-Based Clustering of Image Pixels with Modified Differential Evolution, **IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans,** vol. 38, n.1, 2008.
- [10] S. Das, A. Konar, Automatic Image Pixel Clustering with An Improved Differential Evolution, **Applied Soft Computing.** 9 (2009) 226–236.

- [11] M. Omran, A. Engelbrecht, A. Salman, Particle Swarm Optimization Method for Image Clustering, **Int. J. Pattern Recognit. Artificial Intell.** 19 (3) (2005) 297–322.
- [12] S. Saha, S. Bandyopadhyay, Application of a New Symmetry-Based Cluster Validity Index for Satellite Image Segmentation, **IEEE Geoscience and Remote Sensing Letters**, vol. 5, issue 2, pp. 166-170, 2008.