



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E  
DE COMPUTAÇÃO



# **Segmentação de Imagens por Texturas Através de um Método de Agrupamento de Dados**

**Taiane Viana de Carvalho**

Orientador: Allan de Medeiros Martins

Número de Ordem do PPgEEC: M526  
Natal, RN, Junho de 2018

Universidade Federal do Rio Grande do Norte - UFRN  
Sistema de Bibliotecas - SISBI  
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Carvalho, Taiane Viana de.

Segmentação de imagens por texturas através de um método de agrupamento de dados / Taiane Viana de Carvalho. - 2018.

74 f.: il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica e de Computação. Natal, RN, 2018.

Orientador: Prof. Dr. Allan de Medeiros Martins.

1. Processamento de imagens - Dissertação. 2. Agrupamento de dados - Dissertação. 3. Segmentação de imagens - Dissertação. 4. Textura - Dissertação. 5. Coeficiente de variação espacial - Dissertação. 6. Separação da Negentropia Quadrática de Rényi - Dissertação. I. Martins, Allan de Medeiros. II. Título.

RN/UF/BCZM

CDU 004.932



Universidade Federal do Rio Grande do Norte

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE  
COMPUTAÇÃO

ATA Nº 526

Aos 28 dia do mês de junho do ano de dois mil e dezoito, foi realizada a 526ª sessão de defesa de dissertação de mestrado do Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN, na qual o mestrando Taiane Viana de Carvalho apresentou o trabalho que tem como título: Segmentação de imagens por texturas através de um método de agrupamento de dados. A sessão teve início às 09h00min, tendo a banca examinadora sido constituída pelos seguintes participantes: Allan de Medeiros Martins (Dr. UFRN, Presidente), Pablo Javier Alsina (Dr. UFRN, Examinador Interno ao Programa), Agostinho de Medeiros Brito Junior (Dr. UFRN, Examinador Externo ao Programa) e Aluisio Igor Rego Fontes (Dr. IRFN, Examinador Externo à Instituição). Após a apresentação do trabalho e o exame pela banca, o mestrando foi considerado APROVADO, tendo sido lavrada a presente ata, que vai assinada pelos examinadores e pelo mestrando. A versão final da dissertação deverá ser entregue ao programa, no prazo máximo de 60 dias, contendo as modificações sugeridas pela banca examinadora e constante na folha de correção anexa. Conforme o Artigo 49 da Resolução 197/2013 - CONSEPE, o candidato não terá o título se não cumprir as exigências acima.

*Aluisio Igor R. Fontes*

Dr. ALUISIO IGOR REGO FONTES, IRFN

Examinador Externo à Instituição

*Agostinho Brito*  
Dr. AGOSTINHO DE MEDEIROS BRITO JUNIOR, UFRN

Examinador Externo ao Programa

*Pablo Javier Alsina*  
Dr. PABLO JAVIER ALSINA, UFRN

Examinador Interno

*Allan de Medeiros Martins*  
Dr. ALLAN DE MEDEIROS MARTINS, UFRN

Presidente

*Taiane Viana Carvalho*  
TAIANE VIANA DE CARVALHO

Mestrando

---

# Resumo

---

A segmentação é uma das etapas mais complexas em processamento de imagens e o foco em pesquisas nessa área tem sido cada vez maior. Segmentar, significa subdividir uma imagem em suas regiões distintas a partir de propriedades dos píxeis, tais como nível de cinza e textura. Este trabalho apresenta um método de segmentação com base na extração de características de texturas em imagens.

O método consiste em gerar um conjunto de dados gerado a partir da quantificação das texturas presentes na imagem e depois agrupá-los em classes que representarão as texturas que constituem a imagem. A realização do agrupamento dos píxeis ocorre através da aplicação de uma técnica de agrupamento de dados que inicialmente agrupa os pontos do conjunto de dados em classes auxiliares fazendo o uso de uma técnica de quantização vetorial. Essas classes auxiliares serão ligadas formando uma única classe dependendo do grau de compartilhamento de características da textura. Os métodos tradicionais que calculam a similaridade ou dissimilaridade entre grupos possuem algumas limitações quanto ao aumento da distância entre as médias desses grupos. Uma alternativa utilizada neste trabalho é de um novo método que empregando o conceito de mistura de gaussianas calcula a divergência entre um conjunto de dados modelado por gaussianas com o modelo de uma única gaussiana que descreve todo o conjunto. O objetivo dessa abordagem é separar automaticamente regiões com diferentes texturas em uma imagem, utilizado um algoritmo eficiente e de baixo custo computacional.

---

# Abstract

---

Segmentation is one of the most complex process in image processing and the focus on research in this area has been increasing. Segmentation, consists of dividing an image into distinct regions from pixel properties, as gray level and texture. This paper presents a new method of images segmentation based on extraction of textures features in images.

The method consists of forming a dataset based on the pixel values in the image and then group the data into classes that represent the textures that make up the image. Pixel grouping occurs through the clustering application technique. This technique group the points in the data set in auxiliary centers using vector quantization. The distance between the auxiliary centers is estimated and all centers that have a distance less than a threshold are linked together. Conventional distances used in clustering basically grow with the square of the distance in means. An alternative used in this paper is a new method based on gaussian mixture concept, witch estimates the separation between a data set modeled by Gaussians with model of one single Gaussian. The purpose of this approach is to perform the separation of regions with different textures in an image using an efficient algorithm.

**Keywords:** image processing, image segmentation, textures, clustering

---

# Sumário

---

<b>Sumário</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Abreviaturas e siglas</b>	<b>vi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Objetivo . . . . .	3
1.3 Organização do Trabalho . . . . .	4
<b>2 Fundamentação Teórica</b>	<b>5</b>
2.1 Segmentação por Textura . . . . .	5
2.2 Coeficiente de Variação Espacial . . . . .	6
2.3 Quantização Vetorial . . . . .	8
2.3.1 K-Means . . . . .	9
2.4 Medidas de distância . . . . .	10
2.4.1 Distância Euclidiana . . . . .	11
2.4.2 Distância de Mahalanobis . . . . .	12
2.5 Método de Agrupamento de Dados . . . . .	13
2.5.1 Mistura de Gaussianas . . . . .	14
2.5.2 Separação de Gaussianas . . . . .	16
2.5.3 Aplicação do Método QRNS em Agrupamento de Dados . . . . .	22
<b>3 Abordagem Proposta</b>	<b>27</b>
3.1 Desenvolvimento . . . . .	27
<b>4 Resultados</b>	<b>30</b>
4.0.1 Mosaicos de Texturas Naturais . . . . .	30
4.0.2 Texturas em Parede . . . . .	36
4.0.3 Imagens de Satélite . . . . .	40

4.0.4	Imagens Médicas . . . . .	43
4.0.5	Mais exemplos de resultados pelo método proposto . . . . .	44
4.0.6	Testes com Rede Neural Competitiva . . . . .	53
4.0.7	Testes com Filtro de Gabor . . . . .	56
<b>5</b>	<b>Conclusão</b>	<b>60</b>
	<b>Referências</b>	<b>62</b>

---

# Lista de Figuras

---

2.1	Exemplo de classes de distâncias para uma amostra de 7x7 píxeis nos canais R,G e B respectivamente . . . . .	7
2.2	Exemplo de distâncias de Mahalanobis em um meio com estatística conhecida. (MARTINS, 2005) . . . . .	12
2.3	Duas distribuições gaussianas com variância 3 e diferentes valores de média. (MARTINS et al., 2015) . . . . .	16
2.4	Valor da divergência entre as distribuições em função da separação entre suas médias. (MARTINS et al., 2015) . . . . .	17
2.5	Medidas da negentropia para conjuntos de dados com distribuições diferentes. (MARTINS, 2005) . . . . .	18
2.6	Comparação entre um conjunto de dados formado por duas distribuições gaussianas com o modelo gaussiano equivalente. . . . .	20
2.7	QRNS em função da separação das médias para $\Sigma_1=\Sigma_2 = \mathbf{I}$ e $P_1 = P_2 = 0.5$ . (MARTINS, 2005) . . . . .	22
2.8	Etapas do algoritmo de clustering . . . . .	26
3.1	Fluxograma de etapas da abordagem proposta . . . . .	27
3.2	Exemplo janela de 5x5 . . . . .	28
4.1	Mosaico de texturas de pedras . . . . .	31
4.2	Imagem nas suas componentes R, G e B . . . . .	31
4.3	Resultado da segmentação para o mosaico de texturas de pedras . . . . .	32
4.4	Resultado da segmentação do mosaico de texturas de pedras com filtro de média . . . . .	32
4.5	Mosaico de texturas de madeiras . . . . .	33
4.6	Resultado da segmentação para o mosaico de texturas de madeiras . . . . .	33
4.7	Resultado da segmentação do mosaico de texturas de madeiras com filtro de média . . . . .	33
4.8	Mosaico de texturas de vegetações . . . . .	34
4.9	Resultado da segmentação para o mosaico de texturas de vegetações . . . . .	34

4.10	Resultado da segmentação para o mosaico de texturas de vegetações com filtro de média . . . . .	35
4.11	Mosaico de texturas de água . . . . .	35
4.12	Resultado da segmentação . . . . .	36
4.13	Mosaico de texturas de paredes . . . . .	36
4.14	Resultado da segmentação . . . . .	37
4.15	Mosaico de texturas de paredes na cor amarela . . . . .	37
4.16	Resultado da segmentação . . . . .	38
4.17	Mosaico de texturas de paredes na cor verde . . . . .	38
4.18	Resultado da segmentação . . . . .	39
4.19	Mosaico de texturas de paredes na cor azul . . . . .	39
4.20	Resultado da segmentação . . . . .	40
4.21	Imagem de Auckland pelo satélite Landsat . . . . .	40
4.22	Resultado segmentação da Figura 4.21 . . . . .	41
4.23	Imagens de Incheon pelo satélite Landsat . . . . .	41
4.24	Resultado da segmentação das Figuras 4.23(a) e Figura 4.23(b) . . . . .	42
4.25	Contorno do resultado da segmentação das Figuras 4.23(a) e Figura 4.23(b)	42
4.26	Imagens de melanomas . . . . .	43
4.27	Resultados da segmentação das Figuras 4.26(a) e 4.26(b) . . . . .	43
4.28	Contornos do resultado da segmentação das Figuras 4.26(a) e 4.26(b) . . . . .	44
4.29	Mosaico de texturas de papeis . . . . .	44
4.30	Resultado da segmentação do mosaico de texturas de papeis . . . . .	45
4.31	Mosaico de texturas de superfícies rachadas . . . . .	45
4.32	Resultados da segmentação para o mosaico de texturas de superfícies rachadas . . . . .	46
4.33	Resultado da segmentação da Figura 4.33(a) . . . . .	46
4.34	Resultado da segmentação da Figura 4.34(a) . . . . .	47
4.35	Imagem de células . . . . .	47
4.36	Resultado da segmentação da Figura 4.35 . . . . .	48
4.37	Resultado da segmentação da Figura 4.37(a) . . . . .	48
4.38	Resultado do contorno da segmentação com filtro de média da Figura 4.37(a) . . . . .	49
4.39	Resultado da segmentação para a Figura 4.39(a) . . . . .	50
4.40	Resultado da segmentação da Figura 4.40(a) . . . . .	51
4.41	Resultado da segmentação da Figura 4.41(a) . . . . .	51

4.42	Resultado do contorno da segmentação com filtro de média da Figura 4.41(a)	52
4.43	Imagem de texturas naturais e artificiais	52
4.44	Resultado da segmentação da Figura 4.0.5	53
4.45	Resultados da segmentação para a Figura 4.45(a)	54
4.46	Imagem Original	54
4.47	Resultados da segmentação da Figura 4.46	55
4.48	Resultados	55
4.49	Resultado da segmentação da Figura 4.49(a)	56
4.50	Resultados da segmentação da Figura 4.50(a)	57
4.51	Resultados da segmentação da Figura 4.51(a)	58

---

# Lista de Abreviaturas e siglas

---

EM: *Expectation Maximization*

FDP: Função Densidade de Probabilidade

QRNS: Separação da Negentropia Quadrática de Rényi

VA: Variável Aleatória

---

# Capítulo 1

## Introdução

---

### 1.1 Motivação

Devido ao grande avanço tecnológico dos últimos tempos, verifica-se a utilização cada vez maior de sistemas computacionais inteligentes, no meio industrial e científico, nas tarefas de extração de informações de imagens para auxiliar na eficiência da solução de diversos problemas práticos.

Para um ser humano, basta olhar para uma árvore para perceber que seu tronco é áspero ou olhar um quadro branco para notar que sua textura é extremamente suave, porém, aproximar esse sistema visual humano da visão computacional não é uma tarefa fácil. Várias dificuldades estão presentes ao tentar traduzir para o computador o processo realizado pela visão humana na identificação de diferentes regiões em uma imagem, tais como a complexidade da textura, a não-homogeneidade da iluminação (sombras próximas à borda), as irregulares e as imprecisões das regiões das bordas, etc (NUNES; CONCI, 2007). Por conseguinte, existe atualmente os mais variados tipos de métodos para auxiliar na solução de problemas computacionais que necessitam da extração de informações em uma imagem.

A segmentação é uma das principais etapas para a análise automatizada de imagens (BALAN, 2003). Segmentar, significa subdividir uma imagem em regiões ou objetos que a compõe (GONZALEZ; WOODS, 2000) e a partir daí utilizar estes dados para um processamento subsequente, tal como classificação, descrição, contagem, reconhecimento, etc.

Diversas abordagens tem sido propostas para segmentação, tais como métodos de detecção de bordas de objetos, agrupamento de regiões com características semelhantes, abordagens semânticas e a utilização da dependência espacial entre os elementos (píxeis) da imagem (SCHWARTZ; PEDRINI, 2005).

Uma vez que a segmentação visa particionar uma imagem em regiões com proprie-

dades comuns, normalmente as técnicas de segmentação estão associadas ao conceito de região. Uma região em uma imagem pode ser vista como um conjunto de píxeis conectados envolvidos por uma borda de píxeis criando um contorno, ou ainda, uma região pode ser definida como um conjunto de píxeis conectados por meio de uma condição de uniformidade. Na primeira definição, a segmentação é realizada com base na descontinuidade, enquanto na segunda definição, com base na similaridade (GONZALEZ; WOODS, 2000).

Como resultado da segmentação, cada um dos píxeis em uma mesma região é similar com referência a alguma característica ou propriedade computacional, tais como cor, intensidade e textura. Regiões adjacentes devem possuir diferenças significativas com respeito a(s) mesma(s) característica(s) (JUCA; MELLO, 2011).

A textura é considerada um descritor regional importante e apesar de ser um termo intuitivo muito empregado, não existe uma definição única e precisa. Para Sklansky (1978), “uma região em uma imagem apresenta uma textura única se um conjunto de estatísticas ou outra propriedade local da imagem for constante, de variação suave ou aproximadamente periódica”. Para Sonka (1993), textura pode ser definida como algo que consiste de elementos mutuamente relacionados (a primitiva de textura, que pode ser um píxel ou um conjunto de píxeis). Já McGrogan (1997) define textura como uma estrutura composta de um grande número de elementos similares mais ou menos ordenados. Nunes; Conci (2007) relaciona textura à impressão de rugosidade e contraste ou a repetição de padrões visuais sobre uma região.

O desenvolvimento de métodos de segmentação que utilizam texturas de uma imagem são motivadas, principalmente, pela alta eficiência do sistema visual humano em identificá-las e diferenciá-las. Por isso, o processamento de imagens digitais visa desenvolver ferramentas que possibilitem às máquinas não apenas automatizar procedimentos realizados pelas funções visuais dos seres humanos, mas também superar seu desempenho, pois são capazes de processar grande quantidade de informação com muito mais rapidez (DOBLER, 2015).

Devido a essas características, vários métodos com diferentes abordagens foram desenvolvidas para utilizar a textura como base na segmentação e descrição de uma imagem (COLLET; MURTAGH, 2004; JUCA; MELLO, 2011; MA et al., 2013; RANDEN; HUSOY, 1999; SCHWARTZ; PEDRINI, 2005). Tais métodos têm sido amplamente utilizados no meio científico e industrial, possuindo aplicações em análise de imagens médicas, sensoriamento remoto, controle de qualidade, dentre outras. Uma das possíveis aplicações em análise de imagens médicas, por exemplo, é a detecção de tumores em mamografias (MASCARO et al., 2009), melhorando a resposta do diagnóstico auxiliado

por computador. Já no controle de qualidade, a segmentação pode ser usada pra detectar automaticamente defeitos de fabricação e arranhões em peças de couro (LOVERGINE et al., 1997).

Todos os métodos desenvolvidos visam um resultado preciso da segmentação, pois é esse resultado que determina o sucesso ou o fracasso final dos procedimentos posteriores da análise computadorizada. Assim, um método que permita segmentar regiões baseadas nas suas diferentes texturas considerando suas complexidades, de forma precisa e eficiente é de grande interesse para as diversas áreas de aplicação e vem a ser a proposta deste trabalho.

## 1.2 Objetivo

De forma geral, este trabalho tem como objetivo aplicar um método de agrupamento de dados para realizar a segmentação automática de imagens baseado em suas texturas. Diante das dificuldades apresentadas anteriormente, esta abordagem procura a utilização de métodos eficientes na quantificação das texturas e separação entre as regiões, que possuam um algoritmo de simples compreensão e baixo custo computacional.

Para quantificar as texturas presentes em uma imagem, o método denominado Coeficiente de Variação Espacial (CVE) considera a posição espacial dos píxeis na região de textura e as faixas de cor através de medidas estatísticas de média e desvio padrão. Esse método é utilizado por não necessitar de uma área da imagem excessivamente grande para análise da textura, não emprega um grande número de características para representá-la e, assim, não necessita de um esforço computacional considerável. Ele foi proposto por Nunes e Conci (2003, 2006, 2007) e é aplicado neste trabalho.

Já o método de agrupamento de dados empregado consiste de um método não - supervisionado e é composto por um conjunto de técnicas distintas com a finalidade de realizar a tarefa de agrupamento. Basicamente o método tem como base a divisão dos dados em um número excessivo de pequenas classes auxiliares para posterior agrupamento dos mesmos em clusters maiores que representam as regiões de texturas presentes em uma imagem.

Este método foi proposto por Martins (2005, 2015) e sua principal contribuição está na técnica de separação que calcula a divergência entre um conjunto de dados para realizar a ligação ou não entre esse conjunto. A técnica de separação faz uso do conceito de mistura de gaussianas e negentropia e calcula a divergência entre um conjunto de dados modelado por gaussianas com o modelo de uma única gaussiana que descreve todo o conjunto. Se o valor da divergência for menor que um limiar estabelecido então o conjunto possui

similaridade e será ligado. Esse método ainda apresentada como vantagem independência em relação a escala dos dados.

No decorrer do trabalho serão apresentadas todas as métricas, comparações necessárias e resultados para o completo entendimento do método de segmentação.

### **1.3 Organização do Trabalho**

Este trabalho encontra-se estruturado da seguinte forma: no capítulo 2 é apresentada a fundamentação teórica necessária para o completo entendimento da abordagem proposta; o capítulo 3 apresenta a abordagem proposta, assim como cada uma das etapas presentes no processo; o capítulo 4 reúne os testes e consequentes resultados obtidos; e por fim, no capítulo 5 são sumarizados as conclusões obtidas e trabalhos futuros.

---

# Capítulo 2

## Fundamentação Teórica

---

Nesse capítulo encontra-se o detalhamento da teoria necessária para possibilitar a realização deste trabalho. A seguir, são apresentados todos conceitos necessários para a realização do mesmo.

### 2.1 Segmentação por Textura

No campo do processamento digital de imagens uma imagem pode ser definida como uma função bidimensional,  $f(x,y)$ , em que  $x$  e  $y$  são coordenadas espaciais (plano), e a amplitude de  $f$  em qualquer par de coordenadas  $(x,y)$  é chamada de intensidade, sendo os valores de  $x$ ,  $y$  e de intensidade de  $f$  quantidades finitas e discretas. Uma imagem digital é composta de um número de elementos, cada um com uma localização e valores específicos (GONZALEZ; WOODS, 2000). Esses elementos são chamados de píxeis.

O interesse em métodos de processamento digital de imagens provém de duas áreas principais de aplicação: melhora das informações visuais para a interpretação humana e processamento de dados de imagens para armazenamento, transmissão e representação, considerando a percepção automática por máquinas.

Nesses métodos uma das etapas fundamentais é a segmentação. O procedimento de segmentação divide uma imagem em suas partes ou objetos constituintes, e a eficiência obtida na identificação e separação das regiões de interesse influencia os resultados das etapas subsequentes (GONZALEZ; WOODS, 2000). Como a segmentação não é um fim em si, sua qualidade é determinada pela adequação das regiões extraídas aos tratamentos que serão realizados nas próximas etapas (NUNES, 2006).

Na segmentação, o nível de subdivisão da imagem depende do objetivo que se pretende alcançar e somente se encerra quando as partes constituintes de interesse forem isoladas, o que requer o conhecimento prévio da natureza da imagem. Assim, a segmentação é um problema orientado à aplicação, pois depende do contexto (BENDER, 2003).

Atualmente, existe uma grande variedade de técnicas de segmentação de imagens e uma delas é a segmentação por textura. Texturas se caracterizam pela repetição de um modelo sobre uma região. Este modelo pode ser repetido de forma exata ou com pequenas variações sobre um mesmo tema. Tamanho, formato, cor e orientação dos elementos do modelo podem variar sobre as regiões. A variação encontrada na forma como os modelos se relacionam é suficiente para diferenciar duas texturas (NUNES; CONCI, 2003).

Reconhecer sutilezas entre diferentes texturas não é um processo simples de se reproduzir utilizando algoritmos automáticos, devido aos mais diversos fatores como: iluminação, orientação, etc. Para que esse processo seja executado de forma devida é necessário o uso de algoritmos capazes de analisar na imagem propriedades dos píxeis referentes a textura e a partir daí realizar a separação das diferentes regiões.

Existe três principais abordagens utilizadas no processamento de imagens para descrever a textura de uma região que são a abordagem estatística, a estrutural e a espectral. A próxima seção descreve um método de abordagem estatística que quantifica textura.

## 2.2 Coeficiente de Variação Espacial

O Coeficiente de Variação Espacial (CVE) caracteriza a textura de uma imagem baseado nas medidas de posição e suas variações ou dispersões. O CVE quantifica a característica de textura através de medidas estatísticas que descrevem as variações espaciais de intensidade ou cor. Duas medidas são empregadas: uma medida de posição (média) e uma medida de dispersão (desvio padrão) dos píxeis pertencentes à região da imagem (NUNES; CONCI, 2003; NUNES, 2006).

O cálculo do CVE leva em consideração não só apenas a distribuição de intensidades, mas também a distribuição espacial pelas classes de distâncias, pois a textura de uma região não se caracteriza só pela tonalidade dos píxeis mas também pelos relacionamentos espaciais entre eles.

O cálculo do CVE inicia com a definição de um conjunto de amostras de treinamento para cada região de interesse. Essas amostras são extraídas da imagem, através de janelas de tamanho  $M \times M$  definidas pelo usuário. O modelo de cores que o método utiliza é o RGB.

Para cada amostra de tamanho  $M \times M$ , determina-se o conjunto de intensidades dos píxeis por classe de distância e por canal RGB considerado. A métrica de distância  $D_4$  (também conhecida como métrica Manhattan, cityblock ou quarteirão) é utilizada para calcular a distância de cada píxel ao píxel central da amostra. Para exemplificar, a Figura 2.1 representa com as mesmas cores os píxeis com mesma distância  $D_4(p, q)$  de cada píxel

$q(s,t)$  em relação ao píxel central  $p(x,y)$ , calculadas por meio da equação 2.1, sendo  $x$  e  $y$  as coordenadas do píxel central e  $s$  e  $t$ , as coordenadas do píxel considerado.

$$D_4(p,q) = |x - s| + |y - t| \quad (2.1)$$

Figura 2.1: Exemplo de classes de distâncias para uma amostra de 7x7 píxeis nos canais R,G e B respectivamente



Na Figura 2.1 é possível notar seis grupos correspondendo às seis diferentes distâncias possíveis em relação ao píxel central, cada canal RGB, cada um representado por uma cor. Esses grupos formarão as classes que agrupam os píxeis que possuem a mesma distância  $D_4$  em relação ao píxel central, sendo que a classe seguinte engloba os valores das classes anteriores ( $Classe_n =$  conjunto de píxeis com  $D_4 \leq n$ ).

O próximo passo consiste em obter a média e o desvio padrão para cada classe de distâncias. O cálculo do desvio padrão é dado pela equação 2.2, onde  $x_i$  são as medidas e  $n$  o número de medidas. Em seguida, as medidas de posição e de dispersão são combinadas através de 2.3, cujo valor único (CVE, da classe) preserva a informação de ambas as medidas obtidas.

$$dp = \sqrt{\frac{\sum_{i=1}^n x_i^2}{n} - \left(\frac{\sum_{i=1}^n x_i}{n}\right)^2} \quad (2.2)$$

$$CVE = \arctg\left(\frac{md}{dp+1}\right) \sqrt{md^2 + (dp+1)^2} \quad (2.3)$$

Onde  $md$  é a média das intensidades dos valores e  $dp$  seu desvio padrão. Por fim, o valor médio e o desvio padrão dos CVEs obtidos de cada classe são combinados e, utilizando novamente a equação 2.3, é obtido o resultado do CVE da amostra de textura da região de interesse. A Tabela 2.1 apresenta a média, o desvio padrão e o CVE para os dados da Figura 2.2 no canal Red.

Para cada amostra de textura do conjunto de treinamento é calculado o CVE para os

Tabela 2.1: Dados para o cálculo do CVE no canal R

Classe Distância	md	dp + 1	CVE	md CVE	dp+1 CVE	CVE Textura
1	102,500	12,522	149,651	142,754	4,791	219,572
2	100,500	15,880	143,878			
3	101,375	25,327	138,552			
4	104,167	31,938	138,729			
5	107,273	35,226	141,532			
6	110,313	38,712	144,182			

canais RGB. O valor do CVE em cada canal define uma coordenada do espaço euclidiano para amostra considerada. Assim, uma amostra com os canais RGB possuirá uma coordenada ( $CVE_{red}$ ,  $CVE_{green}$ ,  $CVE_{blue}$ ), caracterizando um ponto no espaço tridimensional que quantifica o valor da textura naquela região.

Ainda que o usuário tenha selecionado amostras representativas das classes, é possível que ocorra uma zona de amostras com características de CVEs comuns, devido a complexidade dos padrões das texturas. Assim, é necessário definir um procedimento para realizar a separação das amostras em grupos distintos, baseado em algum critério de similaridade. Nunes, (2006) implementou dois procedimentos para realizar a separação das amostras em grupos distintos: algoritmos genéticos e clusterização com algoritmo K-Means. Este trabalho utiliza um novo método para realizar essa separação.

### 2.3 Quantização Vetorial

A quantização vetorial é um método usualmente utilizado na compressão de dados e também encontra aplicações no campo de reconhecimento de padrões em sinais, classificação e extração de dados. Esse método tem como objetivo representar certa distribuição de dados utilizando um número de protótipos significativamente menor que o número de dados (VASCONCELOS, 2004).

O papel desses protótipos é aproximar estatisticamente o conjunto de dados original. Imagine que haja um conjunto de dados de entrada formado por  $N$  vetores de dimensão  $d$ . A quantização vetorial consiste em mapear esses dados em outro conjunto de  $M < N$  protótipos, também de dimensão  $d$  (VIOLATO; ALL, 2009).

Ao definir conjuntos de dados similares, as técnicas de agrupamento de dados podem ser utilizadas para quantização vetorial, de forma que todos os dados de um grupo passam a ser representados por um protótipo, por exemplo, o centróide do grupo (VIOLATO; ALL, 2009). Neste caso, escolhe-se  $N_a$  representantes para quantizar todo o conjunto  $M$ .

Essa escolha pode ser feita de diversas formas, as técnicas mais conhecidas são o K-means e as redes neurais competitivas.

Obtidos os  $N_a$  centros, realiza-se a classificação de cada vetor do conjunto de dado, ou seja, atribui-se cada vetor do conjunto de dados a uma única classe dentre as  $N_a$  disponíveis. Esta atribuição pode ser feita com base em diversas regras. Uma das mais utilizadas é o classificador de distância mínima utilizando o método da distância euclidiana (BORGES; SILVA; CASTRO, 2007). O objetivo dessas técnicas é que dados do mesmo grupo apresentem alta similaridade entre si e possuam pouca similaridade com dados de outros grupos.

O classificador compara a distância euclidiana de um vetor a cada um dos  $N_a$  centros. Um vetor será atribuído ao agrupamento que apresenta a menor distância dentre todos os  $N_a$  grupos. O resultado final do processo de classificação constitui-se de todos os vetores classificados como pertencentes a algum dos  $N_a$  grupos.

### 2.3.1 K-Means

O K-Means é uma das heurísticas mais populares usadas para agrupamento ou quantização de dados. Este algoritmo tenta fornecer uma classificação de acordo com os próprios dados, sendo a classificação feita por similaridade de grupos, onde o dado é atribuído ao grupo (cluster) ao qual é mais semelhante.

É um método de aprendizagem não-supervisionado. A ideia por trás deste algoritmo é particionar os dados em  $N_a$  clusters, onde  $N_a$  é definido previamente. Cada cluster é representado por um dado, chamado de centro do cluster (PEREIRA; ALL, 2014).

Inicialmente, os  $N_a$  centros são gerados e os dados são classificados de forma a minimizar a distância quadrática média de cada padrão ao centro mais próximo. Em seguida, calcula-se os novos centros (fazendo a média dos pontos classificados) dos atuais padrões calculados. Este processo é repetido até que não se tenha variação na classificação dos dados ou uma quantidade suficiente de iterações tenha sido alcançada.

O algoritmo do K-Means pode ser melhor descrito na forma de pseudocódigo como demonstrado no Algoritmo 1.

**Algoritmo 1:** K-means

---

**Entrada:**  $X = \{x_1, \dots, x_N\} \in \mathfrak{R}^D$  (NxD dados de entrada)

**Saída:**  $C = \{c_1, \dots, c_{N_a}\} \in \mathfrak{R}^D$  ( $N_a$  centros)

**início**

    Gere aleatoriamente  $N_a$  centros;

**repita**

        (re) Atribua cada elemento ao cluster ao qual o elemento é mais similar de acordo com a métrica adotada;

        Atualize o valor dos centros

**até** que não haja mudanças de elementos de um cluster para outro;

**fim**

---

A escolha da posição inicial dos clusters pode ser realizada de várias formas, entre elas podemos destacar a escolha randômica, onde os centros iniciais são escolhidos aleatoriamente (PEREIRA; ALL, 2014).

Em suma, o algoritmo K-Means é extremamente veloz, geralmente convergindo em poucas iterações para uma configuração estável, na qual nenhum elemento está designado para um cluster cujo centro não lhe seja o mais próximo.

## 2.4 Medidas de distância

Muitos algoritmos agem baseados em cálculos e comparações de algum tipo de distância entre instâncias/atributos dos dados. Dependendo da medida de distância utilizada os algoritmos podem apresentar mudanças significativas em seus resultados como, por exemplo, quando essas medidas são utilizadas para identificação de outliers, formato dos clusters na camada de apresentação, e até mesmo na formação de vizinhanças entre os grupos de pontos de dados (CLÉSIO, 2012).

As medidas de distância, de uma maneira geral, podem ser definidas como medidas de similaridade e ou dissimilaridade; na qual a primeira é para definir o grau de semelhança entre as instâncias e realizam o agrupamento de acordo com a sua coesão, e a segunda de acordo com as diferenças dos atributos das instâncias (DEZA MICHEL MARIE; DEZA, 2009). A medida mais comum é a distância Euclidiana entre dois padrões (vetores).

Essas medidas possuem algumas propriedades definidas. Essas propriedades nos informam que a distância entre dois pontos é sempre um valor positivo e a distância entre o ponto  $x$  até o ponto  $y$  é igual a distância de  $y$  até o ponto  $x$ .

Em geral, há sempre uma estatística envolvida na distribuição dos padrões em um conjunto de dados como um todo e cada classe possui sua distribuição (MARTINS, 2005). A utilização de uma métrica de distância que incorpore esta estatística apresenta uma grande vantagem em relação a medida de distância Euclidiana comum.

Em um classificador, as medidas de distância são fundamentais e afetam diretamente seja no seu custo computacional, complexidade e representação gráfica para análise. Um outro ponto importante destacado por Martins (2005), quando se trata da aplicação de medidas de divergência em métodos de agrupamento de dados, é em relação a invariância à escala dos dados. Esses métodos não devem ser sensíveis a transformações lineares que venham a ocorrer nos dados como, por exemplo, se houver uma mudança de unidade o algoritmo não deve apresentar modificações no seu resultado. Por causa disso, é importante que as medidas de dissimilaridade utilizadas em algoritmos de agrupamento sejam invariantes pelo menos a transformações lineares, garantindo assim, que mudanças de unidade, ou escala, não interfiram no resultado final do algoritmo.

### 2.4.1 Distância Euclidiana

No plano  $\mathbb{R}^2$ , a distância euclidiana entre dois pontos  $\mathbf{x}$  e  $\mathbf{y}$  pode ser calculada pelo teorema de Pitágoras que fornece o tamanho do segmento de reta que une os dois pontos (BORTOLOSSI, 2003), como pode ser visto na equação 2.4.

$$d(x, y) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2} \quad (2.4)$$

Generalizando para dois pontos com  $n$  amostras no espaço euclidiano  $\mathbb{R}^n$  o cálculo da distância euclidiana é dado por

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.5)$$

A distância euclidiana é uma métrica de dissimilaridade (ou similaridade) que não leva em conta a estatística dos dados onde a medida será realizada. Além disso essa distância é variante a transformações lineares. A sua utilização torna os métodos onde ela foi aplicada dependente da escala. Por exemplo, se os dados fossem medidas de comprimento e estivessem em metros, a distância limiar estaria em metros. A simples mudança para centímetros, faria com que a distância limiar aumentasse em 100 vezes, o que torna o problema como crítico.

### 2.4.2 Distância de Mahalanobis

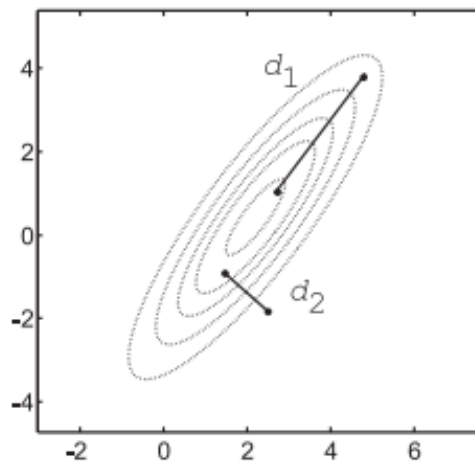
A distância de Mahalanobis entre dois pontos  $x$  e  $y$  é definida como mostra a equação 2.6.

$$d_m(x,y) = \sqrt{(x-y)'\Sigma^{-1}(x-y)} \quad (2.6)$$

Onde  $\Sigma$  é a matriz de covariância associada à estatística onde os dois vetores estão imersos. Pode-se observar que a distância de Mahalanobis é uma generalização da distância Euclidiana onde as direções são ponderadas. Fazer uso dessa distância corrige algumas das limitações da distância Euclidiana (LINDEN, 2009).

Esta medida fornece valores baixos quando os pontos estão alinhados com a direção da variância dos dados. A Figura 2.2, por exemplo, representa os contornos de uma gaussiana onde a distância de Mahalanobis  $d_2$  é maior do que a distância  $d_1$ , embora com as mesmas distâncias medidas utilizando a métrica Euclidiana ocorra o contrário.

Figura 2.2: Exemplo de distâncias de Mahalanobis em um meio com estatística conhecida. (MARTINS, 2005)



Pode-se usar também a métrica de Mahalanobis para medir a distância entre um elemento  $x$  e um cluster de elementos que possua média  $\mu$  e uma matriz de covariância dada por  $\Sigma$ . Neste caso, a distância é definida pela equação 2.7.

$$d_m(x,\mu) = \sqrt{(x-\mu)'\Sigma^{-1}(x-\mu)} \quad (2.7)$$

Conceitualmente, é como se estivéssemos avaliando a pertinência de um elemento não só por sua distância ao centro do cluster, mas também pela distribuição do mesmo,

determinando assim a distância do elemento  $x$  em termos de uma comparação com o desvio padrão do cluster. Quanto maior for o valor desta métrica, maior o número de desvios padrões que um elemento está distante do centro, e menor sua chance de ser um elemento do mesmo.

Um dos usos mais comuns da distância de Mahalanobis é para a detecção de outliers, pois um valor alto determina que um elemento está a vários desvios padrões do centro e, por conseguinte, é provavelmente um outlier. Já se fizermos a comparação com vários grupos existentes, cujas distribuições sejam conhecidas, esta métrica pode ser usada como maneira de determinar a qual grupo um elemento mais provavelmente deve pertencer.

É importante observar que a distância de Mahalanobis, quando utilizada para medir a distância entre dois conjuntos de dados, é invariante a transformações lineares. Quando ocorre uma transformação linear que aumente ou diminua a escala, os valores dos pesos das diferenças é compensado no inverso da matriz de covariância analisada.

## 2.5 Método de Agrupamento de Dados

O agrupamento de dados é uma técnica de análise não-supervisionada, que oferece uma maneira de entender e extrair informações relevantes de grandes conjuntos de dados. Essa técnica tem como objetivo separar os dados em grupos (clusters) de acordo com certo critério. A ideia geral é que dados em um mesmo grupo tenham mais características em comum entre si do que dados de outros grupos (OLIVEIRA, 2008).

A principal diferença entre os métodos supervisionados e não supervisionados é que no primeiro, os dados devem ser atribuídos a grupos já conhecidos previamente. Enquanto que nos métodos não-supervisionados, deve-se descobrir esses grupos extraíndo características escondidas dos dados e desenvolvendo as hipóteses a respeito de sua natureza (LINDEN, 2009).

Na literatura podem ser encontrados muitos trabalhos que abordam a utilização de métodos de agrupamento de dados para segmentar imagens de acordo com suas diferentes texturas (BASTOS; LIATSI; CONCI, 2008; CHAN; VASCONCELOS, 2008; JAIN; FARROKHANIA, 1990; NUNES; CONCI, 2007; OLIVEIRA, 2004; YAN; SHEN; LI, 2009). Este trabalho também faz uso de um método de agrupamento para extrair as características das diferentes texturas presentes em uma imagem. A seguir, será apresentado os conceitos necessários para o entendimento do método utilizado e sua aplicação.

### 2.5.1 Mistura de Gaussianas

Uma maneira de modelar distribuições complexas de dados consiste na utilização da misturas de gaussianas. A mistura gaussiana é uma forma de modelagem que consiste no agrupamento de densidades de probabilidade gaussianas com características individuais (OLIVEIRA; ALL, 2011). Na literatura, existem muitos trabalhos que utilizam misturas de gaussianas ((GOODMAN; MCMICHAEL, 1996), (FILHO; JR; MARCOLINO, 2013)). Em alguns trabalhos essa ferramenta é aplicada em problemas de agrupamento ((TIPPING, 1999), (PERMUTER; FRANCO; JERMYN, 2003)).

Uma mistura de gaussianas é definida como uma distribuição cuja função densidade de probabilidade é dada por uma soma ponderada de gaussianas, como mostrado na equação 2.8

$$p(x) = \sum_{i=1}^{N_g} P_i N(x, \mu_i, \Sigma_i) \quad (2.8)$$

Onde  $p(x)$  é a função densidade de probabilidade que se deseja modelar,  $N_g$  é o número de gaussianas presentes na mistura e  $P_i$  são as probabilidades a priori de cada gaussiana que satisfazem a seguinte condição

$$\sum_{i=1}^{N_g} P_i = 1 \quad (2.9)$$

Uma distribuição normal com vetor média  $\mu_i$  e matriz de covariância  $\Sigma_i$ , representada por  $N(x, \mu_i, \Sigma_i)$ , é dada pela equação 2.10

$$N(x, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu)\Sigma^{-1}(x - \mu)^t\right) \quad (2.10)$$

Onde  $D$  é a dimensão do vetor de dados  $x$ , neste caso,  $x \in \mathbb{R}^D$ .  $\mu$  e  $\Sigma$  são a média e a matriz de covariância respectivamente da distribuição.

Para obtermos o modelo da mistura de gaussianas, um requisito necessário é o número de gaussianas que farão parte da mistura. No caso de modelar várias classes, também é importante o número de gaussianas em cada classe. Além desta quantidade, para modelar todas as classes presentes nos dados, precisa-se dos parâmetros de cada gaussiana como média, probabilidade a priori e matriz de covariância.

Talvez a maneira mais tradicional de estimar esses parâmetros de uma mistura de gaussianas seja utilizando o algoritmo *Expectation Maximization* ou *EM*, como é conhecido na literatura.

### Expectation Maximization

O algoritmo *EM* consiste em um algoritmo para estimar parâmetros de uma mistura de gaussianas que melhor modele um conjunto de dados (DEMPSTER; LAIRD; RUBIN, 1977). Basicamente esse algoritmo encontra um vetor de parâmetros  $\Theta$  de uma função densidade de probabilidade. Em resumo, é necessário encontrar  $P(x|\Theta)$ , que é a probabilidade de ocorrer um vetor  $x$  dados os parâmetros  $\Theta$ . No caso gaussiano,  $\Theta$  é formado por médias, matrizes de covariância e probabilidades a priori. O algoritmo EM encontra os parâmetros da distribuição maximizando a verossimilhança dos dados expressa pela equação 2.11.

$$L(\Theta) = \prod_{i=1}^N P(x_i|\Theta) \quad (2.11)$$

Não se maximiza diretamente  $L(\Theta)$  e sim o seu logaritmo. Por ser uma função estritamente crescente, o logaritmo da verossimilhança terá um máximo no mesmo ponto que teria a verossimilhança em si, e torna o problema de maximização mais simples de se trabalhar, pois transforma o produtório em um somatório formando a expressão dada a seguir

$$\log(L(\Theta)) = \sum_{i=1}^N \log(P(x_i|\Theta)) \quad (2.12)$$

A solução analítica para o problema de maximização da verossimilhança consiste em encontrar o máximo de  $\log(L(\Theta))$  (derivando e igualando a zero) solucionando o sistema de equações dado por

$$\frac{\partial L(\Theta)}{\partial \Theta_i} = 0 \quad (2.13)$$

O algoritmo *EM* é um algoritmo iterativo que parte de uma solução inicial para o vetor de parâmetros e melhora esta solução a cada iteração. A atualização dos parâmetros é feita calculando-se os valores que resolvem o sistema da equação 2.13.

Para misturas de gaussianas, obter uma expressão isolada para os parâmetros nestas equações é muito difícil. A solução é realizar uma aproximação e calcular uma função mais simples que aproxime o valor das verossimilhança, dados os vetores de dados atuais. Este passo é denominado na literatura de Expectation. O próximo passo é encontrar quais os parâmetros que maximizam a função encontrada na fase de expectation. Este passo recebe o nome de maximization. Mesmo com essas simplificações, o algoritmo *EM* possui um alto custo computacional e nem sempre encontra um máximo global para

verossimilhança (MARTINS, 2005).

## 2.5.2 Separação de Gaussianas

Tradicionalmente, diferentes medidas de distância tem sido aplicadas para medir a similaridade entre gaussianas. Dado dois conjuntos de dados modelados por gaussianas com médias  $\mu_1$  e  $\mu_2$  e matrizes de covariância dada por  $\Sigma_1$  e  $\Sigma_2$ , a similaridade entre elas poderia ser calculada usando a distância de Mahalanobis que é definida por

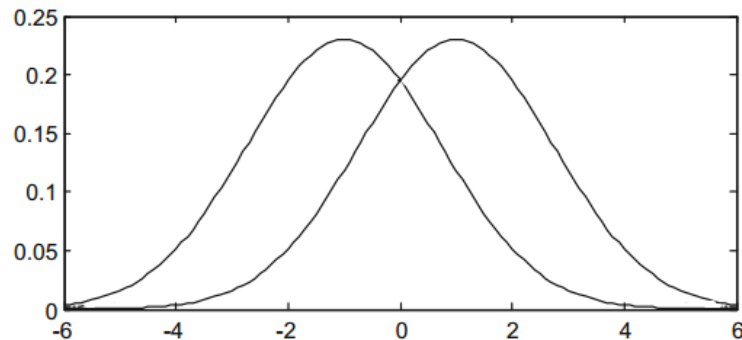
$$d_m = (\mu_1 - \mu_2)^t (\Sigma_1 + \Sigma_2)^{-1} (\mu_1 - \mu_2) \quad (2.14)$$

Ou ainda, poderia-se usar a divergência de Kullback-Leibler definida pela equação 2.15

$$d_{kl} = \frac{1}{2} (\mu_1 - \mu_2)^t (\Sigma_1^{-1} + \Sigma_2^{-1}) (\mu_1 - \mu_2) + \frac{1}{2} \text{trace}(\Sigma_1^{-1} \Sigma_2 + \Sigma_2^{-1} \Sigma_1 - 2\mathbf{I}) \quad (2.15)$$

Todas essas medidas, com pequenas diferenças, exibem um crescimento quadrático com a distância entre as médias, que são maximizadas na direção expressa em função das matrizes de covariância. Para exemplificar, considere duas gaussianas com variância igual a 3 e com médias -1 e 1 respectivamente, como pode ser visto na Figura 2.3.

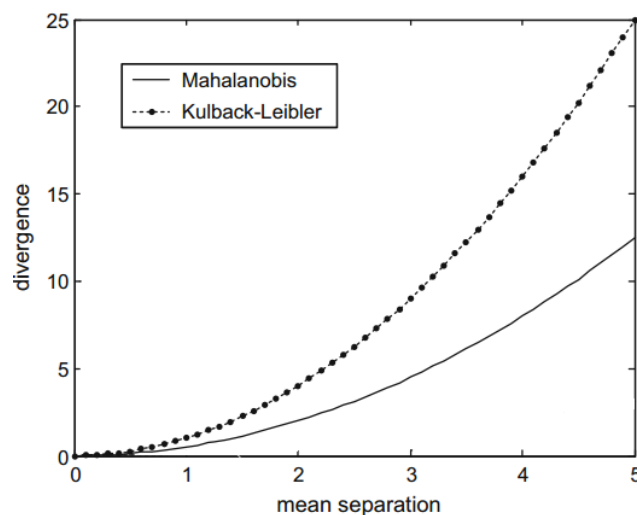
Figura 2.3: Duas distribuições gaussianas com variância 3 e diferentes valores de média. (MARTINS et al., 2015)



A Figura 2.4 mostra a estimativa do valor de dissimilaridade entre as duas gaussianas, pelas medidas de distância acima citadas, em função da diferença entre as médias. É visível o aumento quadrático da dissimilaridade com o aumento da distância entre as médias das gaussianas. Essas medidas, portanto, são insensíveis ao fato de que, quando

a diferença entre as médias é pequena (em comparação com a variância) os dados ainda possam compartilhar informações entre si.

Figura 2.4: Valor da divergência entre as distribuições em função da separação entre suas médias. (MARTINS et al., 2015)



Pensando em uma método alternativo Martins (2005, 2015) propõe uma nova medida de separação entre duas gaussianas, abordando o conceito de teoria da informação, onde substitui a entropia por negentropia e estima a separação entre gaussianas de uma maneira compatível com a análise visual. A ideia é comparar pares de grupos e decidir ligá-los ou não, com base nessa medida.

A medida proposta não é uma medida de distância e sim de separação entre gaussianas. A diferença entre distância e separação está no sentido de que, em geral, as distâncias são calculadas usando principalmente o centro das gaussianas. Duas Gaussianas podem ser distante, mas não separadas e, com a finalidade de assumir que essas duas gaussianas fazem parte de um único e maior grupo, é preciso distinguir entre esses dois termos. Uma medida de separação deve crescer mais lentamente do que o comportamento quadrático da dissimilaridade, visto nas medidas de distância anterior, até que os gaussianas sejam visualmente bem separadas.

A seguir, será abordado o conceito de negentropia para um completo entendimento do funcionamento do método de separação apresentado em seguida.

### Negentropia

Entropia é um conceito fundamental da Teoria da Informação que avalia o grau de informação que pode ser obtido através da observação de uma variável. Neste sen-

tido, quanto mais imprevisível for a variável observada, maior será sua Entropia (MORIMITSU; TUESTA, 2015).

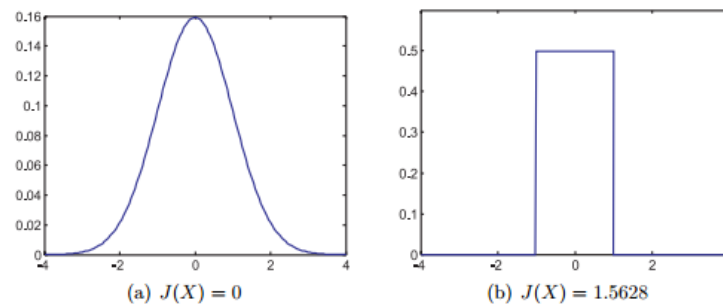
Uma variável gaussiana possui a maior Entropia entre variáveis de mesma variância. Neste sentido, a Entropia pode ser, então, utilizada como uma medida de não-gaussianidade ou como é conhecida na literatura como cálculo da negentropia entre duas distribuições. A negentropia é definida da seguinte forma:

$$J(X) = H(X_g) - H(X) \quad (2.16)$$

Essa medida calcula a diferença de entropia ( $H(X)$ ) existente entre uma variável observada e uma variável gaussiana com os mesmos parâmetros ( $H(X_g)$ ) (mesma média e matriz de covariância da variável aleatória (VA)  $X$ ). A negentropia é uma medida não-negativa, e se torna igual a zero, quando a VA  $X$  tem distribuição gaussiana. Ou seja, quanto mais bem modelado por uma gaussiana um conjunto todo for, menor a sua negentropia.

A Figura 2.4 mostra o valor da negentropia para conjuntos de dados com distribuições diferentes.

Figura 2.5: Medidas da negentropia para conjuntos de dados com distribuições diferentes. (MARTINS, 2005)



Observando a Figura 2.5 vemos que a negentropia de distribuições gaussianas possuem valor igual a zero, enquanto distribuições não gaussianas possuem negentropia com valor maior que zero.

Em termos gerais, a negentropia mede o quanto a distribuição de uma VA não se parece com uma gaussiana, pois retorna o valor zero se a fdp é uma gaussiana. Mas esse fato só é verdadeiro para entropia de Shannon, (onde a distribuição gaussiana tem a maior entropia para uma variância fixa) pelo Princípio Máximo de Entropia. O problema é que a entropia de Shannon é difícil calcular diretamente a partir de amostras (MARTINS, 2005).

Uma alternativa proposta em Principe (1999) e empregada por Martins (2005, 2015)

é de utilizar a definição de entropia de Rényi ao invés da entropia de Shannon devido a simplicidade da estimativa. A entropia de Rényi é uma medida de entropia generalizada, parametrizada por uma constante  $\alpha$ , e é definida como

$$H_\alpha(p_X(X)) = \frac{1}{1-\alpha} \int_{-\infty}^{\infty} p_X(X)^\alpha dx \quad (2.17)$$

A escolha de  $\alpha > 0$  refere-se à escolha da norma para medir a distância da fdp para a origem sendo que para qualquer valor positivo de  $\alpha$ , as propriedades gerais para entropia de uma VA se mantêm.

### Método de Separação da Negentropia Quadrática de Rényi

Utilizando o conceito de negentropia de um conjunto de dados modelados por duas distribuições gaussianas Martins (2005, 2015) propôs um novo método para o cálculo da separação entre estas distribuições. O método tem como base a divergência entre a distribuição do conjunto com o modelo de uma única gaussiana que descreve todo o conjunto de dados, utilizando o conceito de mistura de gaussianas.

Quanto mais informações compartilham as gaussianas originais mais a gaussiana equivalente modela bem o conjunto todo. Quanto mais diferem, pior é o modelo com uma única gaussiana. Dessa forma, quanto mais bem modelado por uma gaussiana o conjunto todo for, menor a sua negentropia.

A negentropia da mistura de gaussianas é definida pela fórmula 2.18

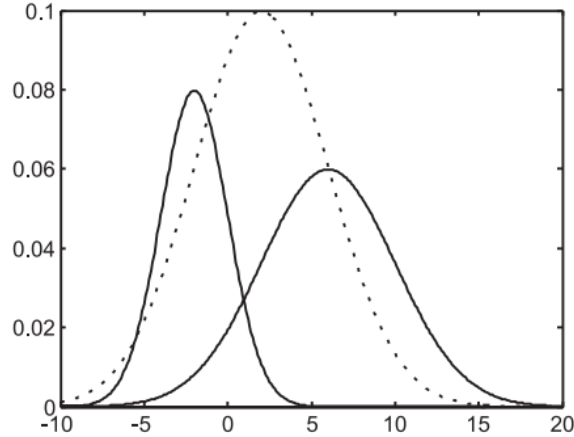
$$J(X) = H(g_X(x)) - H(p_X(x)) \quad (2.18)$$

Onde  $H(g_X(x))$  é a entropia da aproximação gaussiana gerada pelo modelo de mistura, e  $H(p_X(x))$  é a entropia de um modelo de distribuição gaussiana estimada utilizando o conjunto inteiro. A Figura 2.5 mostra como age a comparação entre duas distribuições utilizando o conceito de mistura de gaussianas.

Na Figura 2.6 a linha pontilhada representa o modelo gaussiano que "se ajusta" ao conjunto todo (considerando as duas distribuições). Desta forma compara-se a quantidade de variância presente no conjunto todo, com a variância de uma gaussiana que modela o mesmo. Se separarmos as distribuições mantendo sua variância, a variância da gaussiana equivalente aumentará e quanto mais dispersa for a distribuição maior a sua incerteza.

Para o cálculo da negentropia da mistura de gaussianas Martins, (2005, 2015) fez uso da Entropia de Rényi com ordem 2 ( $\alpha = 2$ ) porque nessa situação encontra-se uma formulação analítica para o negentropia de uma mistura de duas gaussianas. Com essa

Figura 2.6: Comparação entre um conjunto de dados formado por duas distribuições gaussianas com o modelo gaussiano equivalente.



escolha para o valor de  $\alpha$ , a integral resultante ocorre somente no quadrado do somatório das gaussianas e não no logaritmo do somatório, como é o caso da entropia de Shannon, o que simplifica bastante o cálculo analítico do novo método.

Este método tem a vantagem de ser insensível a distância entre as médias de  $g_x(x)$  e  $p_x(x)$ , dependendo apenas da diferença de gaussianidade entre a mistura e o conjunto, que são definidos formalmente por

$$p_x(x) = P_1N(x, \mu_1, \Sigma_1) + P_2N(x, \mu_2, \Sigma_2) \quad (2.19)$$

$$g_x(x) = N(x, \mu_q, \Sigma_q) \quad (2.20)$$

Onde  $P_1$  e  $P_2$  são a probabilidade a priori de cada gaussiana da mistura e  $N(x, \mu_i, \Sigma_i)$  é uma gaussiana com seus respectivos parâmetros. Os parâmetros para a mistura de gaussianas (definida pelo índice  $q$ ) são dados por

$$\mu_q = P_1\mu_1 + P_2\mu_2 \quad (2.21)$$

$$\Sigma_q = P_1\Sigma_1 + P_2\Sigma_2 + P_1P_2(\mu_1 - \mu_2)(\mu_1 - \mu_2)' \quad (2.22)$$

O método proposto calcula o quanto dois conjuntos possuem características em comum, tendo como base os conceitos apresentados acima. Quanto menor o valor retornado pelo método, maior o compartilhamento de informações entre os conjuntos. Essa medida foi denominada Separação da Negentropia Quadrática de Renyi (QRNS). A dedução desse método pode ser vista em (MARTINS, 2005) e é dada pela expressão

$$QRNS = \log \left( P_1^2 \sqrt{\frac{|\Sigma_q|}{|\Sigma_1|}} + \frac{2P_1P_2\sqrt{|\Sigma_q|} e^{-\frac{1}{2}(\mu_1 - \mu_2)^t (\Sigma_1 + \Sigma_2)^{-1} (\mu_1 - \mu_2)}}{\sqrt{|\frac{\Sigma_1 + \Sigma_2}{2}|}} + P_2^2 \sqrt{\frac{|\Sigma_q|}{|\Sigma_2|}} \right) \quad (2.23)$$

Onde  $P_1$  e  $P_2$  são as probabilidades a priori de cada conjunto da mistura definidas por  $P_i = \frac{N_i}{N}$ , onde  $N$  é o total de números de pontos dos conjuntos.  $\mu_1$  e  $\mu_2$  são as médias dos conjuntos e  $\Sigma_1$  e  $\Sigma_2$  são as matrizes de covariâncias definidas por

$$\Sigma_i = \frac{1}{N_i} \sum_{k \in \mathbf{K}_i} (x_k - \mu_i)(x_k - \mu_i)^t \quad (2.24)$$

Onde  $\mu_i$  é a média e  $N_i$  e número de pontos de dados  $x_k$  no conjunto  $\mathbf{K}_i$ . O operador  $||$  representa o determinante da matriz.

É interessante notar que a nova métrica dada na equação 2.23 é composta por três termos. O primeiro e o último termo, contabilizam a diferença entre as matrizes de covariância de cada gaussiana que compõe a mistura ( $\Sigma_1$  e  $\Sigma_2$ ), com a gaussiana equivalente do conjunto ( $\Sigma_q$ ).  $\Sigma_q$  depende da separação entre as médias bem como a matriz de covariância de cada distribuição, assim, quando as médias são separadas o valor do seu determinante cresce.

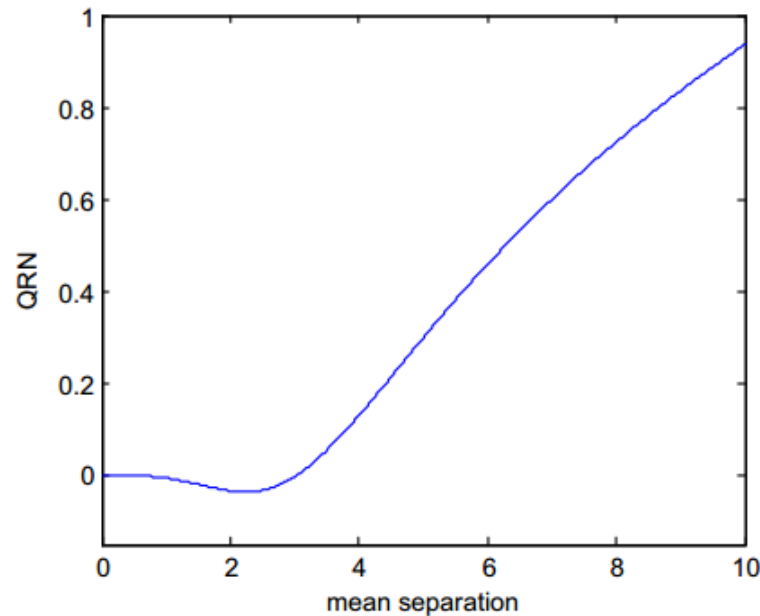
Já analisando o termo central, observa-se que o mesmo possui uma ponderação que tende a diminuir quando as médias se afastam (termo negativo na exponencial). Estes efeitos geram uma compensação no crescimento do QRNS até algum ponto de inflexão. Após este ponto, o QRNS cresce logarithmicamente com a diferença de médias entre as distribuições.

Assim, quando duas distribuições podem ser bem modeladas por uma única gaussiana o valor da medida de separação é próxima de zero, e quando as distribuições podem ser visualmente identificadas como independentes, a separação aumenta logarithmicamente. A Figura 2.6 mostra este comportamento para duas gaussianas com matrizes de covariância iguais a identidade e a mesma probabilidade a priori.

Uma região "estável" pode ser vista na Figura 2.7 entre 0 e 3 indicando que a estrutura das distribuições não se alterou muito e de fato, a mistura pode ainda ser bem modelada por uma única Gaussiana. Nesta região, devido ao crescente valor de  $\Sigma_q$  contra o decaimento da expressão exponencial do QRNS gera-se um efeito visual de "joelho" no gráfico. A partir dessa região o QRNS aumenta logarithmicamente.

Esse efeito de joelho no QRNS ocorre devido ao fato do uso da entropia quadrática de Rényi na dedução do método, pois o termo dentro do logaritmo pode ser menor do que um, obtendo-se valores negativos para o a medida de separação. Por causa disso a nova

Figura 2.7: QRNS em função da separação das médias para  $\Sigma_1 = \Sigma_2 = \mathbf{I}$  e  $P_1 = P_2 = 0.5$ . (MARTINS, 2005)



medida não pode ser declarada como uma distância.

O valor negativo no QRNS tem o mesmo significado do valor zero, indicando que as duas distribuições não devem ser separadas.

Essa medida é invariante a transformações lineares e vale ressaltar que a expressão 2.23 não se destina ao cálculo da negentropia diretamente. A negentropia foi apenas utilizada como inspiração para a medida da não-gaussianidade da mistura.

### 2.5.3 Aplicação do Método QRNS em Agrupamento de Dados

Nesta seção será descrito o uso do QRNS em um algoritmo de agrupamento de dados proposto por Martins (2005, 2015). Para este fim é utilizado um método de quantização vetorial, ferramentas e técnicas como o cálculo da separação da mistura de gaussianas (QRNS).

Esse método foi baseado em um algoritmo proposto por Tyree e Long (1999) onde o primeiro passo desse algoritmo consiste em uma simples quantificação vetorial em todo o conjunto de dados com  $N_a$  centros. Depois disso, a dissimilaridade entre os centros são estimadas e todos os centros que têm uma distância inferior a um limiar são agrupados em uma classe.

A principal diferença entre os dois métodos é que Tyree e Long (1999) utilizam a distância euclidiana para calcular a dissimilaridade entre os grupos e Martins (2005, 2015)

utiliza a medida QRNS.

O método de agrupamento com QRNS propõe basicamente dois algoritmos que dividem o método em duas etapas. A primeira etapa é um algoritmo que agrupa os pontos de um conjunto de dados em grupos. A segunda consiste de um algoritmo para modelar os grupos definidos na etapa anterior utilizando mistura de gaussianas e medir a quantidade de informações compartilhadas entre os mesmos, ligando os que possuem mais similaridade (MARTINS, 2005).

A primeira etapa consiste em realizar uma quantização vetorial em todo o conjunto de dados com  $N_a$  centros, que serão chamados de centros auxiliares. Assim, teremos um total de  $N_a$  clusters (grupos) formado. O valor de  $N_a$  é o primeiro parâmetro a ser fornecido ao algoritmo.

A escolha do número de centros auxiliares para a quantização vetorial não é crítica, porém é uma medida empírica. A escolha segura do número de centros pode se dar começando com uma quantidade elevada de centros, relativa ao número de pontos, de maneira a alocar em cada um dos  $N_a$  conjuntos, em média, um número maior do que a dimensão dos dados.

O número de pontos em cada conjunto deve ser maior que a dimensão dos dados para garantir que a matriz de covariância utilizada na próxima etapa seja inversível (isso garante que seus autovalores sejam diferentes de zero). O autor sugere uma escolha para  $N_a$  como sendo uma porcentagem do número total de pontos, neste caso, até 5 por cento do número total de pontos do conjunto.

Na segunda etapa, é gerada uma matriz U de separação que contabiliza a divergência entre cada par dentre os  $N_a$  clusters auxiliares, formando assim uma matriz  $N_a \times N_a$ . Esta matriz é simétrica e tem diagonal igual a zero (divergência entre uma classe auxiliar e ela mesma). Desta maneira, o elemento  $u_{ij}$  ou o elemento  $u_{ji}$  representa a divergência entre o cluster  $i$  e o cluster  $j$ .

A métrica de divergência utilizada para o cálculo da matriz U é o QRNS, onde cada par dos  $N_a$  clusters serão modelados cada um por uma gaussiana e gerado a mistura do conjunto como um todo.

Para estimar os parâmetros das gaussianas, fazemos uso do resultado do algoritmo de quantização vetorial realizada na etapa anterior. Cada centro auxiliar encontrado pelo algoritmo será tido como a média de cada uma das  $N_a$  gaussianas. Assim, tendo o valor da média e os pontos pertencentes a cada um dos clusters gerados, podemos determinar o grau de informação entre os pares de gaussianas utilizando o QRNS como descrito na seção 2.4.2.

Este processo de estimativa das gaussianas não corresponde ao processo ótimo. En-

tretanto, escolhendo um número suficiente de centros, o erro gerado pode chegar a ser desprezível e compensa quando observado a simplicidade computacional em relação a algoritmos que minimizariam completamente o erro como é o caso do algoritmo EM (MARTINS, 2005).

Após o cálculo da matriz  $U$ , é realizada a associação de cada cluster à uma determinada classe. Para associar um cluster a uma classe, é apresentado o conceito de matriz de ligação  $L$ . Esta matriz é definida como

$$\mathbf{L} = [l_{ij}]_{N_a \times N_a}$$

$$l_{ij} = \begin{cases} 1 & \text{gaussiana } i \text{ ligada a gaussiana } j \\ 0 & \text{caso contrario} \end{cases} \quad (2.25)$$

A matriz  $L$  possui diagonal preenchida com 1 e é simétrica. Os valores  $L_{i,j}$  são sempre 0 ou 1. O conceito de ligação entre clusters é utilizado para denominar gaussianas que pertencem a mesma classe.

Os valores da matriz  $L$  são gerados, comparando-se cada valor de  $U$  com um limiar  $d_t$  definido. Esse limiar é o responsável por decidir se um par de conjuntos qualquer será ligado ou não. Cada par (Cluster  $i$   $C_i$ , Cluster  $j$   $C_j$ ) é então testado; se sua divergência for menor ou igual do que o valor  $d_t$ , os conjuntos  $C_i$  e  $C_j$  serão ligados, ou seja  $L_{i,j} = 1$ , caso contrário teremos  $L_{i,j} = 0$  e os conjuntos não serão considerados ligados. Este limiar é então o segundo parâmetro a ser fornecido ao algoritmo.

Existe duas alternativas para definir o valor do limiar  $d_t$ . Uma delas é definida quando se sabe o número de classes que se deseja agrupar. Neste caso, os limiares possíveis são os próprios valores existentes na matriz  $U$ , variando do menor para o maior valor até chegar no número de classes desejado. Para exemplificar essa alternativa, considere a matriz  $U$  dada a seguir sabendo que o número total de classes formado pelos dados é igual a 2.

$$U = \begin{bmatrix} 0 & 0.24 & 0.9 & 0.012 \\ 0.24 & 0 & 0.76 & 0.55 \\ 0.9 & 0.76 & 0 & 0.89 \\ 0.012 & 0.55 & 0.89 & 0 \end{bmatrix}$$

Neste exemplo existe 6 valores possíveis para o limiar  $d_t$ : 0.012, 0.24, 0.55, 0.76, 0.89 e 0.9, na ordem do menor para o maior. Para o limiar  $d_t = 0.012$ , a matriz de ligação obtida é dada por

$$L = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Neste caso, o cluster 1 se ligou ao cluster 4 gerando uma classe e os clusters 2 e 3 formam sozinhos outras duas classes. No total têm-se 3 classes, sendo que esse valor não corresponde ao número esperado. Passamos então para o próximo valor possível para  $d_t$  que é o valor 0.24. A matriz de ligação obtida agora é dada por

$$L = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

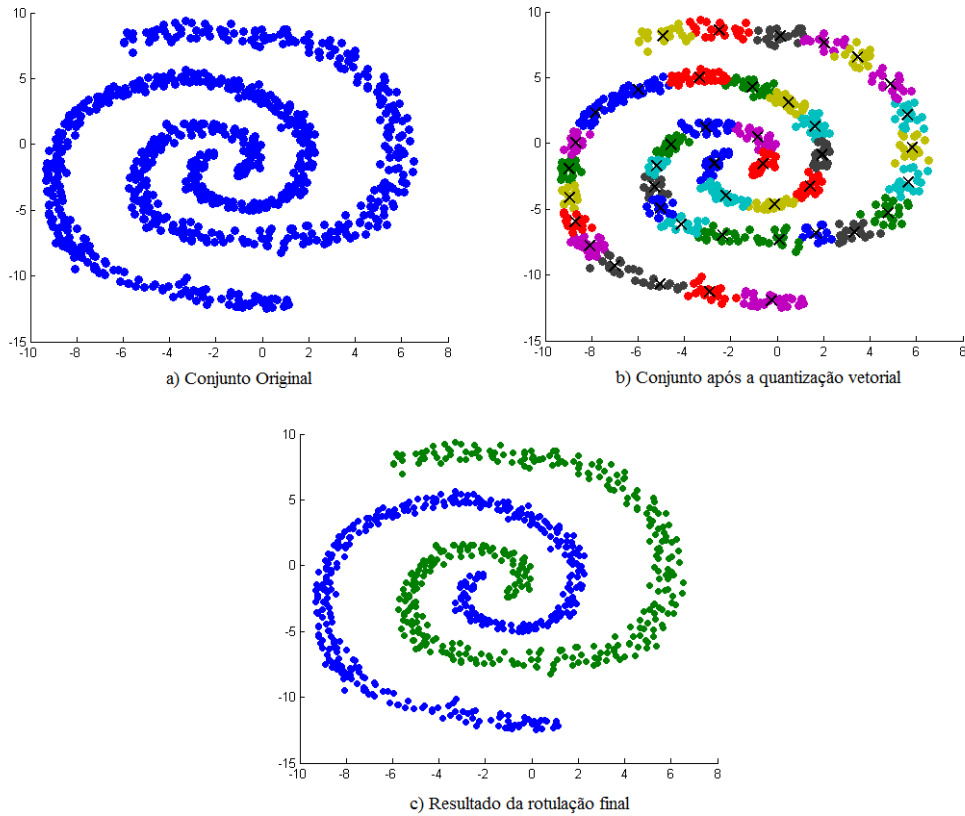
Temos agora 2 classes definidas, onde o cluster 1 está ligado aos clusters 2 e 4 gerando uma classe e a segunda classe é formada pelo cluster 3. Neste caso, temos o número de classes equivalente ao desejado e o valor do limiar final é  $d_t = 0.24$ .

É possível que, para algum valor de limiar  $d_t$ , o algoritmo forneça o número correto de classes, porém com o agrupamento errado. Porém este caso raramente ocorre devido à utilização das métricas que levam em conta a distribuição estatística e espacial dos conjuntos quantizados.

Após gerada a matriz de ligação, serão formadas as classes. A formação se dá associando classes iguais a clusters ligados, de maneira que se o cluster A está ligado ao cluster B e o cluster C ligado à B, então todos fazem parte da mesma classe. Após separadas as classes, cada ponto do conjunto de dados é rotulado com a classe atribuída ao seu cluster.

A Figura 2.7 ilustra o comportamento do conjunto de dados durante a execução do algoritmo e um resumo da técnica de agrupamento com a alternativa para o limiar  $d_t$  é visto no Algoritmo 2.

Figura 2.8: Etapas do algoritmo de clustering




---

**Algoritmo 2:** \*

---

**Entrada:** Conjunto de dados,  $N_a$ , número de classes desejado ( $\alpha$ )

**Saída:** Imagem segmentada

**início**

Realizar a quantização vetorial, separando os  $N_a$  clusters

Gerar a matriz de distâncias U:

**para** cada par dentre os  $N_a$  clusters **faça**

$U_{i,j} = QRNS(Cluster_i, Cluster_j)$

**fim**

**repita**

    Fazer  $d_t$  = menor valor em U que ainda não foi testado

    Gerar a matriz de ligação L:

**para** cada valor de U **faça**

**if**  $U_{i,j} \leq d_t$  **then**

$L_{i,j} = 1$

**else**

$L_{i,j} = 0$

**end**

**fim**

    Agrupar os clusters ligados em classes;

**até** número de classes gerado ser igual ao valor de  $\alpha$ ;

    Rotular cada ponto de cada cluster com sua respectiva classe

**fim**

Algoritmo 2: Método de Agrupamento de Dados

---

---

# Capítulo 3

## Abordagem Proposta

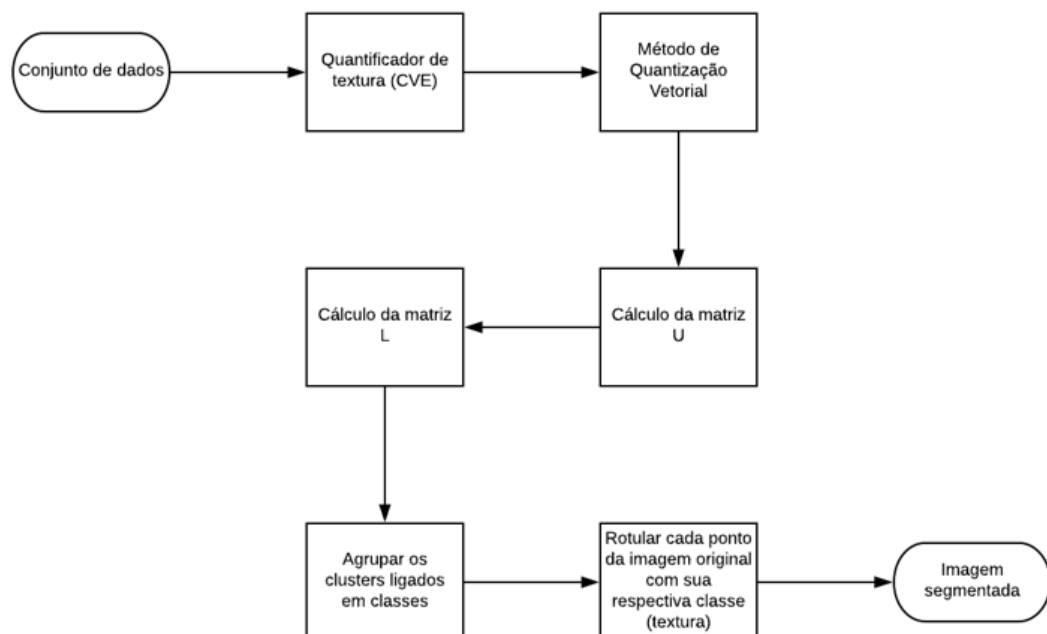
---

Este trabalho tem como proposta a integração das técnicas científicas apresentadas na seção de Fundamentação Teórica para segmentar imagens com base em suas texturas.

### 3.1 Desenvolvimento

A partir da obtenção de um conjunto de dados, oriundos de uma imagem, o propósito é agrupar os pontos amostrados da imagem em classes que irão corresponder às regiões de texturas diferentes. Uma vez agrupados os pontos, cada píxel da imagem é classificado e a imagem segmentada é gerada. A Figura 3.1 resume a abordagem proposta.

Figura 3.1: Fluxograma de etapas da abordagem proposta



O fluxo da abordagem é iniciado obtendo-se amostras da imagem que formarão o conjunto de dados inicial. Em cada região que representa uma textura na imagem um total de  $N$  amostras são extraídas. Essas amostras são obtidas por janelas de tamanho  $M \times M$ . Assim, cada amostra selecionada passa a representar um ponto no espaço euclidiano tridimensional e as coordenadas dos pontos são definidas pelo valor do CVE calculados para cada canal RGB ( $CVE_{red}$ ,  $CVE_{green}$ ,  $CVE_{blue}$ ). O valor para a janela é definido de forma empírica. A Figura 3.2(b) ilustra uma janela de tamanho  $5 \times 5$ , obtida da imagem original 3.2(a), em que será realizado o cálculo do valor do CVE para cada canal RGB e assim formar um dado para o conjunto inicial.

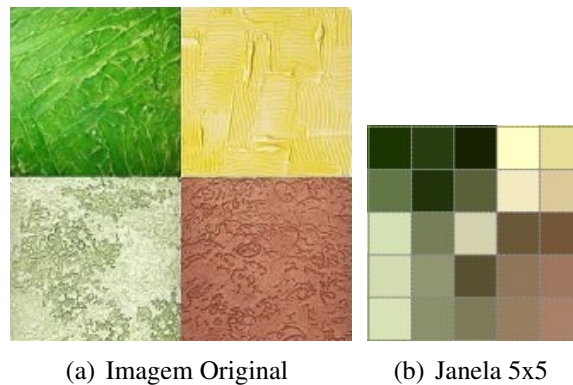


Figura 3.2: Exemplo janela de  $5 \times 5$

Gerado o conjunto de dados, o próximo passo é aplicá-lo na primeira etapa do método de agrupamento, que consiste em realizar a quantização vetorial nos dados. Nesta etapa, inicialmente defini-se o valor de  $N_a$ , que equivale ao total de clusters (grupos) que o conjunto de dados será representado. Neste trabalho foi utilizado o método K-means para realizar a quantização vetorial. Como visto na seção 2.1, tanto o k-means quanto as redes neurais competitivas fornecem valores de quantização bastante parecidos, sendo assim, o k-means foi escolhido por ser um método mais simples.

Após obter os  $N_a$  grupos e seus centros, segue-se para a segunda etapa do método de agrupamento. No início dessa etapa temos a realização do cálculo da matriz  $U$ , onde é necessário modelar cada grupo por uma gaussiana e calcular a divergência entre cada par de gaussianas e a sua correspondente mistura, utilizado o método QRNS.

Com o resultado da matriz de dissimilaridade  $U$  é possível obter a matriz de ligação  $L$ . É nesse ponto onde é realizada a ligação de grupos que possuem maior informações entre seus dados. Dentre as duas formas possíveis para o cálculo da matriz  $L$ , foi escolhida a abordagem informando o número total de classes, pois já sabemos a priori quantas regiões

de diferentes texturas possui a imagem. No final dessa etapa temos formadas as classes geradas pelo método de agrupamento que representam as possíveis texturas da imagem.

Por fim, para segmentar toda a imagem, é realizado a classificação de cada píxel da mesma como pertencente a uma das classes geradas na etapa anterior. Nesse passo, a distância de Mahalanobis foi utilizada para calcular a distância dos píxeis para os centros das classes. Essa métrica para a classificação é escolhida por ser uma medida não-Euclidiana que leva em consideração a estatística dos dados.

---

# Capítulo 4

## Resultados

---

O objetivo deste capítulo é avaliar o método proposto analisando o seu desempenho e aplicabilidade em diversas situações. A implementação da proposta foi realizada no *software* MATLAB e os testes executados em uma máquina com processador Core i5-4200U 1.60 Ghz com 8 Gb de memória RAM. As seções a seguir apresentam os resultados obtidos.

### 4.0.1 Mosaicos de Texturas Naturais

Texturas naturais são aquelas que resultam da intervenção natural do meio ambiente ou que caracterizam o aspecto exterior das formas e coisas existentes na natureza (cascas de troncos de árvores, madeira, papel, água, etc). Testes com diferentes texturas naturais reais em forma de mosaico foram realizados e serão apresentadas a seguir.

#### Texturas de Pedras

O método proposto foi aplicado na Figura 4.1 que possui 9 diferentes texturas encontradas em pedras retirada da base de dados do trabalho de Nunes (2006). O resultado da imagem nas suas componentes R, G e B pode ser visto na Figura 4.2.

Como entrada para o método proposto foi gerado um conjunto de dados com 9000 amostras, sendo 1000 amostras de cada textura, utilizando o método CVE como quantificador de textura. Cada amostra possui o CVE dos componentes R, G e B gerados a partir de uma janela 5x5. Para a quantização vetorial foi utilizado  $N_a = 45$  que é um valor menor do que 5 por cento do total de dados e que apresentou bons resultados. O tempo total necessário pelo método de agrupamento foi de 21,348 segundos.

A classificação da imagem segmentada foi gerada classificando-se cada pixel da mesma como pertencente a uma das classes geradas pelo método de agrupamento. A Figura 4.3 mostra o resultado do método para o número de classes definido igual a 9.

Figura 4.1: Mosaico de texturas de pedras

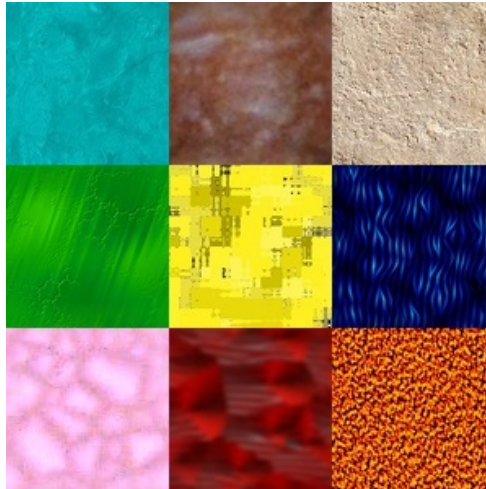
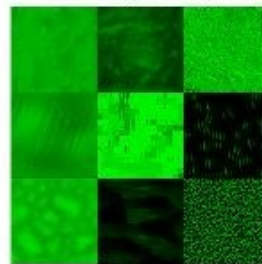
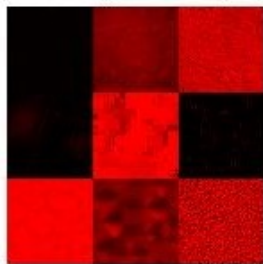


Figura 4.2: Imagem nas suas componentes R, G e B

**Componente R (vermelha) dos pixels**      **Componente G (verde) dos pixels**



**Componente B (azul) dos pixels**

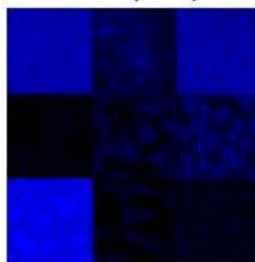
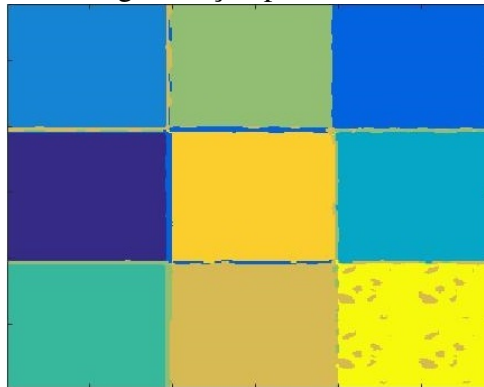
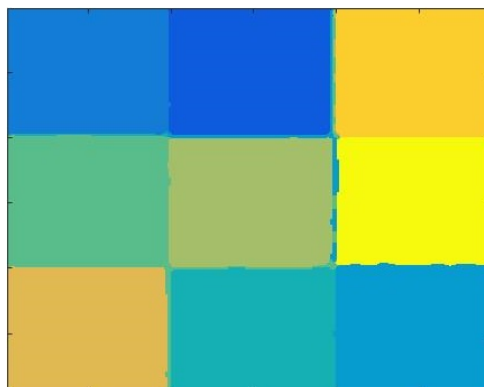


Figura 4.3: Resultado da segmentação para o mosaico de texturas de pedras



Para suavizar alguns ruídos presentes no resultado, como na classe 9 da Figura 4.3, foi utilizado um filtro de média com uma janela  $5 \times 5$  resultando em uma imagem com boa definição dos limites das classes, que pode ser vista na Figura 4.4.

Figura 4.4: Resultado da segmentação do mosaico de texturas de pedras com filtro de média



### Texturas de Madeiras

A Figura 4.5 apresenta um mosaico com 9 diferentes texturas de madeiras retirado da base de dados do trabalho de Nunes (2006). Como entrada para o método proposto foram selecionadas 1000 amostras de cada textura com uma janela  $5 \times 5$ . Para a quantização vetorial foi utilizado  $N_a = 50$ . O tempo total necessário pelo método de agrupamento foi de 22,60 segundos e o resultado pode ser visto na Figura 4.6.

Para suavizar alguns ruídos presentes no resultado um filtro de média foi utilizado com uma janela  $5 \times 5$  resultando em uma imagem com boa definição dos limites das classes, que pode ser vista na Figura 4.7.

Figura 4.5: Mosaico de texturas de madeiras



Figura 4.6: Resultado da segmentação para o mosaico de texturas de madeiras

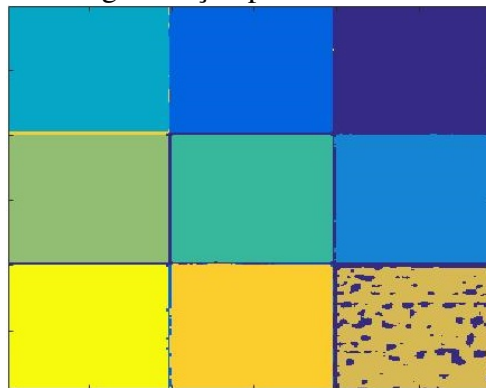
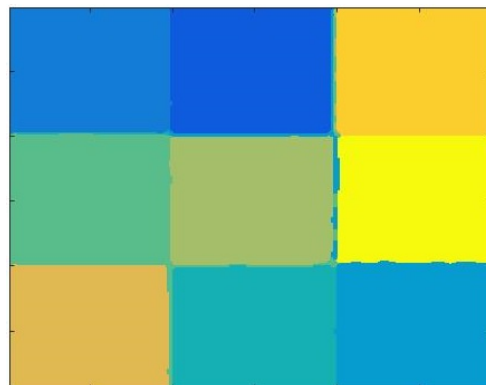


Figura 4.7: Resultado da segmentação do mosaico de texturas de madeiras com filtro de média



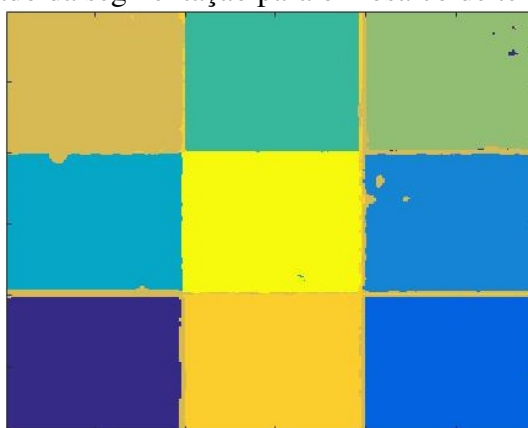
### Texturas de Vegetações

A Figura 4.8 apresenta um mosaico com 9 diferentes texturas de vegetações retirado da base de dados do trabalho de Nunes (2006) que foram extraídas de imagens do satélite Landsat -7 TM. Como entrada para o método proposto foram selecionadas 1000 amostras de cada textura com uma janela  $7 \times 7$ . Para a quantização vetorial foi utilizado  $N_a = 50$ . O tempo total necessário pelo método de agrupamento foi de 22,10 segundos e o resultado pode ser visto na Figura 4.9.

Figura 4.8: Mosaico de texturas de vegetações

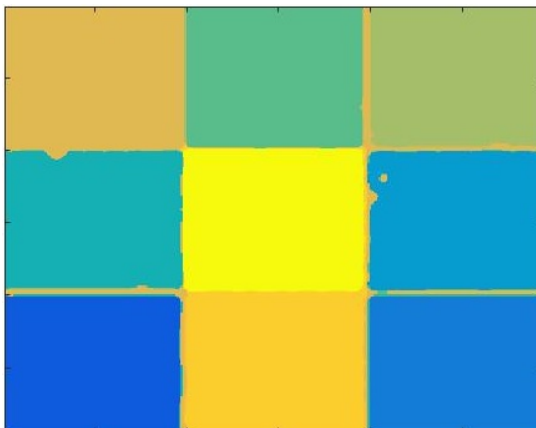


Figura 4.9: Resultado da segmentação para o mosaico de texturas de vegetações



Para suavizar alguns ruídos presentes no resultado um filtro de média foi utilizado com uma janela  $5 \times 5$  resultando em uma imagem com boa definição dos limites das classes, que pode ser vista na Figura 4.10.

Figura 4.10: Resultado da segmentação para o mosaico de texturas de vegetações com filtro de média

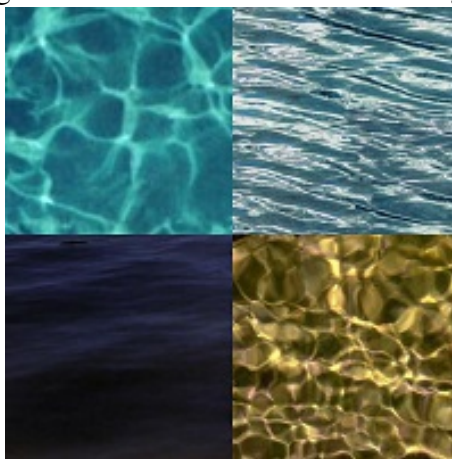


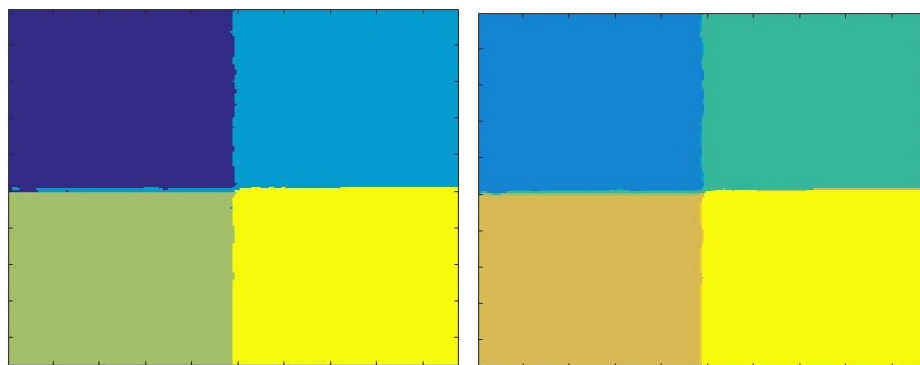
Neste resultado é notável que apesar do mosaico conter algumas texturas em tonalidades verde, o algoritmo conseguiu fazer as devidas distinções, pois o quantificador leva em consideração também o relacionamento espacial entre os píxeis.

### Texturas de Água

A Figura 4.11 apresenta um mosaico com 4 diferentes texturas de água montado a partir da base de dados Texture Library. Como entrada para o método proposto foram selecionadas 1000 amostras de cada textura com uma janela  $7 \times 7$ . Para a quantização vetorial foi utilizado  $N_a = 20$ . O tempo total necessário pelo método de agrupamento foi de 6,164 segundos e o resultado pode ser visto na Figura 4.12(a). E para suavizar alguns ruídos presentes no resultado um filtro de média foi utilizado com uma janela  $3 \times 3$  e o resultado pode ser visto na Figura 4.12(b).

Figura 4.11: Mosaico de texturas de água





(a) Resultado Segmentação da Figura 4.11 (b) Resultado da segmentação com filtro de média para a Figura 4.12(a)

Figura 4.12: Resultado da segmentação

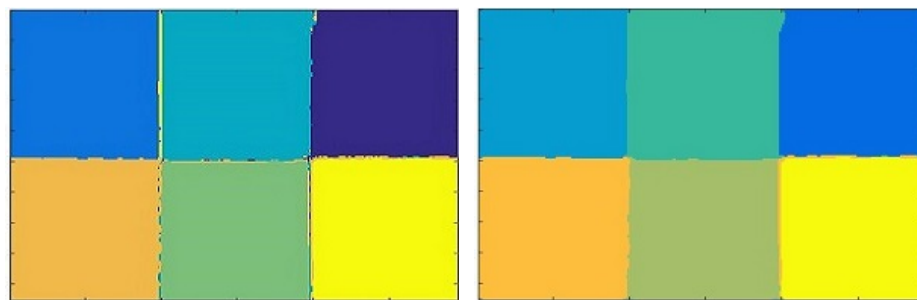
## 4.0.2 Texturas em Parede

Texturas de paredes são exemplos de texturas artificiais que resultam da intervenção humana através da utilização de materiais e instrumentos devidamente manipulados. A Figura 4.13 apresenta um mosaico com 6 diferentes texturas.

Como entrada para o método proposto foram selecionadas 1000 amostras de cada textura com uma janela  $5 \times 5$ . Para a quantização vetorial foi utilizado  $N_a = 30$ . O tempo total necessário pelo método de agrupamento foi de 11,484 segundos e o resultado pode ser visto na Figura 4.38(a). O filtro de média com janela  $7 \times 7$  apresentou o resultado que pode ser visto na Figura 4.28(b).

Figura 4.13: Mosaico de texturas de paredes





(a) Resultado Segmentação para a Figura 4.13 (b) Resultado da segmentação com filtro de média

Figura 4.14: Resultado da segmentação

### Texturas em Parede com mesma cor

A seguir será mostrado o resultado de testes realizados com diferentes texturas em paredes da mesma cor mas com variações tonais, com a finalidade de analisar a eficiência do método proposto quando as texturas possuem tonalidades próximas.

### Texturas Amarelas

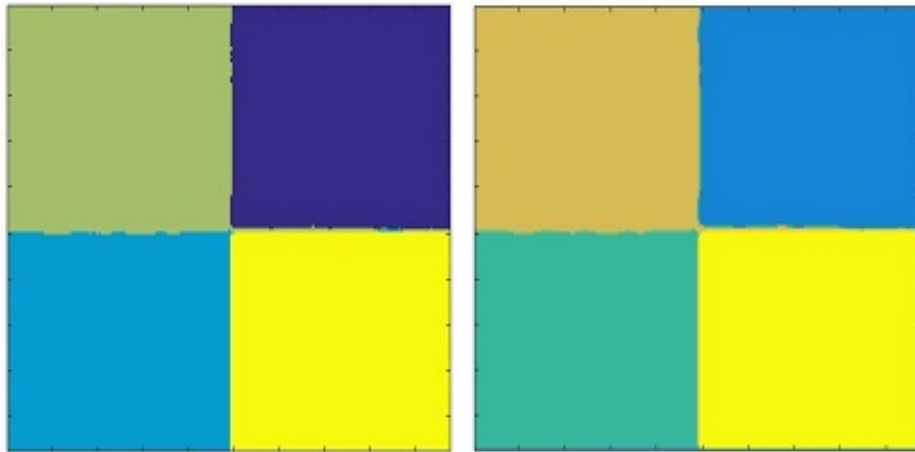
A Figura 4.15 mostra um mosaico com 4 diferentes texturas amarelas encontradas em paredes.

Figura 4.15: Mosaico de texturas de paredes na cor amarela



Foram utilizadas 1000 amostras de cada textura com uma janela  $5 \times 5$ . Para a quantização vetorial foi utilizado  $N_a = 20$ . O tempo total necessário pelo método de agrupamento foi de 9,58 segundos e o resultado pode ser visto na Figura 4.16(a). O filtro de média

com janela  $4 \times 4$  apresentou o resultado que pode ser visto na Figura 4.16(b) que suaviza o resultado da segmentação nas fronteiras das texturas.



(a) Resultado Segmentação para a Figura 4.15  
(b) Resultado da segmentação da Figura 4.15 com filtro de média

Figura 4.16: Resultado da segmentação

### Texturas Verdes

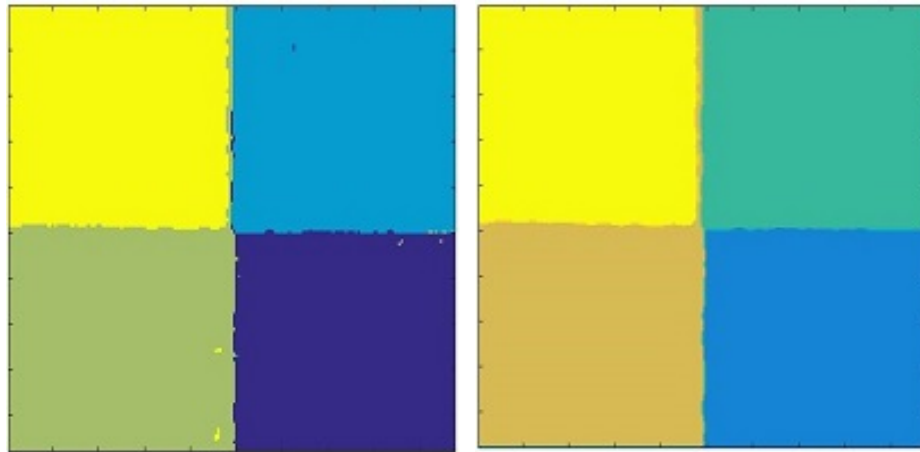
A Figura 4.17 mostra um mosaico com 4 diferentes texturas verdes encontradas em paredes.

Figura 4.17: Mosaico de texturas de paredes na cor verde



Foram utilizadas 1000 amostras de cada textura com uma janela  $5 \times 5$ . Para a quantização vetorial foi utilizado  $N_a = 20$ . O tempo total necessário pelo método de agrupamento foi de 7,683 segundos e o resultado pode ser visto na Figura 4.18(a). O filtro de média

com janela  $4 \times 4$  apresentou o resultado que pode ser visto na Figura 4.18(b) onde suaviza o resultado da segmentação nas fronteiras das texturas.



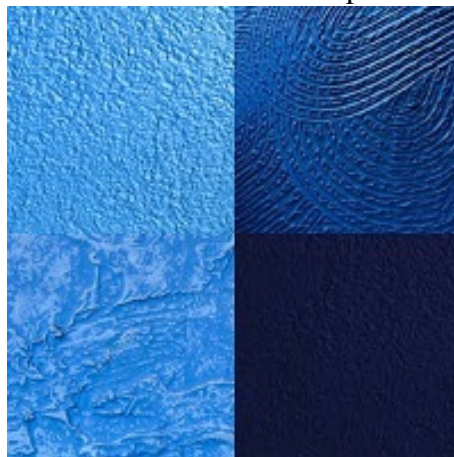
(a) Resultado Segmentação da Figura 4.17 (b) Resultado da segmentação da Figura 4.17 com filtro de média

Figura 4.18: Resultado da segmentação

### Texturas Azuis

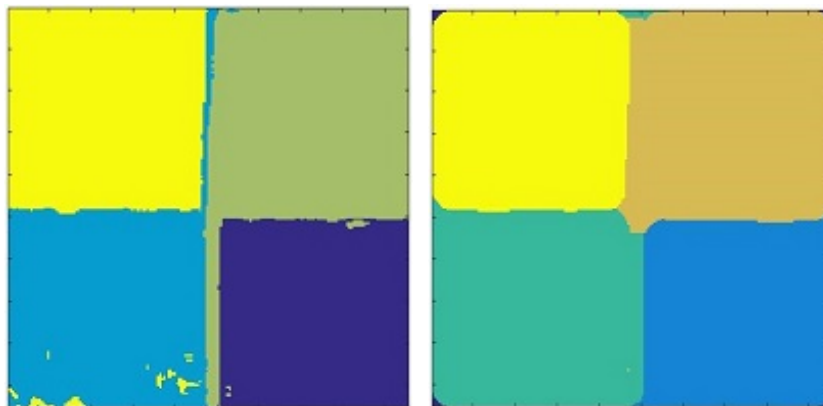
A Figura 4.19 mostra um mosaico com 4 diferentes texturas azuis encontradas em paredes.

Figura 4.19: Mosaico de texturas de paredes na cor azul



Foram utilizadas 1000 amostras de cada textura. O melhor resultado obtido nesse caso foi com uma janela  $11 \times 11$ . Para a quantização vetorial foi utilizado  $N_a = 50$ . O tempo total necessário pelo método de agrupamento foi de 10,69 segundos e o resultado pode

ser visto na Figura 4.20(a). Para suavizar os ruídos obtidos no resultado um filtro com janela 15x15 apresentou o melhor resultado que pode ser visto na Figura 4.20(b).



(a) Resultado Segmentação da Figura 4.19  
(b) Resultado da segmentação da Figura 4.19 com filtro de média

Figura 4.20: Resultado da segmentação

Com esses testes, é possível notar novamente que o método de quantização CVE conseguiu gerar amostras eficientes e o método de agrupamento fez as devidas distinções, pois o CVE leva em consideração também o relacionamento espacial entre os píxeis.

### 4.0.3 Imagens de Satélite

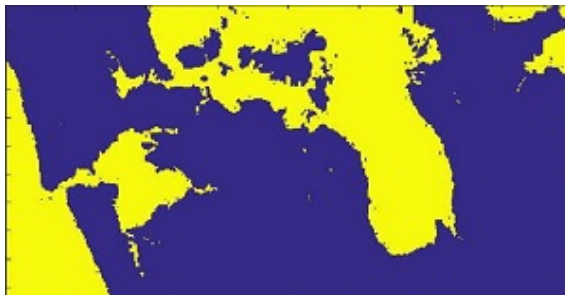
O teste a seguir foi realizado em imagens retiradas da base de dados do satélite Landsat. A Figura 4.21 mostra uma imagem obtida pelo satélite no dia 2 de janeiro de 2014 em Auckland na Nova Zelândia.

Figura 4.21: Imagem de Auckland pelo satélite Landsat



Para a segmentação foram selecionadas duas classes, água e solo, com 1500 amostras de cada classe, com janela de dimensão 3x3. Para a quantização vetorial foi utilizado  $N_a = 12$ . O tempo total necessário pelo método de agrupamento foi de 3,20 segundos.

O resultado pode ser visto na Figura 4.22(a). Para melhor análise do resultado a Figura 4.22(b) apresenta o contorno do resultado da segmentação.



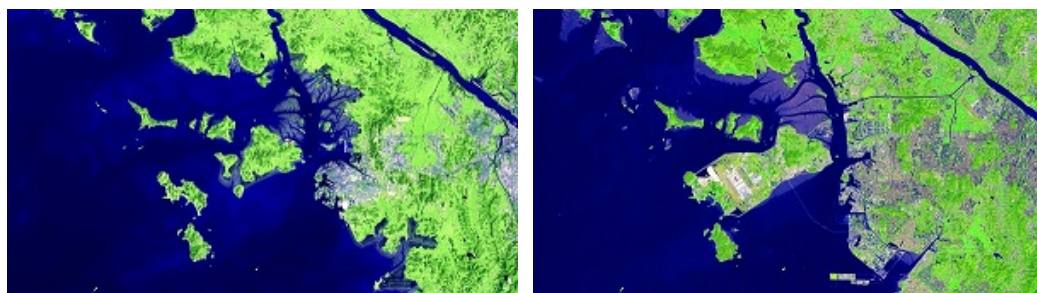
(a) Resultado da segmentação



(b) Contorno do Resultado da segmentação

Figura 4.22: Resultado segmentação da Figura 4.21

Já a Figura 4.23(a) mostra uma costa de Incheon, cidade da Coreia do Sul, tirada no dia 5 de setembro de 1981 e a Figura 4.23(b) foi obtida no dia 16 de setembro de 2013 (32 anos depois).



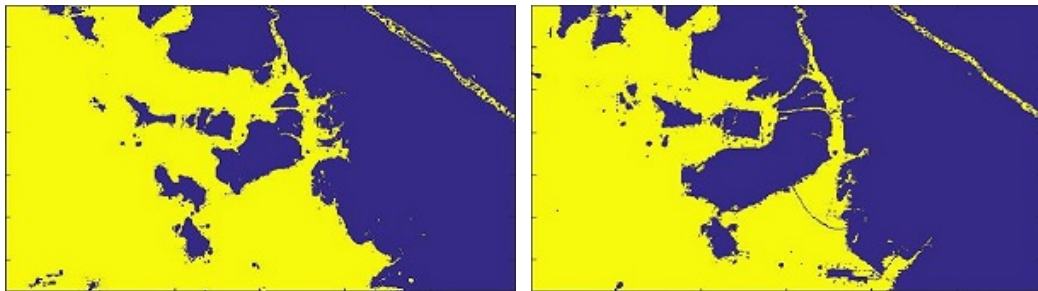
(a) Imagem obtida pelo satélite Landsat em 05/09/1981

(b) Imagem obtida pelo satélite Landsat em 16/09/2013

Figura 4.23: Imagens de Incheon pelo satélite Landsat

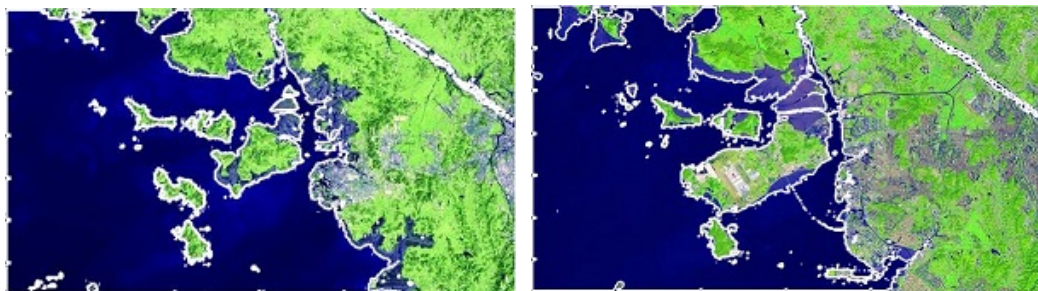
Foram selecionadas duas classes, água e solo, com 2000 amostras de cada classe para formar o conjunto de dados, com janela de dimensão  $3 \times 3$ . Para a quantização vetorial

foi utilizado  $N_a = 12$ . O tempo total necessário pelo método de agrupamento foi de 5,55 segundos. O resultado pode ser visto nas Figuras 4.24(a) e 4.24(b). Para melhor análise do resultado as Figuras 4.25(a) e 4.25(b) apresentam o contorno do resultado da segmentação.



(a) Resultado da segmentação da Figura 4.23(a) (b) Resultado da segmentação da Figura 4.23(b)

Figura 4.24: Resultado da segmentação das Figuras 4.23(a) e Figura 4.23(b)



(a) Contorno do resultado da segmentação da Figura 4.23(a) (b) Contorno do resultado da segmentação da Figura 4.23(b)

Figura 4.25: Contorno do resultado da segmentação das Figuras 4.23(a) e Figura 4.23(b)

Imagens de satélite possuem texturas complexas e grande tamanho, mesmo assim, é possível observar nos resultados uma boa classificação dos temas e definição dos limites.

#### 4.0.4 Imagens Médicas

Uma das possíveis aplicações de métodos de segmentação é em imagens médicas. Para exemplificar, o método proposto foi aplicado em imagens reais que mostram uma região da pele que contém melanoma (câncer de pele) em diferentes estágios, que podem ser vistas na Figura 4.26.

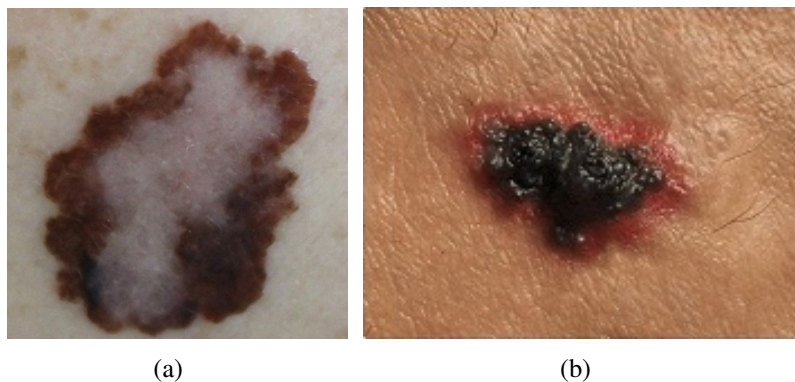
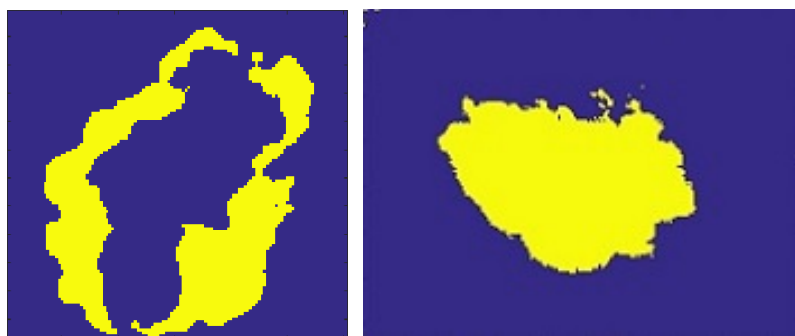


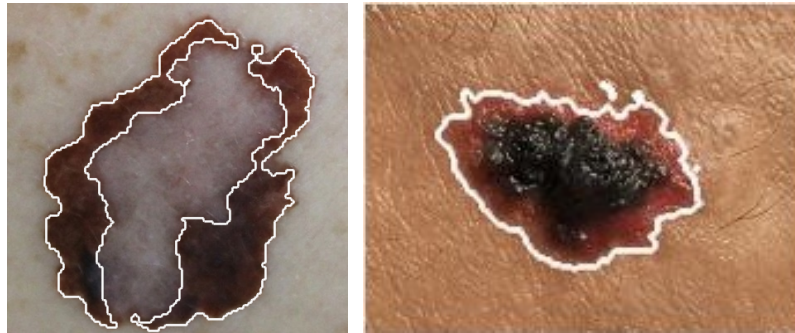
Figura 4.26: Imagens de melanomas

O método de segmentação é utilizado, neste caso, para identificar a região da pele afetada pela doença. Neste teste, foram selecionadas 2000 amostras, sendo 1000 da região da pele afetada e 1000 da região não afetada, e uma janela  $3 \times 3$  para o cálculo do CVE. Para a quantização vetorial foi utilizado  $N_a = 12$ . A Figura 4.27 apresenta o resultado da segmentação, onde é possível ver a identificação do melanoma em ambos os casos. Já a Figura 4.28 apresenta o contorno dos resultados da segmentação, resultando numa imagem com boa definição dos limites da doença visíveis na pele.



(a) Resultado da segmentação da Figura 4.26(a) (b) Resultado da segmentação da Figura 4.26(b)

Figura 4.27: Resultados da segmentação das Figuras 4.26(a) e 4.26(b)



(a) Contorno do resultado da segmentação da Figura 4.26(a)  
(b) Contorno do resultado da segmentação da Figura 4.26(b)

Figura 4.28: Contornos do resultado da segmentação das Figuras 4.26(a) e 4.26(b)

#### 4.0.5 Mais exemplos de resultados pelo método proposto

Nesta seção será demonstrado mais aplicações do método em diferentes tipos de imagens reais.

##### Texturas de Papel

A Figura 4.29 apresenta um mosaico com 9 diferentes texturas de papel retirado da base de dados do trabalho de Nunes (2006).

Figura 4.29: Mosaico de texturas de papeis



Como entrada para o método proposto, foram selecionadas 1000 amostras de cada textura com uma janela  $3 \times 3$ . Para a quantização vetorial foi utilizado  $N_a = 40$ . O tempo total necessário pelo método de agrupamento foi de 21,87 segundos e o resultado pode

ser visto na Figura 4.30(b) Para suavizar alguns ruídos presentes no resultado um filtro de média foi utilizado com uma janela  $3 \times 3$  resultando em uma imagem com boa definição dos limites das classes, que pode ser vista na Figura 4.30(b).

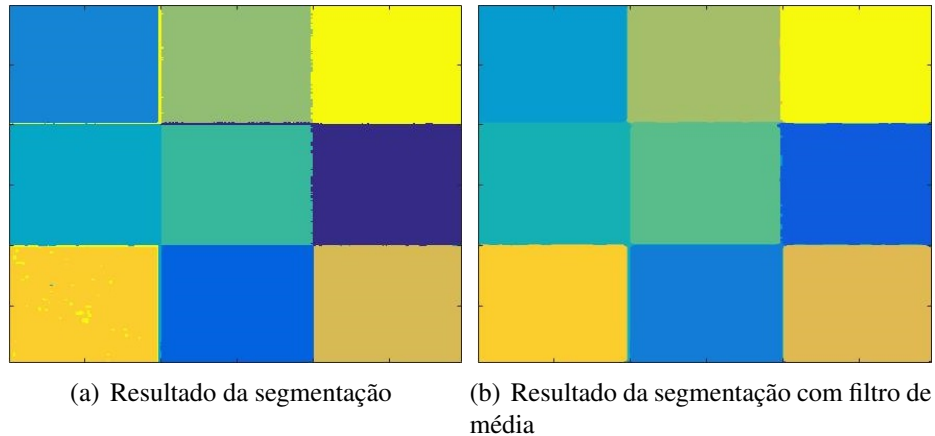
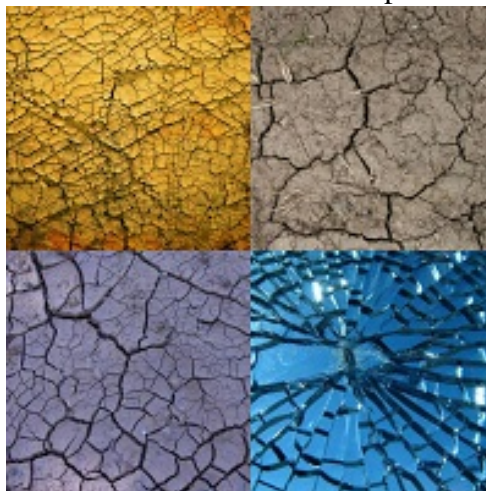


Figura 4.30: Resultado da segmentação do mosaico de texturas de papeis

### Texturas de Superfícies Rachadas

A Figura 4.31 apresenta um mosaico com 4 texturas de diferentes superfícies rachadas retirado da base de dados Describable Textures Dataset (DTD) (CIMPOI et al., 2014).

Figura 4.31: Mosaico de texturas de superfícies rachadas



Como entrada para o método proposto, foram selecionadas 1000 amostras de cada textura com uma janela  $7 \times 7$ . Para a quantização vetorial foi utilizado  $N_a = 30$ . O tempo total necessário pelo método de agrupamento foi de 9,77 segundos e o resultado pode ser

visto na Figura 4.32(a). Para suavizar alguns ruídos presentes no resultado um filtro de média foi utilizado com uma janela  $5 \times 5$  resultando em uma imagem com boa definição dos limites das classes, que pode ser vista na Figura 4.32(b).

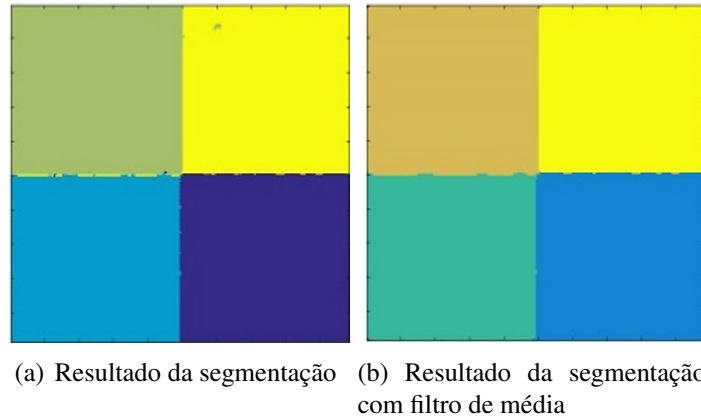


Figura 4.32: Resultados da segmentação para o mosaico de texturas de superfícies rachadas

### Texturas em Tecidos

A Figura 4.33(a) apresenta um mosaico com 4 texturas de diferentes tecidos retirado da base de dados Describable Textures Dataset (DTD) (CIMPOI et al., 2014). Como entrada para o método proposto, foram selecionadas 1000 amostras de cada textura com uma janela  $5 \times 5$ . Para a quantização vetorial foi utilizado  $N_a = 20$ . O tempo total necessário pelo método de agrupamento foi de 14,86 segundos e o resultado pode ser visto na Figura 4.33(b).

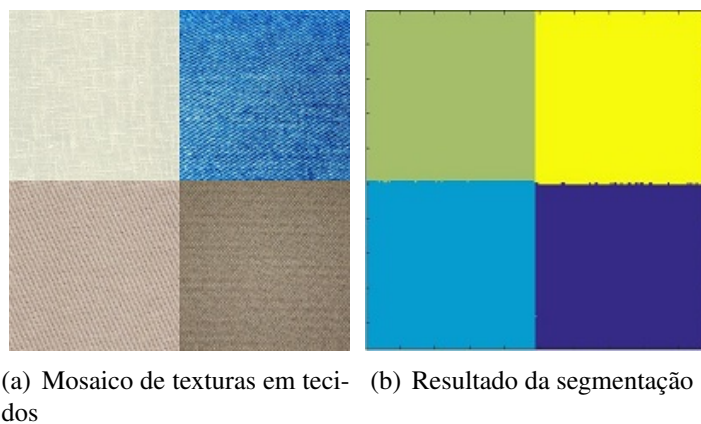


Figura 4.33: Resultado da segmentação da Figura 4.33(a)

### Texturas em Superfícies Listradas

A Figura 4.34(a) apresenta um mosaico com 4 texturas de diferentes superfícies retirado da base de dados Describable Textures Dataset (DTD) (CIMPOI et al., 2014). Como entrada para o método proposto, foram selecionadas 1000 amostras de cada textura com uma janela  $9 \times 9$ . Para a quantização vetorial foi utilizado  $N_a = 10$ . O tempo total necessário pelo método de agrupamento foi de 10,66 segundos e o resultado pode ser visto na Figura 4.34(b).

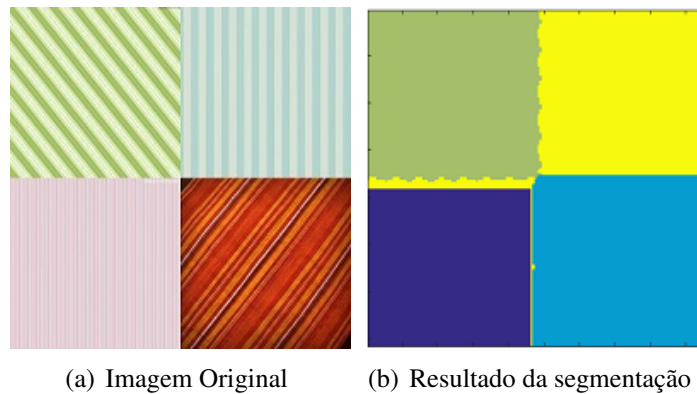
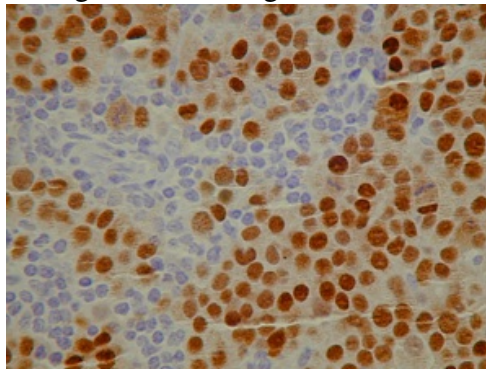


Figura 4.34: Resultado da segmentação da Figura 4.34(a)

### Imagem de Células

A Figura 4.35 apresenta uma imagem com dois grupos de células. Como entrada para o método proposto, foram selecionadas 500 amostras de cada textura com uma janela  $3 \times 3$ . Para a quantização vetorial foi utilizado  $N_a = 12$ . O resultado da segmentação pode ser visto na Figura 4.36(a) e uma melhor visualização do contorno em 4.36(b).

Figura 4.35: Imagem de células



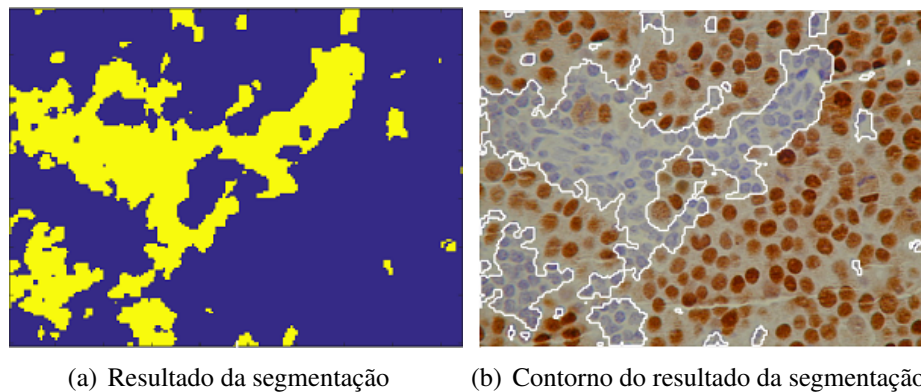


Figura 4.36: Resultado da segmentação da Figura 4.35

### Contornos em Texturas Naturais

Testes em imagens com texturas naturais também foram realizados para identificar seus diferentes objetos. A Figura 4.37(a) apresenta uma imagem real retiradas da base de dados Semantic Benchmark (LI J. CAI, 2013). O objetivo neste caso é identificar o contorno dos cavalos na imagem. Como entrada para o método proposto, foram selecionadas 500 amostras de cada textura com uma janela  $5 \times 5$ . Para a quantização vetorial foi utilizado  $N_a = 25$ . O tempo total necessário pelo método de agrupamento foi de 3,52 segundos e o resultado pode ser visto na Figura 4.37(b).

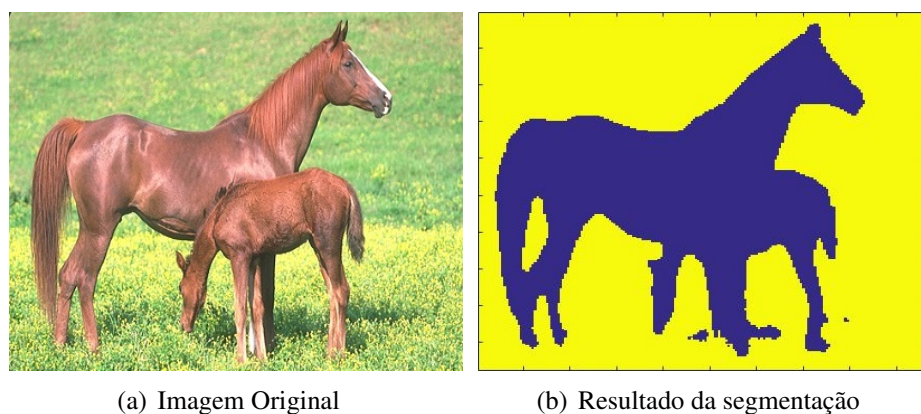


Figura 4.37: Resultado da segmentação da Figura 4.37(a)

Para suavizar alguns ruídos presentes no resultado um filtro de média foi utilizado com uma janela  $5 \times 5$  e para melhor visualização o contorno da segmentação é mostrado na Figura 4.38(b)

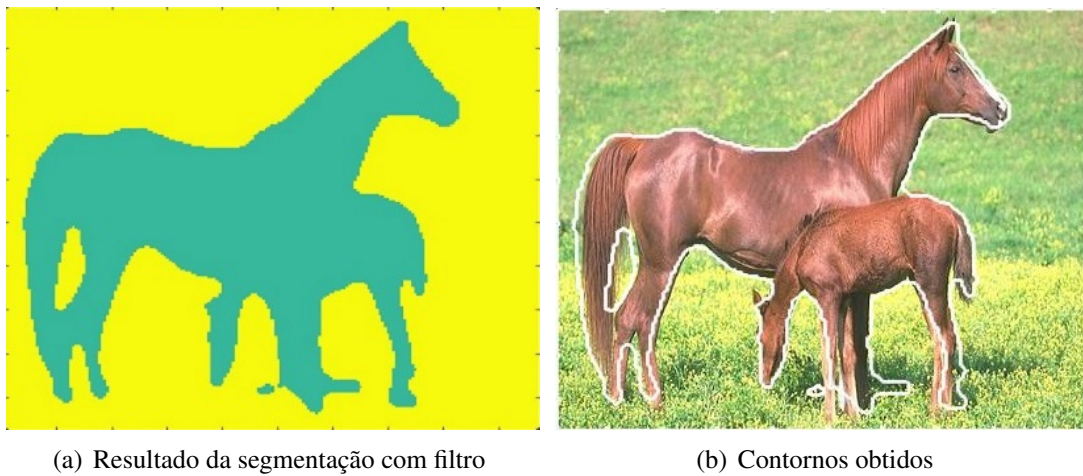


Figura 4.38: Resultado do contorno da segmentação com filtro de média da Figura 4.37(a)

A Figura 4.39(a) também foi retiradas da base de dados Semantic Benchmark (LI J. CAI, 2013). O objetivo neste caso é identificar o contorno do tigre. Como entrada para o método proposto, foram selecionadas 1000 amostras de cada classe com uma janela  $5 \times 5$ . Para a quantização vetorial foi utilizado  $N_a = 30$ . A Figura 4.39(b) apresenta o resultado da segmentação e a Figura 4.39(c) apresenta o contorno do resultado da segmentação para melhor análise do resultado.



(a) Imagem Original

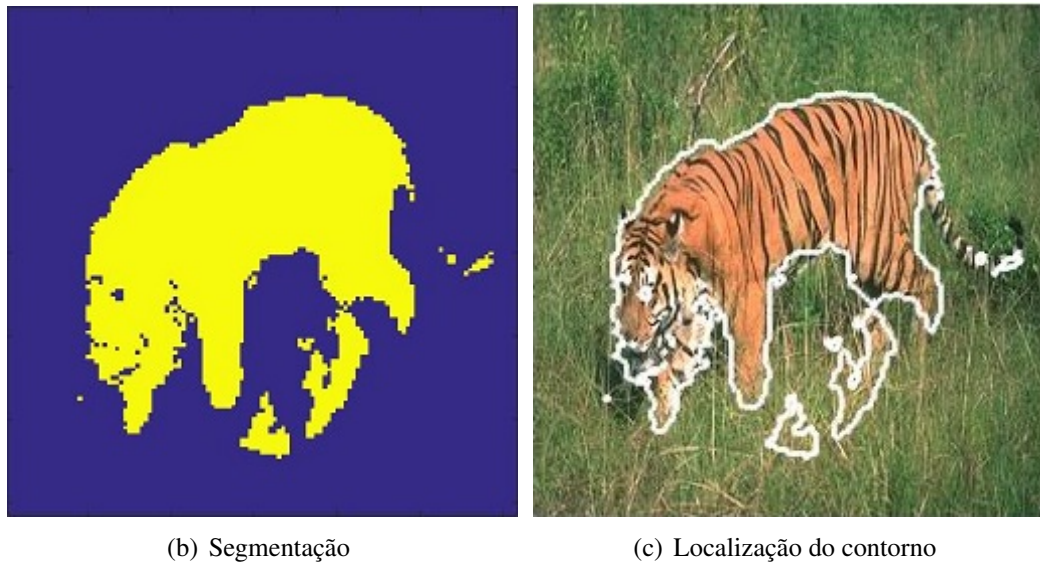
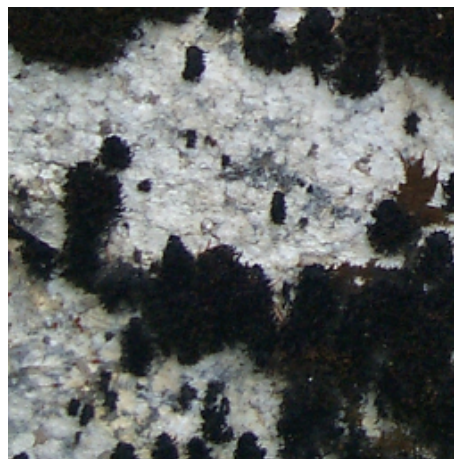


Figura 4.39: Resultado da segmentação para a Figura 4.39(a)

A Figura 4.40(a) foi retirada da base de dados Prague Dataset do trabalho de Yuan et. all (2015). Foram selecionadas um total de 4000 amostras e uma janela  $5 \times 5$ . Para a quantização vetorial foi utilizado  $N_a = 30$ . A Figura 4.40(b) apresenta o resultado da segmentação e a Figura 4.40(c) apresenta o contorno do resultado da segmentação para melhor análise do resultado.



(a) Imagem Original

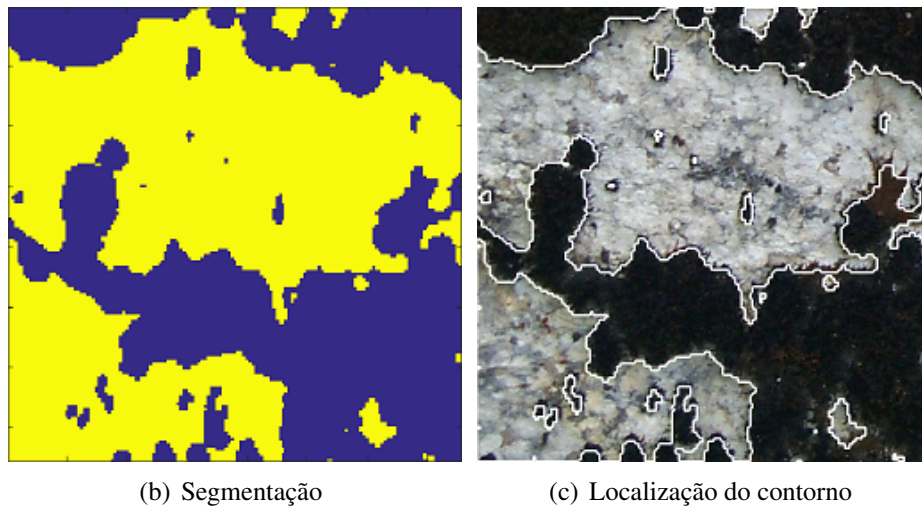


Figura 4.40: Resultado da segmentação da Figura 4.40(a)

Já como entrada para o método proposto em relação a Figura 4.41(a) foram selecionadas 1000 amostras de cada textura com uma janela  $5 \times 5$ . Para a quantização vetorial foi utilizado  $N_a = 12$ . O tempo total necessário pelo método de agrupamento foi de 3,04 segundos e o resultado pode ser visto na Figura 4.41(b).

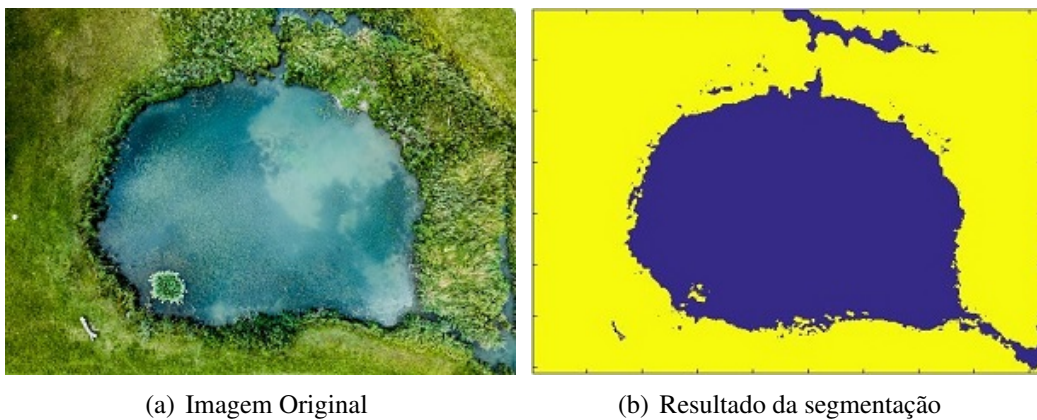


Figura 4.41: Resultado da segmentação da Figura 4.41(a)

Para suavizar alguns ruídos presentes no resultado um filtro de média foi utilizado com uma janela  $4 \times 4$  e para melhor visualização o contorno da segmentação é mostrado na Figura 4.42(b).

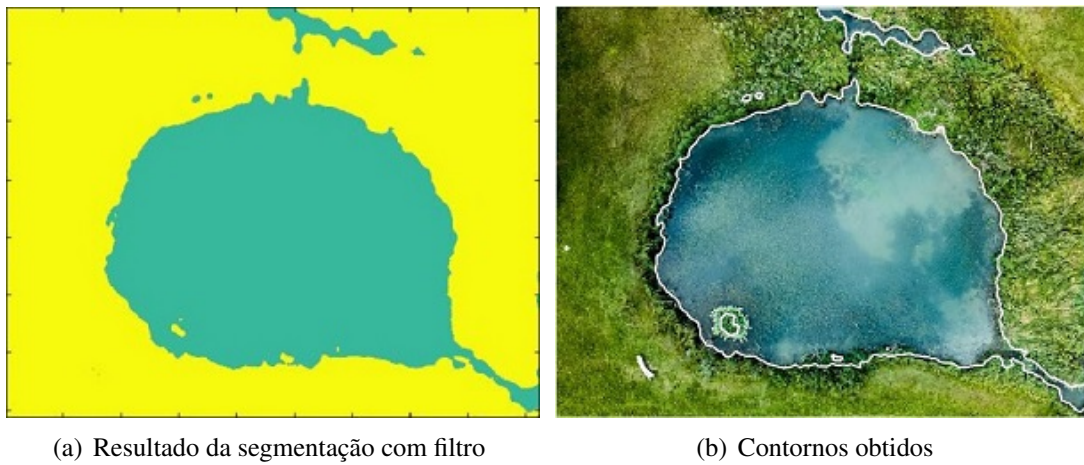
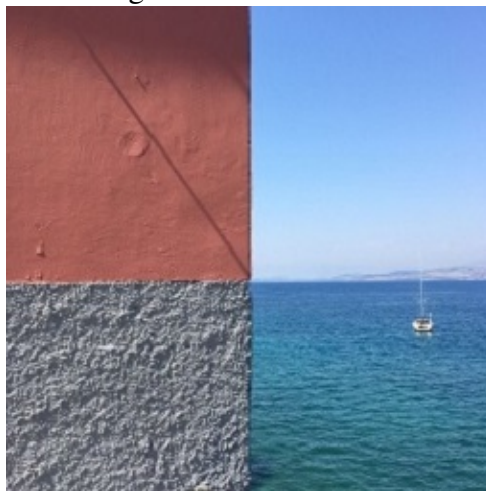


Figura 4.42: Resultado do contorno da segmentação com filtro de média da Figura 4.41(a)

### Imagem com texturas naturais e artificiais

Na Figura 4.43 é uma imagem que apresenta texturas naturais e artificiais. Para segmentar foram seleccionadas 4 classes: parede lisa, parede áspera, céu e mar. Foram seleccionadas 1000 amostras de cada textura com uma janela  $5 \times 5$ . Para a quantização vetorial foi utilizado  $N_a = 30$ . O tempo total necessário pelo método de agrupamento foi de 10,14 segundos.

Figura 4.43: Imagem de texturas naturais e artificiais



A Figura 4.44(a) apresenta o resultado da segmentação e a Figura 4.44(b) apresenta o resultado com filtro de média de janela  $10 \times 10$  para suavizar os ruídos.

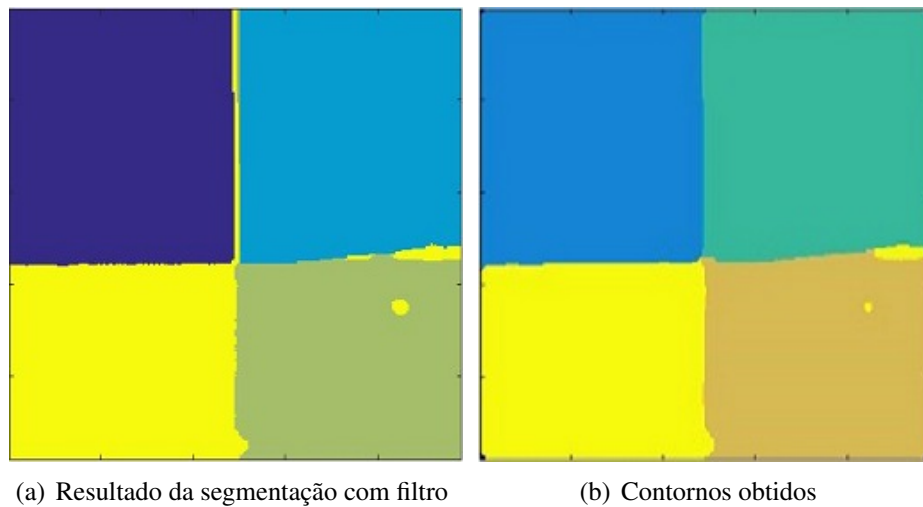


Figura 4.44: Resultado da segmentação da Figura 4.0.5

#### 4.0.6 Testes com Rede Neural Competitiva

Para apresentar a eficiência em relação ao baixo custo computacional do método proposto, foram realizados testes com um outro método de agrupamento que é a rede neural competitiva. Inicialmente, foi realizada a segmentação da imagem 4.45(a), que é uma imagem com imagens de satélite retirada do Prague Dataset do trabalho de Yuan et. al (2015). Para essa comparação foi utilizado o algoritmo de agrupamento implementado nesse trabalho e uma rede neural competitiva. O conjunto de dados foi gerado com  $N = 9000$ , 3000 dados de cada classe, e utilizado nos dois métodos. No método proposto ainda foram definidos  $n = 5$  e  $N_a = 12$ , já na rede competitiva foram realizadas 100 iterações com suas devidas definições. Os resultados obtidos podem ser vistos na Figuras 4.45(b) e 4.45(c).



(a) Imagem Original

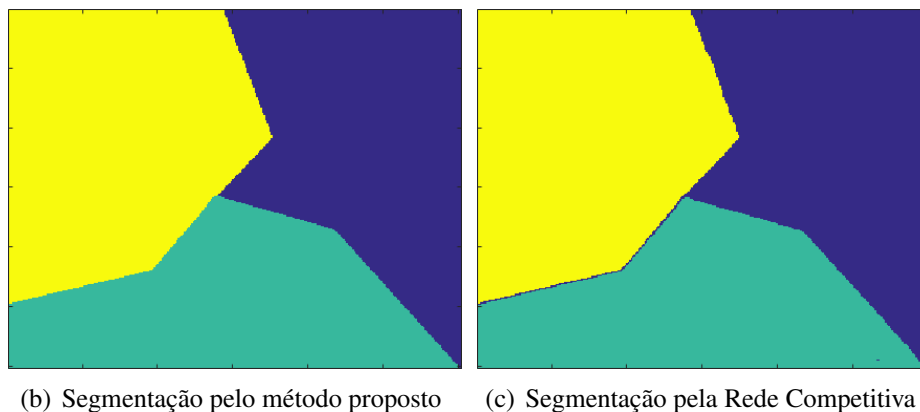


Figura 4.45: Resultados da segmentação para a Figura 4.45(a)

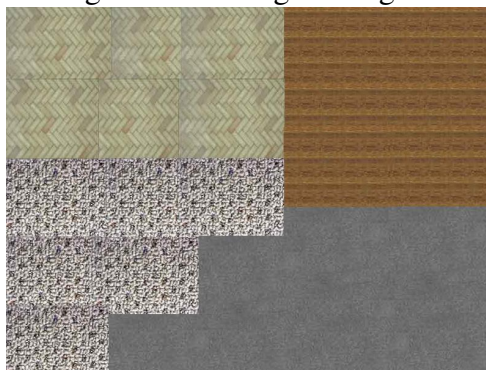
A Tabela 4.1 apresenta o tempo de treinamento dos algoritmos. Lembrando que nesse tempo de treinamento não está incluso o tempo para obter o conjunto de dados inicial e o tempo de classificação da imagem segmentada, que é o mesmo para ambos os casos.

Tabela 4.1: Tempo de Treinamento

Algoritmo	Tempo (s)
Algoritmo Proposto	7,25
Rede Competitiva	491

Um novo teste foi realizado com a imagem 4.46. O conjunto de dados foi gerado com  $N = 8000$ , 2000 dados de cada classe, e utilizado nos dois métodos. No método proposto ainda foram definidos  $n = 5$  e  $N_a = 15$ , já na rede competitiva foram realizadas 100 iterações com uma taxa de aprendizado de 0,0001.

Figura 4.46: Imagem Original



A Tabela 4.2 apresenta o tempo de treinamento dos algoritmos. Lembrando novamente que nesse tempo de treinamento não está incluso o tempo para obter o conjunto

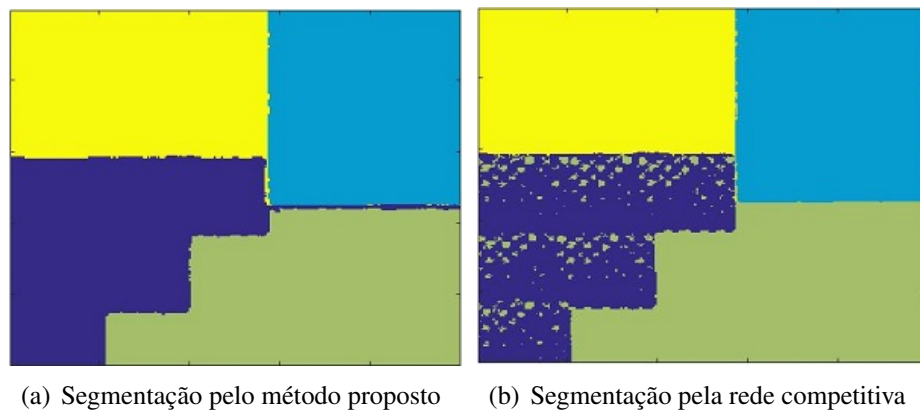


Figura 4.47: Resultados da segmentação da Figura 4.46

de dados inicial e o tempo de classificação da imagem segmentada, que é o mesmo para ambos os casos.

Tabela 4.2: Tempo de Treinamento

Algoritmo	Tempo (s)
Algoritmo Proposto	8,59
Rede Competitiva	914

Um filtro de média foi utilizado nos resultados para eliminar os ruídos presentes e suavizar os contornos das regiões de fronteiras. Em ambos os casos foi utilizada uma janela 10x10 e os resultados são mostrados na Figura 4.48.

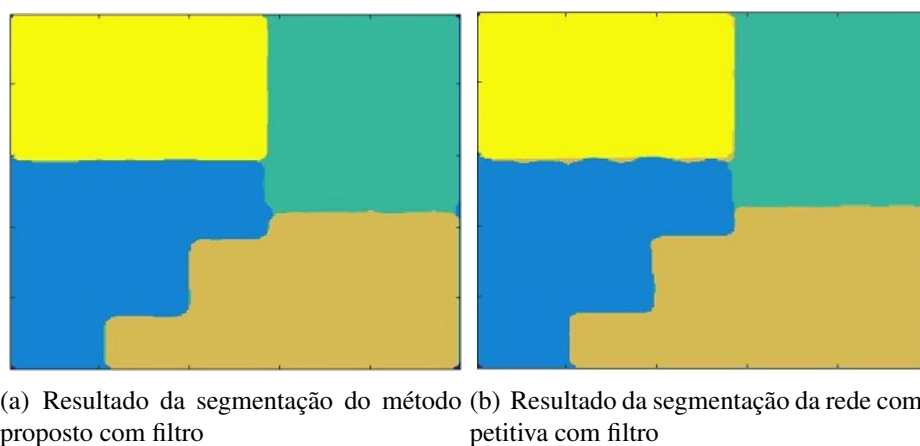
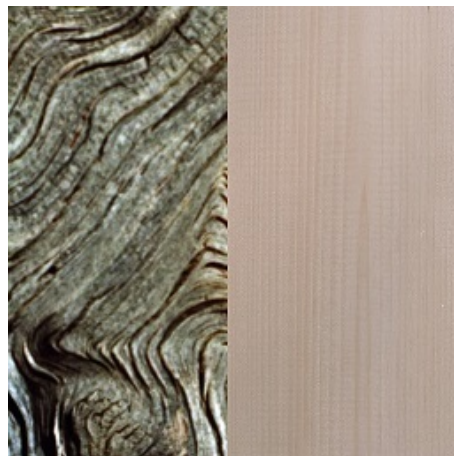


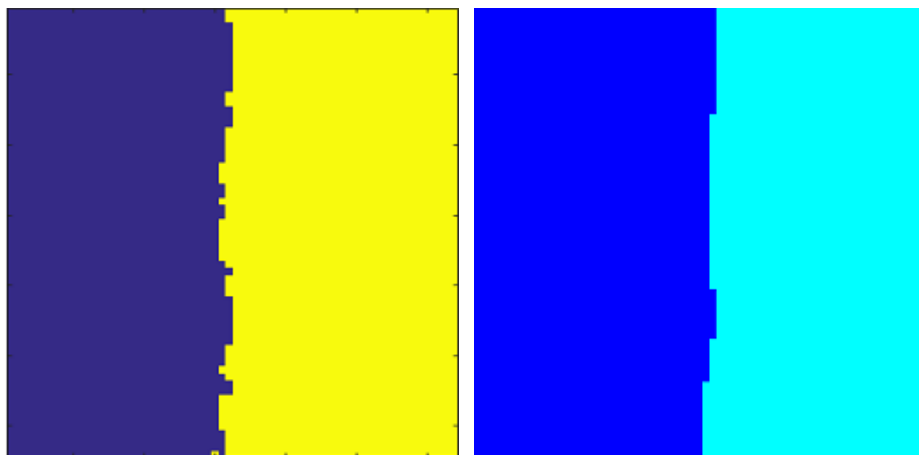
Figura 4.48: Resultados

### 4.0.7 Testes com Filtro de Gabor

Este exemplo compara o tempo de processamento entre o método proposto e outra abordagem, muito conhecida na literatura de segmentação, que quantifica textura utilizando o Filtro de Gabor (JAIN; FARROKHANIA, 1990; WANG; ALL, 2014). Um Toolbox do Matlab foi utilizado sendo o K-Means como método de agrupamento utilizado. O primeiro teste foi realizado com a Figura 4.49(a) que representa texturas diferentes de madeiras, retirada da base de dados VisTex at the MIT Media Lab. Os resultados obtidos podem ser vistos na Figuras 4.49(b) e 4.49(c).



(a) Imagem Original



(b) Segmentação pelo método proposto

(c) Segmentação com Filtro de Gabor

Figura 4.49: Resultado da segmentação da Figura 4.49(a)

A comparação foi realizada entre o tempo de obtenção dos dados pelos algoritmos até

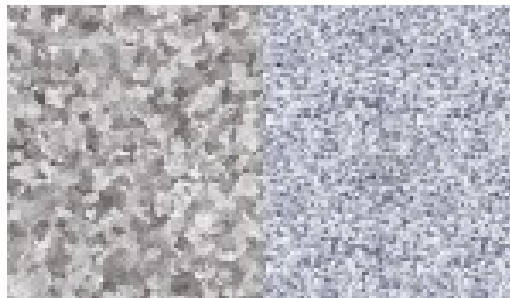
o treinamento dos métodos. Esse resultado pode ser visto na Tabela 4.3.

Tabela 4.3: Tempo

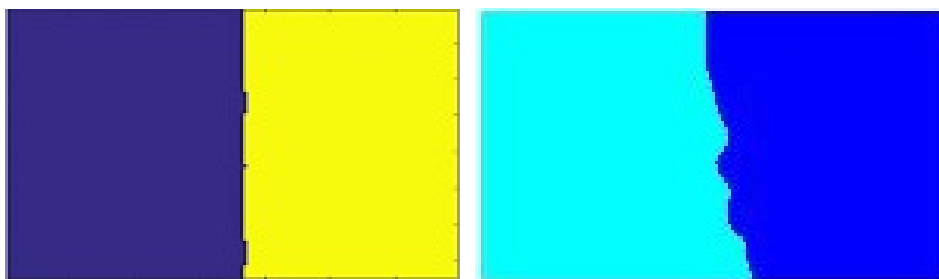
Algoritmo	Tempo (s)
Algoritmo Proposto	14,92
Filtro de Gabor	29,45

O método proposto teve um menor tempo de processamento, com um total de acerto de 93,13%. Já o método com Filtro de Gabor obteve um total de acerto de 88,42%.

O segundo teste foi realizado com a Figura 4.50(a) retirada da base de dados Describable Textures Dataset (DTD) (CIMPOI et al., 2014). Os resultados obtidos podem ser vistos na Figuras 4.50(b) e 4.50(c).



(a) Imagem Original



(b) Segmentação pelo método proposto

(c) Segmentação com Filtro de Gabor

Figura 4.50: Resultados da segmentação da Figura 4.50(a)

A comparação novamente foi realizada entre o tempo de obtenção dos dados pelos algoritmos até o treinamento dos métodos. Esse resultado pode ser visto na Tabela 4.4.

Tabela 4.4: Tempo

Algoritmo	Tempo (s)
Algoritmo Proposto	7,997
Filtro de Gabor	11,1

O método proposto novamente teve um menor tempo de processamento, com um total de acerto de 98,2%. Já o método com Filtro de Gabor obteve um total de acerto de 88,98%. Analisando os resultados de ambos os testes é visível a eficiência do método proposto no resultado final da segmentação e no seu baixo custo computacional.

Um terceiro teste foi realizado com a Figura 4.51(a) retirada da base de dados Prague Dataset do trabalho de Yuan et. all (2015). Os resultados obtidos podem ser vistos na Figuras 4.51(b) e 4.51(c).



(a) Imagem Original

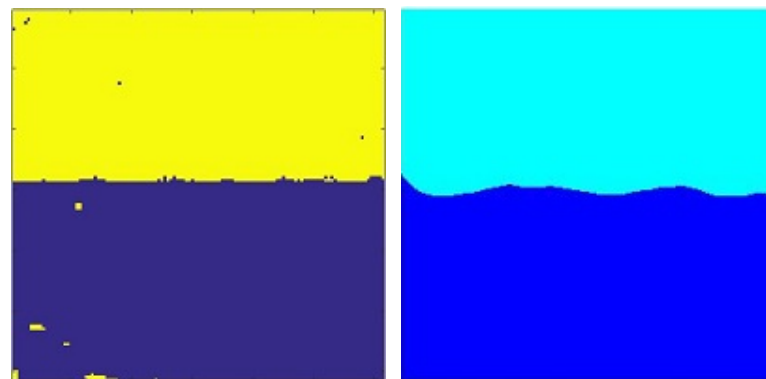
(b) Segmentação pelo método pro-  
posto(c) Segmentação com Filtro de Ga-  
bor

Figura 4.51: Resultados da segmentação da Figura 4.51(a)

A comparação novamente foi realizada entre o tempo de obtenção dos dados pelos algoritmos até o treinamento dos métodos. Esse resultado pode ser visto na Tabela 4.5.

Tabela 4.5: Tempo

Algoritmo	Tempo (s)
Algoritmo Proposto	13.673
Filtro de Gabor	19.126

Analisando os resultados dos testes é visível a eficiência do método proposto no resultado final da segmentação e no seu baixo custo computacional. O método proposto apresentou uma maior porcentagem no acerto, gerando melhor o resultado das fronteiras entre as classes, porém, o método com Filtro de Gabor não gera ruído nas regiões que não pertencem as fronteiras.

---

# Capítulo 5

## Conclusão

---

A segmentação de imagens por textura tem sido amplamente utilizada em análise de imagens médicas, interpretação visual de imagens de sensoriamento remoto, inspeção de qualidade industrial de produtos manufaturados, segmentação de documentos, busca de imagens, sistemas de reconhecimento, entre outros e é uma peça essencial na busca por uma visão computacional que consiga separar regiões com a eficiência e precisão do sistema visual humano.

Neste sentido, este trabalho analisou a aplicação de uma técnica de segmentação de imagens baseada em texturas, utilizando um método de agrupamento de dados para realizar a separação das diferentes regiões. O método de quantificação de textura é baseado na posição espacial dos píxeis e o método de agrupamento é formado pela utilização conjunta de técnicas distintas consolidadas. Todas essas técnicas foram descritas na seção de fundamentação teórica para uma melhor compreensão do algoritmo.

De uma forma geral, o método de agrupamento baseia-se na quantização vetorial do conjunto e na posterior ligação dos clusters formados. Também foram apresentadas sugestões para a definição dos parâmetros empíricos do método. Para o número de clusters auxiliares,  $N_a$ , o valor é estabelecido com base no número de pontos do conjunto, sempre sendo menor do que cinco por cento do número total de pontos. Já para o valor do limiar  $d_t$ , foi apresentada uma heurística que pode ser adotada.

A técnica de segmentação apresentada foi aplicada em imagens que apresentavam diferentes texturas. O conjunto de dados obtidos como entrada para essa técnica consistia de  $N$  pontos extraídos em cada região da imagem com diferentes texturas, utilizando uma janela de tamanho  $M \times M$  para que o valor da textura naquele ponto fosse bem definido pelo cálculo do CVE.

Os resultados obtidos mostram que o método se mostrou robusto, separando bem regiões de texturas diferentes e tonalidades próximas, conseguindo uma boa classificação dos dados. Para verificar a aplicabilidade do método, foram feitas comparações com

outras técnicas disponíveis na literatura, que comprovou a eficácia do mesmo no que foi proposto neste trabalho.

A técnica mostrou eficiência na segmentação de imagens com características bem distintas e cumpriu o objetivo deste trabalho sendo um método de baixo custo computacional e simplicidade no algoritmo.

---

## Referências Bibliográficas

---

- BALAN, A. G. R. *Técnicas de segmentação de imagens aéreas para contagem de população de aves*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, 2003.
- BASTOS, L. O.; LIATSI, P.; CONCI, A. Automatic texture segmentation based on k-means clustering and efficient calculation of co-occurrence features. In: *2008 15th International Conference on Systems, Signals and Image Processing*. [S.l.: s.n.], 2008. p. 141–144. ISSN 2157-8672.
- BENDER, T. C. *Classificação e recuperação de imagens por cor utilizando técnicas de inteligência artificial*. Universidade do Vale do Rio do Sinos, 2003.
- BORGES, R. de O.; SILVA, R. A. A. da; CASTRO, S. S. de. Utilização da classificação por distância euclidiana no mapeamento dos focos de arenização no setor sul da alta bacia do rio araguaia. *Florianópolis: Anais XIII Simpósio Brasileiro de Sensoriamento Remoto*, 2007.
- BORTOLOSSI, H. J. Book;. *Cálculo Diferencial a Várias Variáveis*. [S.l.]: Rio, PUC, 2003. ISBN 85–15-02442-X.
- CHAN, A. B.; VASCONCELOS, N. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 30, n. 5, p. 909–926, 2008.
- CIMPOI, M. et al. Describing textures in the wild. In: *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2014.
- CLÉSIO, F. *Data mining: Medidas de distância*. 2012.
- COLLET, C.; MURTAGH, F. Multiband segmentation based on a hierarchical markov model. *Pattern Recognition*, Elsevier, v. 37, n. 12, p. 2337–2347, 2004.
- DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, JSTOR, p. 1–38, 1977.
- DEZA MICHEL MARIE; DEZA, E. Book;. *Encyclopedia of Distances*. [S.l.]: Springer, Berlin, 2009. ISBN 978-3-642-00233-5.

- DOBLER, J. A. D. Segmentação de texturas em imagens digitais utilizando wavelets redundantes. Universidade Estadual Paulista (UNESP), 2015.
- FILHO, S. da S.; JR, P. D.; MARCOLINO, L. F. V. Mistura de gaussianas: uma abordagem rápida para modelar nuvem de pontos. 2013.
- GONZALEZ, R.; WOODS, R. E. Book;. *Processamento de Imagens Digitais*. [S.l.]: São Paulo: Edgard Blucher, 2000.
- GOODMAN, G. L.; MCMICHAEL, D. W. A gaussian mixture model classifier using supervised and unsupervised learning. In: . [S.l.: s.n.], 1996.
- JAIN, A. K.; FARROKHANIA, F. Unsupervised texture segmentation using gabor filters. In: *1990 IEEE International Conference on Systems, Man, and Cybernetics Conference Proceedings*. [S.l.: s.n.], 1990. p. 14–19.
- JUCA, V.; MELLO, C. Texture-based segmentation using image fidelity indexes. *IEEE Latin America Transactions*, v. 9, n. 3, p. 423–428, June 2011. ISSN 1548-0992.
- LI J. CAI, T. N. J. Z. H. A benchmark for semantic image segmentation. In: *ICME*. [S.l.: s.n.], 2013.
- LINDEN, R. Técnicas de agrupamento. *Revista de Sistemas de Informação da FSMA*, v. 1, n. 4, p. 18–36, 2009.
- LOVERGINE, F. P. et al. Leather inspection by oriented texture analysis with a morphological approach. In: *Proceedings of International Conference on Image Processing*. [S.l.: s.n.], 1997. v. 2, p. 669–671 vol.2.
- MA, W. et al. Image segmentation based on differential immune clone clustering algorithm. *International Journal of Intelligent Computing and Cybernetics*, Emerald Group Publishing Limited, v. 6, n. 1, p. 83–102, 2013.
- MARTINS, A. et al. A new clustering separation measure based on negentropy. *Journal of Control, Automation and Electrical Systems*, Springer US, v. 1, n. 26, p. 28–45, 2015.
- MARTINS, A. d. M. Contribuições aos processos de clustering com base em métricas não-euclidianas. Universidade Federal do Rio Grande do Norte, 2005.
- MASCARO, A. A. et al. Mammographic images segmentation using texture descriptors. In: *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. [S.l.: s.n.], 2009. p. 3653–3653. ISSN 1094-687X.
- MORIMITSU, H.; TUESTA, E. F. Análise comparativa das abordagens de estimativa do modelofastica por maximização da negentropia e da verossimilhança. 2015.
- NUNES, E. O.; CONCI, A. Técnicas de descrição regional baseadas em textura. *2º Congresso Temático de Aplicações de Dinâmica e Controle da Sociedade Brasileira de Matemática Aplicada e Computacional (SBMAC)*., v. 2, June 2003.

- NUNES, E. O.; CONCI, A. Texture segmentation and edge location of regions in multiband images. *IEEE Latin America Transactions*, v. 5, n. 3, p. 184–191, June 2007. ISSN 1548-0992.
- NUNES Éldman de O. *Segmentação por Textura em Imagens Multibandas*. 2006.
- OLIVEIRA, C. M.; ALL et. Aplicação de misturas gaussianas na análise e modelagem de tráfego voip. 2011.
- OLIVEIRA, P. R. *Métodos Auto-Organizáveis para Segmentação de Imagens*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação - ICMC - USP, 2004.
- OLIVEIRA, T. B. S. d. *Clusterização de dados utilizando técnicas de redes complexas e computação bioinspirada*. Tese (Doutorado) — Universidade de São Paulo, 2008.
- PEREIRA, R. L.; ALL et. Classificação de modos de operação em sistemas de bombeio mecânico baseado em pca e k-means. 2014.
- PERMUTER, H.; FRANCOS, J.; JERMYN, I. H. Gaussian mixture models of texture and colour for image database retrieval. In: *IEEE. Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*. [S.l.], 2003. v. 3, p. III–569.
- RANDEN, T.; HUSOY, J. H. Filtering for texture classification: a comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 21, n. 4, p. 291–310, Apr 1999. ISSN 0162-8828.
- SCHWARTZ, W. R.; PEDRINI, H. Segmentação de imagens de terrenos baseada na associação de características de texturas com dependência espacial modelada por campo aleatório de markov. In: . [S.l.]: Goiania, Brasil, 2005. p. 4311–4318.
- TIPPING, M. E. Deriving cluster analytic distance functions from gaussian mixture models. In: *IET. Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*. [S.l.], 1999. v. 2, p. 815–820.
- VASCONCELOS, C. H. Mapeamento do uso e cobertura da terra a partir da segmentação e classificação de imagens. *Acta Amazônica*, SciELO Brasil, v. 34, n. 3, p. 487–493, 2004.
- VIOLATO, R. P. V.; ALL et. Mapeamento do uso e cobertura da terra a partir da segmentação e classificação de imagens. *Congresso Ibero-Latino-Americano de Métodos Computacionais em Engenharia*, (CILAMCE), 2009.
- WANG, M. Y.; ALL at. Study on multi-layered classification of image. *Control Engineering and Information Systems: Proceedings of the 2014 International Conference on Control Engineering and Information Systems (ICCEIS)*, 2014.

YAN, Y.; SHEN, Y.; LI, S. Unsupervised color-texture image segmentation based on a new clustering method. In: IEEE. *New Trends in Information and Service Science, 2009. NISS'09. International Conference on*. [S.l.], 2009. p. 784–787.