



Allan Alves de Moura

**Segmentação de banhistas utilizando algoritmos
de agrupamento com seleção automática do
número de grupos em regiões litorâneas**

Recife

2019

Allan Alves de Moura

Segmentação de banhistas utilizando algoritmos de agrupamento com seleção automática do número de grupos em regiões litorâneas

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE
Departamento de Computação
Curso de Bacharelado em Ciência da Computação

Orientador: Valmir Macario Filho

Recife
2019

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

M929s Moura, Allan Alves de.
Segmentação de banhistas utilizando algoritmos de agrupamento
com seleção automática do número de grupos em regiões litorâneas
/ Allan Alves de Moura. - Recife, 2019.
107 f.: il.

Orientador(a): Valmir Macario Filho.
Trabalho de Conclusão de Curso (Graduação) – Universidade
Federal Rural de Pernambuco, Departamento de Computação,
Recife, BR-PE, 2019.
Inclui referências.

1. Segmentação de imagens 2. Número de grupos automático
3. Detecção de banhistas 4. Sistema de cores I. Macario Filho,
Valmir, orient. II. Título

CDD 004




MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

<http://www.bcc.ufrpe.br>

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por Allan Alves de Moura às 16 horas do dia 14 de janeiro de 2019, no Auditório do CEAGRI-02 – Sala 07, como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado “**Segmentação de banhistas utilizando algoritmos de agrupamento com seleção automática do número de grupos em regiões litorâneas**”, orientado por Valmir Macario Filho e aprovado pela seguinte banca examinadora:



Valmir Macario Filho

DC/UFRPE



Danilo Ricardo Barbosa de Araújo
DC/UFRPE

Agradecimentos

Agradeço a todos aqueles que de alguma forma influenciaram o meu caminho até onde cheguei. Agradeço a minha mãe, Tania Lucia, e meu pai, Ubirajara Moura por me apoiarem durante essa jornada, pela compreensão nos momentos mais difíceis e pelos momentos de alegria compartilhado nas vitórias.

Agradeço a todos os amigos que participaram na construção da minha formação, troca de experiências, conselhos e muito mais.

Agradecer aos professores com quem aprendi tudo durante o curso, mesmo nos momentos difíceis, pois estes são de maior crescimento, afinal, bom marinho só se faz em mar de tempestades. Agradecimento especial para meu orientador, Valmir Macário, aquele que me acompanhou na maior parte do curso.

”Julgue seu sucesso pelas coisas que você teve que renunciar para conseguir.”

Dalai-lama

Resumo

O crescente aumento de ataques de tubarões vem assustando a população das regiões litorâneas, impossibilitando a tomada de banho em certos locais. Em caso de ataque, muitas vezes uma ação para salvar a vítima é tomada somente após o ocorrido do incidente, em que um guarda-vidas tentará socorrer-la. Para mitigar estes acontecimentos, foi pensado em uma ferramenta de auxílio aos guarda-vidas que permitissem aos mesmos agir antes que acidentes pudessem ocorrer, avisando o profissional caso algum banhista ultrapasse uma determinada zona de segurança. O primeiro passo para realização desta ferramenta é a segmentação das imagens de praia em busca de agrupar partes com características visualmente semelhantes a fim de encontrar as pessoas dentro do mar. Para isso, o objetivo deste trabalho é estudar e encontrar um bom algoritmo de segmentação de imagens com seleção automática do número de grupos para evitar a necessidade do controle de parâmetros a fim de separar a foto em regiões ou segmentos. Este algoritmo será utilizado para a implementação da primeira etapa desta ferramenta de auxílio aos guarda-vidas em busca de regiões da imagem que representem banhistas. Técnicas de pré-processamento de imagens como a remoção da faixa de praia foram avaliadas, como também a avaliação de sistemas de cores diferentes na representação da imagem a fim de realizar a comparação entre todos esses fatores. A combinação dos algoritmos e sistemas de cores foram avaliados com e sem a remoção da faixa de praia. Os algoritmos analisados foram: Hierárquico aglomerativo, hierárquico divisivo, *X-means*, *Auto group segmentation* e segmentação automática de imagens coloridas. Todos eles foram aplicados pra três tipos diferentes de vetores de características, compostos por sistema de cor RGB(*red, green e blue*), LAB e a combinação de RGB + LAB. O resultado mais promissor, após avaliação visual das imagens, comportamento dos algoritmos e resultados do índice de *Dunn*, foi obtido utilizando o algoritmo de segmentação de imagens coloridas com vetor de características composto de RGB + LAB, totalizando para o índice de *Dunn* 1.5245 de média para todas as imagens, aplicando a remoção de praia depois da execução do algoritmo.

Palavras-chave: Segmentação de imagens, Sistema de cores, número de grupos automático, Detecção de banhistas.

Abstract

The increasing number of shark attacks has been frightening people that lives in coastal areas, making it impossible to bath in certain places. In an attack situation, most of the time a course of action to save the victim's life is only taken after the incident already has occurred, which a lifeguard tries to help her. An auxiliary tool for lifeguards was thought in order to mitigate these events and allow the lifeguards to act before the incident happens, alerting the professional if someone tries to surpass a delimited zone. The first step to bring this auxiliary tool to life is the technique of image segmentation on beach photos in search for regions that share visual similarities in order to find people inside the sea. Therefore the objective of this work is to study and find a good image segmentation algorithm capable of automatically selecting the best number of groups without the parameter control necessity. The selected algorithm will be used to implement the first phase of the lifeguard auxiliary tool in search for image regions that represent bathers. Image pre-processing techniques like beach removal were evaluated, as well as characteristics vectors selection used to compare elements. The combination between algorithms and characteristics vectors were evaluated with and without beach removal. The analyzed algorithms were: hierachical aglomerative, hierarchival divisive, X-means, auto group segmentation and automatic colored image segmentation. All of them were applied to three different characteristics vectors composed by the color system RGB (red, green and blue), LAB and the combination of RGB + LAB. The most promising result, taking into account the visual analysis and the algorithm comportamental analysis, was obtained by the automatic color image segmentation with RGB + LAB, composed characteristics vector, with value of 1.5245 extracted from Dunn's index, using the beach removal as post-processing technique.

Keywords: Image Segmentation, Color system, automatic number of groups, bathers detection.

Lista de ilustrações

Figura 1 – Pessoas que atravessassem a linha pontilhada seriam considerados em risco.	17
Figura 2 – Representação de uma imagem em tons de cinza e seus valores associados.	26
Figura 3 – Imagem original (esquerda), imagem após executar um processo de realce de nível baixo (centro), imagem resultado após dividir a imagem em regiões (direita).	27
Figura 4 – Imagem digital representando um conjunto de <i>pixels</i>	28
Figura 5 – Representação do espaço de cores XYZ, cada coordenada possui uma cor associada.	28
Figura 6 – (A) sólido de cor do sistema RGB, (B) Sistema de coordenadas que representam o subespaço RGB.	29
Figura 7 – Imagem RGB e seus canais divididos em <i>Red</i> , <i>Green</i> e <i>Blue</i>	30
Figura 8 – Exemplos de cores em formato RGB, o primeiro número representa o canal <i>Red</i> , o segundo <i>Green</i> e o terceiro <i>Blue</i>	31
Figura 9 – Sólido de cor do sistema HSV.	32
Figura 10 – Sólido de cor do sistema CIELAB.	33
Figura 11 – Exemplo de imagem em RGB convertida para tons de cinza.	34
Figura 12 – Conjunto de tuplas de uma imagem em tons de cinza com 256 tons de cor.	36
Figura 13 – Histograma e sua imagem em tons de cinza.	37
Figura 14 – Imagem original em RGB.	37
Figura 15 – Histograma de imagem RGB juntos e separados.	38
Figura 16 – Imagem original (A), Imagem após executar um filtro passa-baixa (B), imagem após executar filtro passa-alta (C).	39
Figura 17 – Representação da máscara que sobrepõe a imagem original e executa a filtragem.	40
Figura 18 – Exemplo de um filtro de mediana de tamanho 3x3.	41
Figura 19 – Remoção de ruído de sal e pimenta através do filtro de mediana.	41
Figura 20 – Imagem resultante e seus diferentes valores de sigma para o filtro gaussiano.	42
Figura 21 – Execução de uma operação logica OR e seu resultado.	43
Figura 22 – Execução de uma operação logica AND e seu resultado.	43
Figura 23 – Exemplo de binarização de uma imagem em tons de cinza.	44
Figura 24 – Limiarização para uma imagem em tons de cinza.	44
Figura 25 – Contorno de um componente conexo.	45

Figura 26 – Segmentação de imagem com apenas dois grupos representados em binário.	47
Figura 27 – Imagem Segmentada utilizando a representação colorida.	47
Figura 28 – Um conjunto de dados e agrupamento baseados na característica de forma e cor.	48
Figura 29 – Representação de um agrupamento hierárquico em forma de dendrograma.	49
Figura 30 – Imagem binarizada de um dos grupos (esquerda) e resultado após extrair contornos (direita)	52
Figura 31 – Máscara binária (esquerda) e região da imagem original (direita). . .	52
Figura 32 – Ilustração dos acontecimentos no algoritmo <i>X-means</i>	56
Figura 33 – Exemplo de contagem de uma GLCM levando em consideração os <i>Pixels</i> na horizontal.	61
Figura 34 – Exemplo do GLCM com seus máximos locais na diagonal destacados e a contagem dos máximos locais.	61
Figura 35 – Imagens com (direita) e sem (esquerda) ruídos.	65
Figura 36 – Imagens com (esquerda) e sem (direita) arrecifes de corais.	65
Figura 37 – Imagens com sobreposição.	65
Figura 38 – Imagens com maré baixa (direita) e alta (esquerda).	66
Figura 39 – Imagens com areia molhada.	66
Figura 40 – Imagens com alta (esquerda) e baixa(direita) incidência de luz. . . .	66
Figura 41 – Imagens com longe (esquerda) e próximo (direita) distanciamento. .	67
Figura 42 – Imagem rotulada.	67
Figura 43 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado para a combinação de RGB e 20 sementes.	74
Figura 44 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas (Direita) para um resultado ruim, misturando regiões da água com as pessoas.	75
Figura 45 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado para a combinação de LAB e 30 sementes.	75
Figura 46 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado para a combinação de RGB e 40 sementes.	76
Figura 47 – Imagem Original (esquerda), imagem totalmente segmentada (direita), grupo de cor verde melhor representou os banhistas para GLCM 20 níveis e sistema de cor RGB.	77

Figura 48 – Imagem Original (esquerda), imagem totalmente segmentada (direita), grupo de cor vermelha melhor representou os banhistas para GLCM 20 níveis e sistema de cor RGB.	78
Figura 49 – Imagem Original (esquerda), imagem totalmente segmentada (direita), grupo de cor vermelha melhor representou os banhistas para GLCM 20 níveis e sistema de cor RGB.	78
Figura 50 – Imagem Original (esquerda), imagem totalmente segmentada (direita), exemplo de resultado ruim para GLCM 20 níveis e sistema de cor RGB.	79
Figura 51 – Imagem Original (esquerda), imagem totalmente segmentada (direita), grupo de cor rosa melhor representou os banhistas para GLCM 40 níveis e sistema de cor LAB.	79
Figura 52 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado do algoritmo hierárquico divisivo com vetor de característica RGB.	81
Figura 53 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um razoável resultado do algoritmo hierárquico divisivo com vetor de características RGB.	81
Figura 54 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado do algoritmo <i>Auto group segmentation</i> com vetor de características LAB.	82
Figura 55 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado do algoritmo <i>xmeans</i> com vetor de características combinado de RGB e LAB.	82
Figura 56 – Imagem original (esquerda), máscara binária (direita), Imagem Resultando da operação <i>AND</i> (baixo).	83
Figura 57 – Imagem original (esquerda), imagem resultante (direita).	84
Figura 58 – Imagem original (esquerda), imagem resultante (direita).	84
Figura 59 – Imagem original (esquerda), imagem resultante (direita).	84
Figura 60 – (1) Imagem Original, Melhores grupos para 20 sementes: (2) LAB, (3) RGB, (4) RGB + LAB.	86
Figura 61 – (1) Imagem Original, Melhores grupos para 30 sementes: (2) LAB, (3) RGB, (4) RGB + LAB.	86
Figura 62 – (1) Imagem Original, Melhores grupos para 40 sementes: (2) LAB, (3) RGB, (4) RGB + LAB.	87
Figura 63 – (1) Imagem Original, Melhores grupos entre categorias:(2) 20 sementes - RGB + LAB, (3) 30 sementes - RGB + LAB, (4) 40 sementes - RGB e LAB (ambos produziram grupos muito parecidos).	87

Figura 64 – (1) Imagem Original, Melhores segmentações para 20 níveis de cinza: (2) LAB representado pelo grupo azul, (3) RGB representado pelo grupo azul, (4) RGB + LAB representado pelo grupo amarelo. . . .	89
Figura 65 – (1) Imagem Original, Melhores segmentações para 30 níveis de cinza: (2) LAB representado pelo grupo rosa, (3) RGB representado pelo grupo amarelo, (4) RGB + LAB representado pelo grupo rosa. . . .	89
Figura 66 – (1) Imagem Original, Melhores segmentações para 40 níveis de cinza: (2) LAB representado pelo grupo verde, (3) RGB representado pelo grupo verde, (4) RGB + LAB representado pelo grupo amarelo. . . .	90
Figura 67 – (1) Imagem Original, Melhores grupos entre categorias:(2) 20 níveis de cinza - RGB representado pelo grupo vermelho, (3) 30 níveis de cinza - RGB + LAB representado pelo grupo verde, (4) 40 níveis de cinza - RGB representado pelo grupo vermelho.	90
Figura 68 – <i>X-means</i> - (1) Imagem Original, Melhores grupos que representaram as pessoas: (2) LAB, (3) RGB, (4) RGB + LAB.	92
Figura 69 – <i>Auto Group Segmentation</i> - (1) Imagem Original, Melhores grupos que representaram as pessoas: (2) LAB, (3) RGB, (4) RGB + LAB.	92
Figura 70 – <i>Hierárquico Divisivo</i> - (1) Imagem Original, Melhores grupos que representaram as pessoas: (2) LAB, (3) RGB, (4) RGB + LAB.	93
Figura 71 – Remoção da faixa de praia como pós-processamento para imagem totalmente segmentada, grupo de pessoas de cor verde.	99
Figura 72 – Remoção da faixa de praia como pós-processamento para imagem do grupo de pessoas.	99

Lista de tabelas

Tabela 1 – Tabela apresentando os resultados da operação lógica OU (OR) e E (AND) entre dois <i>pixels</i> binários.	42
Tabela 2 – Experimento A-1 - Resultados de média para o índice de <i>Dunn</i> variando o número de sementes e os vetores de características.	73
Tabela 3 – Experimento A-2 - Resultados de média para o índice de <i>Dunn</i> variando o número de níveis de tons de cinza da GLCM e os vetores de características.	76
Tabela 4 – Experimento A-3 - Resultados de média para o índice de <i>Dunn</i> variando os vetores de características para os algoritmos de <i>X-means</i> , Hierárquico divisivo e <i>Auto group segmentation</i> , Hierárquico Aglomerativo e Segmentação Automática Colorida	80
Tabela 5 – Experimento B - Resultados de média para o índice de <i>Dunn</i> variando o número de sementes e os vetores de características.	85
Tabela 6 – Experimento B-2 - Resultados de média para o índice de <i>Dunn</i> variando o número de níveis de tons de cinza da GLCM e os vetores de características com a remoção da faixa de praia.	88
Tabela 7 – Experimento B-3 - Resultados de média para o índice de <i>Dunn</i> variando os vetores de características para os algoritmos de <i>X-means</i> , Hierárquico divisivo e <i>Auto group segmentation</i> , com adição dos melhores resultados dos experimentos B-1 e B-2.	91
Tabela 8 – Resultados de média para o índice de <i>Dunn</i> do experimento A para o vetor de características RGB.	94
Tabela 9 – Resultados de média para o índice de <i>Dunn</i> do experimento A para o vetor de características LAB.	94
Tabela 10 – Resultados de média para o índice de <i>Dunn</i> do experimento A para o vetor de características RGB + LAB.	95
Tabela 11 – Resultados de média para o índice de <i>Dunn</i> do experimento B para o vetor de características RGB.	96
Tabela 12 – Resultados de média para o índice de <i>Dunn</i> do experimento B para o vetor de características LAB.	96
Tabela 13 – Resultados de média para o índice de <i>Dunn</i> do experimento B para o vetor de características RGB + LAB.	96
Tabela 14 – Comparação dos Resultados de média para o índice de <i>Dunn</i> sem e com remoção de praia para vetor de características RGB.	97
Tabela 15 – Comparação dos Resultados de média para o índice de <i>Dunn</i> sem e com remoção de praia para vetor de características LAB.	98

Tabela 16 – Comparação dos Resultados de média para o índice de *Dunn* sem e com remoção de praia para vetor de características RGB + LAB. . 98

Sumário

	Lista de ilustrações	8
1	INTRODUÇÃO	16
1.1	Problema de Pesquisa	18
1.2	Justificativa	18
1.3	Objetivos	25
1.4	Estrutura do trabalho	25
2	PROCESSAMENTO DE IMAGEM	26
2.1	Introdução	26
2.2	Pixel	27
2.3	Sistema de cores	28
2.3.1	Modelo RGB de cores	29
2.3.2	Modelo HSV de cores	31
2.3.3	Modelo CIELAB de cores	32
2.3.4	Conversão de RGB para tons de cinza	33
2.3.5	Conversão de RGB para HSV	34
2.3.6	Conversão de RGB para XYZ	35
2.3.7	Conversão de RGB para CIELAB	35
2.4	Histograma	36
2.5	Filtro espacial	38
2.5.1	Filtro de Mediana	40
2.5.2	Filtro Gaussiano	41
2.6	Operações lógicas	42
2.7	Limiarização	43
2.8	Contornos	44
2.9	Segmentação de imagens	46
2.10	Agrupamento	47
2.10.1	Algoritmos hierárquicos	48
2.10.2	Algoritmo <i>K-means</i>	49
3	SEGMENTAÇÃO DE IMAGENS DE PRAIA	51
3.1	Remoção da faixa de praia	51
3.2	Criação do vetor de características	54
3.3	Segmentação por algoritmos de agrupamento com escolha automática do número de grupos	54

3.3.1	X-Means	54
3.3.1.1	Bayesian Information Criterion	57
3.3.2	Determinação do número de <i>clusters</i> utilizando <i>k-means</i> e aplicação em segmentação de imagens coloridas	58
3.3.3	Segmentação automática de imagens coloridas	60
4	AVALIAÇÃO EXPERIMENTAL	64
4.1	Base de Dados	64
4.2	Metodologia experimental	68
4.2.1	Experimento A	68
4.2.1.1	Experimento A-1	68
4.2.1.2	Experimento A-2	69
4.2.1.3	Experimento A-3	69
4.2.2	Experimento B	69
4.2.2.1	Experimento B-1	70
4.2.2.2	Experimento B-2	70
4.2.2.3	Experimento B-3	70
4.3	Ambiente experimental	70
4.4	Métricas de validação	71
5	RESULTADOS	73
5.1	Resultados dos experimentos A	73
5.1.1	Experimento A-1	73
5.1.2	Experimento A-2	76
5.1.3	Experimento A-3	79
5.2	Resultados dos experimentos B	82
5.2.1	Experimento B-1	85
5.2.2	Experimento B-2	88
5.2.3	Experimento B-3	91
5.3	Discussão	93
6	CONCLUSÃO E TRABALHOS FUTUROS	102
6.1	Trabalhos futuros	103
6.2	Contribuições	104
	REFERÊNCIAS	105

1 Introdução

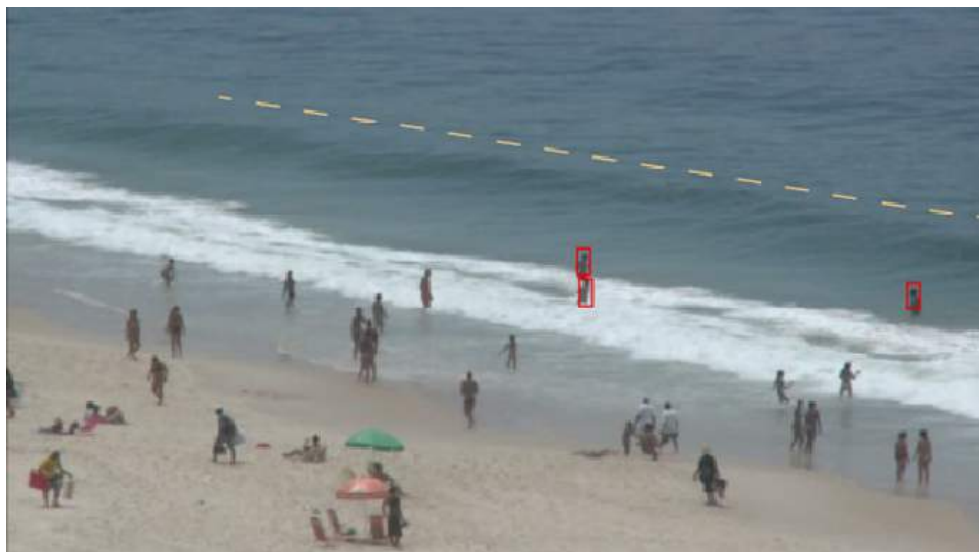
Devido ao grande crescimento tecnológico, o monitoramento de pessoas em espaços públicos se tornou uma atividade constante, tanto por motivos de segurança, como também para análise dos indivíduos. A supervisão dessas imagens normalmente é realizada através de uma pessoa intermediária que irá averiguar a situação registrada. Esta forma de observação não se difere da utilizada pelos guarda-vidas em regiões de risco com eminentes ataques de tubarões ou afogamento de banhistas.

Desde a década de 1990, após o início do funcionamento em pleno vapor do porto de Suape, notou-se um aumento frequente dos ataques de tubarões nas regiões litorâneas do Recife (VASCONCELOS, 2018). De acordo com Fábio Hanzin, biólogo da Universidade Federal de Pernambuco, existe o reconhecimento de que tubarões tendem a seguir grandes embarcações, sendo este um dos principais motivos pelo crescente número de ataques. Além disso, existem outros fatores que podem explicar o problema dos ataques, como a crescente pesca de camarão, havendo despejo de restos da pescaria no mar, topografia do relevo submarino e também certas condições climáticas (VASCONCELOS, 2018).

O Jornal do comércio (2018) afirma que desde 1992, o litoral Pernambucano já soma cerca de 65 ataques de tubarão registrados, sendo 25 fatais. Muitas vezes uma ação para salvar a vítima é tomada somente após o ocorrido do incidente, em que um guarda-vidas tentará socorrê-la. A extensão das regiões litorâneas também dificulta a ação dos guarda-vidas, pois o litoral Pernambuco possui cerca de 187 quilômetros, onde 32 são considerados inapropriados para o banho.

Com a intenção de aprimorar a percepção destes profissionais, além de aumentar o alcance do monitoramento das regiões litorâneas, afim de evitar ainda mais acidentes, é necessário o auxílio de ferramentas para identificar mais facilmente as situações de risco (Figura 1). Com o suporte da ferramenta, o guarda-vidas poderá ser alertado caso algum indivíduo entre em uma zona perigosa, poupando o banhista de uma eventual fatalidade. Na criação de tal ferramenta, um dos principais passos para a detecção de elementos em uma imagem é a sua segmentação em busca de regiões que representam banhistas.

Figura 1 – Pessoas que atravessem a linha pontilhada seriam considerados em risco.



Fonte – O Autor

A segmentação através de agrupamento de dados tem como base encontrar elementos na imagem que possuam certo tipo de característica visual similar, determinada por uma métrica de cálculo de distância, e agrupá-los em conjuntos, de forma que elementos parecidos pertençam ao mesmo grupo e elementos distintos estejam em grupos diferentes. Separando os elementos, são criados um ou mais conjuntos que possam ser analisados de forma individual. O agrupamento de dados possui diversas aplicabilidades, entre elas: pesquisa de mercado (PUNJ; STEWART, 1983), reconhecimento de voz (GORRIZ et al., 2006), classificação de tempo (CLIFTON; LUNDQUIST, 2012) etc.

Um dos desafios da segmentação utilizando a técnica de agrupamento de dados é a dificuldade para encontrar uma certa quantidade de números de grupos, pois é preciso ter conhecimento prévio do problema, no entanto, na maioria dos casos reais, este valor não é encontrado facilmente. Portanto, o objetivo deste projeto é analisar e propor diferentes algoritmos de agrupamento com seleção automática do número de grupos na busca por segmentos de imagens que possam representar banhistas em regiões perigosas. Tendo em mente a premissa de evitar ataques de tubarão e acidentes nas regiões litorâneas. Este projeto servirá como uma primeira etapa de uma ferramenta de detecção e monitoramento de banhistas que auxiliará os guarda-vidas na proteção das pessoas.

Os algoritmos apresentados no decorrer do trabalho são: Hierárquico aglomerativo, hierárquico divisivo (MAIMON; ROKACH, 2010), *auto group segmentation* (RAY; TURI, 2000), *X-means* (PELLEG; MOORE, 2000) e segmentação automática de imagens coloridas (KALAIVANI; CHITRAKALA, 2014). Além disso, uma técnica de remo-

ção da faixa de praia também foi aplicada e analisada nos resultados.

1.1 Problema de Pesquisa

De que forma é possível aplicar algoritmos de segmentação de imagens baseados em agrupamentos que tenham seleção do número de grupos automática, afim de evitar a necessidade de entradas do usuário, para a detecção de banhistas em regiões litorâneas com risco de ataques de tubarões?

Para resolver este problema, várias atividades são necessárias, elas precisam ser devidamente analisadas e testadas de diversas maneiras. Uma das atividades é a geração de vetores característica, este depende diretamente das possíveis formas que as imagens podem ser representadas, conseqüentemente tornando tal representação uma atividade importante. Além disso, a escolha, análise e implementação de diversos algoritmos em busca dos possíveis candidatos para utilização é a base para a resolução do problema.

1.2 Justificativa

A maioria dos métodos de segmentação de imagens são classificados em três categorias, sendo elas métodos baseados em bordas, métodos baseados em regiões, e por fim os métodos de agrupamento. A segmentação baseada em bordas trabalha com a detecção de limites entre diferentes regiões da imagem. Já os métodos baseados em regiões focam em juntar regiões com características similares para formar um novo grupo. Por fim, os métodos de agrupamento separam a imagem em regiões que possuem características similares, como também a separação em diferentes grupos quando não possuem muita semelhança (HOSANG; BENENSON; SCHIELE, 2014).

Das três técnicas citadas, duas delas se destacam mais devido aos resultados (HOSANG; BENENSON; SCHIELE, 2014): Abordagem baseada em detecção de bordas (ARBELAEZ et al., 2014) e a abordagem baseada em busca seletiva que tem como base os métodos de agrupamento (SANDE et al., 2011). Devido à naturalidade dos algoritmos de agrupamento para formar regiões similares, estes apresentam bons resultados para a aplicação da segmentação das imagens.

O número de grupos é um dos parâmetros mais comuns e complexos a serem definidos em algoritmos de agrupamento clássicos como o *K-means* e o *Fuzzy C-Means*. Na segmentação de uma imagem, não é simples estabelecer o número de regiões diferentes presentes, além disso, por conta da grande quantidade de *pixels* em uma imagem de alta resolução, escolher o número de regiões ou grupos para um algoritmo de agrupamento não é uma tarefa trivial, dependendo da abordagem do al-

goritmo, pode até aumentar consideravelmente a duração da execução. Devido a isso, neste trabalho serão estudados algoritmos de agrupamento com seleção automática do número de grupos contidos em cada imagem, pois estes possuem uma maior facilidade de adaptação dependendo dos elementos contidos na imagem.

Ao aplicar tais técnicas de segmentação de imagem baseado em agrupamentos, o objetivo é separar as regiões contidas na imagem em em grupos de elementos. Desta forma, o ideal é que um dos agrupamentos possa representar a maior quantidade de elementos caracterizados como possíveis banhistas no mar. Após a imagem ser dividida em regiões através da segmentação, cada agrupamento estará pronto para passar por um classificador, permitindo a um algoritmo classificar se aquela região da imagem é uma pessoa ou não.

Uma proposta que visa automatizar a seleção do número de grupos é o *X-Means* (PELLEG; MOORE, 2000). Este requer como parâmetro de execução o número máximo de possíveis grupos que existirão na imagem, tendo como resposta o melhor valor para K (número de grupos) determinado por um critério do modelo de avaliação. O algoritmo consiste em iniciar a análise dos agrupamentos com uma quantidade pequena de grupos e tentar dividir estes em novos conjuntos, caso estes grupos separados possuam um valor de score melhor que o anterior, o antigo é substituído, dando lugar aos dois novos grupos, aumentando a quantidade de agrupamentos até que atinja o número máximo de grupos permitidos, ou caso os conjuntos atuais não se dividam.

O trabalho proposto por Ray e Turi (2000) tem o foco em analisar imagens coloridas, este propõe uma nova forma de medição para calcular a validação de um grupo de elementos. A análise desta validação leva em consideração o quão compacto é um agrupamento e também quão distante é este conjunto dos demais. Utilizando esta métrica da compactação em razão da distância entre os grupos, a intenção é diminuir o valor resultante para uma melhor formação destes agrupamentos. O algoritmo inicia todos os elementos como um único conjunto e então vai dividindo os agrupamentos baseado na métrica de validação até que atinja um número máximo de segmentos na imagem. Cada solução encontrada será analisada de forma individual e a que apresentar melhor resultado será escolhida como solução resposta.

Outro algoritmo que tem como estratégia a seleção automática do número de grupos é o *K*-means* (CHEUNG, 2003), o algoritmo consiste em dois passos distintos. O primeiro passo é um pré-processamento que tem como objetivo alocar uma semente (centro de grupo) para cada grupo. O segundo passo consiste em ajustar as sementes adaptativamente por uma regra de aprendizagem, baseada em sanções positivas e negativas, que penalizam a chance de vitória de cada semente rival (centro de grupo que não recebeu um novo elemento ao grupo) nas competições subsequentes

enquanto aprimora a semente vencedora (recebeu um novo elemento ao grupo) para se adaptar ao dado de entrada, desta forma algumas sementes ficarão sem elementos no grupo e serão descartadas, sobrando apenas aquelas na qual formam grupos de fato.

A proposta de Hamerly e Elkan (2004) se baseia no algoritmo *K-means*, criando uma forma de descobrir o possível número de grupos para um determinado conjunto de dados. A quantidade de conjuntos é um dos principais problemas encontrados na literatura referente a agrupamentos, pois dependendo dos dados o número de grupos pode variar. Escolher um número exato para a quantidade de agrupamentos não é uma tarefa trivial, é necessário um grande conhecimento prévio sobre o problema, em que na maioria dos casos a informação não existe. Contornando estes problemas, Hamerly e Elkan (2004) propuseram o algoritmo denominado *G-means*, consistindo de um teste estatístico para analisar os conjuntos de dados e se certificar se os mesmos possuem uma distribuição gaussiana, caso contrário o agrupamento será dividido em dois novos grupos. Após uma rodada de análises e possíveis particionamentos, é aplicado o algoritmo *K-means* para melhorar e refinar a solução atual. Devido a aplicação do teste de distribuição gaussiana, o algoritmo *G-means* assume que o conjunto de dados possui uma forma gaussiana multidimensional, podendo possuir uma matriz de covariância fixa ou não. Além disso, o teste leva em consideração o tamanho da massa de dados do conjunto que está sendo analisado para evitar tomadas de decisão ruins devido a quantidade de elementos no grupo.

Utilizando também um método de sanções existe o *k'-means* (RIZMAN, 2008), este propõe uma nova forma de avaliar a função de custo. Este método avalia um elemento e decide a qual grupo ele vai pertencer, também atualizando o centro do grupo vencedor e penalizando o perdedor levando em consideração duas principais características, a primeira diz que regiões com densidade de dados elevada tende a atrair centros e a segunda é de que cada centro empurra os demais centros para longe dele, visando aumentar a informação dos padrões contidos naquele conjunto de dados. Quando um centro é movido para longe, significa que ele não obteve elementos enquanto os demais centros obtiveram, empurrando o mesmo para longe do conjunto de dado. Desta forma, este centro sem elementos atrelados a ele poderá ser excluído, restando apenas os centros que contém elementos e formarão o conjunto de regiões da imagem.

Outra estratégia encontrada foi utilizando histograma de cores no espaço de cores HSV (*hue, saturation e value*) (CHEN; CHIEN, 2008), esta divide todos os *pixels* da imagem em *bins* do histograma de forma a acelerar o processo, além de estimar uma quantidade de grupos sem a necessidade de testar várias possibilidades. Basicamente em vez de analisar elemento por elemento da imagem, o algoritmo agrupa

os elementos baseado no histograma, permitindo uma maior análise dos dados para gerar grupos e alocá-los para o seu determinado conjunto.

O Rek-means (BLOISI; LOCCHI, 2008) foi desenvolvido visando mitigar certos problemas encontrados com o algoritmo *k-means*. Este algoritmo trabalha sem a necessidade do conhecimento prévio do número de grupos encontrados em um conjunto de dados, realizando o cálculo automaticamente. Além disso, também levou em consideração a necessidade de ser robusto em relação a inicialização aleatória dos centros iniciais do algoritmo, sendo este passo muito importante, pois dependendo da inicialização, o algoritmo pode convergir para mínimos locais. A ideia do *Rek-means* trabalha em conjuntos de dados provenientes de uma distribuição gaussiana. Ele utiliza pequenas áreas em formato de retângulo para primeiro agrupar os dados, em seguida para eficientemente combina-los nos agrupamentos finais. O algoritmo consiste em 5 passos principais. O primeiro é aplicar o algoritmo padrão *K-means* para $K = n/4$; em que n é o número total de elementos pertencentes a base de dados. No segundo passo, todos os agrupamentos gerados que possuem dois ou menos elementos atribuídos a ele são descartados. O terceiro passo consiste em criar áreas retangulares que contenham todos os elementos pertencentes ao seu agrupamento específico. Após gerar esses retângulos, é calculada a distância entre eles, caso ela seja menor do que uma distância especificada D , eles são combinados, tornando-se um único grupo. Por fim, para cada grupo restante após a combinação, é aplicado um teste de validação *Anderson-Darling*. Caso falhe no teste, a combinação é desfeita.

O trabalho proposto por Li *et al* (2008) utiliza dois passos principais: Segmentação de primeiro plano usando MID (*Mosaic Image Difference*) e detecção de ombros e cabeças através do descritor HOG (*Histogram of Oriented Gradients*). O algoritmo leva em consideração a movimentação das pessoas, de forma que em longos períodos, esses movimentos satisfazem uma distribuição uniforme temporal e espacial, pois tais movimentos são aleatórios, permitindo a detecção através do MID. São analisados pequenos blocos da imagem (*grid*), em que caso os três métodos estatísticos extraídos do MID forem satisfatórios para este bloco, ele será considerado como área de primeiro plano. O conjunto dos blocos que satisfizeram as métricas se tornam a região segmentada de pessoas. A partir desta região, é extraída informações através do descritor de imagens HOG, dividindo a área segmentada em janelas, cada janela passará por um classificador treinado para detecção de ombro e cabeça que irá definir se aquela janela é uma pessoa ou não.

O algoritmo proposto por Antic *et al* (2009) utiliza uma câmera zenital para realizar a contagem de pessoas no decorrer de *frames* de um vídeo. Devido a fixação da câmera, é aplicada uma técnica de subtração da imagem de fundo através de blocos, permitindo reduzir o custo computacional. Após a remoção do fundo, é aplicado

um algoritmo de agrupamento como o *K-means* para extrair as possíveis pessoas que possam estar passando sob a câmera. Através da remoção do fundo, o trabalho realizado pelos algoritmos de agrupamento é extremamente facilitado, pois as pessoas são evidenciadas por formas concentradas. O número de grupos para a solução também é definido através de uma métrica, levando em consideração o tamanho médio das pessoas. Para realizar o rastreamento, é feita uma comparação de grupos gerados entre dois *frames* sequenciais, combinando e rotulando como mesmo grupo aqueles que estiverem mais próximos entre os *frames*.

O algoritmo *Dip-Means* possui um método incremental de agrupamento que combina três componentes principais (KALOGERATOS; LIKAS, 2012). O primeiro é uma técnica de clusterização de busca local que recebe um modelo de agrupamento para otimizar os seus parâmetros (os parâmetros do modelo são os centros dos grupos). Em sequência, o passo considerado mais importante é a tomada de decisão onde será definido se um subconjunto dos dados possui mais de uma estrutura de grupos utilizando a métrica *dip-dist*, definida pelo autor. Por fim, entra em ação um procedimento de partição que dividirá um subconjunto de dados em dois novos grupos, como também definindo os seus novos centros. O *dip-dist* consiste em um teste de distribuição de densidade empírica para unimodalidade (KALOGERATOS; LIKAS, 2012), o objetivo é checar se existe mais de uma estrutura de grupos em um subconjunto de dados, utilizando somente a distância entre todos os elementos, em vez dos próprios elementos. O teste assume que o subconjunto de dados consiste de uma grande região onde a densidade alcança um limite máximo, e a partir disto, quanto mais se distanciar desta região, é observado que tal limite só decai. Caso este fato não seja verdadeiro, é preciso realizar a divisão do subconjunto. Tal técnica permite que diversas distribuições unimodais podem ser testadas, como por exemplo a gaussiana, *student-t*, entre outros (KALOGERATOS; LIKAS, 2012).

O algoritmo *Improvements Over K-Means* (ABUBAKER; ASHOUR, 2013) propõe uma nova forma de definir a densidade dos grupos baseado em outro algoritmo chamado de *K-Nearest-Neighbor* (KNN). De acordo com o artigo, este novo procedimento contorna alguns problemas do *K-Means* tradicional como por exemplo a sua fácil influencia por *outliers* e ruídos, além de ter uma seleção de bons protótipos iniciais e também a escolha automática de centros. Este novo modo de calcular a densidade leva em consideração o raio do círculo criado para buscar os *K* elementos mais próximos, como também o volume gerado pelo mesmo. O algoritmo em geral inicia calculando a densidade de todos os elementos do conjunto de dados utilizando o KNN, a partir daí, é feita uma ordenação decrescente baseada nos resultados obtidos através deste cálculo. O primeiro ponto mais denso será declarado como o primeiro protótipo, os demais pontos serão encontrados analisando essa mesma lista, levando sempre em consideração os protótipos previamente selecionados para que não haja sobreposição.

Depois de descoberto os protótipos, é aplicado o *K-means*.

Já Shahbaba e Beheshti (2014) afirmam que na maioria dos algoritmos de agrupamento não existe a total confiança nas estruturadas de dados proveniente dos resultados encontrados. Eles acreditam que uma das principais razões para a inconsistência entre os resultados e as suposições são por causa dos erros de inicialização, pois os algoritmos de agrupamento dependem bastante da estrutura inicial dos grupos. Muitos dos algoritmos não detectam possíveis problemas iniciais, acarretando em um acúmulo de erros que afetará o resultado final. Então Shahbaba e Beheshti (2014) propuseram o novo algoritmo denominado de *MACE-means*. Este algoritmo penaliza os erros de agrupamento e de otimização baseados em uma abordagem probabilística, possibilitando uma forma de avaliar o erro de agrupamento através de uma medida quantitativa. Outro aspecto apresentado pelo *MACE-means* é a capacidade de estimar o número correto de grupos através de uma modelagem e análise rigorosa dos valores de *Mean Square Errors* de diversas quantidades de centros de grupos. Então foi definido o *Average central error* (ACE), que calcula a diferença entre os centros verdade e os centros estimados, de forma que, quanto menor for o valor de ACE, melhor. O objetivo central do algoritmo é a minimização do valor obtido através do cálculo ACE (*Minimum ACE-means*), para diversos valores de números de grupos, de forma que o menor, será o melhor resultado.

O algoritmo *Automatic color image segmentation* (KALAIVANI; CHITRAKALA, 2014) trabalha em torno da seleção automática do número de grupos baseado em máximos locais encontrados a partir de uma matriz de co-ocorrência em tons de cinza. O algoritmo é dividido em duas etapas. A primeira consiste em definir a quantidade de regiões no qual a imagem será particionada, servindo de entrada para a segunda etapa. Em seguida, um algoritmo rápido de *K-means* é executado para segmentar a imagem no número de partições definidas previamente. Para definir o número de partições, a imagem de entrada é transformada para o espaço de cores de tons de cinza. No passo seguinte, a imagem em tons de cinza passa por um pré-processamento, aplicando filtro de mediana (removendo ruídos, porém mantendo as bordas da imagem intacta), para remoção de possíveis ruídos gerados pela transformação do espaço de cores. Com o fim do pré-processamento, é extraído a matriz de co-ocorrência de níveis de cinza (método estatístico usado para examinar texturas baseados na relação espacial dos *pixels*), a partir da diagonal desta matriz, é avaliado cada valor em busca de máximos locais, ou seja, valores que são maiores do que os seus vizinhos. Aqueles que tiverem esta característica são considerados como picos. O número de partições da imagem será equivalente ao número de picos encontrados na diagonal da matriz. Em sequência é aplicada uma versão modificada do *K-means* com alocação dos centros iniciais por amostragem dos dados, parando quando os centros não modificarem.

O algoritmo *MAP-DP* também é capaz de realizar o agrupamento sem precisar especificar a quantidade de grupos. É utilizado uma abordagem probabilística na qual o número de grupos é definido pela quantidade de elementos existentes no conjunto de dados, também conhecido como *Bayesian non-parametric* (BNP), dentro dessa categoria, é especificamente utilizado o *dirichlet process mixture model*, no entanto, da forma original este procedimento é considerado bastante custoso computacionalmente (RAYKOV et al., 2016). O algoritmo então utiliza este método de forma diferente, acrescentando também um parâmetro de controle de crescimento do número de grupos para administrar o crescimento deles em relação ao total de elementos da base dados. Diferente do algoritmo *K-means*, o *MAP-DP* pode ser adaptado para diferentes formas dos agrupamentos através de modelos probabilísticos, permitindo modifica-los baseados em distribuições preditivas e seus hiper-parâmetros específicos para cada tipo de distribuição e dados (RAYKOV et al., 2016).

Khandelwal e Awekar (2017) afirmam que os algoritmos baseados em *k-means* tendem a buscar a melhora da compacticidade dos grupos e sua velocidade de execução, no entanto a maioria deles calculam a distância de cada ponto para cada centro em todas as iterações. Para resolver tal problema, eles propuseram uma nova heurística de rápida execução que irá aumentar a velocidade de execução em até 3 vezes com o custo de uma pequena piora na compacticidade dos agrupamentos. A heurística trabalha em torno da observação de que, na grande maioria das vezes, os elementos do conjunto de dados raramente trocam entre grupos muito distintos, se mantendo entre uma certa quantidade de agrupamentos. Portanto, a proposta apresentada é de que para cada elemento do conjunto de dados, é armazenada uma lista com os centros de grupos mais próximos deste elemento, de forma que a quantidade de centros desta lista é menor em relação a quantidade total de grupos, reduzindo consideravelmente a quantidade de cálculos de distância. O algoritmo se desenrola normalmente na primeira iteração, de forma que a base de dados seja dividida entre os possíveis grupos. No início da segunda iteração, ao invés de utilizar todos os possíveis centros de grupos, é criada uma lista para cada elemento contendo os centros mais próximos, e a partir de então, checa se há a necessidade de mudança de grupo somente dentro daquela lista.

Utilizando diversas técnicas aplicadas especificamente para segmentação de imagens de praia, até então não encontrada na literatura, a contribuição deste trabalho será a análise de algoritmos de segmentação de imagens em regiões litorâneas com técnicas de agrupamento que permitem a seleção automática do número de grupos. Tendo em vista pessoas que não possuem total conhecimento sobre o problema de segmentação de imagens de praia, dificilmente terá a capacidade de identificar o número correto de grupos para gerar os segmentos de imagem corretos. Desta forma, a característica de seleção automática do número de grupos se faz importante, pois

esta possui a principal vantagem de não precisar inserir manualmente entrada de dados, sendo uma característica difícil de avaliar em outros algoritmos de agrupamento. Além disso, uma técnica de pré-processamento para remover a faixa de praia das imagens foi aplicada para avaliar se houve melhora ou não nos resultados, assim como a alteração dos sistemas de cores que descrevem como os elementos são para o agrupamento, visando encontrar melhores resultados

1.3 Objetivos

Objetivo Geral:

- Analisar algoritmos de agrupamento com seleção automática do número de grupos e implementá-los para aplicações em imagens de região litorânea.

Objetivos Específicos:

1. Implementar diversos Sistemas de cores para representação das imagens de praia.
2. Implementar algoritmos de agrupamento com seleção automática do número de grupos.
3. Analisar combinações de sistemas de cores para geração de vetores de características.
4. Avaliar resultados de diversas combinações entre vetores de características e algoritmos com e sem remoção da faixa de praia.
5. Encontrar a combinação com resultados mais promissores.

1.4 Estrutura do trabalho

O Capítulo 2 apresenta conceitos básicos sobre técnicas utilizadas no percorrer deste trabalho, Capítulo 3 contem informações sobre técnicas diretamente utilizadas, como algoritmos para segmentação e remoção de faixa de praia. No Capítulo 4 é especificado a avaliação experimental, contendo a metodologia aplicada para a análise. Capítulo 5 é responsável pelos resultados encontrados, além da discussão e informações extraídas a partir dos resultados. Por fim, o Capítulo 6 apresenta as conclusões e trabalhos futuros.

2 Processamento de Imagem

O processamento de imagens consiste em um conjunto de técnicas para manipulação de imagens digitais. Tais técnicas recebem como entrada uma imagem digital e o resultado também será outra imagem digital alterada baseada na manipulação da técnica (GONZALEZ; WOODS, 2017). Atualmente, o processamento de imagens possui duas vertentes: processamento de dados em imagens para facilitar o entendimento por linguagem de máquina para um determinado problema e melhoria das informações digitais de uma imagem para aumentar a percepção visual, facilitando a interpretação humana (GONZALEZ; WOODS, 2017). Este capítulo apresentará um conjunto de conceitos básicos para facilitar o entendimento do leitor acerca das aplicações deste trabalho, além de técnicas utilizadas para chegar ao resultado final.

2.1 Introdução

Uma imagem digital pode ser definida através de uma função bidimensional $f(x, y)$, de forma que x e y representam coordenadas espaciais de um plano, de forma que cada dupla de coordenadas possua um valor de intensidade finito e discreto associado. Quando falamos em imagens digitais, a sua representação computacional é feita através de uma matriz multidimensional de tamanho $M \times N$, de forma que cada índice da matriz possui uma localização exata (através das coordenadas) e um valor representando a intensidade da cor (GONZALEZ; WOODS, 2017). Como exemplificado na Figura 2, uma matriz com dimensões de 4×4 , em que cada combinação de coordenadas possui um valor de cor.

Figura 2 – Representação de uma imagem em tons de cinza e seus valores associados.

100	150	100	0
150	100	150	100
100	150	100	150
0	100	150	100

Fonte – <https://www.inf.pucrs.br/flash/progbn/trab/t1bn-181-87345rdghsugszx/>

A partir da representação computacional de imagens, é possível realizar diversas manipulações delas para diferentes objetivos. Tais manipulações podem ser ca-

racterizadas em duas principais áreas, sendo elas processamento de imagens e visão computacional. Considerando uma linha contínua, onde as extremidades se encontram essas duas áreas, existem três processos computacionais, distribuídos em tal linha, considerando a manipulação de imagens: processos de nível baixo, médio e alto (GONZALEZ; WOODS, 2017).

Os processos de nível baixo são representados pelo fato de que tanto a entrada como a saída do processo são imagens. Operações de pré-processamento de imagens como remoção de ruído, realçar contraste e aguçamento de imagens estão classificados neste nível. Já os processos de nível médio são caracterizados pela entrada como imagens e a saída como extração de certos atributos da imagem (bordas, contornos, regiões similares, etc.). Por fim, os processos de alto nível, são responsáveis por dar um significado relevante a elementos extraídos da imagem (GONZALEZ; WOODS, 2017).

A área de processamento de imagens é representada pelos processos de nível baixo e médio principalmente, representados na Figura 3, indicando que este campo trabalha com entradas e saídas como imagens e, além disso, extração de atributos de imagens. No entanto, existe a sobreposição das áreas de processamento de imagens e visão computacional a partir dos processos de nível médio.

Figura 3 – Imagem original (esquerda), imagem após executar um processo de realce de nível baixo (centro), imagem resultado após dividir a imagem em regiões (direita).



Fonte – <http://www.dca.fee.unicamp.br/courses/IA725/1s2006/program/img/realce.jpg>

2.2 Pixel

O menor detalhe discernível em uma imagem e que possui uma cor associada é denominado de *pixel*, o conjunto de *pixels* formam uma imagem digital (GONZALEZ; WOODS, 2017). A Figura 4 representa uma imagem digital com dimensões 4x4, totalizando 16 *pixels*, em que cada *pixel* tem seu valor específico, exemplificado através dos retângulos vermelhos, para o pixel (0, 0) a intensidade de cor 100, pixel (1, 2) a cor 150 e por fim, pixel (3, 0), a cor 0.

Figura 4 – Imagem digital representando um conjunto de *pixels*.

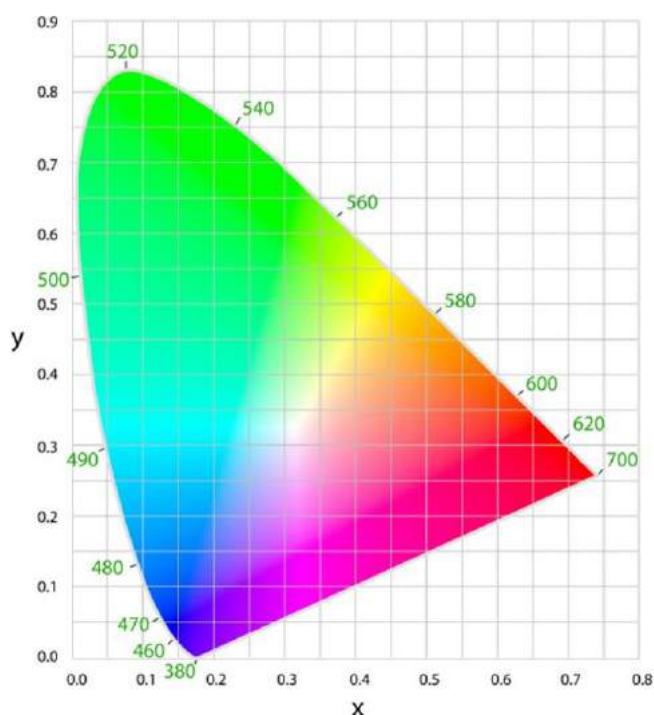
100	150	100	0
150	100	150	100
100	150	100	150
0	100	150	100

Fonte – Adaptado de <https://www.inf.pucrs.br/flash/progbn/trab/t1bn-181-87345rdghsugszx/>

2.3 Sistema de cores

Também conhecido como modelo de cores e espaço de cores, os sistemas de cores têm a função de representar um sistema de coordenadas, comumente retratado por um modelo matemático, e um subespaço de forma que cada cor seja representada somente por um único ponto dentro deste espaço, conforme na Figura 5. Além disso, um dos objetivos de possuir sistemas de cores é criar uma forma de padronizar a representação das cores, permitindo seu uso amplo em diversas fontes (GONZALEZ; WOODS, 2017).

Figura 5 – Representação do espaço de cores XYZ, cada coordenada possui uma cor associada.



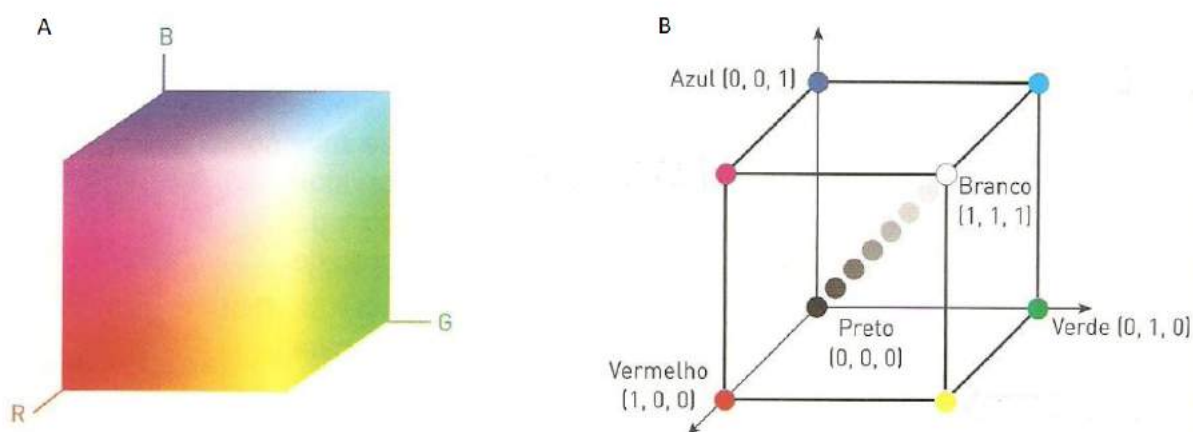
Fonte – <https://gerenciamentodecor.files.wordpress.com/2016/03/3accd-xyz.jpg>

A cor de elementos contidos em uma imagem pode ser uma ótima característica para distinguir objetos, tanto para sistemas computacionais como também para a percepção humana.

2.3.1 Modelo RGB de cores

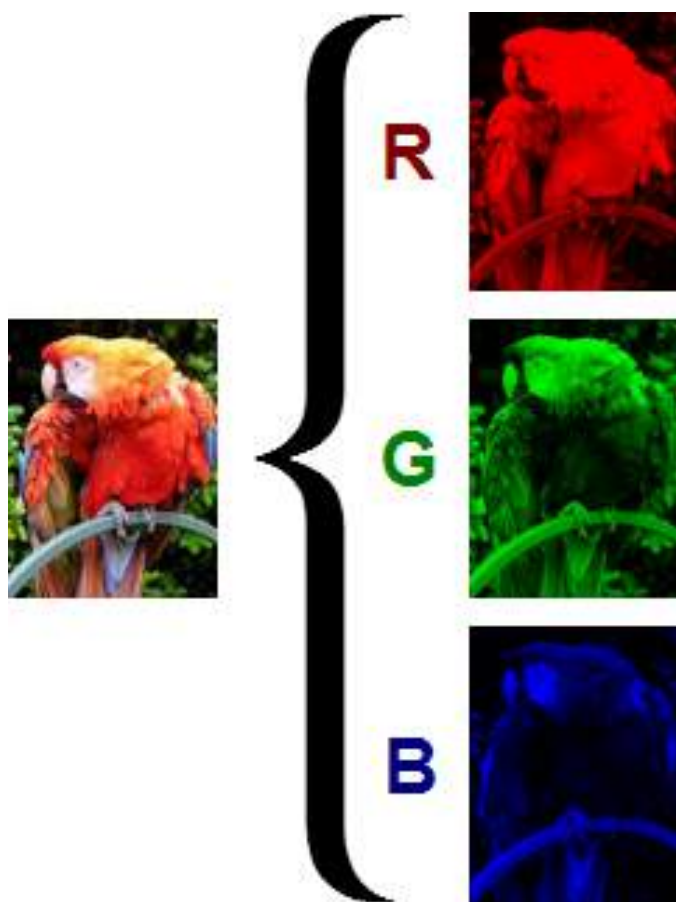
O sistema de cor RGB consiste das três cores primárias: vermelho (*Red*), verde (*Green*) e azul (*Blue*). O subespaço de cores encontradas neste sistema possui um formato de cubo (GONZALEZ; WOODS, 2017), apresentado na Figura 6, em que o eixo $(1, 0, 0)$ representa a cor vermelha, o eixo $(0, 1, 0)$ a cor verde, e o eixo $(0, 0, 1)$ a cor azul. A combinação de valores desses eixos representa uma mistura das cores primárias, podendo gerar diferentes tons de diversas cores.

Figura 6 – (A) sólido de cor do sistema RGB, (B) Sistema de coordenadas que representam o subespaço RGB.



Fonte – Adaptado de [http : //bwulex.blogspot.com/2016/11/modelos - de - cor - rgbcmymk - hsv - yuv.html](http://bwulex.blogspot.com/2016/11/modelos-de-cor-rgbcmymk-hsv-yuv.html)

Na imagem digital, a forma como o sistema de cores RGB é implementado computacionalmente consiste de um conjunto de 3 matrizes de dimensões similares, denominada de canais, porém cada uma delas irá representar um dos eixos do sistema de cores RGB. A imagem resultante da combinação dessas três matrizes gera a imagem original representada pelo sistema RGB, exemplificado na Figura 7.

Figura 7 – Imagem RGB e seus canais divididos em *Red*, *Green* e *Blue*.

Fonte – Adaptado de <http://gameautomators.com/book/ImageProcessing/2-color.html>

No sistema computacional, o *pixel* associado a uma mesma coordenada nos canais RGB nas três matrizes de cor possuem um valor de intensidade do respectivo canal, variando de 0, representando a ausência de luz (cor preta), até 255, em que dependendo do canal, representará a cor máxima do mesmo. Por exemplo, se o canal *red* possuir intensidade zero, o *green* também zero e o *blue* 255, isso significa que a cor daquele pixel será de um azul puro. A Figura 8 mostra um conjunto de cores e seus respectivos valores RGB.

Figura 8 – Exemplos de cores em formato RGB, o primeiro número representa o canal *Red*, o segundo *Green* e o terceiro *Blue*.

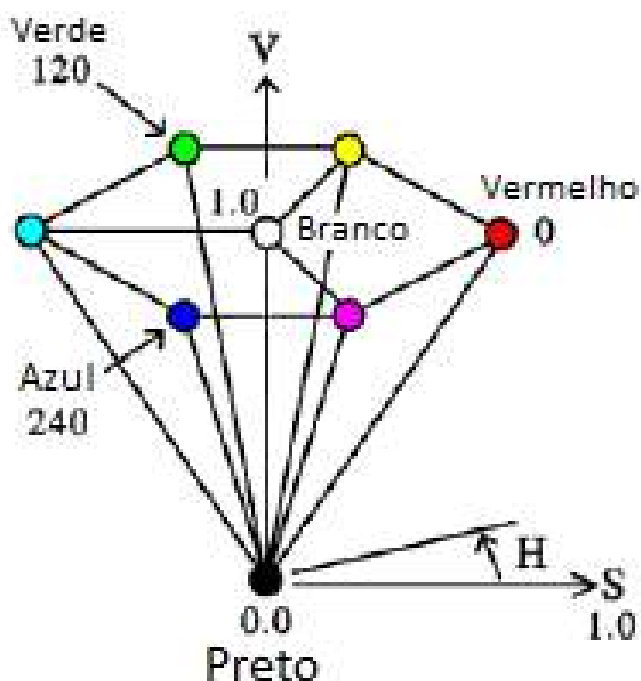
0 0 0	96 96 96	160 160 160	255 255 255	64 0 0	128 0 0	192 0 0	255 0 0
0 64 0	0 128 0	0 192 0	0 255 0	0 0 64	0 0 128	0 0 192	0 0 255
64 255 0	96 255 0	128 255 0	192 255 0	255 64 0	255 128 0	255 192 0	255 255 0
64 0 255	96 0 255	128 0 255	192 0 255	255 0 64	255 0 128	255 0 192	255 0 255
0 64 255	0 96 255	0 128 255	0 192 255	0 255 64	0 255 128	0 255 192	0 255 255
128 64 128	0 64 64	64 64 0	64 64 128	128 64 64	128 64 0	128 0 64	64 0 128

Fonte – Adaptado de <http://www.virtualkennelclub.com/vkc-manual/91-color-and-tint>

2.3.2 Modelo HSV de cores

O sistema de cor HSV é formado por três elementos: Matiz (*Hue*), Saturação (*Saturation*) e valor (*Value*), possuindo a mesma representação computacional matricial do sistema RGB. O sólido de cor representado por esse sistema apresenta um formato de cone hexagonal, como representado na Figura 9.

Figura 9 – Sólido de cor do sistema HSV.



Fonte – Adaptado de <https://www.slideshare.net/TejaRamPooniya/computer-image-processing-lecture-notes>

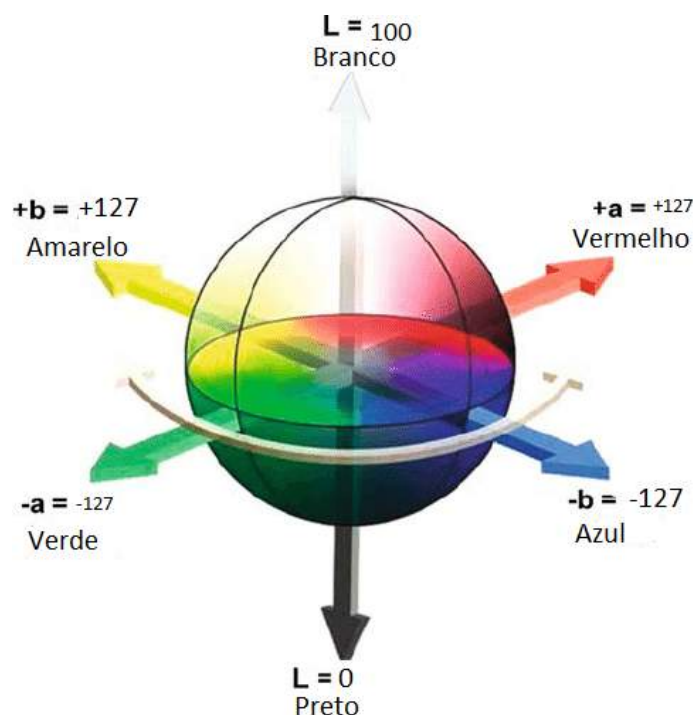
Baseado no cone hexagonal, o eixo central representa a intensidade da cor (*value*), podendo variar de valores entre 0 e 1, de forma que 0 significa a ausência da cor e 1 a total presença dela. A matiz é definida por angulação em relação a cor vermelha, variando de 0 até 360 graus, definindo qual tonalidade vai ser representada (Figura 9). Por fim, a saturação (*Saturation*) representa a pureza da cor, definido pela distância radial entre o centro do cone hexagonal até a superfície externa, variando de 0 a 1, 0 representando a mistura de várias cores, e 1 a cor mais concentrada possível (SURAL; QIAN; PRAMATIK, 2002).

2.3.3 Modelo CIELAB de cores

O modelo CIELAB possui uma característica diferencial dos sistemas até então apresentados, devido a sua conversão a partir do sistema XYZ, considerado como aquele que abrange todas as cores, o espaço de cores do CIELAB é distribuído em um espaço aproximadamente uniforme e tridimensional, de forma que cada cor possui uma localização única que a identifica em termos de coordenadas cartesianas em respeito aos eixos L^* , a^* e b^* (WEATHERALL; BERNARD, 1992), tornando a utilização deste sistema bastante eficiente para comparação de cores, pois a distância entre os pontos cartesianos refletem diretamente na similaridade das cores.

O eixo L^* é o componente de luminosidade, variando de 0 até 100, sendo as cores pretas e brancas respectivamente, e os valores intermediários representando tons de cinza de forma uniformemente distribuída. O eixo a^* está relacionado a variações de cores entre os tons vermelho e verde, enquanto o eixo b^* varia entre os tons amarelo e azul. Ambos os eixos variam entre valores de -127 até 127 (OPENCV, 2015), como representado na Figura 10.

Figura 10 – Sólido de cor do sistema CIELAB.



Fonte – Adaptado de <https://www.flickr.com/photos/greenmambagreenmamba/4236391637>

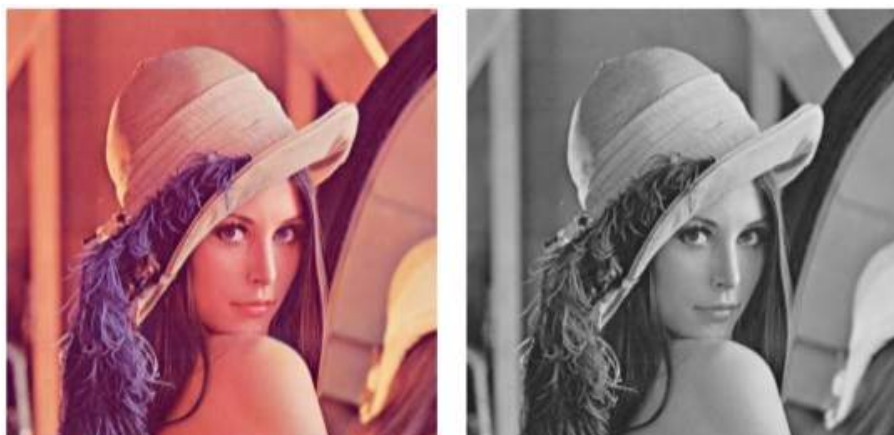
2.3.4 Conversão de RGB para tons de cinza

Muitas técnicas envolvendo processamento de imagens requer que elas estejam especificamente em tons de cinza, desta forma, se faz necessário utilizar um método que transforme as imagens padrões no sistema de cores RGB para tons de cinza, permitindo a execução dos algoritmos específicos.

Para converter uma imagem em RGB para tons de cinza basta aplicar a fórmula 2.1 para cada pixel (OPENCV, 2015), levando em consideração que o pixel resultante em tons de cinza é uma combinação com pesos específicos dos três canais RGB. O resultado da conversão é exemplificado na figura 11.

$$Y = 0.299.R + 0.587.G + 0.114.B \quad (2.1)$$

Figura 11 – Exemplo de imagem em RGB convertida para tons de cinza.



Fonte – O Autor

O resultado final é uma soma ponderada de todas as cores RGB, e cada *pixel* é multiplicado por um valor de acordo com a importância do canal específico.

2.3.5 Conversão de RGB para HSV

Para realizar a conversão do espaço de cores RGB para o HSV, é preciso realizar um pré-processamento de normalização dos valores de cada pixel. Para cada canal RGB, os elementos que podem variar de 0 até 255 serão normalizados para se encaixar em valores de 0 a 1. Então, cada pixel será processado pelas formulas a seguir, gerando como resultado um pixel no espaço de cor HSV (OPENCV, 2015).

$$V = \max(R, G, B) \quad (2.2)$$

$$S = \begin{cases} V - \min(R, G, B)/V, & V \neq 0 \\ 0, & V = 0 \end{cases} \quad (2.3)$$

$$H = \begin{cases} 60(G - B)/(V - \min(R, G, B)), & \text{se } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)), & \text{se } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)), & \text{se } V = B \end{cases} \quad (2.4)$$

Se $H < 0$, $H = H + 360$

2.3.6 Conversão de RGB para XYZ

Antes de converter uma imagem em RGB para CIELAB, é necessário previamente realizar uma conversão de RGB para XYZ, pois o XYZ foi criado para abranger todas as cores visíveis. A conversão é dada pelas formulas a seguir (OPENCV, 2015) sendo aplicadas para cada pixel RGB.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.5)$$

$$X = X/0.950456 \quad (2.6)$$

$$Z = Z/1.088754 \quad (2.7)$$

2.3.7 Conversão de RGB para CIELAB

Após a obtenção da imagem em formato XYZ, é possível transformá-la para o espaço de cores CIELAB seguindo as equações a seguir para cada pixel da imagem (OPENCV, 2015).

$$L = \begin{cases} 116 * Y^{1/3} - 16 & \text{para } Y > 0.008856 \\ 903.3 * Y & \text{para } Y \leq 0.008856 \end{cases} \quad (2.8)$$

$$A = 500(f(X) - f(Y)) + delta \quad (2.9)$$

$$B = 200(f(Y) - f(Z)) + delta \quad (2.10)$$

$$f(t) = \begin{cases} t^{1/3} & \text{para } t > 0.008856 \\ 7.787t + 16/116 & \text{para } t \leq 0.008856 \end{cases} \quad (2.11)$$

$$delta = \begin{cases} 128 & \text{para imagens de 8 bits} \\ 0 & \text{para imagens de ponto flutuante} \end{cases} \quad (2.12)$$

Caso as imagens estejam sendo dimensionadas por 8 bits, significa que os *pixels* são representados por números inteiros, necessitando realizar a normalização a seguir:

$$L = L * 255/100 \tag{2.13}$$

$$A = A + 128 \tag{2.14}$$

$$B = B + 128. \tag{2.15}$$

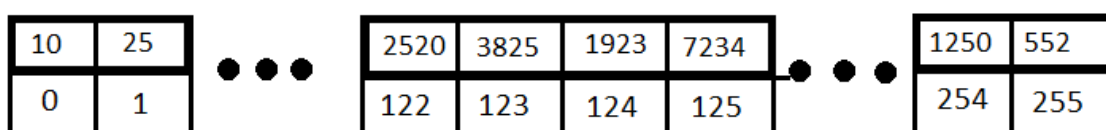
2.4 Histograma

Histogramas são utilizados como base para diversas técnicas que circundam o processamento de imagens. Técnicas para realce de imagem, extração de certas características que possibilitam a sua aplicação em segmentação para divisão da imagem em regiões, ou até mesmo prover uma descrição da aparência global da imagem através da distribuição dos tons de cores. Outro aspecto importante sobre histogramas é a simplicidade do cálculo, tornando-os ótimas ferramentas para processamento em tempo real (GONZALEZ; WOODS, 2017).

Para calcularmos um histograma, é preciso levar em consideração a quantidade de possíveis tons de cor, definidos através do intervalo entre $[0, L-1]$ em que L representa o total de tons de cor encontrados na imagem (para uma imagem composta por 8 bits, o número total de tons de cor seria de 256). O histograma é uma função discreta $h(r_k) = n_k$, na qual r_k é o k -ésimo valor de tom de cor e n_k é a quantidade de repetições de tal tom representado por r_k (GONZALEZ; WOODS, 2017). Basicamente os histogramas apresentam um conjunto de tuplas em que o primeiro elemento corresponde a um tom de cor e o segundo elemento a quantidade de vezes que este tom se repetiu por toda a imagem (Figura 12).

Figura 12 – Conjunto de tuplas de uma imagem em tons de cinza com 256 tons de cor.

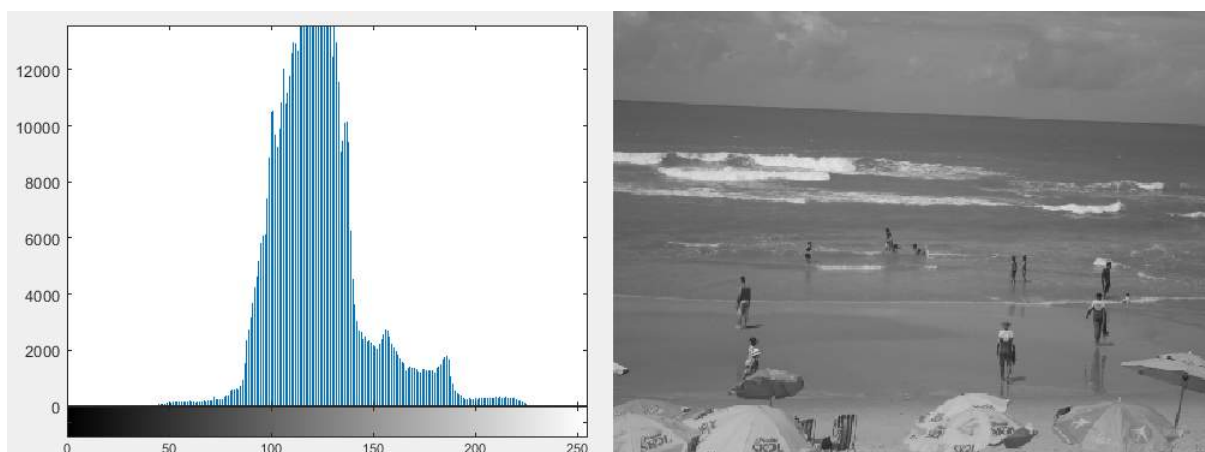
Contagem de repetições



Tons de cor

Apesar da representação computacional mostrada na Figura 12, muitas vezes os histogramas são apresentados em formas gráficas para facilitar o entendimento e a visualização da distribuição dos tons da imagem (Figura 13).

Figura 13 – Histograma e sua imagem em tons de cinza.



Fonte – O Autor

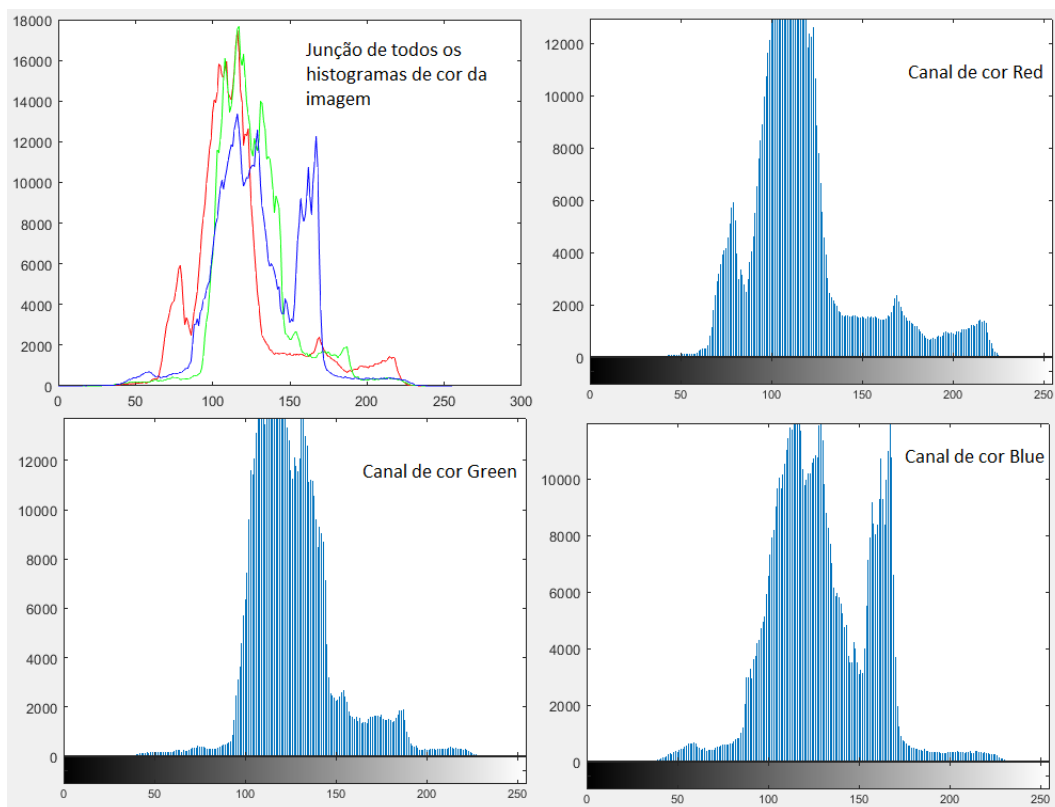
Apesar de muitas vezes os histogramas serem apresentados de imagens em tons de cinza, também é possível criar histogramas de imagens coloridas. Exemplificando uma imagem RGB, será necessário criar um histograma a parte para cada canal de cor existente na imagem (*red*, *green* e *blue*) como mostra a Figura 14 e 15.

Figura 14 – Imagem original em RGB.



Fonte – O Autor

Figura 15 – Histograma de imagem RGB juntos e separados.



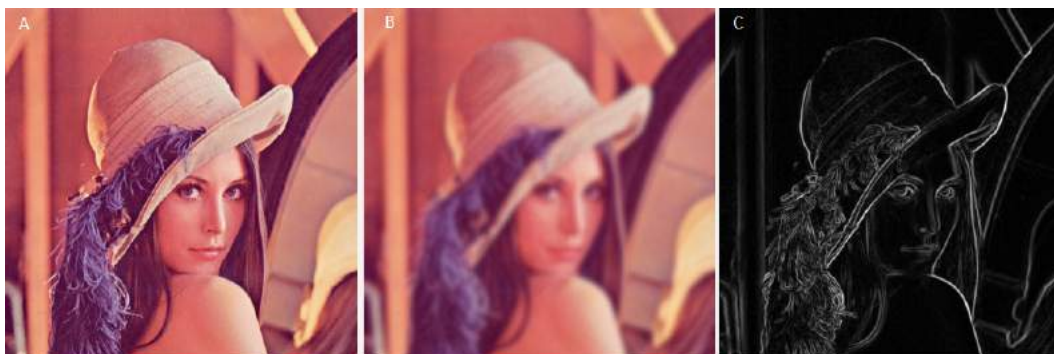
Fonte – O Autor

2.5 Filtro espacial

A filtragem espacial é uma das técnicas de processamento de imagem mais utilizadas, pois possui uma vasta gama de aplicações, como realce de uma imagem, destacamento de bordas, remoção de ruídos, entre outros (GONZALEZ; WOODS, 2017).

O termo filtro é utilizado emprestado do processamento no domínio da frequência, significando que ao filtrar, se refere a permissão ou rejeição de certos componentes de frequência. Os tipos de filtros também foram divididos levando em consideração o domínio da frequência, possuindo os filtros passa-baixa e passa-alta. O passa-baixa significa que o filtro permite que baixas frequências passem e não sofram modificação, enquanto as altas frequências sofrerão atenuação, gerando uma imagem na qual as áreas onde há grande mudança de tons seja borrada (bordas). Já os filtros passa-alta são o inverso, permite que altas frequências passem pelo filtro sem sofrer modificação, enquanto as baixas frequências sofrem alteração, desta forma, aumentando o detalhamento de regiões que há grande diferença de tons (bordas). A diferença entre os filtros pode ser exemplificada na Figura 16.

Figura 16 – Imagem original (A), Imagem após executar um filtro passa-baixa (B), imagem após executar filtro passa-alta (C).

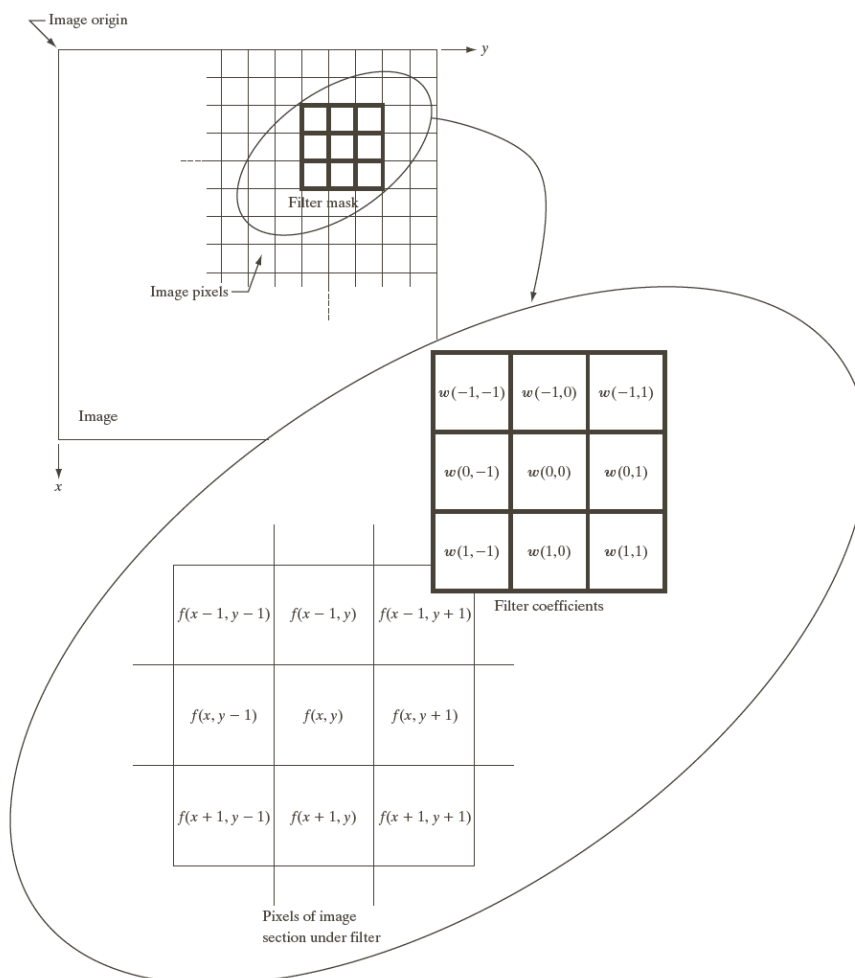


Fonte – O Autor

A execução de um filtro em uma imagem consiste em possuir uma vizinhança, normalmente um pequeno retângulo, e uma operação previamente definida que será realizada sobre os *pixels* encontrados na vizinhança (GONZALEZ; WOODS, 2017). Após executar a operação, um novo *pixel* é criado exatamente no centro da vizinhança, cujo valor é resultante da operação de filtragem. O processo de filtragem só termina quando o centro da vizinhança percorre cada *pixel* encontrado na imagem original.

Normalmente, a vizinhança e a operação predefinida são representadas através de um elemento chamado de máscara. A máscara consiste de uma matriz dimensional (formato da vizinhança) que é sobreposta a matriz da imagem, de forma que o centro da máscara case com o *pixel* analisado. O resultado final será a soma dos produtos dos coeficientes do filtro com os valores dos *pixels* que foram sobrepostos, representando a operação predefinida (Figura 17) (GONZALEZ; WOODS, 2017).

Figura 17 – Representação da máscara que sobrepõe a imagem original e executa a filtragem.

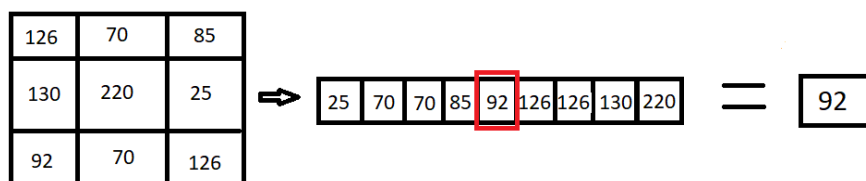


Fonte – (GONZALEZ; WOODS, 2017)

2.5.1 Filtro de Mediana

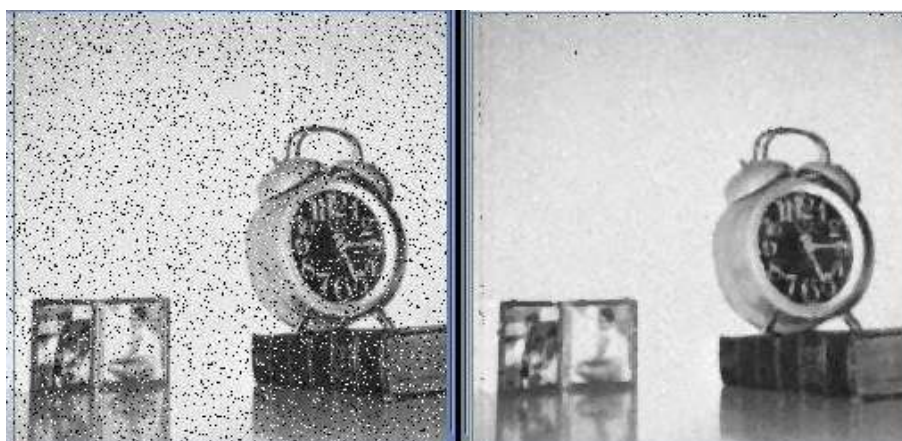
O filtro de mediana é do tipo passa-baixa, realizando uma suavização da imagem, ele consiste na ordenação dos *pixels* casados na máscara em ordem crescente do valor de intensidade de cor. A partir da ordenação, é selecionado o valor de intensidade que estiver no meio da lista ordenada, atribuindo tal valor ao *pixel* resultante, exemplificado na Figura 18. Uma das aplicabilidades de tal filtro é a remoção de ruídos conhecidos como sal e pimenta, pontos brancos e pretos distribuídos na imagem(Figura 19) (GONZALEZ; WOODS, 2017).

Figura 18 – Exemplo de um filtro de mediana de tamanho 3x3.



Fonte – O Autor.

Figura 19 – Remoção de ruído de sal e pimenta através do filtro de mediana.



Fonte – <https://sites.google.com/site/lemared/filtrado-espacial-suavizamiento>

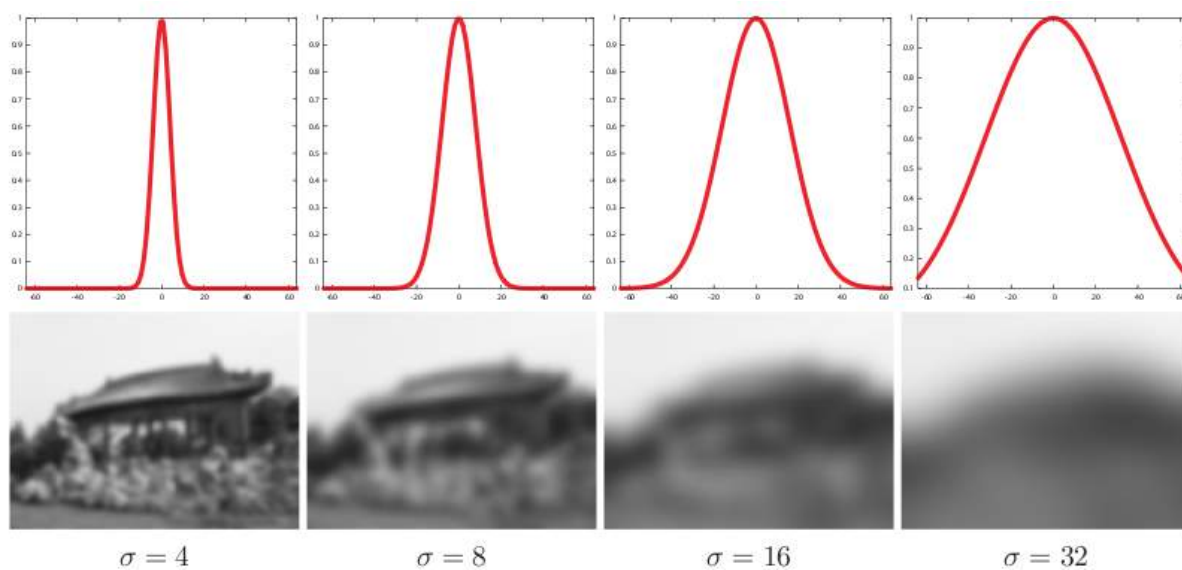
2.5.2 Filtro Gaussiano

O filtro gaussiano é um filtro passa-baixa, ou seja, ele atenua as altas frequências (detalhes da imagem, como contornos) e passa as baixas frequências. A imagem após ser aplicada se torna visivelmente borrada (Figura 20). Este filtro possui dois parâmetros: o tamanho da máscara que irá percorrer os *pixels* da imagem e seus vizinhos e o valor de desvio padrão máximo que definirá a largura de alcance da gaussiana (sigma), de forma que quanto maior for o sigma, mais borrada a imagem resultante será (Figura 20). O filtro gaussiano pode ser calculado por meio da Equação 2.16.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.16}$$

Em que σ representa o desvio padrão da curva gaussiana, x sendo a distância da origem no eixo horizontal e y a distância da origem no eixo vertical.

Figura 20 – Imagem resultante e seus diferentes valores de sigma para o filtro gaussiano.



Fonte – <http://lvelhoimpa.br/ip09/demos/jbu/filtros.html>

2.6 Operações lógicas

As operações lógicas normalmente só podem ser utilizadas no processamento de imagens quando uma imagem estiver binarizada. Uma imagem binarizada é quando os *pixels* presentes só podem assumir dois valores, zero ou um.

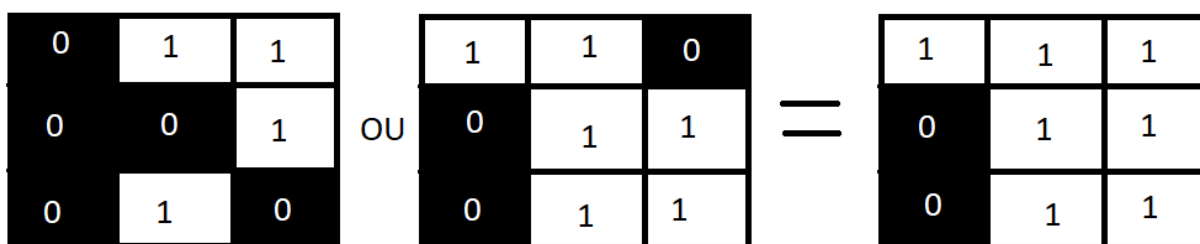
Supondo que duas imagens A e B estão binarizadas e possui dimensões iguais, uma operação lógica é aplicada *pixel a pixel*, gerando um resultado que também será binário. No decorrer do projeto, a operação lógica OU (OR) e E (AND) foram utilizadas, e o resultado final da operação entre as possíveis combinações de valores dos *pixels* são apresentadas a seguir:

Pixel A	Pixel B	Op. OU lógica	Op. AND lógica
1	1	1	1
1	0	1	0
0	0	0	0
0	1	1	0

Tabela 1 – Tabela apresentando os resultados da operação lógica OU (OR) e E (AND) entre dois *pixels* binários.

A Figura 21 apresenta um resultado de duas imagens que realizaram uma operação lógica OU (OR).

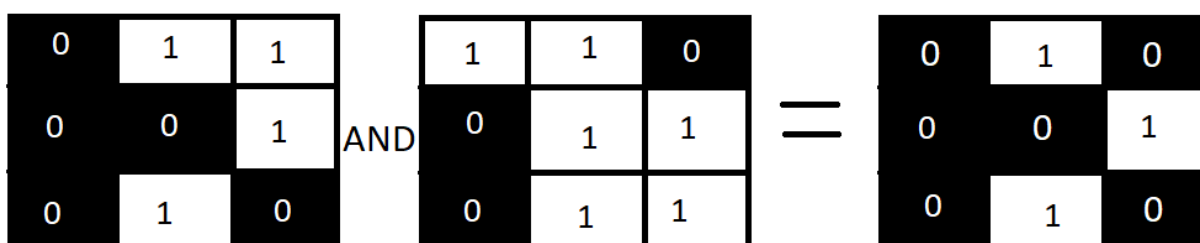
Figura 21 – Execução de uma operação logica OR e seu resultado.



Fonte – O Autor

A Figura 22 apresenta um resultado de duas imagens que realizaram uma operação lógica AND.

Figura 22 – Execução de uma operação logica AND e seu resultado.



Fonte – O Autor

2.7 Limiarização

Devido a redução do custo computacional causado pela diminuição de dados analisados em uma imagem, a limiarização tem um papel fundamental para aplicações de processamento de imagens (GONZALEZ; WOODS, 2017). A limiarização é uma técnica que separa regiões de imagens através dos seus valores de intensidade, normalmente trabalhadas em imagens em tons de cinza. A limiarização é comumente utilizada para separar o fundo de possíveis objetos de interesse contidos na imagem, onde as partes consideradas fundo se tornam o número 0 (cor preta), enquanto os objetos de interesse se tornam 1 (branco), transformando a imagem em tons de cinza em binária (Figura 24).

A limiarização se dá a partir da definição de um certo limiar T de intensidade de cor que será decidido através da imagem de forma que melhor separe as intensidades de cores do fundo em relação aos objetos de interesse. Intensidades de cores que estiverem abaixo do limiar T serão considerados parte do fundo, tornando-se 0 (preto), enquanto elementos acima do limiar se tornarão objetos de interesse de cor branca, 1 (GONZALEZ; WOODS, 2017). Formalizando, a limiarização do tipo global funciona da seguinte forma, em que $f(x, y)$ equivale a intensidade de cor do pixel analisado:

$$g(x, y) = \begin{cases} 1 & , f(x, y) > T \\ 0 & , f(x, y) \leq T \end{cases} \quad (2.17)$$

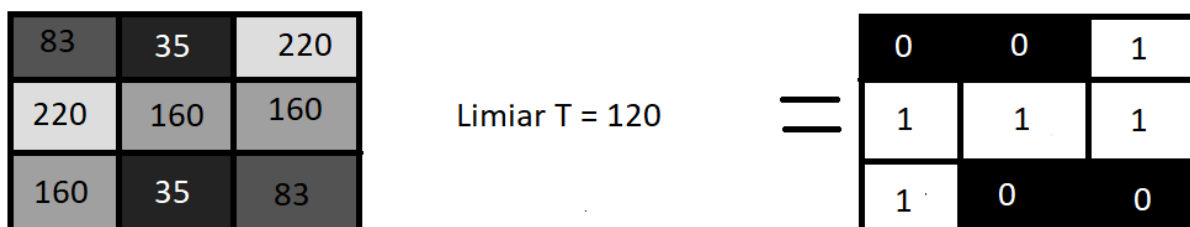
A Figura 23 a seguir mostra um exemplo de limiarização de uma imagem em tons de cinza:

Figura 23 – Exemplo de binarização de uma imagem em tons de cinza.



Fonte – <https://blogs.mathworks.com/steve/2016/05/16/image-binarization-new-r2016a-functions/>

Figura 24 – Limiarização para uma imagem em tons de cinza.



Fonte – O Autor

2.8 Contornos

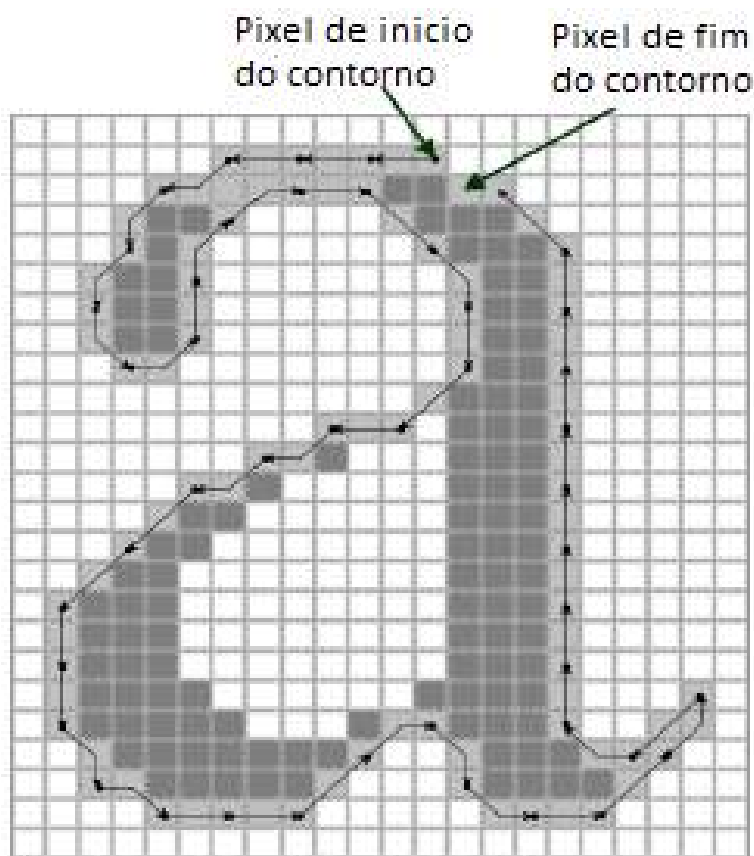
Contornos podem ser explicados através de uma curva que une diversos pontos contínuos (bordas), possuindo intensidades de cores similares. As técnicas de contor-

nos possuem diversas aplicabilidades, entre elas: análises de formas, detecção de objetos e reconhecimento (SUZUKI; KEIICHI, 1985).

Um das principais formas de extração de contornos é através da definição de um componente conexo. Um componente conexo é definido através de um subconjunto de *pixels* S que qualquer *pixel* pertencente a este conjunto pode criar um caminho através de uma regra de vizinhança para outro *pixel* também pertencente ao conjunto, desta forma, se todos os *pixels* presentes neste subconjunto respeitar a regra, S é definido como um componente conexo.

A partir do componente conexo, o contorno pode ser definido como os *pixels* que constituem a borda de tal componente conexo, de forma que a regra de vizinhança levada em consideração não é verdade para todos *pixels* em torno do possível *pixel* de borda (Figura 25).

Figura 25 – Contorno de um componente conexo.



Fonte – Adaptado de <http://www.rroi.com/open-access/an-improved-edge-adaptive-grid-technique-to-authenticate-grey-scale-images.php?aid=50901>

2.9 Segmentação de imagens

A segmentação de imagens tem como função a subdivisão de uma imagem em diversas regiões ou objetos de interesse, agrupando partes da imagem que se parecem em uma única região, enquanto partes diferentes, em regiões distintas a partir de um conjunto de critérios predefinidos (GONZALEZ; WOODS, 2017).

Dado uma imagem e seu conjunto de *pixels* R , pode-se definir uma segmentação de imagem através de um processo que segregue R em n sub-regiões $R_1, R_2, R_3, \dots, R_n$ de tal forma que siga as regras a seguir (GONZALEZ; WOODS, 2017):

- A) A união de todas as sub-regiões R_i forma a imagem R original.
- B) Todas as sub-regiões R_i é um conjunto conectado.
- C) Não existe *pixels* que possam estar entre sub-regiões distintas.
- D) Todos os *pixels* dentro de uma sub-região devem cumprir com as propriedades da sub-região.
- E) Sub-regiões não podem compartilhar da mesma propriedade

A regra (A) afirma que a segmentação deve ser completa, significa que todos os *pixels* de uma imagem necessariamente precisam estar pertencendo a uma das sub-regiões, e a união de todas as sub-regiões formarão a imagem original. A regra (B) requer que os pontos de uma região estejam conectados através de alguma forma predefinida. A condição (C) indica que não pode existir sobreposição entre as sub-regiões, indicando que cada pixel só pode pertencer a um único grupo. A regra (D) afirma que todos os *pixels* presentes dentro de uma sub-região específica devem atender a uma determinada propriedade daquele grupo, como por exemplo o nível de intensidade de cor dos *pixels* precisa ser similar entre os *pixels* presentes no grupo a partir de uma intensidade X . Por fim, a regra (E) indica que sub-regiões distintas não podem possuir uma mesma característica de outra sub-região, como por exemplo dois grupos possuírem a propriedade de somente possuir *pixels* com a mesma intensidade de cor X .

A segmentação é uma das etapas mais importantes quando se trata da extração de características, conseqüentemente uma das mais difíceis, pois a precisão da segmentação determinará o sucesso ou fracasso na análise da imagem, tendo em vista que tais erros se propagarão para as futuras etapas de extração e classificação de dados (GONZALEZ; WOODS, 2017).

A segmentação de imagens pode ser representada de duas formas principais, através de imagens binárias representando as regiões brancas como pertencentes

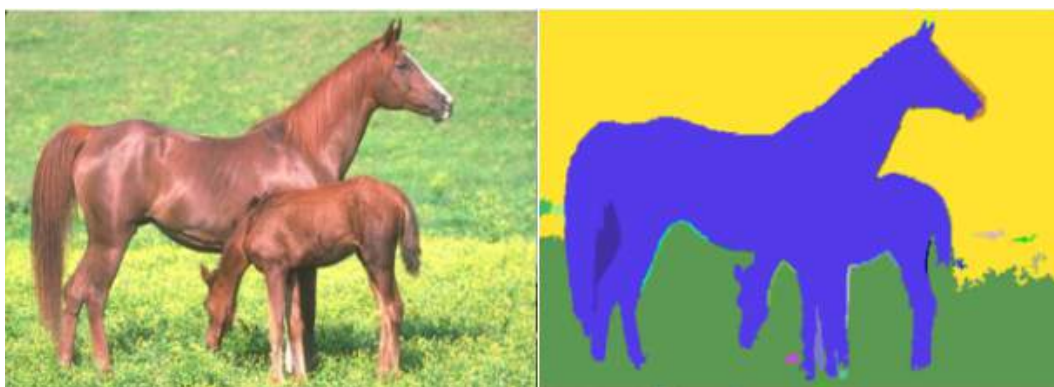
ao grupo (Figura 26), ou através de diferentes cores para cada grupo em uma única imagem (Figura 27).

Figura 26 – Segmentação de imagem com apenas dois grupos representados em binário.



Fonte – O Autor

Figura 27 – Imagem Segmentada utilizando a representação colorida.



Fonte – <http://www.lapix.ufsc.br/wp-content/uploads/2015/10/fh2.jpg>

2.10 Agrupamento

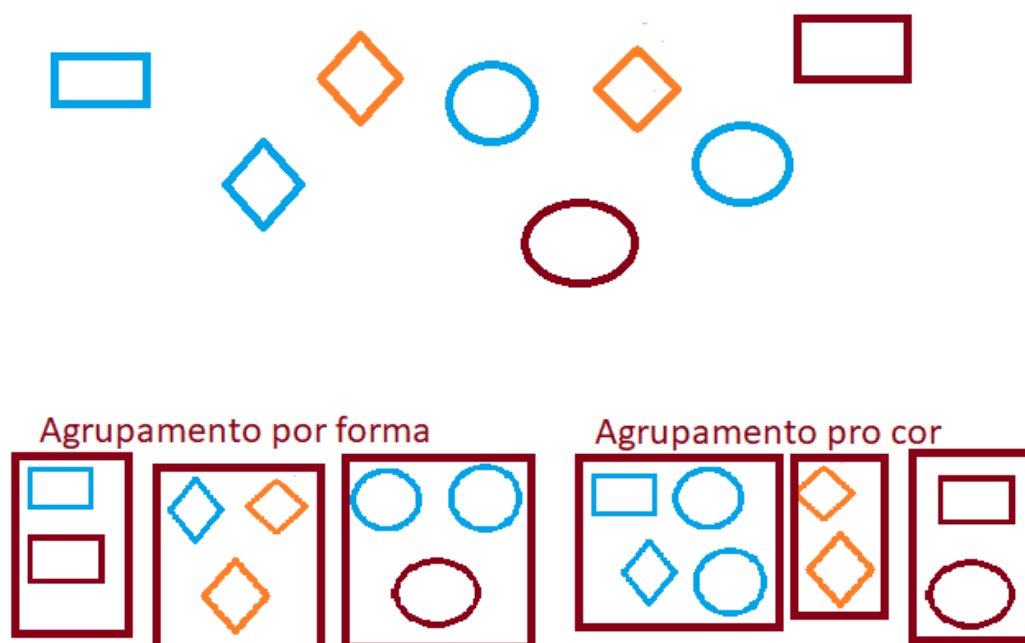
Agrupamentos, também conhecidos como clusterização, é uma técnica que visa analisar um conjunto de dados que possua pouca ou quase nenhuma informação sobre o mesmo. Tais técnicas são aplicadas quando o conjunto de dados não possui uma classificação predefinida para cada elemento do grupo, mas sim quando os elementos aparentam possuir uma divisão natural, permitindo separar os elementos em diferentes conjuntos que compartilham similaridade interna entre os elementos (WITTEN; FRANK, 2005).

No caso de imagens, os elementos analisados serão os *pixels*, onde todos eles serão divididos entre os possíveis grupos, de forma que os *pixels* que possuam uma determinada característica similar participem do mesmo grupo, enquanto *pixels* diferentes fiquem em grupos diferentes. Quanto maior for a similaridade dos elementos

dentro de um grupo e maior for a dissimilaridade entre elementos de diferentes grupos, melhor será o agrupamento.

A similaridade dos agrupamentos é subjetiva, há a necessidade de escolher uma determinada característica para analisar e realizar os agrupamentos baseadas na mesma. A Figura 28 representa um agrupamento realizado de duas formas diferentes devido as características que possam vir a serem analisadas.

Figura 28 – Um conjunto de dados e agrupamento baseados na característica de forma e cor.



Fonte – O Autor

2.10.1 Algoritmos hierárquicos

Os algoritmos que utilizam o método hierárquico constroem os agrupamentos recursivamente particionando o conjunto de dados. Há duas formas de realizar o particionamento: *Bottom-up* ou *Top-Down* (MAIMON; ROKACH, 2010).

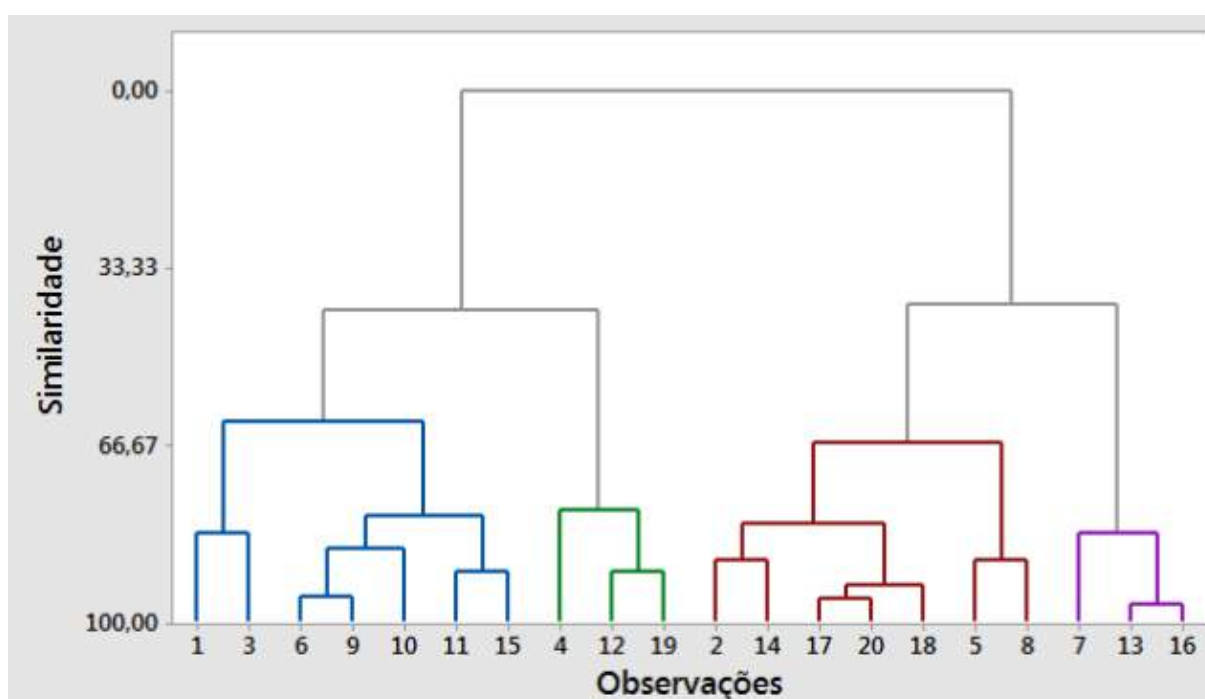
O *Bottom-up*, também chamado de algoritmo de agrupamento hierárquico aglomerativo, consiste em todos os elementos pertencentes ao conjunto de dados serem inicialmente considerados como um grupo, se o conjunto de dados possuir N elementos, inicialmente o algoritmo possuirá N grupos. A partir disso, os grupos são combinados baseados em uma métrica de similaridade até que todos os elementos pertençam a um único grupo.

Já a abordagem *Top-Down*, conhecida também por algoritmo hierárquico divi-

sivo, todos os elementos começam pertencendo a um único grupo e então este será dividido em subgrupos através de uma métrica de dissimilaridade até que todos os elementos possuam seu subgrupo individual.

A solução encontrada através de algoritmos hierárquicos é representada através de um dendrograma (Figura 29). A solução em formato de grupos é definida a partir de um ponto de corte do nível de similaridade escolhido pelo usuário. No exemplo da Figura 29, caso o ponto de corte seja de 0.33 o agrupamento consistirá de dois grupos, o primeiro contendo as observações 1, 3, 6, 9, 10, 11, 15, 4, 12 e 19, enquanto o segundo grupo possuirá os elementos 2, 14, 17, 20, 18, 5, 8, 7, 13 e 16. No mesmo exemplo da Figura 29, caso o corte seja em 0.50, haverá 4 grupos, de forma que o primeiro grupo possuirá os elementos 1, 3, 6, 9, 10, 11 e 15, segundo grupo terá 4, 12 e 19, terceiro grupo conterà os elementos 2, 14, 17, 20, 18, 5 e 8, por fim, o último grupo ficará com 7, 13 e 16.

Figura 29 – Representação de um agrupamento hierárquico em forma de dendrograma.



Fonte – Adaptado de <https://support.minitab.com/pt-br/minitab/18/help-and-how-to/modeling-statistics/multivariate/how-to/cluster-observations/interpret-the-results/all-statistics-and-graphs/dendrogram/>

2.10.2 Algoritmo *K-means*

O algoritmo clássico *K-means*, é uma das técnicas clássicas mais utilizadas. Primeiro, para realizar o agrupamento, o algoritmo precisa da informação do número de grupos no qual o conjunto de dados será particionado, este parâmetro é chamado

de K .

Após a definição do parâmetro K , um elemento fictício será designado aleatoriamente para cada grupo para representar o centro inicial, a partir disto, cada elemento pertencente ao conjunto de dados é atribuído a um dos grupos cujo uma métrica de distância baseado em determinadas características é a menor em relação aos centros dos grupos. Quando todos os elementos do conjunto de dados forem atribuídos a um grupo, os centros dos grupos serão atualizados baseados na média dos elementos pertencentes aos grupos respectivos. Por fim, todo o processo será repetido até que os elementos pertencentes a cada grupo se mantenham iguais entre duas repetições, se tornando estável e jamais alterando independente de quantas repetições seja feita (WITTEN; FRANK, 2005).

O pseudocódigo do algoritmo *K-Means* é apresentado a seguir:

1. Designa um elemento para ser o centro para cada grupo G_1, G_2, \dots, G_k .
2. Para cada elemento x_i pertencente ao conjunto de dados, atribuir o elemento para o grupo que possuir a menor distância entre tal elemento e o centro do grupo.
3. Para cada grupo, calcular o novo centro a partir da média de todos os elementos pertencentes a cada um dos grupos.
4. Repetir os passos 2 e 3 até que os novos centros não mudem entre as iterações.

Apesar de ser bastante eficiente e utilizado, o *K-means* compartilha de algumas limitações intrínsecas ao método. O algoritmo não é invariante a rotação e translação, devido a formação natural do algoritmo de realizar esferas no espaço dos dados e também é suscetível a *outliers*, significando que elementos fora do padrão influenciam diretamente na formação dos grupos. Por fim, um dos principais problemas é que o *K-means* assume que o número de grupos K é conhecido e fornecido, no entanto, na realidade ter o conhecimento de K não é trivial e muitas vezes é determinante para o sucesso ou fracasso do agrupamento.

3 Segmentação de imagens de praia

Neste capítulo serão apresentados os algoritmos que realizam a segmentação de imagens de praia em duas etapas. Primeiro, será apresentada a técnica de remoção da faixa de areia da praia, caracterizada como uma técnica de pré-processamento das imagens, além da criação dos vetores-característica que descrevem os elementos da imagem.

Depois, serão apresentados algoritmos de segmentação por agrupamento sem a necessidade de especificar um número de grupos. Tal característica é importante quando não se possui um conhecimento prévio do problema, pois identificar o número de grupos em uma imagem colorida não é um trabalho trivial. Dessa forma, apresentaremos nas próximas seções os algoritmos: *X-means*, utilizando uma métrica estatística conhecida como BIC para medir a qualidade dos agrupamentos. Outro algoritmo que determina o número de *clusters* utilizando *k-means* na aplicação em segmentação de imagens coloridas, aplicando um método de validação baseado na compacticidade e distância dos grupos, por fim, o algoritmo Segmentação automática de imagens coloridas, extraíndo informações de textura da imagem com matriz de co-ocorrência dos níveis de cinza para determinar a quantidade de grupos.

3.1 Remoção da faixa de praia

O algoritmo de remoção da faixa de praia se baseou no fato de que tanto as regiões de água como do céu tem a cor predominante azul, se diferenciando bastante da região de areia (CARRERA, 2017). Ela é importante, pois como o sistema é focado em pessoas que estão dentro da praia, a remoção da areia permitirá ter um foco maior apenas na região de praia, diminuindo a complexidade e o tamanho da imagem analisada.

O primeiro passo do algoritmo de remoção de praia é a aplicação do algoritmo de agrupamento tradicional (*K-means*), para segmentar a imagem em quatro regiões distintas. Para cada região apresentada como resultado, é criada uma imagem binária representando somente o respectivo grupo, aplicando uma técnica de limiarização. Na sequência, para cada grupo é extraído o seu contorno, visando enquadrar as regiões de interesse, como também preencher possíveis espaços internos gerado pelo agrupamento, então, o tamanho das áreas geradas pelos contornos são medidos, e aqueles que possuem um valor maior do que o parâmetro α serão considerados (Figura 30) (CARRERA, 2017).

Figura 30 – Imagem binarizada de um dos grupos (esquerda) e resultado após extrair contornos (direita)



Fonte – O Autor

Com as máscaras em mãos, é feita uma operação lógica *AND* entre a máscara e sua imagem original, caso o pixel da máscara for 1, o pixel da imagem original manterá a cor, caso pixel da máscara for 0, o pixel da imagem original se tornará preto (Figura 31). Tais imagens são convertidos para o sistema de cor HSV e o canal *H* (*hue*) é extraído.

Figura 31 – Máscara binária (esquerda) e região da imagem original (direita).



Fonte – O Autor

O canal de cor *H* foi escolhido devido a sua propriedade, pois este representa especificamente a cor. Como buscamos a cor azul, as imagens das regiões são analisadas através da extração de um histograma do canal *H*, onde as regiões que possuam tal espectro de cor elevado, como também o desvio padrão dos tons de cor (cores constantes resultam em um desvio padrão baixo), seria considerado como região de mar e/ou céu (CARRERA, 2017).

Por fim, todas regiões que foram classificadas como possíveis de representarem mar/céu sofrem uma operação lógica *OR* para juntar todas elas em uma única

máscara, permitindo a extração da região de mar e céu da imagem original através da operação lógica *AND* similar a utilizada anteriormente.

Em alguns casos, regiões da imagem são perdidas devido ao contato da área com as bordas da imagem. Para contornar tal problema, as bordas laterais da imagem são expandidas acrescentando *pixels* de cor branca (1) na máscara final para que os contornos possam se fechar. Após a conclusão do procedimento, as bordas laterais são removidas para que a máscara tome as dimensões originais novamente (CARRERA, 2017).

O pseudocódigo para o algoritmo vem a seguir:

1. Converte imagem original para sistema de cor HSV.
2. Aplica o filtro gaussiano para suavizar as regiões de transição da imagem com $\sigma = 5$ e tamanho de máscara 9x9.
3. Extraí os grupos através do algoritmo *K-means*.
4. Para cada grupo, faça:
 - Gera imagem binária aplicando uma limiarização.
 - Extraí contornos da imagem binarizada.
 - Para cada contorno, checar tamanho. se menor do que parâmetro α (área do contorno), descartar, se não, manter.
 - Extrair da imagem HSV a região baseada na imagem binária.
 - Calcular histograma da região extraída.
 - Extrair os *bins* do histograma referente a cor azul (valores de 81 até 130 do canal de cor *H*).
 - Calcula desvio padrão do histograma azul, caso for maior do que o limiar, marcar como área de mar/céu.
5. Juntar todas as regiões marcadas como mar/céu através de operação lógica *OU*, gerando máscara final.
6. Adicionar borda a máscara final, extrair e preencher contornos.
7. remove borda da máscara final.
8. extraí da imagem original a região através de operação lógica *AND* da máscara final e imagem original.

3.2 Criação do vetor de características

A etapa de pré-processamento é aplicada em busca de melhoras relevantes para a etapa posterior de aplicação dos algoritmos. Levando em consideração que o objeto de análise é uma imagem e estas são compostas por *pixels*, menor elemento encontrado em uma fotografia, a sua representação se dá pelo sistema de cores no qual a fotografia foi gerada inicialmente.

Normalmente, as câmeras captam imagens no espaço de cores RGB, no entanto através da aplicação de técnicas de pré-processamento de imagens podemos transformar o sistema inicial em diferentes outros sistemas de cor, em busca de um modelo de cor que possa gerar melhores resultados. Desta forma, a conversão dos possíveis espaços de cores em outros é uma técnica de pré-processamento aplicada neste projeto, tendo principalmente as conversões para RGB, XYZ, LAB e HSV.

O vetor-característica é a estrutura de dados que descreve os elementos da imagem, é comparando-os que definimos quão igual ou diferente são os elementos da imagem. Neste trabalho, os vetores-característica dos *pixels* são representados pelo sistema de cores da imagem. Também é possível combinar diversos sistema de cores em um único vetor-característica, criando uma composição de comparação para buscar combinações que possam melhorar o resultado final do agrupamento. São os vetores-característica que servem de entrada para os algoritmos de agrupamento analisarem e decidirem a distribuição entre os grupos da solução.

3.3 Segmentação por algoritmos de agrupamento com escolha automática do número de grupos

3.3.1 X-Means

O algoritmo *X-means* trabalha com uma única entrada de dados, especificando o limite máximo de K no qual o usuário considera que o verdadeiro número de grupos está contido, o resultado do algoritmo apresenta os centros dos agrupamentos e também o número ideal de K dentro do alcance máximo definido pelo usuário. Tal solução é alcançada a partir de um *score* calculado através de um critério de seleção de modelos tal como *BIC* (*Bayesian Information Criterion*), critério adotado pelo *X-means* (PELLEG; MOORE, 2000)

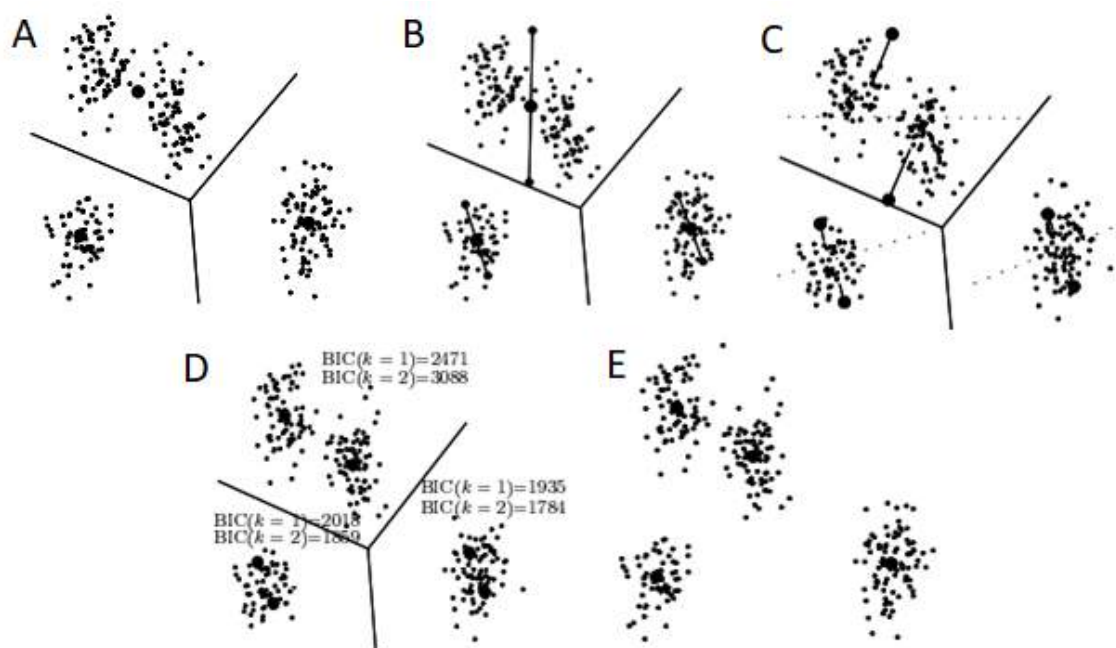
O algoritmo possui uma abordagem *Top-down*, de forma que todos os elementos do agrupamento pertencem a um único grupo inicialmente e no decorrer das iterações, mais agrupamentos são adicionados até que o número máximo de K , correspondente ao número de grupos, seja atingido. Toda vez que um novo grupo é adicionado,

o novo modelo gerado é analisado e aquele que obtiver o melhor *score* para a métrica BIC (Seção 3.3.1.1) será armazenado. A partir do momento que o limite máximo for atingido, aquele modelo armazenado será considerado como solução, contendo os agrupamentos e o número de grupos K ideal.

X-means possui duas operações principais que são executadas diversas vezes até que o limite máximo de K seja alcançado, elas são: *Improve-params* e *Improve-structure*. O *Improve-params* consiste em aplicar o algoritmo *K-means* convencional para alcançar a convergência dos elementos em seus respectivos grupos do modelo atual. O *Improve-structure* atua para encontrar se há a necessidade de criar novos agrupamentos, de forma que os novos grupos serão gerados a partir da divisão de outro agrupamento já existente (PELLEG; MOORE, 2000).

Improve-structure começa dividindo cada grupo existente em dois grupos filhos, ambos os filhos serão movidos numa distância proporcional ao tamanho da região do grupo pai (original) em direções opostas. Em seguida, para cada região do grupo pai será aplicado um *K-means* local (para $k = 2$) envolvendo os seus grupos filhos, desta forma, os grupos filhos competirão somente pelos elementos cujo anteriormente pertenciam ao grupo que os originou. Depois da convergência, é feita a avaliação de modelo para cada grupo antes de ser dividido e para os pares de filhos originados da divisão. Com os resultados em mão, a comparação deles irá definir se há a necessidade de manter a divisão do grupo ou permanecer no estado anterior a partição, caso o grupo original for o vencedor, os filhos são descartados, do contrário, os filhos permanecem e o original é descartado.

Exemplificado na Figura 32 o que acontece com o conjunto de dados ao aplicar *X-means*. Na Figura 32-A) mostra o estado atual do conjunto de dados, Figura 32-B mostra a divisão dos grupos pai em seus dois grupos filhos e o distanciamento do mesmo em proporção a região, já na Figura 32-C já mostra a posição atual dos centros de cada grupo filho após a convergência do *K-means* para $k = 2$. Na Figura 32-D mostra os resultados encontrados pela avaliação do modelo calculado para antes da divisão e depois, por fim, a Figura 32-E mostra o resultado final obtido após decidir quais agrupamentos permanecerão divididos ou unificados.

Figura 32 – Ilustração dos acontecimentos no algoritmo *X-means*.

Fonte – Adaptado de (PELLEG; MOORE, 2000)

O pseudocódigo do algoritmo *X-means* é descrito a seguir:

1. Inserir número máximo de grupos
2. K inicialmente igual a 1
3. Até que o número máximo de grupos seja atingido ou não tenha alteração no número de grupos, faça:
 - Divide cada grupo em dois novos grupos com centros baseado nos elementos do seu conjunto.
 - Aplica o algoritmo *2-Means* localmente até a sua convergência em cada região do grupo previamente dividido.
 - Calcula a métrica *BIC* (*Bayesian Criterion Information*) dos grupos antes de serem divididos e dos novos grupos após a separação.
 - Compara os resultados da métrica *BIC* entre o grupo original e depois da divisão, se o grupo original tiver maior valor da métrica *BIC*, descarta a divisão e mantém o original, se não, descarta o original e mantém a divisão.
 - Armazena o estado atual como uma possível solução.
4. Calcula o *BIC* para as soluções como um todo, a que tiver o maior resultado é a resposta final.

A opção de parar a execução do laço sem chegar ao número máximo de grupos acontece em caso de, após a divisão de todos os grupos, o valor da métrica para nenhuma das divisões forem maiores do que o valor da métrica do grupo original.

3.3.1.1 Bayesian Information Criterion

Dado uma base de dados D e um conjunto de modelos M , cada modelo M_j corresponde a uma solução com diferentes valores de k , uma das formas de calcular o melhor modelo é através do BIC (*Bayesian information criterion*), fornecendo uma probabilidade a posteriori de $P[M_j/D]$ mais próxima da normalização através da Equação 3.1:

$$BIC(M_j) = L(D) - \frac{p_j}{2} \cdot \log R \quad (3.1)$$

Em que $L(D)$ é a log-verossimilhança do conjunto de dados de acordo com o j -ésimo modelo e p_j é o número de parâmetros no modelo M_j e R indica o número total de elementos da base de dados. O parâmetro p_j é obtido através da soma da Equação 3.2:

$$p_j = (K - 1) + dim * K + 1 \quad (3.2)$$

em que dim representa o número de atributos que cada elemento da base de dados possui e K o número total de grupos.

Para realizar o cálculo da Log-verossimilhança para o modelo j , basta somar todos os resultados da log-verossimilhança para os grupos contidos neste modelo através da Equação 3.3

$$L(D_n) = -\frac{R_n}{2} \log(2\pi) - \frac{R_n \cdot dim}{2} \log(\sigma^2) - \frac{R_n - K}{2} + R_n \log R_n - R_n \log R \quad (3.3)$$

R_n indica a quantidade de elementos pertencentes ao grupo analisado, dim indica quantos parâmetros cada elemento possui, K o número total de grupos da solução, R a quantidade total de elementos da base dados e por fim, σ^2 , indicando a estimativa máxima da verossimilhança para a variância com base no modelo gaussiano esférico e pode ser calculado através da Equação 3.4 .

$$\sigma^2 = \frac{1}{R - K} \sum (x_i - \mu_{(i)})^2 \quad (3.4)$$

onde $\mu_{(i)}$ indica o centro do grupo cujo o elemento x_i pertence.

3.3.2 Determinação do número de *clusters* utilizando *k-means* e aplicação em segmentação de imagens coloridas

Esse algoritmo, nomeado neste trabalho de *Auto Group Segmentation*, também possui uma abordagem *Top-Down*, de forma que todos os elementos começam como um único grupo e a partir deste estado, os grupos são divididos, gerando novos conjuntos de agrupamentos. O *Auto group segmentation* é baseado no algoritmo *K-means* clássico, no entanto possui características para sobrepor algumas limitações deste outro, como por exemplo, a necessidade de especificar a quantidade de agrupamentos.

O algoritmo resolve o problema do número de grupos através da incorporação de um método de medida de validação, utilizando medida de distâncias de *intra-cluster* e *inter-cluster*. A medida de *intra-cluster* consiste em calcular a compacticidade dos agrupamentos, ou seja, o quão próximo os elementos de um grupo são do centro do seu agrupamento. Já a medida *inter-cluster* trabalha em calcular a distância entre os grupos, considerando que queremos a maior distância entre eles o possível, pois quanto mais distante um do outro, mais bem dividido foi o agrupamento (RAY; TURI, 2000).

A medida *intra-cluster* pode ser calculada através da Equação 3.5.

$$intra = \frac{1}{N} \sum_{i=1}^K \sum_{x \in C_i} \|x - z_i\|^2 \quad (3.5)$$

Em que N é o número de elementos do conjunto de dados, K é o número de grupos, z_i é o centro do agrupamento C_i . O cálculo basicamente consiste na média das distâncias entre os elementos e seus respectivos centros de grupos, considerando que quanto menor for a média, mais compacto são os agrupamentos.

Para calcular a medida *inter-cluster* é preciso aplicar a Equação 3.6

$$inter = \min(\|z_i - z_j\|^2), i = 1, 2, \dots, K - 1 \text{ e } j = i + 1, \dots, K \quad (3.6)$$

As distâncias entre os centros dos agrupamentos são calculadas e é tomada a menor distância entre eles para ser a medida de *inter-cluster*, levando em consideração que quanto maior for o valor, melhor é a medida.

Utilizando as duas medidas definidas anteriormente, pode-se compô-las de uma forma que nos diga se aquele conjunto de agrupamentos é uma boa solução. O *Auto group segmentation* as combina pela razão demonstrada na Equação 3.7.

$$Validity = \frac{intra}{inter} \quad (3.7)$$

Utilizando essa razão, minimiza-se o valor da validação, pois tenta-se minimizar o valor de *intra-cluster* enquanto se deseja maximizar o valor de *inter-cluster*. Desta forma, o modelo de agrupamento desejado é aquele cujo a medida de validação for a menor possível, determinando o valor ideal do número de grupos (RAY; TURI, 2000).

O algoritmo foi inicialmente produzido aplicando o sistema de cores RGB, no entanto, ele é facilmente modificado para incorporar diversos modelos de cores (RAY; TURI, 2000). O funcionamento do algoritmo é dado a partir da inserção do número máximo de grupos que a solução poderá possuir, a partir disso, em cada iteração, um dos grupos pertencentes a solução atual será particionado em dois, até que o limite máximo de grupos especificado seja alcançado. O critério que define qual grupo será dividido é aquele que possuir a maior taxa de variância dos elementos em relação ao centro do seu grupo, calculado a partir da Equação 3.8.

$$\sigma_{ij}^2 = \frac{1}{N_i} \sum_{x \in C_i} (x - z_{ij})^2 \text{ para } i = 1, 2, \dots, K \text{ e } j = \dim(\text{pixel}) \quad (3.8)$$

N_i representando o número de *pixels* no grupo C_i , x é o vetor com os valores de cada canal de cor e z indica o centro do grupo.

Através da Equação 3.8 será obtida a variância de cada canal de cor existente na imagem. Para finalizar, basta calcular a média dos resultados, desta forma, encontrando o valor de variância final. Com o grupo de maior variância selecionado, este será dividido em dois novos grupos com seus centros inicialmente especificados através da Equação 3.9.

$$\text{centro1}_i = (z_{ij} - a_j, \dots, z_{in} - a_n) \text{ para } j = 1, 2, \dots, n \quad (3.9)$$

$$\text{centro2}_i = (z_{ij} + a_j, \dots, z_{in} + a_n) \text{ para } j = 1, 2, \dots, n$$

j representa o número de canais de cor e a_j são constantes que levam em consideração o maior e o menor valor para seus respectivos canais de cor. As constantes a_j são dadas através do menor valor entre a metade de $z_{ij} - \min_j$ e $z_{ij} - \max_j$, sendo \min_j o menor valor atribuído a um dos *pixels* do j -ésimo canal de cor e \max_j , o maior (RAY; TURI, 2000). Esta etapa irá fornecer dois novos centros de grupos bem divididos, mas ainda pertencentes a área do grupo antes da divisão.

Após a definição das equações, o pseudocódigo do algoritmo *Auto Group Segmentation* é dado a seguir:

1. Inicialmente número de grupos $K = 2$.
2. Número máximo de grupos é fornecido.

3. Enquanto K for menor ou igual ao número máximo de grupos, faça
 - Calcular a variância de todos os grupos separadamente, comparando cada componente ao seu centro.
 - Selecionar o grupo cujo a variância tem o maior valor.
 - Calcular os dois novos centros.
 - Aplicar o *K-Means* para o novo conjunto de grupos de forma global.
 - Calcular o valor da métrica de validação para a nova possível solução
4. Encontrar um máximo local nos valores de validação, de forma que: validação $(K-1) < \text{validação}(K) > \text{validação}(K+1)$ para $K \geq 3$.
5. Retorna a solução que possuir menor valor de validação a partir do máximo local.

Os passos 4 e 5 são necessários pois os valores da métrica de validação eram baixíssimos para agrupamentos com pequenas quantidades de grupos. Isto acontece devido ao grande valor encontrado através da Equação 3.6 para pequenas quantidades de grupos na solução (RAY; TURI, 2000).

3.3.3 Segmentação automática de imagens coloridas

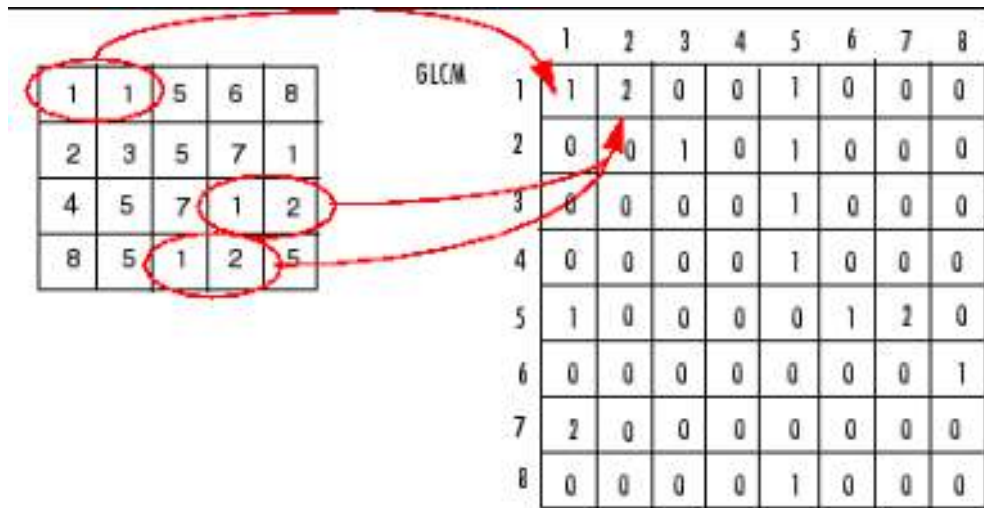
O algoritmo proposto por Kalaivani (2014) consiste de duas principais fases, na primeira é gerado o número de grupos K que será equivalente a quantidade de regiões na qual a imagem será segmentada e, em seguida, K é passado para a segunda fase, responsável por executar um algoritmo de agrupamento chamado de *Fask K-means*, particionando os *pixels* da imagem em K segmentos de região (KALAIVANI; CHITRAKALA, 2014).

O processo para encontrar o número de grupos K recebe como entrada uma imagem no sistema de cores RGB, esta é transformada para tons de cinza, representando principalmente as características de luminosidade presentes na imagem. A partir da imagem em tons de cinza, é aplicado um pré-processamento para remoção de ruídos possivelmente gerados pela conversão da imagem. A técnica de remoção de ruídos foi definida pelo filtro de mediana, pois tal abordagem permite que a imagem permaneça com suas informações de bordas (KALAIVANI; CHITRAKALA, 2014), necessárias para o passo seguinte.

Com a imagem filtrada em mãos, é extraído a matriz de co-ocorrência dos níveis de cinza (*Gray level co-occurrence matrix*, GLCM) (HARALICK; SHANMUGAN; DINS-TEIN, 1973). GLCM é um método estatístico para extrair informações de textura da imagem baseado nas relações espaciais entre os *pixels*. O algoritmo GLCM consiste

em uma contagem de ocorrências de pares de *pixels* com valores de intensidades específicas por toda a imagem. A contagem é armazenada em uma matriz $M \times M$ em que M representa a quantidade de tons de cinza presentes na imagem. A Figura 33 exemplifica uma extração de GLCM para uma imagem com 8 tons de cinza.

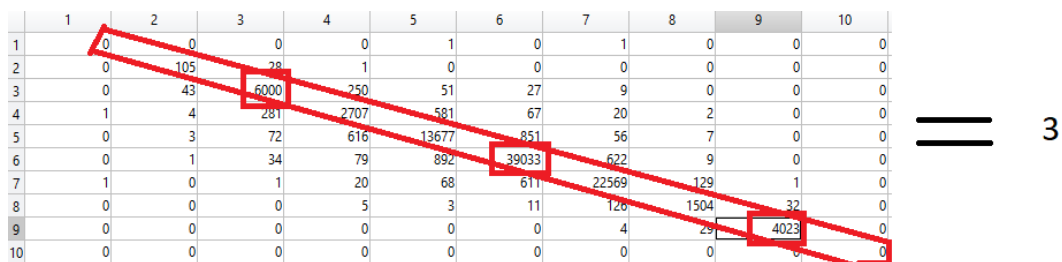
Figura 33 – Exemplo de contagem de uma GLCM levando em consideração os *Pixels* na horizontal.



Fonte – <https://www.mathworks.com/help/images/ref/referencetoh38.gif>

Após a extração da GLCM, somente a contagem dos elementos que residem na diagonal da matriz são relevantes, pois eles representam quando tons de cinza da mesma intensidade se repetem em sequência, caracterizando uma região da imagem de textura similar. Então é feita uma busca nos elementos da diagonal procurando máximos locais, ou seja, quantas vezes uma das contagens da diagonal for maior do que a contagem dos seus vizinhos, exemplificados na Figura 34. A quantidade de grupos K é definida pela quantidade de máximos locais encontrados na diagonal.

Figura 34 – Exemplo do GLCM com seus máximos locais na diagonal destacados e a contagem dos máximos locais.



Fonte – O Autor.

O pseudocódigo da primeira fase do algoritmo é descrito a seguir:

1. Carrega imagem em RGB
2. Converte imagem RGB para tons de cinza
3. Aplica filtro de mediana para remover possíveis ruídos
4. Calcula matriz de co-ocorrência dos níveis de cinza
5. Extrai os valores da contagem da diagonal
6. Contabiliza a quantidade de máximos locais encontrados.

Com o fim da primeira etapa, obtendo o número de grupos que a imagem será dividida, se dá o início da segunda etapa. Esta etapa é responsável pela segmentação da imagem, aplicando um algoritmo de agrupamento conhecido como *Fast K-means Clustering algorithm* (KALAIVANI; CHITRAKALA, 2014). Tal algoritmo foi selecionado devido a natureza das imagens, pois cada pixel contida nela será transformado em um elemento do conjunto de dados que será agrupado. Consequentemente, dependendo das dimensões de largura e altura, o conjunto de dados pode possuir milhões de elementos.

O *Fast K-means Clustering algorithm* tem como função reduzir o custo computacional através do refinamento em dois estágios. O primeiro estágio é para selecionar as melhores posições iniciais para os centros dos grupos através de uma execução do algoritmo *K-means* para uma quantidade reduzida de elementos da base de dados extraídos de forma aleatória. Com o tamanho da base de dados reduzida, o agrupamento tenderá a convergir para um resultado rapidamente.

Após a finalização deste primeiro estágio, são extraídos os centros dos grupos gerados anteriormente, e a partir disto, é executado novamente o *K-means* que recebe como entrada os centros dos grupos e também a base de dados completa. Esta abordagem foi selecionada pois com os centros de grupos previamente definidos, o algoritmo *K-means* tenderá a convergir também mais rápido para o conjunto de dados inteiros. Como resultado, teremos a base de dados rotulada com seus respectivos centros, permitindo a segmentação da imagem.

O pseudocódigo para o algoritmo *Fast K-means* vem a seguir:

1. Extrai aleatoriamente elementos da base de dados definidos por uma porcentagem do total.
2. Aplica *K-means* até a convergência.
3. Extrai os centros de grupos gerados.

4. Aplica novamente o *K-means* recebendo como entrada os centros gerados anteriormente e a base de dados completa.

4 Avaliação Experimental

Neste capítulo serão apresentadas as informações relacionadas a base de dados, metodologia dos experimentos, como também a métrica de avaliação e o ambiente de desenvolvimento utilizado. Serão demonstradas características encontradas nas imagens coletadas, a metodologia do desenvolvimento dos experimentos, além da métrica selecionada para avaliar a qualidade dos agrupamentos e também as tecnologias utilizadas para desenvolver o trabalho.

4.1 Base de Dados

A primeira etapa do projeto consistiu da criação de uma base de dados relevante ligada ao problema em questão, necessitando de imagens de regiões litorâneas contendo banhistas no mar. Portanto, foi realizado a coleta dessas imagens nas regiões litorâneas do Recife em diversas ocasiões, áreas e posicionamento de câmera, permitindo que a base de dados contemple diferentes momentos e possíveis eventos que possam acontecer.

A base de dados contém 136 imagens com resolução de 1000 x 700, o número de imagens foi definido desta forma devido a necessidade de rotulação manual e tempo de execução dos experimentos. As imagens são compostas por diferentes características e componentes presentes, listados a seguir:

- Imagens com pouca e grande interferência na areia (Figura 35);
- Imagens com e sem arrecifes de corais (Figura 36);
- Imagens com Objetos sobrepondo regiões de água (Figura 37);
- Imagens com diferentes níveis de marés (Figura 38);
- Imagens com grandes regiões de praia com areia molhada (Figura 39);
- Imagens com diferentes incidências de luz (Figura 40);
- Imagens com Distanciamento diferenciado, aumentando e reduzindo tamanho dos banhistas (Figura 40);

Figura 35 – Imagens com (direita) e sem (esquerda) ruídos.



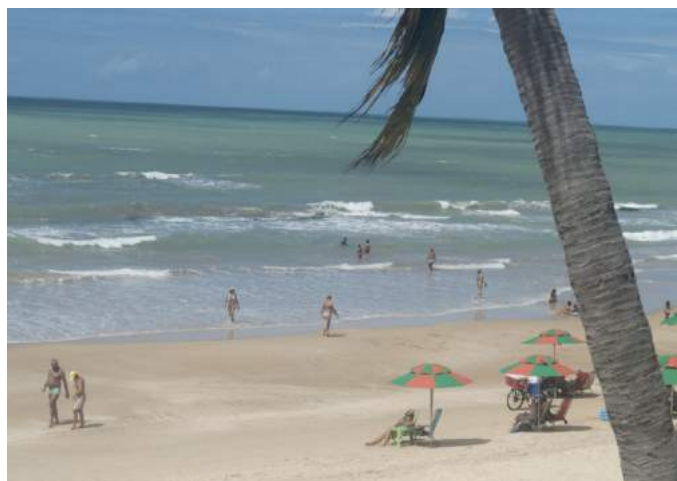
Fonte – O autor

Figura 36 – Imagens com (esquerda) e sem (direita) arrecifes de corais.



Fonte – O autor

Figura 37 – Imagens com sobreposição.



Fonte – O autor

Figura 38 – Imagens com maré baixa (direita) e alta (esquerda).



Fonte – O autor

Figura 39 – Imagens com areia molhada.



Fonte – O autor

Figura 40 – Imagens com alta (esquerda) e baixa(direita) incidência de luz.



Fonte – O autor

Figura 41 – Imagens com longe (esquerda) e próximo (direita) distanciamento.



Fonte – O autor

Após a coleta da base de dados, o processo de rotulação das imagens é iniciado. Com o auxílio da ferramenta chamada *LabelMe* (RUSSELL et al., 2008) para identificar e rotular as pessoas presentes nas imagens, é desenhado um polígono que circunda a região de interesse, destacando os banhistas (Figura 42).

Figura 42 – Imagem rotulada.



Fonte – O autor.

A rotulação se faz necessária para os demais componentes do projeto maior e não será utilizado neste trabalho, pois algoritmos de agrupamento são naturalmente

classificados como aprendizagem não-supervisionada, ou seja, não é utilizado uma rotulação prévia para sua execução.

4.2 Metodologia experimental

Alguns dos algoritmos apresentados para segmentação possuem alguns parâmetros próprios, como o algoritmo hierárquico aglomerativo, com o número de sementes iniciais para realizar a distribuição, e também, o algoritmo de segmentação de imagens coloridas, variando a quantidade dos níveis de tons de cinza disponíveis para o descritor GLCM.

Os experimentos também levaram em consideração como será o vetor de características de cada elemento da imagem (*pixels*). Inicialmente todas as imagens utilizam o sistema de cores RGB e através dos métodos de pré-processamento para conversão, serão avaliados outros modelos de cores em busca daqueles que apresentam resultados mais promissores para o agrupamento. É também possível combinar para o mesmo *pixel* diversos sistemas de cores simultaneamente para representar o seu vetor de características.

Os experimentos ocorreram em torno do controle dos parâmetros individuais dos algoritmos, como também o vetor característica que representará os *pixels* e a remoção ou não da faixa de areia da praia. Esses experimentos serão descritos nas próximas subseções.

Para cada combinação de algoritmo e vetor de características, como também modificações de parâmetros se existir, será aplicado a base de dados inteira, contendo as 136 imagens. Ao fim do agrupamento de cada imagem, o valor do índice de *Dunn* será extraído e armazenado. No fim de todas as imagens, é feito o cálculo da média dos índices de *Dunn*, assim como o desvio padrão, obtendo o resultado, mais detalhes na Seção 4.4.

4.2.1 Experimento A

O experimento A consiste dos testes dos algoritmos e combinações de possíveis vetores de característica para imagens sem a remoção da faixa de areia da praia.

4.2.1.1 Experimento A-1

O experimento A-1 consiste em avaliar os resultados dos agrupamentos gerados pelo algoritmo hierárquico aglomerativo, sem a remoção da faixa de areia, variando o parâmetro de sementes iniciais para valores de 20, 30 e 40 sementes. Além disso, para cada valor de semente avaliado, também será testado o seu desempenho

para com diferentes vetores de característica dos *pixels*, variando entre RGB, LAB e a combinação de RGB e LAB.

Quando se trabalha com imagens de alta resolução, a inicialização do algoritmo com cada *pixel* representando um grupo se torna inviável. Então, a inicialização do algoritmo é realizada pela técnica de sementes, consistindo de escolher aleatoriamente um elemento para cada semente. Em seguida, os demais elementos são distribuídos entre todas elas de acordo com a métrica da menor distância. O restante do algoritmo continua de forma padrão.

4.2.1.2 Experimento A-2

O experimento A-2 visa avaliar os resultados dos agrupamentos gerados pelo algoritmo de segmentação automática de imagens coloridas, sem a remoção da faixa de areia, variando o parâmetro de possíveis tons de cinza no descritor GLCM. Quanto maior a quantidade, mais detalhado será a análise da imagem, variando entre 20, 30 e 40 tons de cinza. Também será avaliado seu desempenho com diferentes vetores de característica dos *pixels*, variando entre RGB, LAB e a combinação de RGB e LAB.

4.2.1.3 Experimento A-3

O experimento A-3 tem como finalidade avaliar os resultados dos agrupamentos gerados pelos demais algoritmos que não possuem parâmetros de inicialização: Hierárquico divisivo, *X-means* e *Auto group segmentation*. São avaliados somente da alteração dos vetores de características para representação dos *pixels*, sem remoção de areia.

Para o algoritmo hierárquico divisivo e o *X-means*, existe a necessidade de selecionar uma quantidade máxima de número de grupos para servir como critério de parada. Para ambos os casos, tal quantidade ficou limitada a 15 grupos, pois mais do que isso, a imagem se tornaria muito particionada, não sendo o objetivo da segmentação deste trabalho.

Em relação ao *X-means* o número de grupos máximos define o limite máximo de possíveis grupos, mas não significa que sempre terá tal quantidade. Já para o hierárquico divisivo, o limite máximo indica a profundidade da árvore gerada, a quantidade de grupos é definido pelo nível de corte de tal árvore, definida para 0.2 a profundidade do corte.

4.2.2 Experimento B

O experimento B consiste na avaliação dos algoritmos nas combinações de possíveis vetores de característica para imagens com a remoção da faixa de areia

da praia com o objetivo de avaliar se a etapa de remover a areia da praia impacta na qualidade dos agrupamentos gerados por esses algoritmos e o possível grupo dos banhistas.

4.2.2.1 Experimento B-1

O experimento B-1 consiste em avaliar os resultados dos agrupamentos gerados pelo algoritmo hierárquico aglomerativo, com a remoção da faixa de areia, variando o parâmetro de sementes iniciais para valores de 20, 30 e 40 sementes. Além disso, para cada valor de semente avaliado, também será testado o seu desempenho para com diferentes vetores de característica dos *pixels*, variando entre RGB, LAB e a combinação de RGB e LAB.

4.2.2.2 Experimento B-2

O experimento B-2 visa avaliar os resultados dos agrupamentos gerados pelo algoritmo de segmentação automática de imagens coloridas, com a remoção da faixa de areia, modificando o parâmetro de possíveis tons de cinza no descritor GLCM, variando entre 20, 30 e 40 tons de cinza. Também será testado o seu desempenho com diferentes vetores de característica dos *pixels*, variando entre RGB, LAB e a combinação de RGB e LAB.

4.2.2.3 Experimento B-3

O experimento B-3 tem como finalidade avaliar os resultados dos agrupamentos gerados pelos demais algoritmos: Hierárquico divisivo, *X-means* e *Auto group segmentation*. Tais algoritmos não possuem um parâmetros de inicialização, então serão avaliadas somente da alteração dos vetores de características para representação dos *pixels*, com a remoção da praia.

Assim como no experimento A-3, o critério de parada para os algoritmos hierárquico divisivo e o *X-means*, ficou definido como 15 grupos.

4.3 Ambiente experimental

Foram utilizadas para implementação deste trabalho a ferramenta *Visual studio* 2017 juntamente com a linguagem C++ e a biblioteca *Opencv (Open Source Computer Vision Library)* (BRADSKI et al., 2000). Também, a ferramenta de desenvolvimento *Matlab®*, assim como a linguagem JAVA através do ambiente de desenvolvimento *Eclipse*. A seguir, uma lista do que foi utilizado para cada etapa:

- O Visual studio 2017 com a linguagem C++ juntamente a biblioteca Opencv foi aplicada para o pré-processamento de remoção das faixas de praia das imagens.
- Para implementar os algoritmos Hierárquicos aglomerativo e divisivo, além do *Auto group segmentation*, foi utilizado o Eclipse juntamente à linguagem JAVA.
- O Matlab® serviu para a implementação do algoritmo de segmentação automática de imagens coloridas.

4.4 Métricas de validação

Para avaliar a qualidade dos agrupamentos, foram implementados 4 métricas específicas para agrupamentos não-supervisionados. Elas foram: Separação, Coesão, silhueta (ROUSSEUW, 1987) e índice de *Dunn* (DUNN, 1973). No entanto, devido a resolução das imagens utilizadas, separação, coesão e silhueta se mostraram inviáveis devido ao tempo necessário para o cálculo das métricas. A coesão é calculada através do somatório de todos os componentes do grupo para os outros elementos pertencentes ao mesmo conjunto, já a separação seleciona o grupo mais próximo do conjunto que estiver sendo analisado e realizar o somatório de todas as distâncias entre os elementos do grupo analisado em relação a todos os elementos do outro agrupamento mais próximo.

Levando em consideração a escala das imagens coloridas com dimensões de 1000x700, para cada imagem terá 700.000 *pixels*. Exemplificando para a métrica de coesão, caso um grupo dentre todos da solução tenha 200.000 *pixels*, seria necessário aproximadamente 40 bilhões de operações de cálculo de distância para um único grupo. Desta forma, foi decidido utilizar o índice de *Dunn* nesse trabalho.

O índice de *Dunn* utiliza das distâncias entre centros dos grupos, selecionando a menor distância encontrada entre todos os grupos existentes e o maior diâmetro dentre os grupos. Então é aplicado uma razão da menor distância entre grupos por maior distância entre elementos e seu centro, de forma que quanto maior o valor, melhor o resultado.

Existem diversas formas para calcular o diâmetro de um grupo, neste trabalho foi optado pela média das distâncias entre o centro do grupo e os elementos pertencentes a ele (Equação 4.1).

$$Diam(G_i) = \frac{\sum_{x \in G_i} d(x - c_i)}{N_i} \quad (4.1)$$

G_i é o i -ésimo grupo do agrupamento, c_i é o centro do i -ésimo grupo, N_i é a quantidade de elementos pertencentes ao i -ésimo grupo.

A Equação 4.2 a seguir mostra como calcular o índice de *Dunn*:

$$Dunn(S) = \frac{\min(d(c_i, c_j))}{\max(Diam(G_i))} \quad (4.2)$$

S representando a solução, c_i e c_j são centros de grupos pertencentes a solução S para $i \neq j$.

5 Resultados

Neste capítulo serão apresentados os resultados encontrados na aplicação dos experimentos A e B definidos no Capítulo 4. A estrutura será baseada em três seções principais: i. experimentos A, consistindo dos testes dos algoritmos de agrupamento sem a remoção da faixa de areia da praia e alteração dos vetores de característica; ii. experimentos B, testes dos algoritmos de agrupamento com a remoção da faixa de areia da praia e alteração dos vetores de característica; iii. uma seção para discussão e análise dos resultados obtidos.

A qualidade dos algoritmos são avaliados através da métrica do índice de *Dunn*, apresentando os valores médios obtidos da execução dos algoritmos para o conjunto da base de dados de imagens de praia, como também o valor do desvio padrão, representando a variação entre os resultados.

5.1 Resultados dos experimentos A

Esta seção apresenta os resultados obtidos para o experimento A descrito na seção 4.2.1, consistindo da execução dos algoritmos sem a remoção da faixa de areia das imagens, além da variação das possíveis representações dos vetores de características dos *pixels* com o sistemas de cores, RGB, LAB e RGB+LAB. Os melhores resultados encontrados em cada Tabela são marcados em negrito.

5.1.1 Experimento A-1

No experimento A-1, foi avaliado o resultado para o algoritmo hierárquico aglomerativo, variando o parâmetro de sementes iniciais para valores de 20, 30 e 40 sementes. Além disso, foram avaliados os sistemas de cores RGB, LAB e a combinação dos dois para gerar os possíveis vetores de características em busca de bons resultados (Tabela 2).

Tabela 2 – Experimento A-1 - Resultados de média para o índice de *Dunn* variando o número de sementes e os vetores de características.

	20 sementes		30 sementes		40 sementes	
	Média	σ	Média	σ	Média	σ
RGB	1.6938	0.3697	1.7467	0.3719	1.8153	0.3647
LAB	1.6840	0.3957	1.6906	0.3700	1.7698	0.3470
RGB + LAB	1.1491	0.2789	1.2211	0.2753	1.2173	0.2807

Para cada valor de semente testado, o algoritmo foi executado para diferentes vetores de características compostos pela variação dos sistemas de cores. Os resultados então são demonstrados através da média e desvio padrão (σ) dos valores obtidos para o índice de *Dunn* entre as combinações.

Os resultados obtidos na Tabela 2 mostram que para o vetor de características RGB e LAB, a quantidade de 40 sementes mostraram os resultados mais promissores, tendo média de 1.8153 e 1.7698 respectivamente. Já para o vetor composto de RGB e LAB juntos, 30 sementes obteve o melhor resultado, tendo a média de 1.2211. Os valores para 40 sementes foram os maiores provavelmente pelo fato de que os agrupamentos sofreram uma divisão mais fina dos *pixels* na etapa inicial do algoritmo, permitindo que os agrupamentos finais sejam bem estruturados.

A seguir, um conjunto de imagens representando alguns resultados encontrados para o algoritmo hierárquico aglomerativo, cada exemplo mostrará a imagem original e o grupo que melhor enquadrou os banhistas (Figuras 43, 44, 45, 46).

Figura 43 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado para a combinação de RGB e 20 sementes.



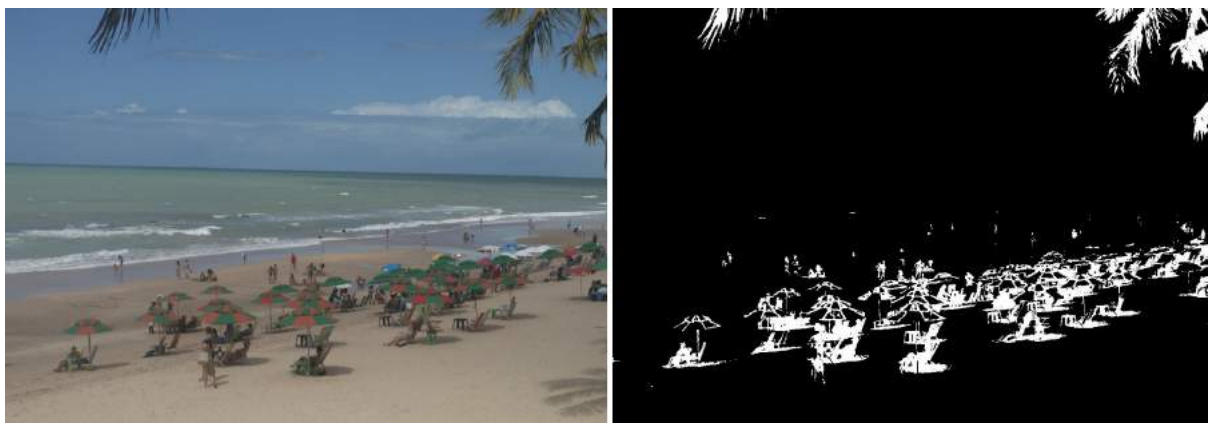
Fonte – O autor

Figura 44 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas (Direita) para um resultado ruim, misturando regiões da água com as pessoas.



Fonte – O autor

Figura 45 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado para a combinação de LAB e 30 sementes.



Fonte – O autor

Figura 46 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado para a combinação de RGB e 40 sementes.



Fonte – O autor

5.1.2 Experimento A-2

No experimento A-2, o algoritmo de segmentação automática de imagens coloridas é avaliado. Para este experimento, foi variado o número de possíveis tons de cinza para a matriz de co-ocorrência dos níveis de tons de cinza do descritor GLCM. Tal variação mostrou que quanto maior a quantidade de tons possíveis, mais detalhado será a imagem, permitindo discernir uma maior quantidade de *pixels* entre diferentes tons de cinza.

Além da variação da quantidade de tons de cinza, também foi variado os possíveis vetores de características, utilizando os sistemas de cores RGB, LAB e a combinação dos dois. A Tabela 3 apresenta os resultados encontrados para o índice de *Dunn*, contendo o valor médio e o desvio padrão (σ) para a base de dados, em relação a combinação das variações entre os níveis de tons de cinza da GLCM e as possíveis representações dos vetores de características.

Tabela 3 – Experimento A-2 - Resultados de média para o índice de *Dunn* variando o número de níveis de tons de cinza da GLCM e os vetores de características.

	20 tons de cinza		30 tons de cinza		40 tons de cinza	
	Média	σ	Média	σ	Média	σ
RGB	2.0640	0.5367	1.6852	0.5016	1.4236	0.4318
LAB	1.7751	0.5387	1.4009	0.5139	1.2070	0.4580
RGB + LAB	1.8156	0.4307	1.5245	0.4479	1.2651	0.4027

Para a métrica aplicada, os resultados para o algoritmo de segmentação automática de imagens coloridas utilizando 20 sementes, em todos os vetores de caracte-

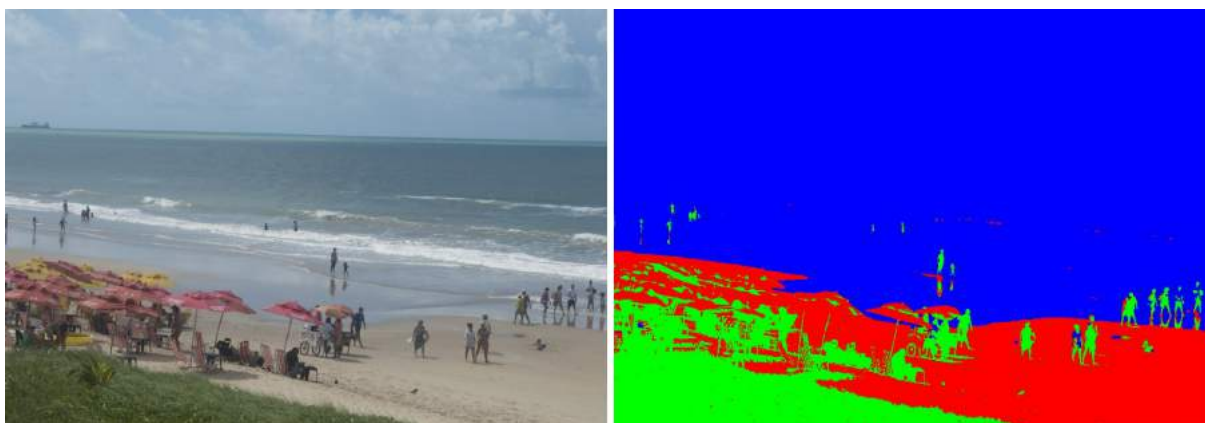
rísticas mostraram os melhores valores, em que RGB teve a média de 2.0640, LAB de 1.7751 e combinação de RGB + LAB de 1.8156.

No entanto, observando a maioria dos resultados de imagens encontrados para a GLCM com 20 tons de cinza, foi percebido que a quantidade de grupos gerados por solução ficou em torno de 1 a 4 grupos. Mesmo que tal solução possua um índice elevado levando em consideração o agrupamento total, uma boa parte das imagens acabaram englobando os banhistas em um grupo maior, sendo eles o nosso principal objetivo.

Como a quantidade dos tons de cinza da GLCM indica a qualidade do detalhamento imagem analisada, quanto maior a quantidade de tons de cinza, mais detalhes são perceptíveis na imagem para o descritor de texturas. Com tal informação em mente, é evidente que o motivo pelo qual uma quantidade baixa de tons de cinza pode ocasionar no englobamento dos banhistas em grupos maiores. Assim sendo, para o algoritmo de segmentação automática de imagens coloridas, não levando em consideração os resultados de 20 tons, os de 30 tons possuem os resultados mais promissores.

A seguir, um conjunto de imagens representando alguns resultados encontrados para o algoritmo de segmentação automática de imagens coloridas, cada exemplo mostrará a imagem original e o resultado completo da segmentação. (Figuras 47, 48, 49, 50, 51).

Figura 47 – Imagem Original (esquerda), imagem totalmente segmentada (direita), grupo de cor verde melhor representou os banhistas para GLCM 20 níveis e sistema de cor RGB.



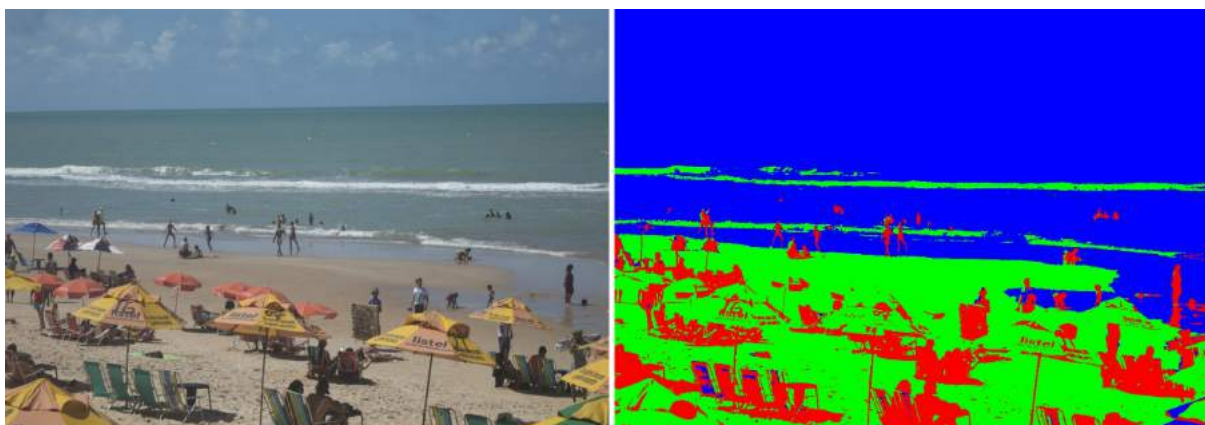
Fonte – O autor

Figura 48 – Imagem Original (esquerda), imagem totalmente segmentada (direita), grupo de cor vermelha melhor representou os banhistas para GLCM 20 níveis e sistema de cor RGB.



Fonte – O autor

Figura 49 – Imagem Original (esquerda), imagem totalmente segmentada (direita), grupo de cor vermelha melhor representou os banhistas para GLCM 20 níveis e sistema de cor RGB.



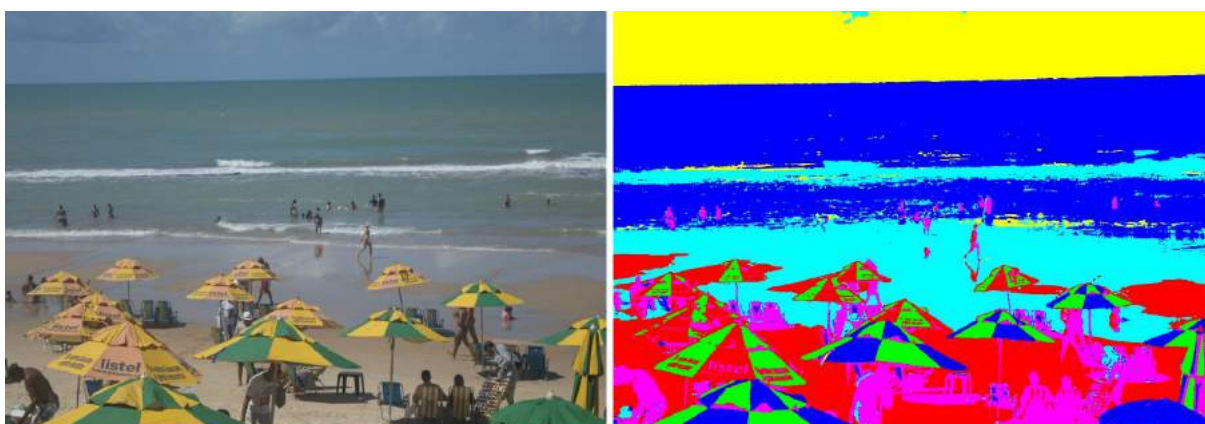
Fonte – O autor

Figura 50 – Imagem Original (esquerda), imagem totalmente segmentada (direita), exemplo de resultado ruim para GLCM 20 níveis e sistema de cor RGB.



Fonte – O autor

Figura 51 – Imagem Original (esquerda), imagem totalmente segmentada (direita), grupo de cor rosa melhor representou os banhistas para GLCM 40 níveis e sistema de cor LAB.



Fonte – O autor

5.1.3 Experimento A-3

Neste experimento, os demais algoritmos que não possuem parâmetros de inicialização são avaliados. Os algoritmos são: *X-means*, Hierárquico divisivo e *Auto group Segmentation*. Para cada um dos algoritmos, foram aplicados os sistemas de cores RGB, LAB e a combinação deles para geração dos possíveis vetores de características que descrevem como os *pixels* são. A Tabela 4 apresenta os resultados encontrados para o índice de *Dunn*, especificando o valor médio e o desvio padrão σ para a base de dados de imagens de praia, variando os vetores de características.

Tabela 4 – Experimento A-3 - Resultados de média para o índice de *Dunn* variando os vetores de características para os algoritmos de *X-means*, Hierárquico divisivo e *Auto group segmentation*, Hierárquico Aglomerativo e Segmentação Automática Colorida

		Média	Desvio Padrão (σ)
X-Means	RGB	1.1669705623036304	0.3335815533946962
	LAB	0.8638510506605973	0.3417794956501355
	RGB + LAB	0.883193331896416	0.38615192601804416
<i>Auto Group Segmentation</i>	RGB	1.689492698865999	0.3707954548385317
	LAB	1.4543577631177707	0.40225784495196193
	RGB + LAB	1.500077602386799	0.3152614421049866
Hierárquico Divisivo	RGB	1.8045059891466342	0.3887619170943952
	LAB	1.7063648912115927	0.448259094086873
	RGB + LAB	1.3090428234181708	0.3596470413168057

Para a métrica aplicada, o algoritmo hierárquico divisivo apresentou os melhores resultados quando utilizado os sistemas de cores RGB e LAB para representar os vetores de características, possuindo média de 1.8045 e 1.7064 respectivamente, enquanto o *Auto Group Segmentation* obteve o resultado mais promissor para a combinação de RGB e LAB como vetor de características, contendo média de 1.5001.

Um aspecto muito importante que deve ser levado em consideração para este experimento são os baixos valores encontrados para o *X-means*. Tais valores foram baixos devido a natureza de divisão dos agrupamentos no âmbito local, forma como o algoritmo resolve a sua divisão de elementos. Como especificado no capítulo 3, o algoritmo *X-means* irá checar para cada grupo existente se há a necessidade de divisão deste grupo em dois novos grupos, caso seja necessário. Os dois grupos gerados pela divisão irão competir para definir a partição de somente os elementos que anteriormente pertenciam ao grupo original.

Desse modo, uma abordagem visando somente os elementos locais, não levará em consideração a solução do agrupamento como um todo, acarretando a redução dos resultados encontrados pelo índice de *Dunn*, pois através da divisão local, a proximidade entre os demais grupos já existentes não é levada em consideração.

A seguir, um conjunto de imagens dos diversos resultados obtidos para os três algoritmos: *X-means*, *auto group segmentation* e hierárquico divisivo (Figuras 52, 53, 54, 55).

Figura 52 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado do algoritmo hierárquico divisivo com vetor de característica RGB.



Fonte – O autor

Figura 53 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um razoável resultado do algoritmo hierárquico divisivo com vetor de características RGB.



Fonte – O autor

Figura 54 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado do algoritmo *Auto group segmentation* com vetor de características LAB.



Fonte – O autor

Figura 55 – Imagem Original (esquerda), imagem binária do grupo que melhor representou as pessoas na água (Direita) para um bom resultado do algoritmo *xmeans* com vetor de características combinado de RGB e LAB.



Fonte – O autor

5.2 Resultados dos experimentos B

Os experimentos B seguem a mesma metodologia aplicada para os experimentos A, adicionado da remoção da faixa de praia, região da imagem que a priori, é desnecessária para a segmentação dos banhistas. É importante salientar que as informações relacionadas a algoritmos e seus comportamentos descritos na seção de experimento A também são válidas para os algoritmos no experimento B.

Como explicado no Capítulo 3, a técnica da remoção da faixa de areia da imagem consiste de um método de pré-processamento, aplicando uma técnica de computação visual para agrupar e analisar as tonalidades de cores presente nos grupos em

busca de regiões com uma presença maior de tons azuis. Os valores de área mínima e desvio padrão para o algoritmo foram especificados empiricamente através de testes, possuindo valor de 10000 e 1200 respectivamente.

É gerado uma máscara binária em que 1 representa regiões de tons azulados e 0 possíveis áreas de areia. Tal máscara permite extrair da imagem original as áreas de mar e céu através de uma operação *AND* entre a máscara e a imagem colorida, de forma que somente os *pixels* com valor 1 na máscara terá cor da imagem colorida na figura resposta, caso contrário, será preto. A seguir, exemplos de resultados encontrados pelo algoritmo de remoção de praia (Figuras 56, 57, 58, 59):

Figura 56 – Imagem original (esquerda), máscara binária (direita), Imagem Resultando da operação *AND* (baixo).



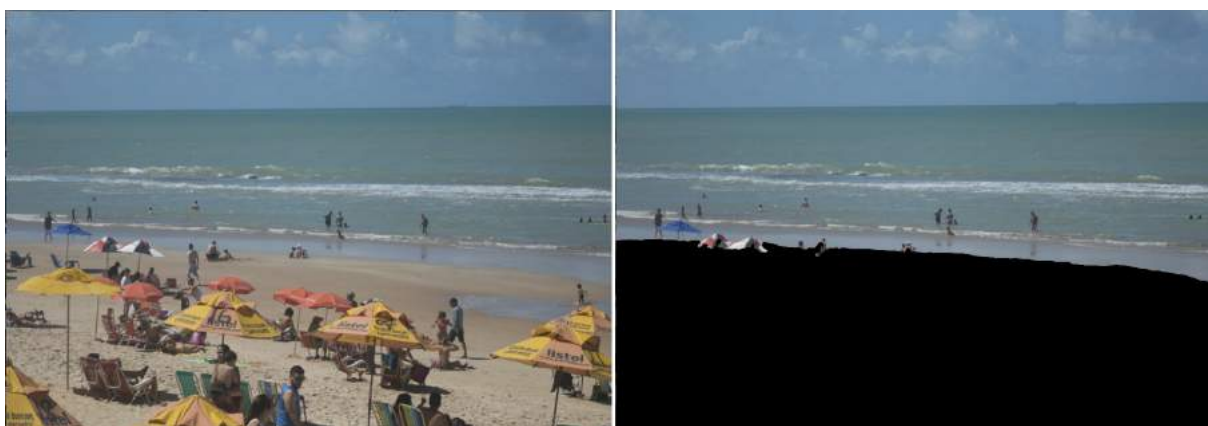
Fonte – O autor

Figura 57 – Imagem original (esquerda), imagem resultante (direita).



Fonte – O autor

Figura 58 – Imagem original (esquerda), imagem resultante (direita).



Fonte – O autor

Figura 59 – Imagem original (esquerda), imagem resultante (direita).



Fonte – O autor

5.2.1 Experimento B-1

No experimento B-1, foi avaliado o resultado para o algoritmo hierárquico aglomerativo, variando o parâmetro de sementes iniciais para valores de 20, 30 e 40 sementes. Além disso, foi aplicado os sistemas de cores RGB, LAB e a combinação dos dois para gerar os possíveis vetores de características em busca de bons resultados (Tabela 5), desta vez aplicando a técnica de remoção das regiões de praia.

Tabela 5 – Experimento B - Resultados de média para o índice de *Dunn* variando o número de sementes e os vetores de características.

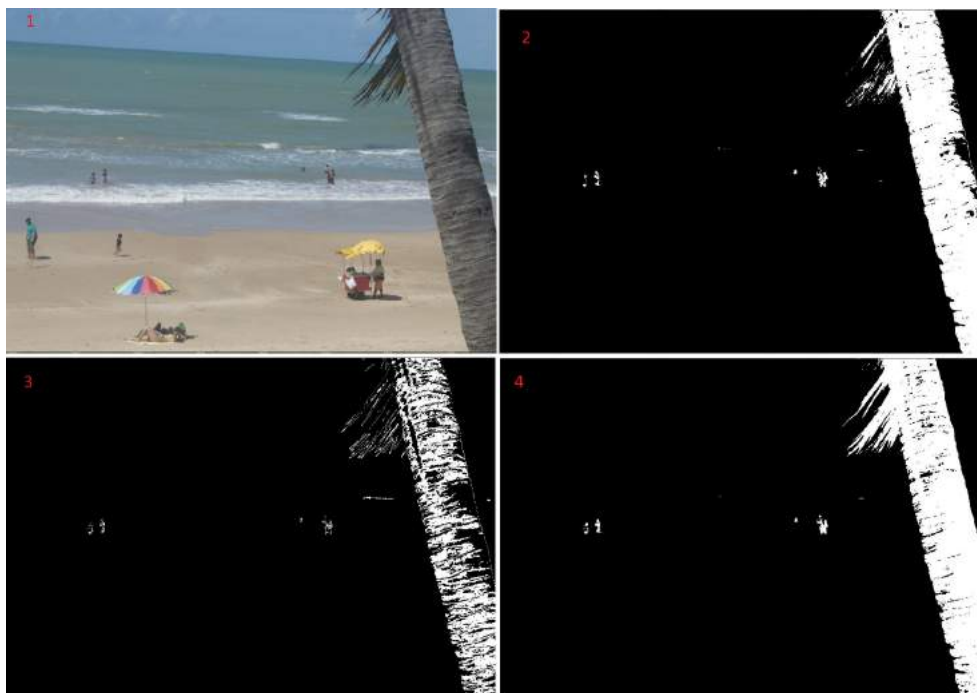
	20 sementes		30 sementes		40 sementes	
	Média	σ	Média	σ	Média	σ
RGB	2.0859	0.8081	2.0854	0.7606	2.0632	0.5106
LAB	2.0514	0.7697	1.9534	0.7351	1.8883	0.6705
RGB + LAB	1.3572	0.4805	1.2822	0.3567	1.2824	0.3552

Para cada valor de semente testado, o algoritmo foi executado para diferentes vetores de características compostos pela variação dos sistemas de cores. Os resultados então são demonstrados através da média e desvio padrão (σ) dos resultados para o índice de *Dunn* entre as combinações, removendo a possível faixa de praia da imagem através de algoritmo de pré-processamento.

Os resultados obtidos na Tabela 5 mostram que para todos os tipos de vetores de características, a utilização de 20 sementes com a remoção da faixa de praia obtiveram os melhores resultados, em que o vetor RGB possui média de 2.0859, LAB de 2.0514 e RGB + LAB tem 1.3572.

A seguir, um conjunto de imagens representando alguns resultados encontrados para o algoritmo hierárquico aglomerativo com remoção de praia, cada exemplo mostrará a imagem original e os grupos que melhor enquadraram os banhistas para cada nível de semente (Figuras 60, 61, 62) e por fim, uma imagem mostrando os melhores agrupamentos dividido por categoria de sementes (Figura 63).

Figura 60 – (1) Imagem Original, Melhores grupos para 20 sementes: (2) LAB, (3) RGB, (4) RGB + LAB.



Fonte – O autor

Figura 61 – (1) Imagem Original, Melhores grupos para 30 sementes: (2) LAB, (3) RGB, (4) RGB + LAB.



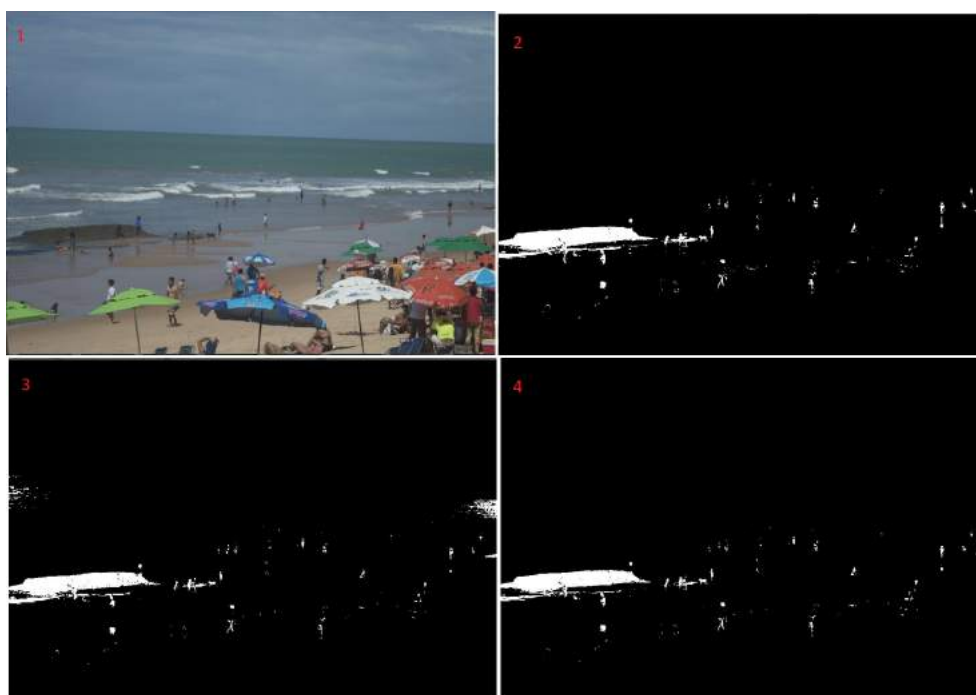
Fonte – O autor

Figura 62 – (1) Imagem Original, Melhores grupos para 40 sementes: (2) LAB, (3) RGB, (4) RGB + LAB.



Fonte – O autor

Figura 63 – (1) Imagem Original, Melhores grupos entre categorias:(2) 20 sementes - RGB + LAB, (3) 30 sementes - RGB + LAB, (4) 40 sementes - RGB e LAB (ambos produziram grupos muito parecidos).



Fonte – O autor

5.2.2 Experimento B-2

No experimento B-2, o algoritmo de segmentação automática de imagens coloridas é avaliado. Para este experimento, foi variado o número de possíveis tons de cinza para a matriz de co-ocorrência dos níveis de tons de cinza. Além da variação da quantidade de tons de cinza, também foi variado os possíveis vetores de características, utilizando os sistemas de cores RGB, LAB e a combinação dos dois.

A Tabela 6 apresenta os resultados encontrados para o índice de *Dunn*, contendo o valor médio e o desvio padrão (σ) para a base de dados, em relação a combinação das variações entre os níveis de tons de cinza da GLCM e as possíveis representações dos vetores de características para imagens com remoção da praia.

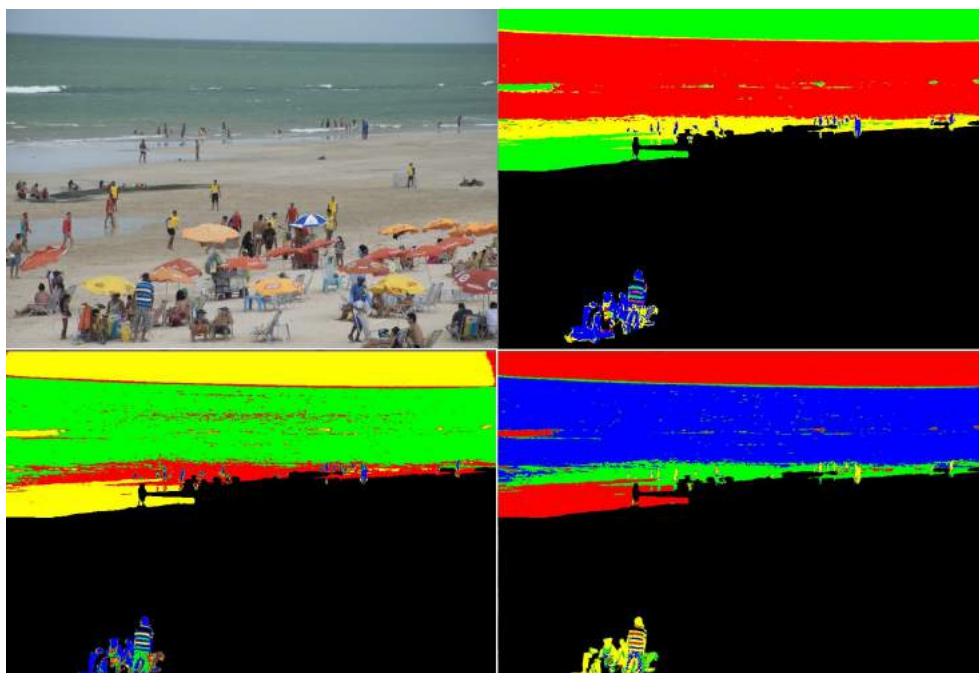
Tabela 6 – Experimento B-2 - Resultados de média para o índice de *Dunn* variando o número de níveis de tons de cinza da GLCM e os vetores de características com a remoção da faixa de praia.

	20 tons de cinza		30 tons de cinza		40 tons de cinza	
	Média	σ	Média	σ	Média	σ
RGB	2.2269	1.4222	1.7786	1.0395	1.2972	0.5621
LAB	1.7872	0.9335	1.4853	0.8103	1.0491	0.5109
RGB + LAB	1.9478	1.1025	1.5857	0.9016	1.1277	0.4660

Para a métrica aplicada, os resultados para o algoritmo de segmentação automática de imagens coloridas levando em consideração todos os vetores de características, 20 tons de cinza obteve os melhores valores, possuindo uma média de 2.2269 para o vetor RGB, 1.7872 para LAB e por fim, 1.9478 quando utilizado RGB + LAB. No entanto, assim como no Experimento A-2, utilizar a quantidade de 20 tons de cinza não foi o ideal, gerando pouquíssimos grupos, de forma que a maioria foram de 2 grupos.

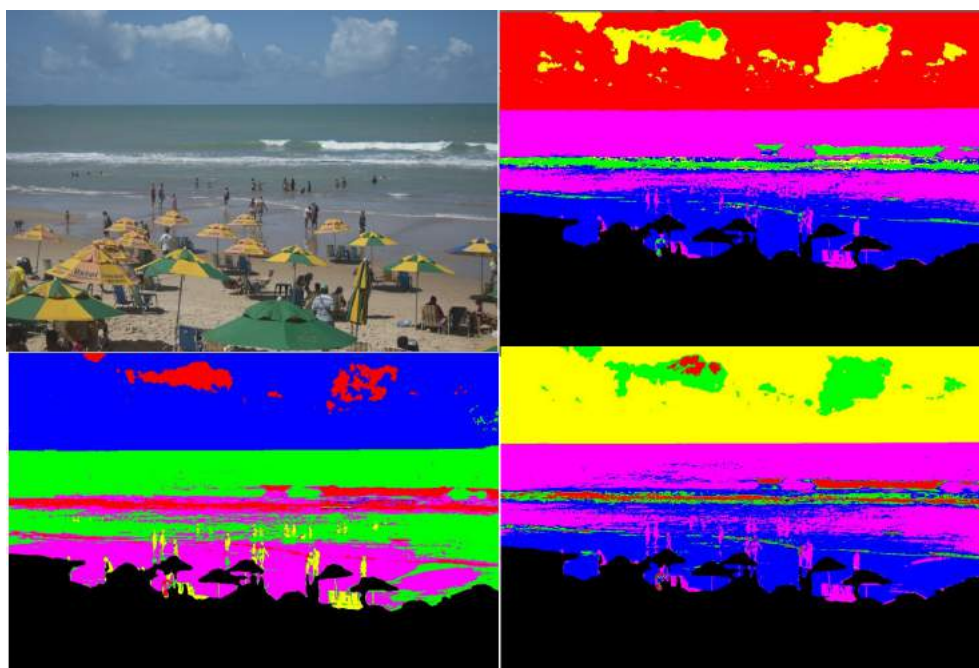
A seguir, um conjunto de imagens representando alguns resultados encontrados para o algoritmo de segmentação automática de imagens coloridas com remoção de praia, cada exemplo mostrará a imagem original e os grupos que melhor enquadraram os banhistas para cada nível de tons de cinza da GLCM (Figuras 64, 65, 66) e por fim, uma imagem mostrando os melhores agrupamentos dividido por categoria de sementes (Figura 67).

Figura 64 – (1) Imagem Original, Melhores segmentações para 20 níveis de cinza: (2) LAB representado pelo grupo azul, (3) RGB representado pelo grupo azul, (4) RGB + LAB representado pelo grupo amarelo.



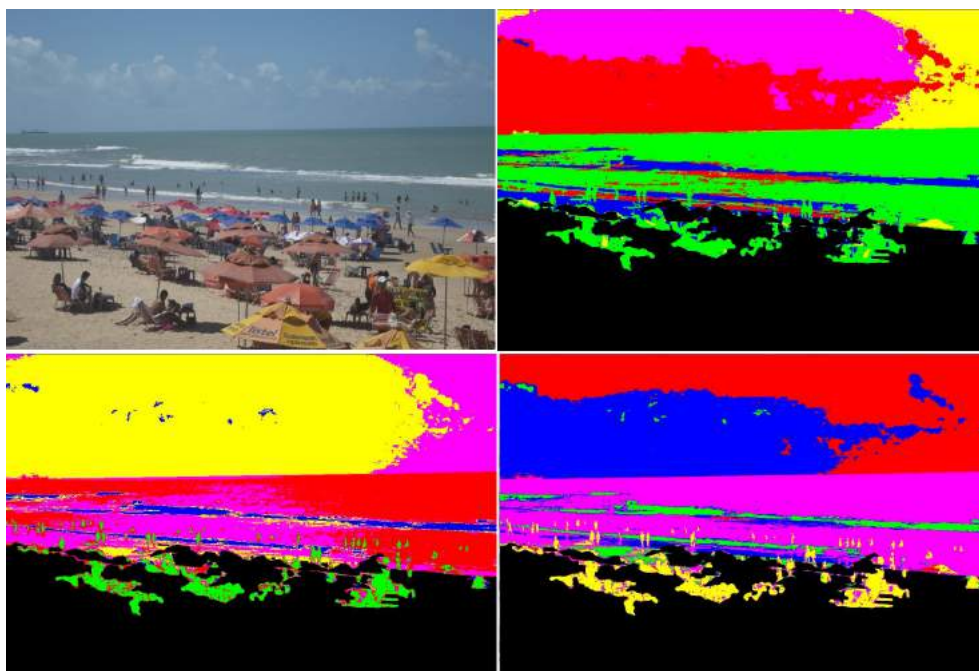
Fonte – O autor

Figura 65 – (1) Imagem Original, Melhores segmentações para 30 níveis de cinza: (2) LAB representado pelo grupo rosa, (3) RGB representado pelo grupo amarelo, (4) RGB + LAB representado pelo grupo rosa.



Fonte – O autor

Figura 66 – (1) Imagem Original, Melhores segmentações para 40 níveis de cinza: (2) LAB representado pelo grupo verde, (3) RGB representado pelo grupo verde, (4) RGB + LAB representado pelo grupo amarelo.



Fonte – O autor

Figura 67 – (1) Imagem Original, Melhores grupos entre categorias:(2) 20 níveis de cinza - RGB representado pelo grupo vermelho, (3) 30 níveis de cinza - RGB + LAB representado pelo grupo verde, (4) 40 níveis de cinza - RGB representado pelo grupo vermelho.



Fonte – O autor

5.2.3 Experimento B-3

Neste experimento, os demais algoritmos que não possuem um controle direto dos parâmetros de execução são avaliados. Os algoritmos são: *X-means*, Hierárquico divisivo e *Auto group Segmentation*. Para cada um dos algoritmos, foram aplicados os sistemas de cores RGB, LAB e a combinação deles para geração dos possíveis vetores de características que descrevem como os *pixels* são, acrescentando a remoção da faixa de praia das imagens. A Tabela 7 apresenta os resultados encontrados para o índice de *Dunn*, especificando o valor médio e o desvio padrão σ para a base de dados de imagens de praia, variando os vetores de características.

Tabela 7 – Experimento B-3 - Resultados de média para o índice de *Dunn* variando os vetores de características para os algoritmos de *X-means*, Hierárquico divisivo e *Auto group segmentation*, com adição dos melhores resultados dos experimentos B-1 e B-2.

		Média	Desvio Padrão (σ)
X-Means	RGB	1.0197344041831722	0.4352018835703881
	LAB	1.09066804731009	0.9243699209132973
	RGB + LAB	0.7664367876659499	0.27513078506988053
<i>Auto Group Segmentation</i>	RGB	1.4067758189139024	0.4581150692375824
	LAB	1.138452120823641	0.43086481214580896
	RGB + LAB	1.221555796777121	0.43680960741308456
Hierárquico divisivo	RGB	2.099065908865676	0.8074498801242361
	LAB	1.847469110563855	1.1140912463867507
	RGB + LAB	1.0603420358358395	0.3657051448646044

Observando a Tabela 7, para os vetores de características RGB e LAB separadamente, o algoritmo hierárquico divisivo apresentou os melhores resultados, contendo a média de 2.0991 e 1.8475 respectivamente. No entanto, para a combinação de RGB e LAB como um único vetor de características, o *auto Group segmentation* mostrou os resultados mais promissores para esse vetor, com uma média de 1.2216.

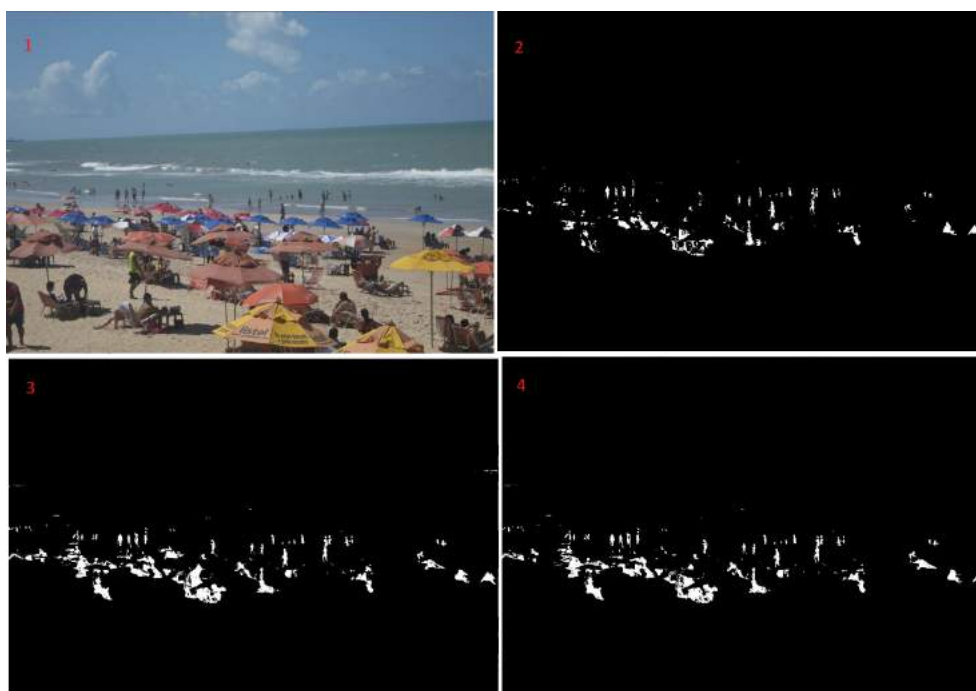
A seguir, um conjunto de imagens de resultados obtidos para os três algoritmos: *X-means*, *auto group segmentation* e hierárquico divisivo (Figuras 68, 69, 70).

Figura 68 – *X-means* - (1) Imagem Original, Melhores grupos que representaram as pessoas: (2) LAB, (3) RGB, (4) RGB + LAB.



Fonte – O autor

Figura 69 – *Auto Group Segmentation* - (1) Imagem Original, Melhores grupos que representaram as pessoas: (2) LAB, (3) RGB, (4) RGB + LAB.



Fonte – O autor

Figura 70 – *Hierárquico Divisivo* - (1) Imagem Original, Melhores grupos que representaram as pessoas: (2) LAB, (3) RGB, (4) RGB + LAB.



Fonte – O autor

5.3 Discussão

Os experimentos realizados neste trabalho tem o objetivo de analisar a qualidade dos agrupamentos automáticos gerado pelos algoritmos. Inicialmente eles foram divididos em dois principais experimentos, os do tipo A e do tipo B.

Durante a execução dos experimentos, foi percebido que os resultados do índice de *Dunn* estão entrelaçados com o vetor de características utilizado, tendo em vista que o cálculo de compacticidade e distanciamento do grupos de uma solução leva em consideração tal vetor. Desta forma, os resultados encontrados devem ser analisados categoricamente divididos entre os vetores de características.

É importante salientar que os resultados encontrados para a GLCM de 20 tons de cinza não foram levados em consideração devido ao baixíssimo números de grupos gerados, contendo uma média de 2 a 4 grupos, ocasionando em valores altos do índice de *Dunn*.

Os experimentos do tipo A tiveram como objetivo o teste dos algoritmos hierárquico aglomerativo, hierárquico divisivo, *auto group segmentation*, *X-means* e segmentação automática de imagens coloridas para as imagens sem o pré-processamento que remove a área da praia da imagem. Em especial, o hierárquico aglomerativo e o de segmentação automática de imagens necessitam de um controle de parâmetro extra

para gerar seus resultados. Os resultados dos experimentos variando os testes de parâmetros estão descritos nas subseções 5.1.1 e 5.1.2.

A seguir, um conjunto de Tabelas (8, 9, 10) particionadas pelo vetor de características utilizados, de forma que para os algoritmos com controle de parâmetros somente será exibido o melhor resultado para aquele determinado vetor para os experimentos do tipo A.

Tabela 8 – Resultados de média para o índice de *Dunn* do experimento A para o vetor de características RGB.

	Média	Desvio Padrão (σ)
<i>X-means</i>	1.1669705623036304	0.3335815533946962
<i>Auto Group Segmentation</i>	1.689492698865999	0.3707954548385317
Hierárquico Divisivo	1.8045059891466342	0.3887619170943952
Hierárquico Aglomerativo	1.815263183488256	0.3647447840108059
Seg. Automática de imagens coloridas	1.6851971487424071	0.5016171433361224

Tabela 9 – Resultados de média para o índice de *Dunn* do experimento A para o vetor de características LAB.

	Média	Desvio Padrão (σ)
<i>X-means</i>	0.8638510506605973	0.3417794956501355
<i>Auto Group Segmentation</i>	1.4543577631177707	0.40225784495196193
Hierárquico Divisivo	1.7063648912115927	0.448259094086873
Hierárquico Aglomerativo	1.7697979561998016	0.3469811110979991
Seg. Automática de imagens coloridas	1.4008743074574268	0.5139085761593

Tabela 10 – Resultados de média para o índice de *Dunn* do experimento A para o vetor de características RGB + LAB.

	Média	Desvio Padrão (σ)
<i>X-means</i>	0.883193331896416	0.38615192601804416
<i>Auto Group Segmentation</i>	1.500077602386799	0.3152614421049866
Hierárquico Divisivo	1.3090428234181708	0.3596470413168057
Hierárquico Aglomerativo	1.2210506817820324	0.27533398668015363
Seg. Automática de imagens coloridas	1.5244897707973126	0.44793821962863706

Observando as Tabelas 8, 9 e 10, podemos avaliar os algoritmos que se mostraram mais promissores para cada vetor de características. O algoritmo hierárquico aglomerativo obteve os resultados mais altos para o índice de *Dunn* tanto para o vetor de características RGB e LAB separadamente, enquanto a combinação de RGB e LAB se mostrou mais promissor quando utilizado o algoritmo de segmentação automática de imagens coloridas.

Em relação aos experimentos do tipo B, da mesma forma do A, foram executados 3 experimentos para avaliar os algoritmos, com a diferença da remoção da faixa de praia das imagens. As mesmas observações discutidas anteriormente nesta seção sobre os algoritmos em específico, também valem para os experimentos do tipo B.

Levando em consideração apenas os resultados encontrados através do índice de *Dunn*, os valores de média mais altos se encontram nos experimentos do tipo B, indicando os melhores resultados de agrupamento em uma perspectiva global da imagem. Além disso, a técnica de remoção da faixa de praia aumentaria o custo computacional, no entanto, tal problema pode ser contornado, pois a ideia é que a etapa de segmentação da praia não seja aplicada para cada quadro do vídeo, mas sim em determinados períodos de tempo. Para o vetor de características RGB + LAB, o algoritmo de segmentação automática de imagens coloridas com 30 tons de cinza é o mais promissor, enquanto para o vetor LAB, o hierárquico aglomerativo obteve o maior resultado, por fim, para o vetor RGB, a vitória ficou com o hierárquico divisivo.

A seguir, um conjunto de Tabelas (11, 12, 13) particionadas pelo vetor de características utilizados, de forma que para os algoritmos com controle de parâmetros somente será exibido o melhor resultado para aquele determinado vetor para os experimentos do tipo B.

Tabela 11 – Resultados de média para o índice de *Dunn* do experimento B para o vetor de características RGB.

	Média	Desvio Padrão (σ)
<i>X-means</i>	1.0197344041831722	0.4352018835703881
<i>Auto Group Segmentation</i>	1.4067758189139024	0.4581150692375824
Hierárquico Divisivo	2.099065908865676	0.8074498801242361
Hierárquico Aglomerativo	2.0858503869488305	0.8080617053852194
Seg. Automática de imagens coloridas	1.7785799806457567	1.039465305425165

Tabela 12 – Resultados de média para o índice de *Dunn* do experimento B para o vetor de características LAB.

	Média	Desvio Padrão (σ)
<i>X-means</i>	1.09066804731009	0.9243699209132973
<i>Auto Group Segmentation</i>	1.138452120823641	0.43086481214580896
Hierárquico Divisivo	1.847469110563855	1.1140912463867507
Hierárquico Aglomerativo	2.051383750019409	0.7696914542732832
Seg. Automática de imagens coloridas	1.4852978320951027	0.8103449070674957

Tabela 13 – Resultados de média para o índice de *Dunn* do experimento B para o vetor de características RGB + LAB.

	Média	Desvio Padrão (σ)
<i>X-means</i>	0.7664367876659499	0.27513078506988053
<i>Auto Group Segmentation</i>	1.221555796777121	0.43680960741308456
Hierárquico Divisivo	1.0603420358358395	0.3657051448646044
Hierárquico Aglomerativo	1.3571652521364053	0.4805276547267213
Seg. Automática de imagens coloridas	1.5857249596315146	0.9015772338370096

Ao olhar os resultados das Tabelas 11, 12 e 13 é percebido um aumento significativo no desvio padrão dos resultados em relação aqueles encontrados nos experimentos do tipo A. Este fato é atribuído a questão de que os resultados se tornaram inconsistentes, principalmente devido a aleatoriedade da inicialização dos centros dos

grupos, principalmente após a remoção da faixa de praia. Tal acontecimento é devido a característica padrão dos algoritmos de agrupamento baseados no *K-means*, que possui uma inicialização aleatória dos centros dos grupos, os algoritmos avaliados também herdaram tal característica. Com a remoção das regiões da faixa de praia, a quantidade de possíveis elementos da imagem que pudessem ocasionar em uma maior semelhança com os banhistas foram removidos, desta forma se tornou mais difícil para que um dos centros de grupos consiga obter um valor similar aos banhistas.

A seguir, um conjunto de Tabelas (14, 15, 16) que mostram a comparação dos resultados encontrados para o índice de *Dunn* com e sem a remoção da faixa de praia categorizados por vetor de características.

Tabela 14 – Comparação dos Resultados de média para o índice de *Dunn* sem e com remoção de praia para vetor de características RGB.

	Remoção da Faixa de praia?	Média	Desvio Padrão (σ)
<i>X-means</i>	Não	1.1669705623036304	0.3335815533946962
	Sim	1.0197344041831722	0.4352018835703881
<i>Auto Group Segmentation</i>	Não	1.689492698865999	0.3707954548385317
	Sim	1.4067758189139024	0.4581150692375824
Hierárquico Divisivo	Não	1.8045059891466342	0.3887619170943952
	Sim	2.099065908865676	0.8074498801242361
Hierárquico Aglomerativo	Não	1.815263183488256	0.3647447840108059
	Sim	2.0858503869488305	0.8080617053852194
Seg. Automática de imagens coloridas	Não	1.6851971487424071	0.5016171433361224
	Sim	1.7785799806457567	1.039465305425165

Tabela 15 – Comparação dos Resultados de média para o índice de *Dunn* sem e com remoção de praia para vetor de características LAB.

	Remoção da Faixa de praia?	Média	Desvio Padrão (σ)
<i>X-means</i>	Não	0.8638510506605973	0.3417794956501355
	Sim	1.09066804731009	0.9243699209132973
<i>Auto Group Segmentation</i>	Não	1.4543577631177707	0.40225784495196193
	Sim	1.138452120823641	0.43086481214580896
Hierárquico Divisivo	Não	1.7063648912115927	0.448259094086873
	Sim	1.847469110563855	1.1140912463867507
Hierárquico Aglomerativo	Não	1.7697979561998016	0.3469811110979991
	Sim	2.051383750019409	0.7696914542732832
Seg. Automática de imagens coloridas	Não	1.4008743074574268	0.5139085761593
	Sim	1.4852978320951027	0.8103449070674957

Tabela 16 – Comparação dos Resultados de média para o índice de *Dunn* sem e com remoção de praia para vetor de características RGB + LAB.

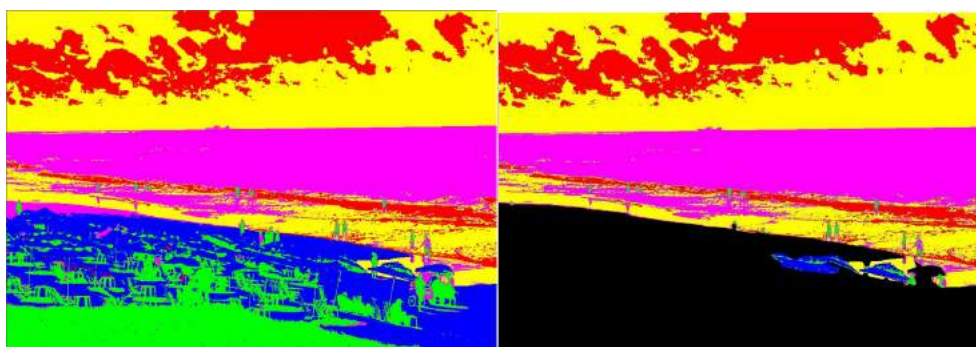
	Remoção da Faixa de praia?	Média	Desvio Padrão (σ)
<i>X-means</i>	Não	0.883193331896416	0.38615192601804416
	Sim	0.7664367876659499	0.27513078506988053
<i>Auto Group Segmentation</i>	Não	1.500077602386799	0.3152614421049866
	Sim	1.221555796777121	0.43680960741308456
Hierárquico Divisivo	Não	1.3090428234181708	0.3596470413168057
	Sim	1.0603420358358395	0.3657051448646044
Hierárquico Aglomerativo	Não	1.2210506817820324	0.27533398668015363
	Sim	1.3571652521364053	0.4805276547267213
Seg. Automática de imagens coloridas	Não	1.5244897707973126	0.44793821962863706
	Sim	1.5857249596315146	0.9015772338370096

Através das Tabelas 14, 15 e 16, é possível observar os resultados em comparação, também é notável a diferença entre os valores de média e principalmente o desvio padrão. Em todos os casos no qual os resultados do experimento B foram maiores, o desvio padrão foi muito mais elevado em relação aos experimentos do tipo A, sendo que na maioria deles, a diferença da média não foi muito expressiva.

Após analisar os resultados encontrados nas imagens, através de uma avaliação visual e também comportamental dos algoritmos, como a aleatoriedade dos centros iniciais e baixa informação de textura para o algoritmo de segmentação automática de imagens, a conclusão é de que a melhor abordagem para segmentação de imagens de banhistas é não utilizar o algoritmo de remoção da faixa de praia como um método de pré-processamento das imagens, e sim como pós-processamento.

A ideia de utilizar a remoção de praia após o término da segmentação sem a remoção, é limpar o resultado final, depois que todos os grupos foram concluídos, basta utilizar a máscara da imagem que foi gerada, aplicando uma operação AND com as imagens binárias dos grupos. Assim, somente aqueles onde na mesma posição na máscara e na imagem binária forem brancas, se tornará branca, removendo toda a região da praia do resultado final (Figuras 71 e 72).

Figura 71 – Remoção da faixa de praia como pós-processamento para imagem totalmente segmentada, grupo de pessoas de cor verde.



Fonte – O autor

Figura 72 – Remoção da faixa de praia como pós-processamento para imagem do grupo de pessoas.



Fonte – O autor

Através da análise visual das imagens resultantes, alguns resultados da segmentação de banhistas se tornaram piores no experimento B, pois, para o algoritmo

de segmentação de imagens coloridas, que utiliza os tons de cinza para extrair informações de textura através do descritor GLCM, houve uma redução geral da qualidade dos agrupamentos em relação aos banhistas, pois com a remoção da faixa de praia, uma boa parte das informações de textura também foram removidas, ocasionando em agrupamentos com menos grupos.

Para os demais algoritmos também foi notável a queda na qualidade visual para segmentação dos banhistas, pois, como explicado previamente, dificultou a inicialização dos centros dos grupos para obter uma semelhança com as informações dos banhistas, devido a remoção da faixa de praia.

Desta forma, como o objetivo principal deste trabalho é buscar grupos que melhor representem os banhistas, os experimentos do tipo B foram descartados, apesar de seus bons valores do índice de *Dunn*, indicando agrupamentos melhores estruturados do ponto de vista global da imagem.

Então, como resultado final serão apresentados os algoritmos mais promissores para cada tipo de vetor de características baseado nos valores do índice de *Dunn* para os experimentos do tipo A. Para o vetor de características utilizando o sistema de cor RGB, o algoritmo hierárquico aglomerativo com 40 sementes apresentou o resultado mais promissor, assim como para o vetor de características LAB. Já para a utilização do vetor de características composto pelo RGB e LAB simultaneamente, o algoritmo de segmentação automática de imagens coloridas foi o vencedor, para GLCM com 30 níveis de tons de cinza.

Dentre os 3 possíveis resultados baseado nos vetores de características, juntamente a seu índice de *Dunn*, como também observando os resultados em imagens encontrados, o vetor de características RGB é o mais instável devido a própria natureza do seu sistema de cor, pois ele é voltado para exibição de imagens coloridas na tela de dispositivos eletrônicos, não se preocupando em criar diferenças proporcionais entre as diferentes cores de uma imagem.

Em relação ao sistema de cores LAB, como explicado na seção 2, ele foi projetado para diferenciar as cores em distâncias proporcionais nos seus 3 canais, tornando-o um ótimo canal de cor com o intuito de calcular distâncias entre as cores, apesar de não aparentar um aspecto bom para a visão humana, ele é mais amigável para a computação de informações.

Descartando o vetor de características RGB, as duas opções restantes foram o algoritmo hierárquico aglomerativo utilizando LAB e o segmentação automática de imagens coloridas usando a combinação de LAB e RGB. Observando os resultados obtidos para os dois algoritmos, ambos apresentaram ótimos resultados para a grande maioria das imagens.

Como critério de desempate, foi levado em consideração o tempo de execução dos algoritmos. O melhor resultado encontrado para o algoritmo hierárquico aglomerativo foi composto de 40 sementes, de forma que quanto maior a quantidade de sementes, mais demorado era a execução, assim sendo, este algoritmo com 40 sementes foi um dos mais demorados para terminar a bateria de testes.

Em contrapartida o algoritmo de segmentação automática de imagens coloridas já foi pensado com a intenção de trabalhar com imagens, apresentando uma abordagem mais amigável para a mesma, seu tempo de execução é menor em relação ao hierárquico aglomerativo, desta forma sendo escolhido neste trabalho como a melhor opção de aplicação.

6 Conclusão e trabalhos futuros

A análise da combinação entre os algoritmos e os vetores de características, além da remoção ou não da faixa de praia, passaram por testes baseados no índice de *Dunn* para avaliar a qualidade dos agrupamentos. Os resultados, juntamente com uma análise visual das imagens e comportamental dos algoritmos, demonstraram que a remoção de praia como uma técnica de pré-processamento acabou diminuindo a qualidade dos agrupamentos referentes aos banhistas. Desta forma, a remoção da praia pode ser feita após o fim do agrupamento, removendo informações desnecessárias sobre a segmentação.

Já entre a combinação de vetores de características e algoritmos, dois resultados se destacaram, eles foram os algoritmos hierárquicos aglomerativos utilizando 40 sementes com vetor de características LAB e o algoritmo de segmentação automática de imagens coloridas com vetor RGB e LAB combinados. Como critério de desempate, foi levado em consideração o tempo de execução dos algoritmos, em que o hierárquico aglomerativo levava mais tempo devido a sua natureza aglomerativa, além de que o algoritmo de segmentação automático de imagens coloridas foi pensado para trabalhar com imagens.

Com tais informações em mente, foi definido que atualmente, a aplicação do algoritmo de segmentação automática de imagens coloridas, juntamente com o vetor de características combinado de RGB e LAB, executando a remoção da faixa de praia após o fim do algoritmo de agrupamento, mostraram os resultados mais promissores.

Este trabalho, dado sua natureza, possui uma vertente social bastante relevante, ele visa segmentar imagens de pessoas em regiões de praia em busca de banhistas, de forma a preparar a imagem para a próxima etapa, definida como a fase de classificação, analisando os segmentos encontrados para classificá-los em banhistas ou não banhistas. A ferramenta final, contendo este trabalho como primeira fase, almeja ajudar os guarda-vidas nas regiões litorâneas a monitorar os banhistas caso eles tentem ultrapassar uma zona de segurança.

Para tal fim, uma série de estudos e análises foram necessárias: i. definir algoritmos de agrupamento para segmentar as imagens com característica especial de automaticamente definir quantas regiões a imagem possui; ii. criar e combinar vetores de características para poder representar o menor elemento encontrado nas imagens (*pixels*) de forma que melhor os pudesse representar computacionalmente e assim agrupá-los corretamente; iii. criação de uma base de dados relevante para o problema em questão, tendo em vista que a abordagem de segmentação de banhistas em re-

giões de praia ainda não tinha sido abordado na literatura; iv. além de observar modificações na própria imagem através de técnicas de pré-processamento, como a remoção ou não da faixa de praia.

Em relação aos algoritmos, cinco foram implementados para comparação, sendo eles os algoritmos hierárquicos aglomerativos (MAIMON; ROKACH, 2010), adaptado para imagens, aplicando sementes para criar grupos iniciais, hierárquico divisivo (MAIMON; ROKACH, 2010), *auto group segmentation* (RAY; TURI, 2000), segmentação automática de imagens coloridas (KALAIVANI; CHITRAKALA, 2014) e o *X-means* (PELLEG; MOORE, 2000). No âmbito dos vetores de características, a aplicação de dois sistemas de cores foram selecionados, o RGB e o LAB. Além disso, também foi aplicado a combinação dos dois sistemas como um único vetor.

Os algoritmos com seleção automática do número de grupos é um problema não-trivial e essencial, pois para definir tal número, é necessário ter um conhecimento prévio sobre o problema, além disso, como o agrupamento é realizado em imagens, aumenta a complexidade do problema, tendo em vista que a quantidade e os elementos que aparecem por imagens pode ser completamente diferente.

6.1 Trabalhos futuros

Para melhorar o trabalho como um todo, 5 ideias principais ocorreram durante a sua execução. A primeira delas foi a avaliação dos resultados das imagens segmentadas através das imagens-ouro com os banhistas previamente rotuladas na criação da base de dados.

Com a primeira ideia em mente, surgiu a segunda: uma forma de avaliar individualmente cada grupo da solução segmentada e realizar uma espécie de filtragem para eliminar o máximo de grupos que não apresentam nenhum traço de banhistas nele, se possível, apontar qual grupo é o que contém mais pessoas. Desta forma, seria possível analisar automaticamente os segmentos das imagens que contém banhistas com as imagens rotuladas da base de dados.

Outra ideia é avaliar os resultados através de uma prova de conceito, recortando partes da imagem para aplicar outras métricas, como a silhueta, apontado na Seção 4.4, de forma que tal recorte se tornará computável em tempo hábil.

Um aspecto interessante para aumentar o alcance destes métodos é enriquecer ainda mais a base de dados, acrescentando imagens que contemplem modificações climáticas, como dias chuvosos. Além de variações na própria faixa de praia, como áreas cheias de pedras.

A última ideia para trabalhos futuros surgiu após os resultados encontrados pe-

los experimentos do tipo B, pois a aleatoriedade da inicialização dos centros interferiu. Assim sendo, uma forma de melhor inicializar os centros, tentando priorizar ao menos um centro de grupo para os banhistas, melhoraria a qualidade dos resultados para a remoção da faixa de praia.

6.2 Contribuições

Esse trabalho surgiu como um projeto de iniciação científica, com o mesmo orientador. No ano de 2017 ocorreu uma apresentada na JEPEX (Jornada de Ensino, Pesquisa e Extensão) realizada na UFRPE. Ainda no mesmo ano foi apresentado um trabalho de classificação relacionado (SILVA et al., 2017) no 29th ICTAI (*International Conference on Tools with Artificial Intelligence*) que foi realizada em Boston nos Estados Unidos e teve um trabalho publicado no IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos) e outro artigo (GOMES; GONÇALVES, 2018) no XIII WVC (Workshop de Visão Computacional) realizado na Universidade Federal do Rio Grande do Norte (UFRN) e teve um artigo publicado nos anais do evento. Por fim, outra iniciação científica englobando os diversos passos do sistema, aplicando a segmentação, classificação e rastreamento dos banhistas foi realizada, desta vez pela FACEPE (Fundação de amparo à ciência tecnologia do estado de Pernambuco), apresentado no ano de 2018.

Referências

- ABUBAKER, M.; ASHOUR, W. Efficient data clustering algorithms: Improvements over kmeans. In: MECS. *International Journal of Intelligent Systems and Applications*. [S.l.], 2013.
- ANTÍC, B. et al. K-means based segmentation for real-time zenithal people counting. In: IEEE. *2009 16th IEE International Conference on Image Processing*. [S.l.], 2009. v. 16, p. 2565–2568.
- ARBELAEZ, P. et al. Multiscale combinatorial grouping. In: . *IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 328–335.
- BLOISI, D.; LOCCHI, L. Rek-means: A k-means based clustering algorithm. In: IEEE. *Science Engineering and Management Research*. [S.l.], 2008. p. 109–118.
- BRADSKI, G. et al. The opencv library (2000). *Dr. Dobb's Journal of Software Tools*, 2000.
- CARRERA, A. Segmentação de multidões para um sistema de monitoramento de banhistas com foco no baixo custo computacional. *Universidade Federal Rural de Pernambuco, Recife*, 2017.
- CHEN, Y.-L. C. T.-W.; CHIEN, S. yi. Fast image segmentation based on k-means clustering with histograms in hsv color space. In: IEEE. *Multimedia Signal Processing*. [S.l.], 2008.
- CHEUNG, Y.-M. K*-means: a new generalized k-means clustering algorithm, pattern recognition letters. In: ELSERVIER. *Pattern recognition Letters*. [S.l.], 2003. v. 24, p. 2883–2893.
- CLIFTON, A.; LUNDQUIST, J. K. Data clustering reveals climate impacts on local wind phenomena. In: *Journal of Applied Meteorology and Climatology*. [S.l.: s.n.], 2012. v. 51.8, p. 1547–1557.
- DUNN, J. C. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, v. 3, p. 32–57, 1973.
- GOMES, R. B.; GONÇALVES, L. M. G. Xiii workshop de visão computacional: anais do 13º workshop de visão computacional realizado na ufrn. Editora da UFRN, 2018.
- GONZALEZ, R. C.; WOODS, R. *Processamento digital de imagens. tradução: Cristina yamagami e leonardo piamonte*. [S.l.]: São Paulo: Pearson Prentice Hall, 2017.
- GORRIZ, J. M. et al. Hard c-means clustering for voice activity detection. In: ELSEVIER. *Speech Communication*. [S.l.], 2006. v. 48, p. 1638–1649.
- HAMERLY, G.; ELKAN, C. Learning the k in k-means. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2004. p. 281–288.

- HARALICK, R. M.; SHANMUGAN, K.; DINSTEN, I. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, IEE, v. 6, p. 610–621, 1973.
- HOSANG, J.; BENENSON, R.; SCHIELE, B. How good are detection proposals, really? In: *British Machine Vision Conference*. [S.l.: s.n.], 2014. p. 1–25.
- JC, O. *Jovem atacado por tubarão em Piedade morre no Hospital da Restauração*. 2018. Disponível através do link <http://jconline.ne10.uol.com.br/canal/cidades/geral/noticia/2018/06/04/jovem-atacado-por-tubarao-em-piedade-morre-no-hospital-da-restauracao-341854.php>. Acessado em 18-06-2018.
- KALAIVANI, A.; CHITRAKALA, S. Automatic color image segmentation. In: *Science Engineering and Management Research (ICSEMR)*. [S.l.]: IEEE, 2014. p. 1–5.
- KALOGERATOS, A.; LIKAS, A. Dip-means: an incremental clustering method for estimating the number of clusters. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 2393–2401.
- KHANDELWAL, S.; AWEKAR, A. Faster k-means cluster estimation. In: SPRINGER. *European Conference on Information Retrieval*. [S.l.], 2017. p. 520–526.
- LI, M. et al. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: IEEE. *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. [S.l.], 2008. p. 1–4.
- MAIMON, O.; ROKACH, L. *Data Mining And Knowledge Discovery Handbook*. [S.l.]: Springer, 2010.
- OPENCV. *OpenCV: Color conversions*. 2015. Acesso em: 15/05/2018. Disponível em: http://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html.
- PELLEG, D.; MOORE, A. X-means: Extending k-means with efficient estimation of the number of clusters. In: ICML. *Seventeenth International Conference on Machine Learning*. [S.l.], 2000.
- PUNJ, G.; STEWART, D. W. Cluster analysis in marketing research: Review and suggestions for application. In: JSTOR. *Journal of Marketing Research*. [S.l.], 1983. v. 20, p. 134–148.
- RAY, S.; TURI, R. H. Determination of number of clusters in k-means clustering and application in colour image segmentation. In: *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*. [S.l.: s.n.], 2000.
- RAYKOV, Y. et al. What to do when k-means clustering fails: a simple yet principled alternative algorithm. *PloS one*, v. 11, p. e0162259, 2016.
- RIZMAN, K. An efficient k'-means clustering algorithm. In: ELSERVIER. *Pattern recognition Letters*. [S.l.], 2008. v. 29, p. 1385–1391.
- ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, v. 20, p. 53–65, 1987.

RUSSELL, B. C. et al. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, Springer, v. 77, n. 1, p. 157–173, 2008.

SANDE, K. E. Van de et al. Segmentation as selective search for object recognition. In: IEEE. *Computer Vision (ICCV), 2011 IEEE International Conference on*. [S.l.], 2011. p. 1879–1886.

SHAHBABA, M.; BEHESHTI, S. Mace-means clustering. In: *Signal Processing*. [S.l.: s.n.], 2014. p. 216–225.

SILVA, R. L. da et al. Detecting people from beach images. In: IEEE. *Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on*. [S.l.], 2017. p. 636–643.

SURAL, S.; QIAN, G.; PRAMATIK, S. Segmentation and histogram generation using the hsv color space for image retrieval. *Proceedings. International Conference on Image Processing*, IEEE, v. 2, p. II–II, 2002.

SUZUKI, S.; KEIICHI, A. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, v. 30, p. 32–46, 1985.

VASCONCELOS, Y. *Por que ocorrem tantos ataques de tubarão em Recife?* 2018. Disponível através do link <https://mundoestranho.abril.com.br/mundo-animal/por-que-ocorrem-tantos-ataques-de-tubarao-em-recife/>. Acessado em 18-06-2018.

WEATHERALL, I. L.; BERNARD, D. C. Skin color measurements in terms of cielab color space values. *Journal of investigative dermatology*, IEEE, v. 99, n. 4, p. 468–473, 1992.

WITTEN, I. H.; FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques*. [S.l.]: Elsevier, 2005.