



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARCOS JOSÉ CANÊJO ESTEVÃO DE AZEVÊDO

**Segmentação de Imagens de Cenas Naturais Baseada no Speed Drawing
Challenge com Aplicação em Segmentação de Mapas e Plantas Baixas**

Recife

2022

MARCOS JOSÉ CANÊJO ESTEVÃO DE AZEVÊDO

**Segmentação de Imagens de Cenas Naturais Baseada no Speed Drawing
Challenge com Aplicação em Segmentação de Mapas e Plantas Baixas**

Tese apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciências da Computação.

Área de Concentração: Inteligência Computacional

Orientador (a): Prof. Dr. Carlos Alexandre Barros de Mello

Recife

2022

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

A994s Azevêdo, Marcos José Canêjo Estevão de
Segmentação de imagens de cenas naturais baseada no *speed drawing challenge* com aplicação em segmentação de mapas e plantas baixas / Marcos José Canêjo Estevão de Azevêdo. – 2022.
102 f.: il., fig., tab.

Orientador: Carlos Alexandre Barros de Mello.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2022.

Inclui referências.

1. Inteligência computacional. 2. Segmentação. I. Mello, Carlos Alexandre Barros de (orientador). II. Título.

006.31

CDD (23. ed.)

UFPE - CCEN 2022-115

Marcos José Canêjo Estevão de Azevêdo

“Segmentação de Imagens de Cenas Naturais Baseada no Speed Drawing Challenge com Aplicação em Segmentação de Mapas e Plantas Baixas”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação. Área de Concentração: Inteligência Computacional.

Aprovado em: 30/03/2022.

Orientador: Prof. Dr. Carlos Alexandre Barros de Mello

BANCA EXAMINADORA

Prof. Dr. Tsang Ing Ren
Centro de Informática / UFPE

Prof. Dr. Cleber Zanchettin
Centro de Informática / UFPE

Profa. Dra. Mylène Chistine Queiroz de Farias
Departamento de Engenharia Elétrica / UnB

Prof. Dr. Francisco Madeiro Bernardino Junior
Escola UNICAP ICAM-TECH / UNICAP

Prof. Dr. Marcelo Walter
Instituto de Informática / UFRGS

Dedico este trabalho a família.

RESUMO

Este trabalho apresenta um método para segmentação de imagens inspirado no *Speed Drawing Challenge*. O *Speed Drawing Challenge* consiste em um desafio em que artistas devem representar um desenho em segundos sem perder as principais características que o tornam fiel ao desenho original. À medida que o tempo é reduzido, o artista se encaminha a produzir um desenho mais simples, se aproximando de um desenho contendo apenas os traços mais representativos, focando nos contornos. Para o problema de detecção de contorno, uma imagem de borda representa um desenho produzido sem uma grande limitação de tempo. Assim, o método proposto faz uso dos conceitos de *superpixel*, detecção de bordas e mapas de saliência para produção de uma imagem, simulando a variação de tempo do *Speed Drawing Challenge*: a modelagem do desafio em poucos segundos expressa uma imagem com apenas os contornos mais relevantes detectados. Diferente das técnicas do estado-da-arte, o método proposto gera uma imagem de bordas já final, sem necessidade de posterior pós-processamento como binarização. Mais do que um algoritmo apenas, propomos uma metodologia que pode ser aplicada de formas diferentes, mas seguindo a mesma ideia, em outros domínios, como na segmentação de mapas e plantas baixas. Os resultados para imagens de cenas naturais são apresentados para o banco de dados BSDS500 e avaliados de forma qualitativa e quantitativa para as métricas ODS, OIS e AP.

Palavras-chaves: speed drawing challenge; segmentação; detecção de borda; mapa de saliência; superpixel; segmentação de plantas baixas.

ABSTRACT

This work introduces a method for image segmentation inspired by the Speed Drawing Challenge. In it, the artists are defied to represent a drawing in seconds without losing the main characteristics that make it loyal to the original drawing. As time gets shorter, the artist gradually produces a more simplified drawing, approaching a sketch containing only the most representative strokes, and focusing on the contours. For the contour detection problem, an edge image represents a drawing produced without a large time. Thus, the proposed method relies on superpixel concepts, edge detection, and saliency maps to produce an image mimicking the variation in time from the Speed Drawing Challenge: modeling the challenge in a few seconds expresses an image with only the most relevant contours detected. Different from state-of-the-art approaches, the proposed method generates a final edge image without the need for further post-processing as binarization. More than just an algorithm, we have proposed a methodology that can be used in different ways based on the same major idea, for example, as in the segmentation of maps and floor plans. The results for natural scenes are presented for the BSDS500 database and evaluated qualitatively and quantitatively for the ODS, OIS, and AP metrics.

Keywords: speed drawing challenge; segmentation; edge detection; saliency map; superpixel; floor plan segmentatio.

LISTA DE FIGURAS

Figura 1 – (a) Imagem original, (b) imagem contendo bordas detectadas pelo algoritmo de Canny e (c) imagem de contorno.	18
Figura 2 – Exemplo de segmentação pelo método de detecção de contorno. A primeira imagem representa a imagem original, ou seja, um vaso esculpido com vários detalhes. Na imagem seguinte, o seu contorno é detectado com interesse maior no objeto, ignorando vários de seus detalhes. Por fim, o resultado da segmentação é apresentada.	19
Figura 3 – Contornos de 5 imagens feito por 3 pessoas diferentes. Por exemplo, para a imagem Elefantes, o contorno da Pessoa 1 possui mais detalhes do que o da Pessoa 3; porém, para a imagem Reunião, a Pessoa 3 produziu mais contornos.	19
Figura 4 – Exemplo do <i>Speed Drawing Challenge (SDC)</i> para 10 minutos, 1 minuto e 10 segundos.	20
Figura 5 – Exemplo do posicionamento do disco para cálculo do gradiente orientado. O diâmetro do disco é de 50 pixels apenas para ilustração.	24
Figura 6 – Orientação dos 17 filtros derivativos Gaussianos.	24
Figura 7 – Exemplo do gradiente orientado para cada canal.	25
Figura 8 – (a) Imagem original, com (b) seu padrão ouro e (c) contorno detectado pelo <i>Globalized Probability of Boundary (gPb)</i> (c).	26
Figura 9 – Etapas do processo de obtenção da <i>Transformada de Watershed Orientada</i> : (a) imagem de contorno $E(x, y, \theta)$ de entrada, (b) arcos resultantes da aplicação da <i>Transformada de Watershed</i> , (c) segmentos de linhas formados pelos pontos de extremidades dos arcos, (d) exemplo do <i>Oriented boundary strength</i> para quatro valores de θ e (e) os arcos finais formados pela associação dos <i>Oriented boundary strength</i> com os segmentos de linhas.	27
Figura 10 – Resultados do gPb-owt-ucm: (a) imagem original), (b) imagem de resposta máxima do contorno detectado pelo gPb, (c) imagem resultante do gPb-ucm com imagem de contorno de entrada obtida pelo gPb e (d) imagem segmentada.	28
Figura 11 – (a) Imagem original I , (b) imagem binária S de (c) contorno criada por um humano e exemplos de <i>patches s</i> extraídos.	29
Figura 12 – Exemplo dos <i>Sketch Tokens (ST)</i>	29
Figura 13 – Exemplo de detecção de 2 tipos de <i>ST</i>	30
Figura 14 – (a) Colunas com imagens originais, (b) padrão ouro e (c) contornos detectados pelo algoritmo <i>ST</i>	30

Figura 15 – (a-c) Imagens de entrada e seus (d-f) resultados para o algoritmo <i>Structured Forest Edges (SFE)</i> .	31
Figura 16 – Proposta multiescala da <i>Holistically-Nested Edge Detection (HED)</i> .	32
Figura 17 – Arquitetura proposta por Maninis <i>et al.</i>	33
Figura 18 – Arquitetura proposta por Maninis <i>et al.</i>	34
Figura 19 – Exemplo de simplificação dos vários pontos do contorno.	34
Figura 20 – Exemplo bordas detectadas para (a) 0°, (b) 90° e (c) todas orientações aprendidas da imagem.	35
Figura 21 – Arquitetura proposta por Kelm <i>et al</i> que busca tirar proveito do processo de refinamento das camadas mais profundas de uma <i>Convolutional Neural Network (CNN)</i> .	36
Figura 22 – Novos blocos na RefineNet propostos pelos autores.	36
Figura 23 – Arquitetura da <i>Dense Extreme Inception Network for Edge Detection (DexiNed)</i> .	37
Figura 24 – Exemplo das imagens nas 6 saídas após a operação de <i>upsample</i> , seguidas da imagem agregada e da imagem da média.	37
Figura 25 – Arquitetura da <i>Bi-Directional Cascade Network for Perceptual Edge Detection (BDCN)</i> .	38
Figura 26 – <i>SDC</i> de 20 minutos, 2 minutos e 2 segundos.	40
Figura 27 – Exemplo de <i>SDC</i> de 10 minutos, 1 minuto e 10 segundos.	40
Figura 28 – Exemplo do <i>SDC</i> modelado para o problema de detecção de contorno através do conceito de <i>superpixel</i> .	41
Figura 29 – Imagem que pode ser interpretada como o resultado do <i>SDC</i> de tempo máximo.	42
Figura 30 – Imagem representando o resultado de um desafio de 1 minuto.	43
Figura 31 – Imagem esboço, representando o desenho feito em 10 segundos no <i>SDC</i> .	43
Figura 32 – Apresentação do método proposto dividido em duas partes.	45
Figura 33 – Exemplo de imagem segmentada com 16, 256, 4096 e 65536 <i>superpixels</i> .	47
Figura 34 – (a-c) Execução do algoritmo de West <i>et al.</i> , (d) ajuste de todas arestas proposta pela nova abordagem (e) com base nas cores e bordas da imagem.	48
Figura 35 – Exemplo da (a) imagem original, suas versões segmentadas pelo <i>Superpixel Hierarchy (SH)</i> com (b) 5 e (c) 3 (com o urso totalmente suprimido).	49
Figura 36 – (a) Imagem original e (b) suas bordas detectadas pelo SFE.	50
Figura 37 – Detecção de borda para 8 diferentes orientações geradas pelo <i>gPb</i> .	51
Figura 38 – Imagens de borda geradas pelo algoritmo de Canny aplicado as imagens de <i>superpixel</i> criadas a partir das modificações propostas.	52
Figura 39 – Exemplo da imagem <i>sh_canny</i> da Figura 37(a).	53

Figura 40 – Exemplo de cena natural no qual o objeto principal (representado pela raposa) tem parte dos seus pixels não capturados, mas relacionados com a borda da imagem.	55
Figura 41 – Visão geral do algoritmo <i>FastMBD</i> . Adaptada de (MBD..., 2019).	55
Figura 42 – (a) Imagem original, (b) sua versão das 8 imagens de <i>superpixel</i> resumida (<i>sh_colored</i>) e (c) mapa de saliência detectado pelo MBD+.	56
Figura 43 – Exemplo de bordas fracas removidas da (a) imagem <i>sh_canny</i> com ajuda da (b) imagem de borda SFE resultando na (c) imagem <i>sh_imp</i>	57
Figura 44 – Exemplo das imagens com (a) as bordas removidas imprecisamente e (b) passo final de recuperação.	57
Figura 45 – Exemplo de bordas detectadas para (a) imagem original com diferentes números de <i>superpixels</i> e (b) imagem de 10 segundos para comparação. O fator de multiplicação é: (c) 3, (d) 5, (e) 7, (f) 10, (g) 13, (h) 15, (i) 17 e (j) 20.	58
Figura 46 – Exemplo da metodologia aplicada na análise quantitativa para (a) imagem original, sendo (b) a imagem de 10 segundos, (c) gPb, (d) gPb sem as bordas detectadas pela imagem de 10 segundos, (e) SFE, (f) SFE também sem as bordas presentes na imagem de 10 segundos e (g) o padrão ouro.	62
Figura 47 – (a) Imagem original e bordas detectadas por (b) SDC10s, (c) SDC1m, (d) SDC10M, (e) COB e (f) HED.	63
Figura 48 – Resultado de 20 imagens criadas por diferentes algoritmos de detecção de borda e contorno, juntamente com as imagens do método proposto (para 10 segundos, 1 e 10 minutos).	64
Figura 49 – Exemplos de imagens utilizadas nos testes.	66
Figura 50 – (a) Imagem resultante do processo de remoção das paredes de traço largo, (b) remoção de linhas finas, (c) classificação dos componentes conectados em texto e (d) remoção de ruídos deixados pelos processos anteriores.	67
Figura 51 – Imagem final da implementação do algoritmo proposto por Ahmed <i>et al.</i>	69
Figura 52 – (a) Imagem obtida pelo processo de remoção dos traços finos e (b) imagem final que deveria apenas conter texto.	69
Figura 53 – (a) Imagem binarizada, (b) remoção de pequenos e (c) grandes elementos do resultado final do passo de segmentação de texto.	70
Figura 54 – (a) Imagem com os componentes maiores que 400 pixels, (b) operação de fechamento, (c) afinamento e (d) resultado final.	71
Figura 55 – Resultado do passo final de recuperação de caracteres.	72

Figura 56 – (a) Imagem de entrada com os detalhes realçados, (b) imagem desfocada pela simulação de afastamento do objeto em relação ao observador e (c) imagem final.	74
Figura 57 – Imagem após o passo de remoção de elementos gráficos.	75
Figura 58 – Imagem final após a recuperação de caracteres.	75
Figura 59 – Detecção hierárquica realizada pela rede proposta.	76
Figura 60 – Ilustração da <i>deep multi-task neural network</i> proposta.	76
Figura 61 – Resultados do método de Zeng <i>et al</i> para o banco R2V em relação a outras técnicas.	77
Figura 62 – Resultados obtidos por Zeng <i>et al</i> para o banco R3D comparados a outras técnicas.	77
Figura 63 – Diagrama de blocos do método para segmentação de informações textuais e gráficas em imagens de mapas e plantas baixas.	78
Figura 64 – Imagem original de planta baixa usada como exemplo.	78
Figura 65 – Resultado do pré-processamento para planta baixa.	79
Figura 66 – Exemplo do passo de divisão entre as informações textuais e gráficas, dando origem as imagens (a) <i>text_layer</i> e (b) <i>graphics_layer</i>	80
Figura 67 – Imagem sketch criada a partir da ideia de <i>smearing</i>	81
Figura 68 – Resultado dos elementos textuais separados.	81
Figura 69 – Exemplo do resultado obtido para uma (a) planta baixa, (b) sua imagem <i>sketch</i> motivada pela ideia do SDC e imagens segmentadas para as informações (c) textuais e (d) gráficas.	83
Figura 70 – Exemplo de segmentação do método proposto para (a) imagem de planta baixa degradada, (b) imagem <i>sketch</i> e resultados obtidos para as informações (c) textuais e (d) gráficas.	84
Figura 71 – Imagem final contendo apenas texto obtido pelo método de (b) (MACHADO; MELLO, pp. 1-8, 2015) comparado com o (b) proposto.	85
Figura 72 – Exemplo de segmentação do método proposto para (a) imagem de planta baixa, (b) imagem <i>sketch</i> e resultados obtidos para as informações (c) textuais e (d) gráficas.	86
Figura 73 – Segmentação de (a) um antigo mapa de Pernambuco, (b) imagem de 10 segundos e suas informações (c) textuais e (d) gráficas.	87
Figura 74 – Exemplo de mapa com muita informação gráfica e textual, sendo (a) mapa original e suas imagens (b) <i>sketch</i> , (c) textual e (d) gráfica.	88
Figura 75 – (a) Resultado da técnica concedida por (MACHADO; MELLO, pp. 1-8, 2015) relacionada com (b) o SDC proposto como solução em outro domínio.	89
Figura 76 – Segmentação de (a) um antigo mapa de Pernambuco, (b) imagem de 10 segundos e suas informações (c) textuais e (d) gráficas.	89

Figura 77 – (a) Imagem original, (b) imagem <i>sh_colored</i> definida na forma antiga	
e (c) a usada neste trabalho, junta com os (d-f) mapas de saliência,	
respectivamente.	92
Figura 78 – (a) Imagem original, (b) padrão ouro e imagens de contorno obtidas	
pelos algoritmos (c) COB, (d) HED e (e) SDC10s.	93
Figura 79 – Dois exemplos de <i>Non-photorealistic Rendering</i> como possibilidade de	
aplicações futuras do trabalho.	94

LISTA DE TABELAS

Tabela 1 – Comparação entre os resultados do SDC proposto com outros algoritmos para o BSDS500.	60
Tabela 2 – Valores obtidos no <i>benchmark</i> após adição (gPb+SDC10s) ou remoção (gPb-SDC10s) das bordas detectadas pelo SDC10s no gPb.	61
Tabela 3 – Valores obtidos no <i>benchmark</i> após adição (SFE+SDC10s) ou remoção (SFE-SDC10s) das bordas detectadas pelo SDC10s no SFE.	61
Tabela 4 – Valores de Weighted- F_β para a nova abordagem adotada para aprimoramento do <i>Superpixel Hierarchy</i> em relação a antiga.	91

LISTA DE ABREVIATURAS E SIGLAS

<i>Deformable</i>	
HOG	<i>Deformable Histogram of Oriented Gradients</i>
AP	<i>Average Precision</i>
BDCN	<i>Bi-Directional Cascade Network for Perceptual Edge Detection</i>
BIPED	<i>Barcelona Images for Perceptual Edge Detection</i>
CNN	<i>Convolutional Neural Network</i>
COB	<i>Convolutional Oriented Boundaries</i>
CRP	<i>Chained Residual Pooling</i>
DB	Davies-Bouldin
DE	Distância Euclidiana
DexiNed	<i>Dense Extreme Inception Network for Edge Detection</i>
DNN	<i>Deep Neural Network</i>
DT	<i>Distance Transform</i>
EC	Elemento Conectado
gPb	<i>Globalized Probability of Boundary</i>
HED	<i>Holistically-Nested Edge Detection</i>
MBD	<i>Minimum Barrier Distance</i>
MRF	<i>Multi-Resolution Fusion</i>
ODS	<i>Optimal Dataset Scale</i>
OIS	<i>Optimal Image Scale</i>
OWT	<i>Oriented Watershed Transform</i>
RB	<i>Residual Blocks</i>
RCN	<i>Refine Contour Net</i>
RCU	<i>Residual Convolution Unit</i>
SDC	<i>Speed Drawing Challenge</i>
SEM	<i>Scale Enhancement Module</i>
SFE	<i>Structured Forest Edges</i>
SH	<i>Superpixel Hierarchy</i>
ST	<i>Sketch Tokens</i>
SVDD	<i>Support Vector Data Description</i>

UCM

Ultrametric Contour Map

SUMÁRIO

1	INTRODUÇÃO	17
1.1	HIPÓTESE	21
1.2	OBJETIVOS	21
1.3	ESTRUTURA DA TESE	21
2	TRABALHOS RELACIONADOS	23
2.1	MAIRE <i>ET AL.</i> (2008)	23
2.2	ARBELÁEZ <i>ET AL.</i> (2011)	26
2.3	LIM <i>ET AL.</i> (2013)	28
2.4	DOLLAR E ZITNICK (2013)	30
2.5	XIE E TU (2016)	31
2.6	MANINIS <i>ET AL.</i> (2017)	32
2.7	KELM <i>ET AL.</i> (2019)	35
2.8	SORIA <i>ET AL.</i> (2020)	36
2.9	HE <i>ET AL.</i> (2020)	37
3	HIPÓTESE	39
3.1	<i>SPEED DRAWING CHALLENGE</i>	39
3.2	SDC APLICADO PARA PROBLEMAS DE DETECÇÃO DE CONTORNO	41
3.3	SDC APLICADO PARA PROBLEMAS DE SEGMENTAÇÃO EM MAPAS E PLANTAS BAIXAS	42
4	MÉTODO PROPOSTO PARA SEGMENTAÇÃO DE IMAGENS DE CENAS NATURAIS	45
4.1	PARTE 1 - AGRUPAMENTO POR SUPERPIXELS	45
4.1.1	Definindo o Número de <i>Superpixel</i> de Forma Automática	48
4.1.2	Orientação de Borda	49
4.2	PARTE 2 - ANÁLISE DE TRAÇOS VIA MAPA DE SALIÊNCIA	54
4.2.1	Detecção de Bordas Fracas	55
4.2.2	Remoção de Traços Fora da Cena Principal	57
4.3	MODELANDO IMAGEM DE 1 MINUTO E 10 MINUTOS	58
5	RESULTADOS	60
5.1	DETECÇÃO DE BORDAS RELEVANTES	60
5.2	RESULTADOS QUALITATIVOS	62
6	APLICAÇÃO: SEGMENTAÇÃO DE MAPAS E PLANTAS BAIXAS	65

6.1	TRABALHOS RELACIONADOS	65
6.1.1	Ahmed <i>et al.</i> (2011)	65
6.1.2	Mello e Machado (2014).	68
6.1.3	Machado e Mello. (2015)	72
6.1.4	Zeng <i>et al.</i> (2019)	76
6.2	MÉTODO DE SEGMENTAÇÃO POR SDC	77
6.2.1	Pré-processamento	78
6.2.2	Segmentação de informações gráficas e textuais	79
6.2.3	<i>Speed Drawing</i>	79
6.3	RESULTADOS	81
6.3.1	Segmentação de plantas baixas	82
6.3.2	Segmentação de mapas	82
7	CONCLUSÃO	90
7.1	CONSIDERAÇÕES FINAIS	90
7.2	PRINCIPAIS CONTRIBUIÇÕES	91
7.3	TRABALHOS FUTUROS	94
7.4	PUBLICAÇÕES	95
	REFERÊNCIAS	96

1 INTRODUÇÃO

Com o desenvolvimento tecnológico dos últimos tempos, a execução automática de tarefas por parte dos computadores tem se tornado essencial em várias aplicações. No campo da visão computacional, destacamos: sistemas de reconhecimento facial (DAVE; VYAS; DESAI, 2018), veículos autônomos (LI et al., 2018) e sistemas de diagnósticos médico (MEHTA et al., 2018a; MEHTA et al., 2018b), onde o objetivo é fornecer informações mais precisas sobre o paciente.

O primeiro passo de um sistema usual de visão computacional (SZELISKI, 2010; DAVIES, 2004) é a aquisição da imagem que, após ser digitalizada, fica disponível para a realização de diversas operações. A imagem resultante é representada por pixels que, normalmente, apresentam a intensidade de luz em um determinado sistema de cores.

Após a digitalização, é bastante comum a aplicação de um pré-processamento, visando preparar a imagem para uma próxima etapa. Exemplos incluem a utilização de filtros para remoção de ruídos ou intensificação de contraste para garantir que a imagem esteja de acordo com as condições necessárias para a aplicação.

Em seguida, dependendo da aplicação, diferentes caminhos podem ser seguidos. Um deles é a extração de características da imagem que visa coletar informações de diferentes complexidades a depender do objetivo. Análise de texturas (LUKASHEVICH; SADYKHOV, 2012) e detecção de formas (DICKINSON; PIZLO, 2013) são possíveis exemplos.

Outra possibilidade é a segmentação da imagem. Em um sistema de reconhecimento facial, por exemplo, uma região relevante da imagem é segmentada e submetida à detecção de rosto para posterior reconhecimento. De forma geral, a segmentação e detecção compõem as etapas mais difíceis em um sistema de visão computacional, podendo determinar um eventual sucesso ou falha de todo o sistema.

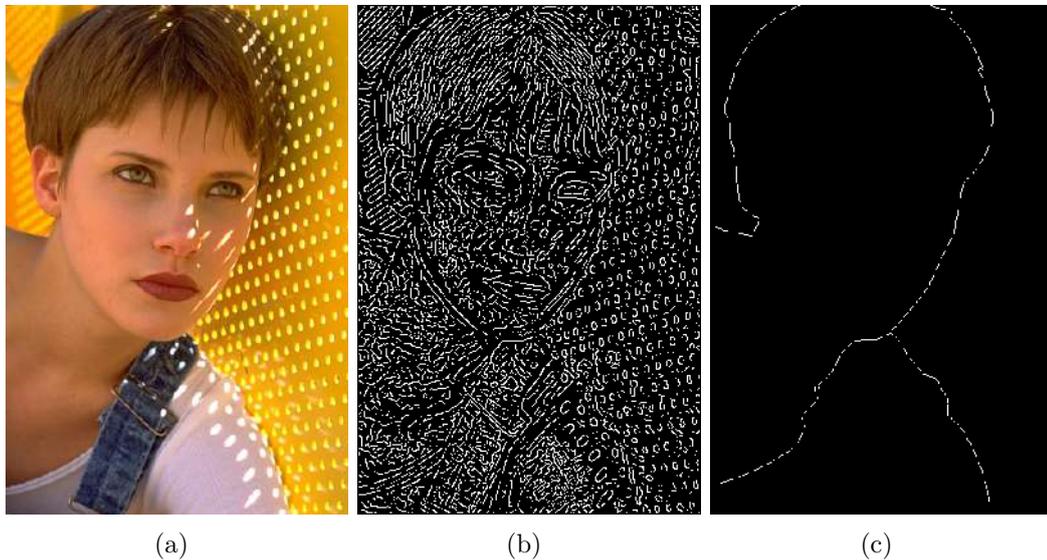
A segmentação de imagens (GONZALEZ; WOODS, 2001) é um dos principais desafios no campo da visão computacional e tido como um passo essencial para outras tarefas da área (AKTER; KWON, 2018; AKININA; NIKIFOROV; SAVIN, 2018; RIOMOROS et al., 2010). Ela tem como objetivo separar a imagem de entrada em determinadas partes, transformando suas informações mais simples de serem analisadas e facilitando passos seguintes de um sistema de visão computacional.

As técnicas de segmentação são divididas basicamente em duas principais categorias: segmentação por descontinuidade ou por similaridade. A segmentação por descontinuidade consiste em dividir a imagem com base nas mudanças bruscas de intensidade, como nas fronteiras entre cores. Já a segmentação por similaridade tem como base realizar a segmentação por meio de regiões que sejam semelhantes a partir de algum critério pré-definido. O crescimento de regiões é um exemplo de técnicas baseadas em similaridade.

A detecção de borda é um tipo de segmentação por descontinuidade que tem como

um dos principais problemas a quantidade exagerada de traços detectados. Na Figura 1, por exemplo, para a imagem de entrada (Figura 2(a)), a sua imagem de borda (Figura 2(b)) exemplifica este problema. Já a Figura 2(c), representa uma imagem de contorno, ou seja, traços detectados a partir da supressão de informações (como textura, cor e variação de iluminação) da imagem de entrada. Este processo é chamado de detecção de contorno. Métodos clássicos de detecção de bordas como: Prewitt (PREWITT, 1970), Sobel (SOBEL, 2014), Canny (CANNY, 1986) e outros (DHARAMPAL; MUTNEJA, 2015), são comumente utilizados na detecção de contorno, simultaneamente com outras características da imagem, tais como, contraste, cor ou textura.

Figura 1 – (a) Imagem original, (b) imagem contendo bordas detectadas pelo algoritmo de Canny e (c) imagem de contorno.



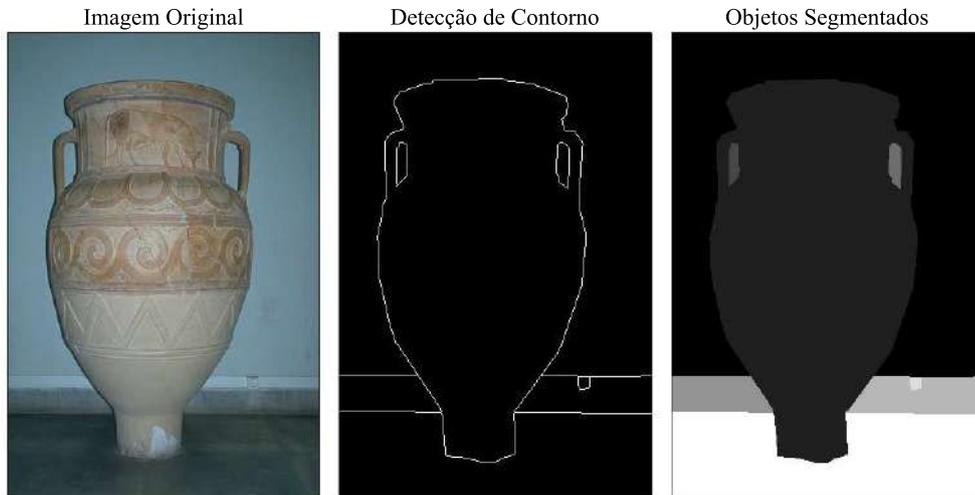
Fonte: Marcos Canêjo (2022)

A Figura 2 (ARBELÁEZ et al., 2011) apresenta outro exemplo de segmentação gerada a partir da detecção de contorno. Observa-se que detalhes do vaso representados por desenhos circulares, triangulares e um animal quadrúpede na parte superior, não fazem parte do conceito de contorno da imagem original.

As técnicas de detecção de contorno podem ser aplicadas em diversos tipos de problemas, como detecção de objetos (GIRSHICK et al., 2014), detecção de placas de carros (QUIROS et al., 2017), sistemas de prevenção de acidentes em estradas (PANDEY; KULKARNI, 2018), etc. Quase sempre, a detecção de contorno se encontra nas primeiras etapas de diversas aplicações.

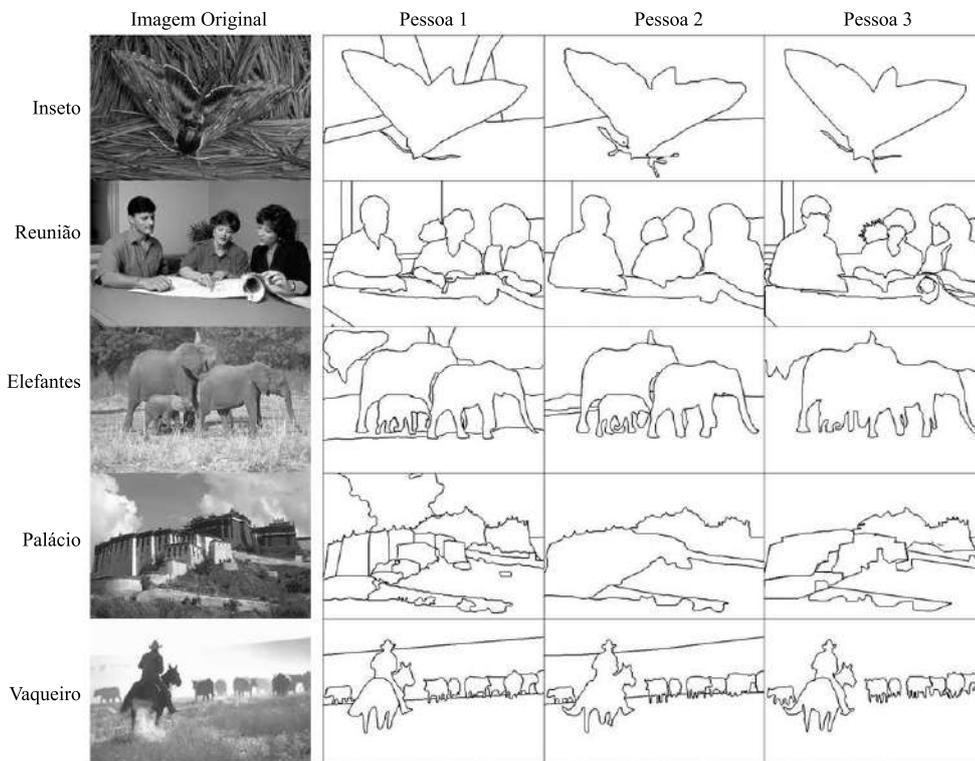
O contorno de uma imagem pode se tornar algo extremamente complexo e relacionado ao processo decisório do ser humano conforme mostra a Figura 3 (MARTIN et al., 2001a). Para cada uma das 5 imagens, 3 pessoas diferentes consideraram contornos diferentes. Nota-se que, a depender da imagem, uma pessoa pode produzir mais contornos ou menos contornos quando comparada com outra pessoa.

Figura 2 – Exemplo de segmentação pelo método de detecção de contorno. A primeira imagem representa a imagem original, ou seja, um vaso esculpido com vários detalhes. Na imagem seguinte, o seu contorno é detectado com interesse maior no objeto, ignorando vários de seus detalhes. Por fim, o resultado da segmentação é apresentada.



Fonte: Marcos Canêjo (2022)

Figura 3 – Contornos de 5 imagens feito por 3 pessoas diferentes. Por exemplo, para a imagem Elefantes, o contorno da Pessoa 1 possui mais detalhes do que o da Pessoa 3; porém, para a imagem Reunião, a Pessoa 3 produziu mais contornos.

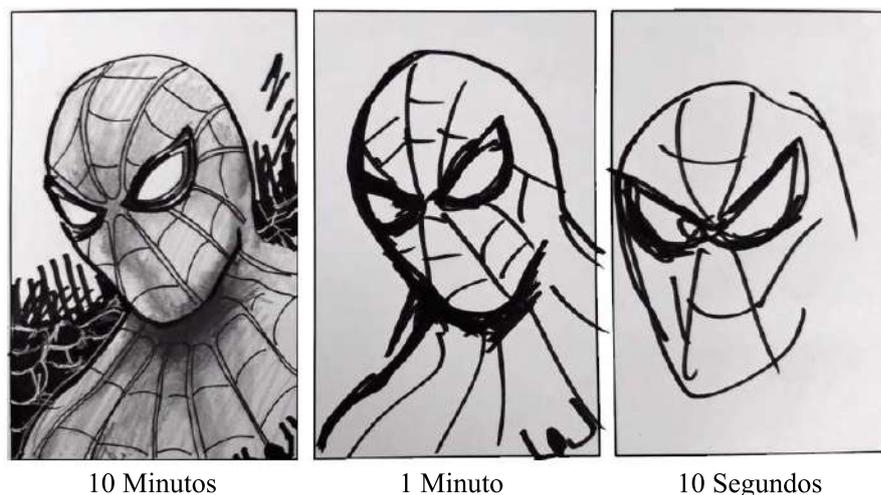


Fonte: MARTIN et al. (2001a)

Uma imagem apenas com contornos, como as apresentadas na Figura 3, pode ser vista como um desenho. Se imaginarmos um artista, fazendo um rascunho de uma imagem de

uma cena natural, sem se preocupar com coloração, ele estará gerando uma imagem de contornos. Dependendo do tempo que tenha para fazer essa imagem, ela poderá ter mais ou menos detalhes. A prática de desafios é algo bastante comum em diversas áreas. Nas artes, o *Speed Drawing Challenge (SDC)* (SPEED..., 2018), exemplificado na Figura 4, pode ser visto como uma forma de elaborar um melhor entendimento sobre a capacidade do ser humano em selecionar certos traços para melhor representação de um determinado desenho, algo relacionado ao problema de detecção de contorno. O objetivo é simples: produzir desenhos, com limitações de tempo, da forma mais semelhante possível. Por exemplo, o artista precisa demonstrar suas habilidades em desenhos de 10 minutos, 1 minuto e 10 segundos, provocado cada vez mais em representar um desenho capaz de ser compreendido independente do tempo disponível. No *SDC*, as imagens de mais tempo são normalmente realizadas primeiro, pois servem como um treinamento para o desenho de 10 segundos.

Figura 4 – Exemplo do *SDC* para 10 minutos, 1 minuto e 10 segundos.



Fonte: ART (2022)

Portanto, com inspiração no *SDC*, que representa em sua natureza a possível maior competência no processo decisório humano por conter artistas especializados na arte de desenhar, este trabalho propõe-se a investigar uma modelagem do *SDC* como possível solução para problemas associados à detecção de contorno.

Além disso, a mesma inspiração é empregada para um problema de segmentação em imagens de plantas baixas e mapas, onde um dos principais desafios é exatamente o cenário em que linhas finas estão sobrepostas nos elementos textuais tornando difícil a separação do texto dos elementos gráficos.

1.1 HIPÓTESE

Fazer o artista realizar um desenho em segundos é o grande objetivo do **SDC**. A partir deste conceito de restrição no tempo ou abstração do desenho, nossa hipótese é que é possível usar esta capacidade do ser humano em selecionar traços expressivos de um desenho como uma inspiração para um melhor entendimento sobre as principais características que diferem uma borda de um contorno. Isto é, em um desenho de 10 segundos, por exemplo, a escolha dos traços que melhor representam o desenho se torna mais evidente, ou seja, uma detecção de contorno é executada pelo artista. Esse conceito pode ser aplicado a diferentes domínios cujo objetivo seja detecção dos principais traços de uma imagem de bordas ou um desenho.

1.2 OBJETIVOS

O objetivo deste trabalho é desenvolver um algoritmo eficiente para a detecção de contorno através da modelagem do **SDC**. A modelagem então consiste em reproduzir o comportamento humano no processo decisório na escolha dos traços principais de um desenho para sua representação. Da mesma forma, e de acordo com o **SDC**, o método deve ser capaz de se comportar de forma diferente a fim de reproduzir os resultados de um desafio de diferentes tempos.

Os objetivos específicos são:

- Propor uma abordagem para o problema de detecção de contornos através da modelagem do **SDC**;
- Desenvolver um algoritmo único que consiga gerar imagens de contornos com diferentes níveis de detalhes;
- Desenvolver um algoritmo para detecção de contornos que gere uma imagem final de contornos, tornando-o apto a aplicações de tempo real;
- Aplicar a ideia do *Speed Drawing Challenge* em outros domínios.

1.3 ESTRUTURA DA TESE

Nos capítulos seguintes, são apresentados os trabalhos analisados com o detalhamento de aspectos importantes de suas implementações, a nova abordagem com inspiração no **SDC** e resultados obtidos para o problema de detecção de borda. No Capítulo **2** o estado da arte é apresentado com as principais técnicas da área. Em seguida, a Tese é apresentada no Capítulo **3** com suas implicações no domínio de detecção de contorno e a descrição do método proposto no Capítulo **4** para segmentação de imagens de cenas naturais com seus resultados expostos no Capítulo **5**. O Capítulo **6** apresenta como a ideia do **SDC** pode

ser usada em outro domínio e, por fim, no Capítulo 7, temos as considerações finais e trabalhos futuros.

2 TRABALHOS RELACIONADOS

Este capítulo apresenta algumas das principais técnicas de detecção de contorno baseadas em diferentes estratégias, destacando algumas técnicas baseadas em redes neurais convolucionais. Os resultados são apresentados de forma qualitativa e obtidos a partir dos artigos originais.

2.1 MAIRE ET AL. (2008)

O [Globalized Probability of Boundary \(gPb\)](#) [\(MAIRE et al., 2008\)](#) faz parte de uma categoria de algoritmos dentro do problema de detecção de contorno que busca uma solução por meio da utilização de informações globais. Ele é uma extensão do detector de contorno proposto por Martin *et al* [\(MARTIN; FOWLKES; MALIK, 2004\)](#), denominado Pb, que faz uso apenas das informações locais de cor, textura e contraste.

Dada uma imagem de entrada I , o algoritmo Pb calcula o gradiente orientado $G(x, y, \theta)$ por meio do posicionamento de um disco dividido em duas partes com um diâmetro definido para cada ângulo θ . Para cada uma das duas partes do disco, um histograma dos valores de intensidade é calculado e a magnitude do gradiente G em (x, y) é definida pela distância X^2 entre esses dois histogramas (g e h), ou seja,

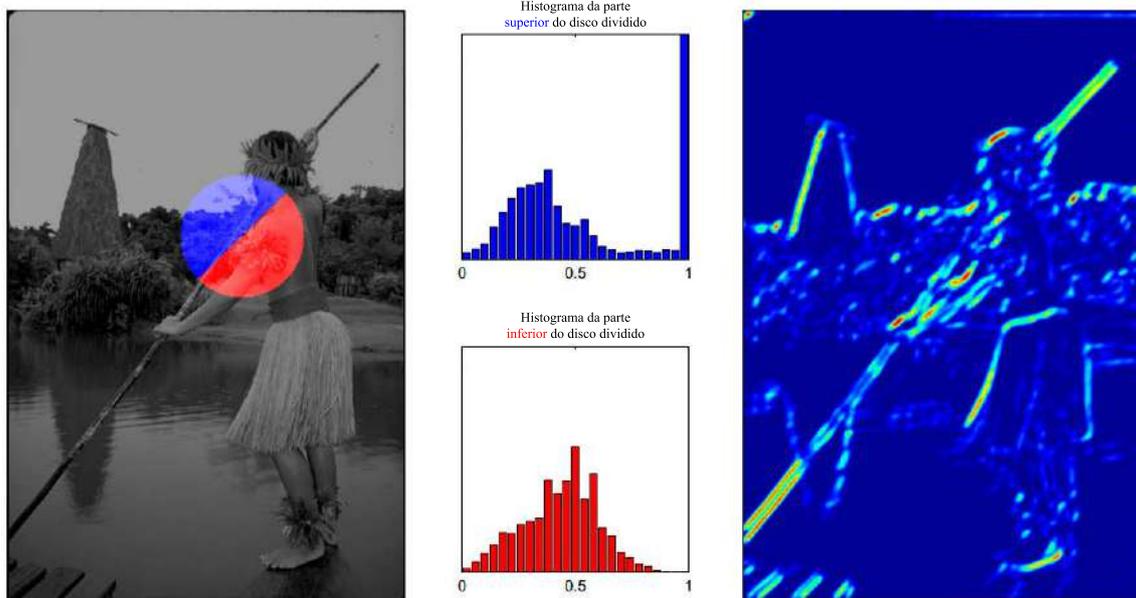
$$X^2(g, h) = \frac{1}{2} \sum_i \frac{(g(i) - h(i))^2}{g(i) + h(i)}. \quad (2.1)$$

A Figura [5](#) ilustra este processo com o disco dividido em duas partes, uma azul e outra vermelha, juntamente com o resultado do cálculo para um disco com diâmetro igual a 50 e $\theta = \frac{\pi}{4}$.

O próximo passo é a aplicação do filtro Savitzky-Golay [\(SAVITZKY; GOLAY, 1964\)](#) para realçar os máximos locais e suavizar os picos para cada orientação de θ . Em seguida, uma função de probabilidade *a posteriori* é aplicada em 4 canais diferentes de I , onde os 3 primeiros correspondem ao modelo de cor CIELAB (L^* , a^* , b^*) nomeados pelo autor como canal de brilho, canal de cor “a” e canal de cor “b”. Para imagens em tons de cinza, o canal de contraste é a imagem em si e nenhum canal de cor é usado. O quarto canal é o de textura que atribui para cada pixel um *id* antes do cálculo do histograma de gradiente orientado. Esta etapa transforma a imagem I em tons de cinza e executa uma convolução com 17 filtros derivativos Gaussianos sendo um “centralizado” como exibe a Figura [6](#). Cada pixel é representado por um vetor de 17 intensidades que são agrupados pelo algoritmo *K-Means*, gerando uma imagem onde cada pixel assume o valor (*id*) que representa o seu grupo. Isto é, os pixels rotulados para o grupo 1, por exemplo, assumem o valor 1; os rotulados para o grupo 2, assumem o valor 2, e assim por diante. A imagem resultante deste processo é submetida ao cálculo do gradiente orientado como os outros

canais. A Figura 7 apresenta um exemplo do cálculo do gradiente orientado para os canais de contraste, cor e textura com $\theta = 0$ e $\theta = \frac{\pi}{2}$.

Figura 5 – Exemplo do posicionamento do disco para cálculo do gradiente orientado. O diâmetro do disco é de 50 pixels apenas para ilustração.



Fonte: ARBELÁEZ et al. (2011)

Figura 6 – Orientação dos 17 filtros derivativos Gaussianos.

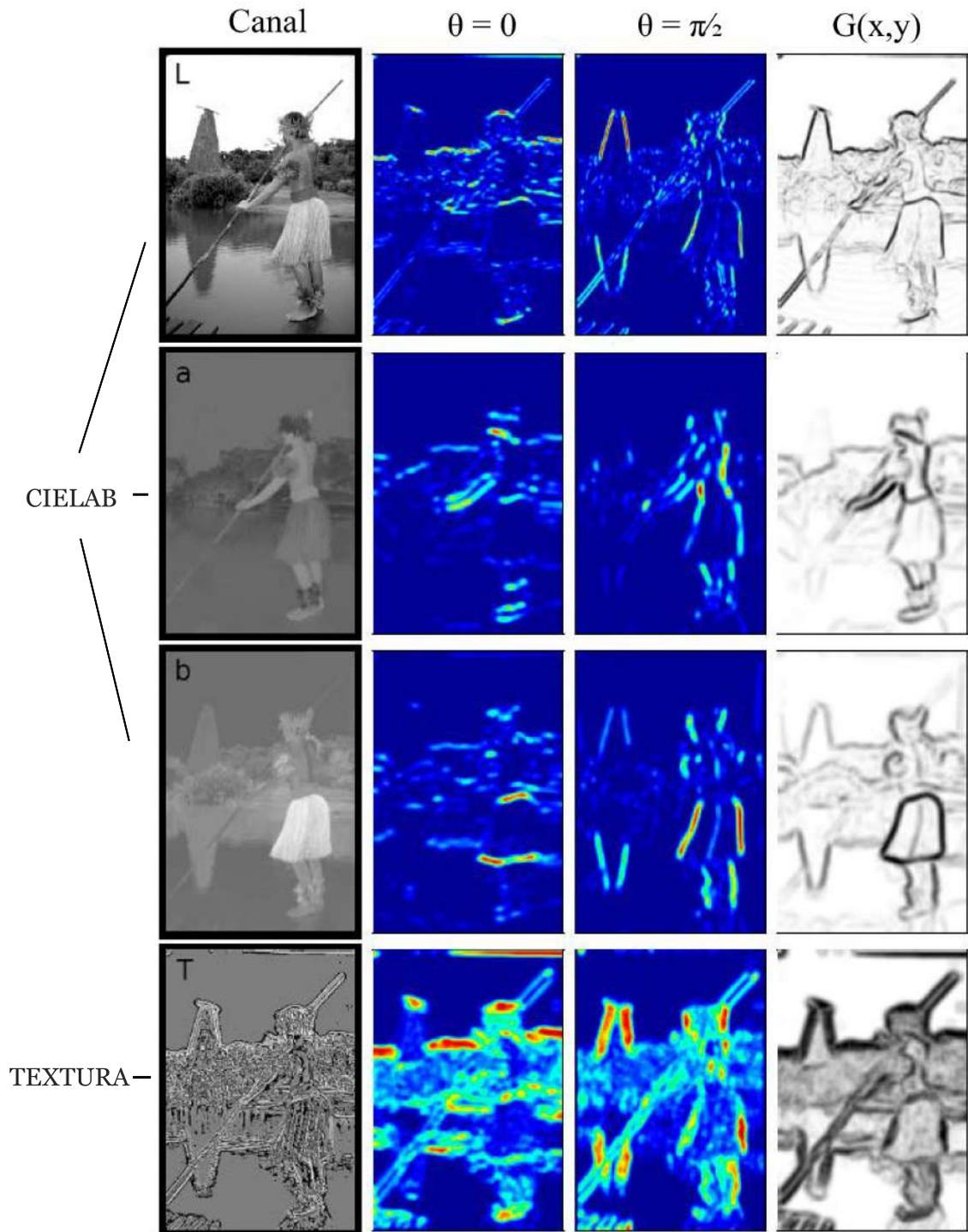


Fonte: ARBELÁEZ et al. (2011)

Assim, o Pb gera uma saída $Pb_{\sigma}(x, y, \theta)$ que representa uma probabilidade *a posteriori* da existência de contorno para cada pixel da imagem calculada em canais diferentes por meio do posicionamento de um disco de raio σ dividido em duas partes e com uma orientação θ . A imagem de contorno é obtida por meio da extração dos valores máximos para cada orientação que são, em seguida, afinados pelo algoritmo de supressão não-máxima (*non-maximum suppression*).

O gPb processa a saída $Pb_{\sigma}(x, y, \theta)$, porém, com o objetivo de detectar estruturas finas e espessas, Maire *et al* propõem a utilização de 3 tamanhos diferentes para o raio σ : $[\frac{\sigma}{2}, \sigma, 2\sigma]$, onde σ representa o valor padrão do algoritmo Pb. Os valores são então obtidos para 8 orientações no intervalo $[0, \pi]$ e unificados em um único sinal multi-escalar por uma soma ponderada que caracteriza a informação local das bordas que é definida por uma matriz esparsa de afinidade que utiliza a resposta máxima de contorno ao longo de uma linha conectada a 2 pixels de borda. No passo seguinte, os 8 autovetores do sistema matricial são tratados como imagens e submetidos a convolução de filtros derivativos

Figura 7 – Exemplo do gradiente orientado para cada canal.



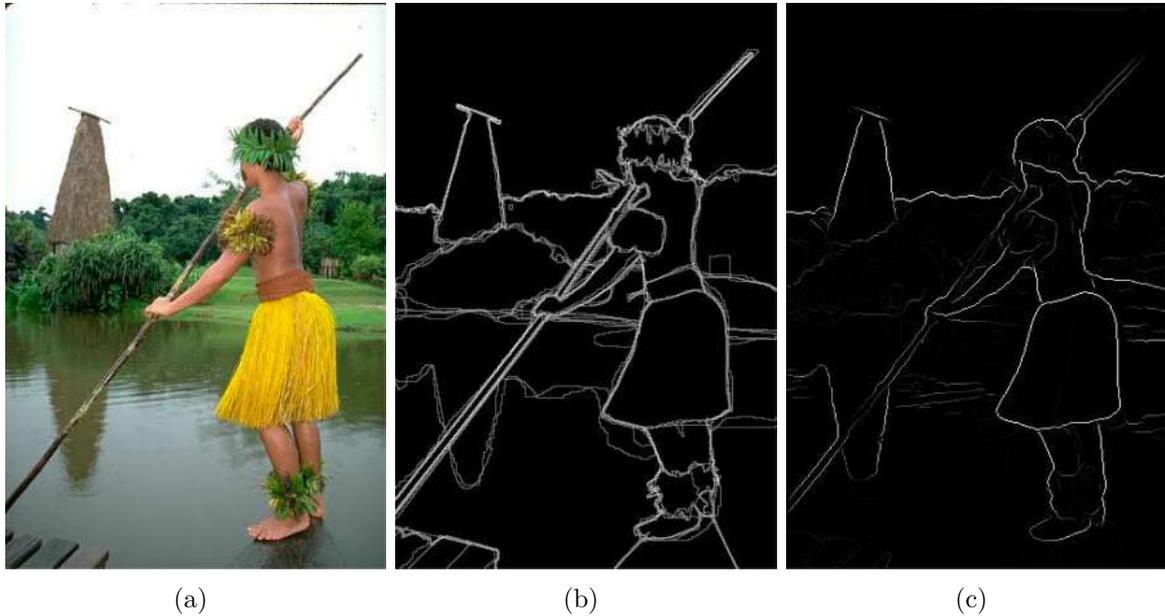
Fonte: [ARBELÁEZ et al. \(2011\)](#)

Gaussianos para cada orientação, combinando suas informações resultantes. Enfim, ambas as informações locais e de borda são unificadas por uma soma ponderada produzindo a detecção de contorno na imagem.

A Figura 8 apresenta o resultado do [gPb](#) com o padrão ouro ([MARTIN et al. 2001b](#)) da imagem. É possível notar que várias bordas pertencentes ao contorno não foram de-

tectadas, principalmente nas regiões mais difíceis da imagem constituídas por um tipo de cocar na cabeça, e ornamentos no braço e no tornozelo da figura indígena.

Figura 8 – (a) Imagem original, com (b) seu padrão ouro e (c) contorno detectado pelo gPb (c).



Fonte: Marcos Canêjo (2022)

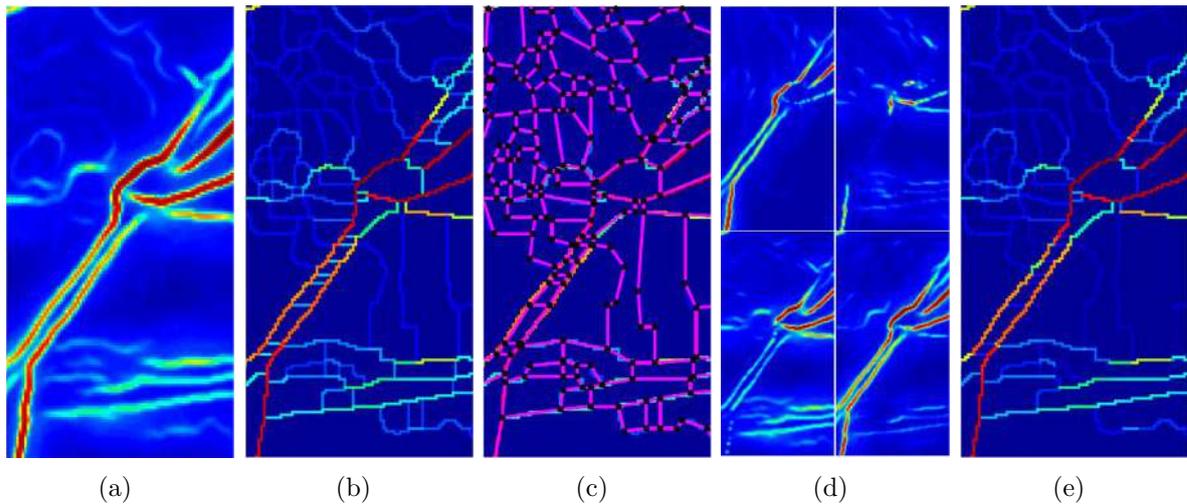
2.2 ARBELÁEZ ET AL. (2011)

Como dito previamente, a detecção de contorno tem um papel fundamental em sistemas que dependem de segmentação. Uma imagem de contorno pode ser a entrada de um algoritmo de segmentação. No método proposto em (ARBELÁEZ et al., 2009), o objetivo é exatamente este: segmentar uma imagem a partir de seus contornos detectados previamente. Os autores propõem a aplicação da *Oriented Watershed Transform (OWT)* e do *Ultrametric Contour Map (UCM)* (ARBELÁEZ, 2006), visando construir uma imagem com contornos fechados, facilitando a sua segmentação. Assim, o algoritmo pode ser empregado tanto para segmentação como também na detecção de contorno. Qualquer imagem de contorno pode ser utilizada como entrada do novo método, mas, por questões de desempenho da época de sua criação, os autores adotam imagens de contorno resultantes do gPb, dando origem ao algoritmo gPb-owt-ucm, mas comumente chamado de gPb-ucm quando empregado para o problema de detecção de contorno.

A imagem de contorno de entrada $E(x, y, \theta)$ contém a probabilidade de existência de contorno no ponto (x, y) com orientação θ . Esses valores são denominados de *Oriented Boundary Strength*, ou seja, um valor que representa a “força” de um determinado contorno existir em uma orientação. O primeiro passo é produzir um conjunto de regiões iniciais a partir dessa imagem de contorno. A resposta máxima para cada orientação

$(\max_{\theta} E(x, y, \theta))$ é obtida, seguida da aplicação da *Transformada de Watershed* (NAJMAN; SCHMITT, 1996), indicada nas Figuras 10(a) e 10(b), respectivamente, produzindo pequenas regiões delimitadas por arcos K_0 que possivelmente representam a localização dos contornos. Posteriormente, os valores do *Oriented Boundary Strength* são integrados aos arcos K_0 , representados por segmentos de linhas, isto é, cada ponto em K_0 é ponderado pelo seu respectivo valor em $E(x, y, \theta)$, dando origem ao que os autores chamam de *Transformada de Watershed Orientada*. A Figura 10(c) apresenta os arcos como segmentos de linhas, a Figura 10(d) exemplos do *Oriented Boundary Strength* para quatro orientações e a Figura 10(e) o resultado final. A Figura 9 é apresentada em um mapa de cor azul (detecção fraca) e vermelho (detecção forte) ou baixa e alta probabilidade de existir borda, respectivamente.

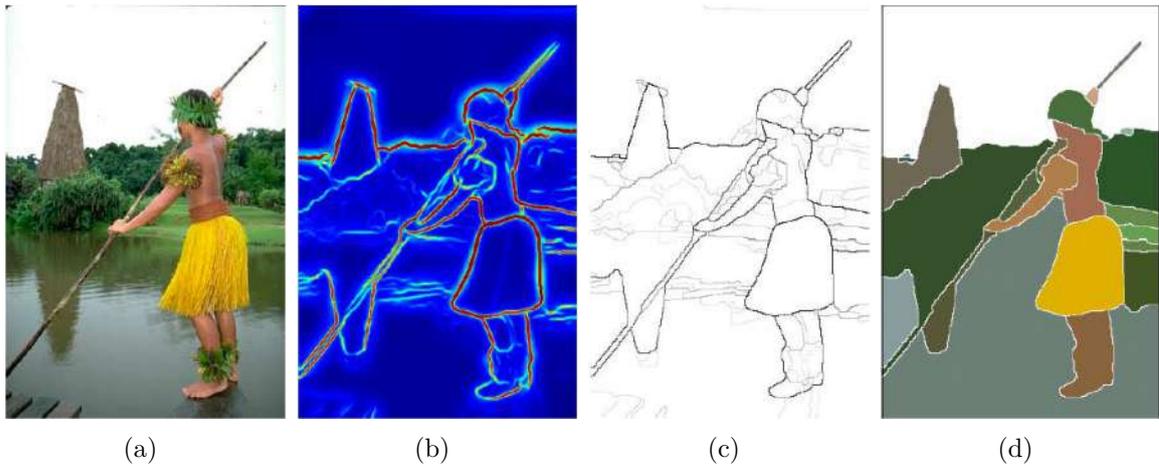
Figura 9 – Etapas do processo de obtenção da *Transformada de Watershed Orientada*: (a) imagem de contorno $E(x, y, \theta)$ de entrada, (b) arcos resultantes da aplicação da *Transformada de Watershed*, (c) segmentos de linhas formados pelos pontos de extremidades dos arcos, (d) exemplo do *Oriented boundary strength* para quatro valores de θ e (e) os arcos finais formados pela associação dos *Oriented boundary strength* com os segmentos de linhas.



Fonte: ARBELÁEZ et al. (2009)

Em seguida, as regiões criadas pela OWT são reunidas de forma hierárquica por um algoritmo guloso de fusão de regiões baseado em um grafo. Inicialmente, um grafo é definido onde os nós são as regiões com ligações a regiões adjacentes ponderadas por uma medida de similaridade. O algoritmo então organiza os nós a partir da medida de similaridade e inicia a fusão das regiões. Por fim, após o processo de fusão, uma imagem de contorno é obtida com contornos fechados, como pode ser visto na Figura 10, por meio de uma representação hierárquica chamada de UCM.

Figura 10 – Resultados do gPb-owt-ucm: (a) imagem original), (b) imagem de resposta máxima do contorno detectado pelo gPb, (c) imagem resultante do gPb-ucm com imagem de contorno de entrada obtida pelo gPb e (d) imagem segmentada.



Fonte: ARBELÁEZ et al. (2009)

2.3 LIM ET AL. (2013)

O trabalho publicado em (LIM; ZITNICK; DOLLAR, 2013) propõe uma abordagem baseada em *Sketch Tokens (ST)*. Isto é, dado um conjunto de imagens I , com sua imagem binária correspondente S , representada por um esboço de contorno criado por um humano; o conjunto de *ST* é composto por um agrupamento de pequenas partes (*patches*) s extraídas da imagem binária S , como pode ser visto na Figura 11. A ideia dos autores é criar um conjunto de representações (*ST*) de diversas estruturas de bordas que possam existir em uma imagem, abrangendo linhas, curvas, linhas paralelas, etc.

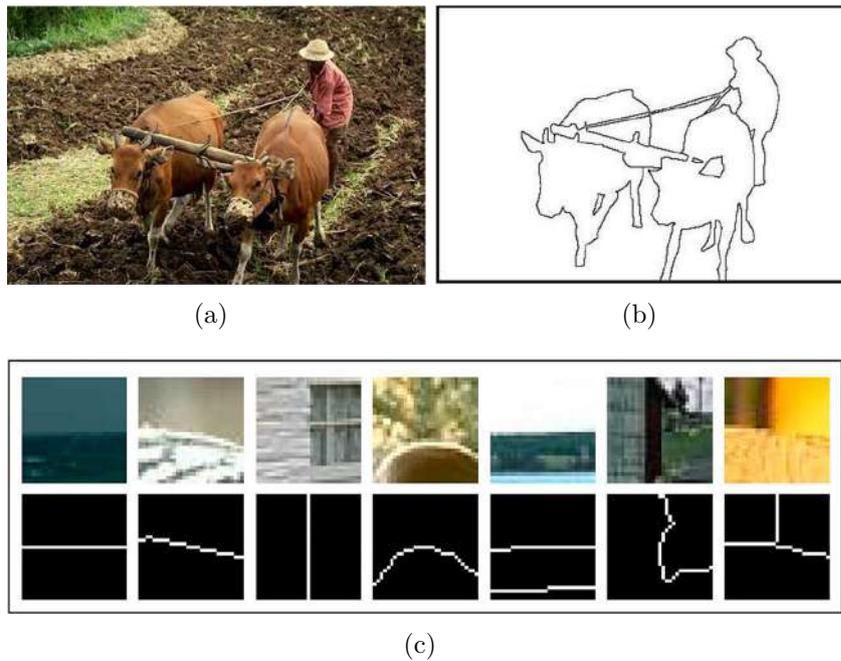
Cada *patch* s_j extraído de sua imagem binária S_i possui o tamanho fixo de 35×35 pixels e somente aqueles que contêm um pixel de contorno posicionado no centro da imagem são escolhidos. A Figura 12 exemplifica *ST* selecionados, variando de linhas retas a estruturas mais complexas que foram submetidos ao descritor DAISY (WINDER; HUA; BROWN, 2009), visando torná-los invariantes a pequenas mudanças da borda.

Com os *ST* definidos, o próximo passo é detectá-los em imagens coloridas de teste. A detecção é feita por meio de um classificador treinado com características obtidas de *patches* coloridos x extraídos das imagens de treino I , relacionados aos *ST* já escolhidos. Dois tipos de características, ambas baseadas no trabalho de Dollár *et al* (DOLLÁR et al., 2009), são obtidas do *patch* colorido: um diretamente indexado aos canais de cor, gradiente e gradiente orientado; e características de auto-similaridade (*self-similarity features*), formando um vetor com 21.350 elementos. A Figura 13 ilustra a detecção de dois *ST* diferentes.

A extração de características proposta por Dollár *et al.* consiste na utilização de vários canais de uma imagem para coletar informações como somatórios locais, histogramas,

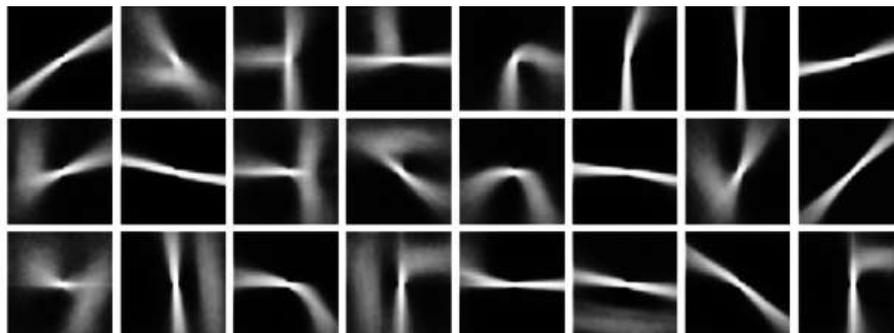
wavelet de Haar, etc, nomeadas *integral channel features*. No algoritmo de Lim *et al.*, 3 canais de cores, respectivos ao espaço de cor CIELAB, são utilizados, juntamente com 3 canais de magnitude de gradiente e 8 canais de gradiente orientado, totalizando 14 canais, que geram um vetor de 17.150 elementos.

Figura 11 – (a) Imagem original I , (b) imagem binária S de (c) contorno criada por um humano e exemplos de *patches* s extraídos.



Fonte: LIM; ZITNICK; DOLLAR (2013)

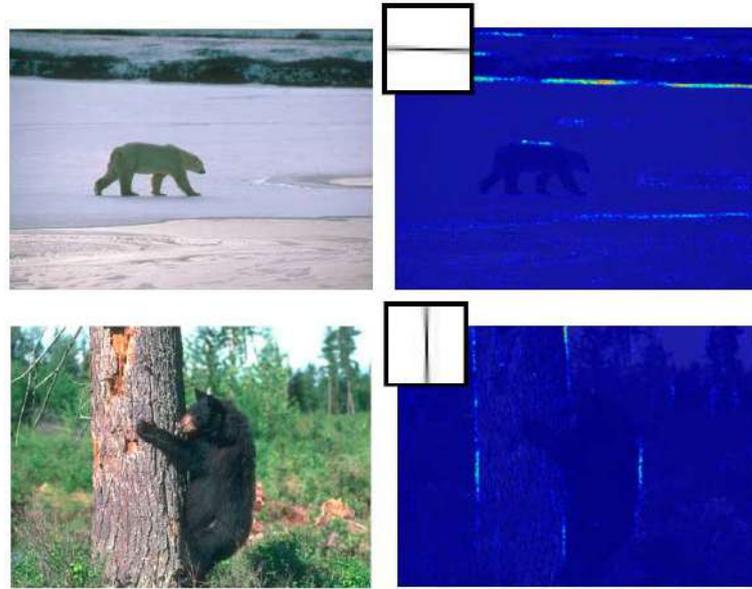
Figura 12 – Exemplo dos **ST**.



Fonte: LIM; ZITNICK; DOLLAR (2013)

As características de auto-similaridade buscam identificar possíveis contornos em regiões texturizadas nos mesmos 14 canais. Para cada canal, 300 novas características são geradas, ou seja, $300 \times 14 = 4.200$ características de auto-similaridade. Em seguida, um vetor de características com 21.350 elementos é formado a partir das duas extrações propostas. Por fim, o vetor é submetido a um classificador (*random forest* (BREIMAN, 2001))

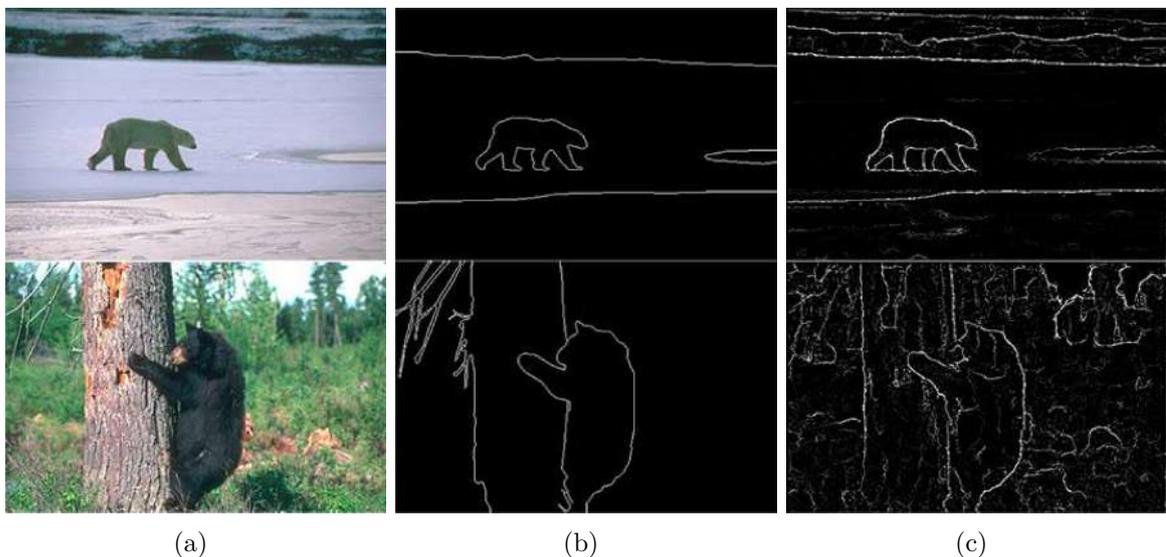
Figura 13 – Exemplo de detecção de 2 tipos de **ST**.



Fonte: LIM; ZITNICK; DOLLAR (2013)

treinado para o conjunto de validação do BSDS500 com seu resultado exemplificado na Figura 14.

Figura 14 – (a) Colunas com imagens originais, (b) padrão ouro e (c) contornos detectados pelo algoritmo **ST**.



Fonte: Marcos Canêjo (2022)

2.4 DOLLAR E ZITNICK (2013)

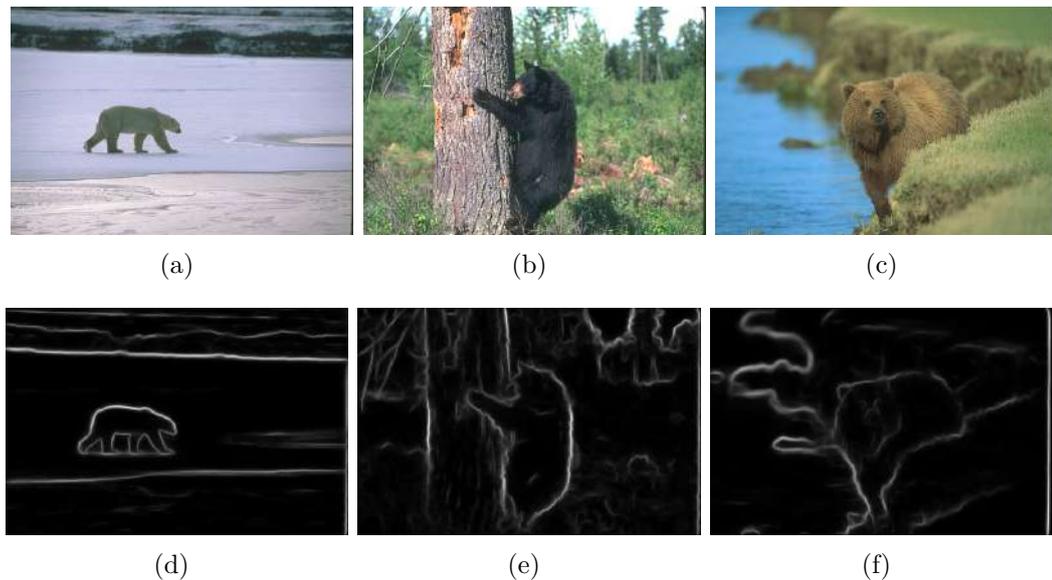
No trabalho de Dollar e Zitnick (DOLLAR; ZITNICK, 2013), um algoritmo para detecção de borda baseado no *structured learning* (NOWOZIN; LAMPERT et al., 2011) é proposto. A

ideia consiste em treinar uma *random forest* (KONTSCHIEDER et al., 2011) para capturar informações de estruturas como linhas retas, linhas paralelas e junções do tipo Y e T (REN; FOWLKES; MALIK, 2006) a partir de segmentos para uma imagem de entrada.

A *random forest* é implementada por múltiplas árvores de decisão treinadas de forma independente. Para cada árvore, um vetor contendo 7.228 características é calculado a partir de uma série de operações envolvendo diversos canais de gradiente, cor, etc, descritas no trabalho, dando origem ao *Structured Forest Edges (SFE)*.

Os autores usam as 200 imagens de treino e 100 imagens de validação do BSDS500 para o treinamento. Cada árvore foi treinada de forma paralela em 30 minutos utilizando um milhão de segmentos. O processo de classificação, ou seja, o *output* da *random forest* também pode ser paralelizado, tornando rápido seu treino e uso. A Figura 15 apresenta resultados obtidos para o conjunto de teste do BSDS500.

Figura 15 – (a-c) Imagens de entrada e seus (d-f) resultados para o algoritmo *SFE*.



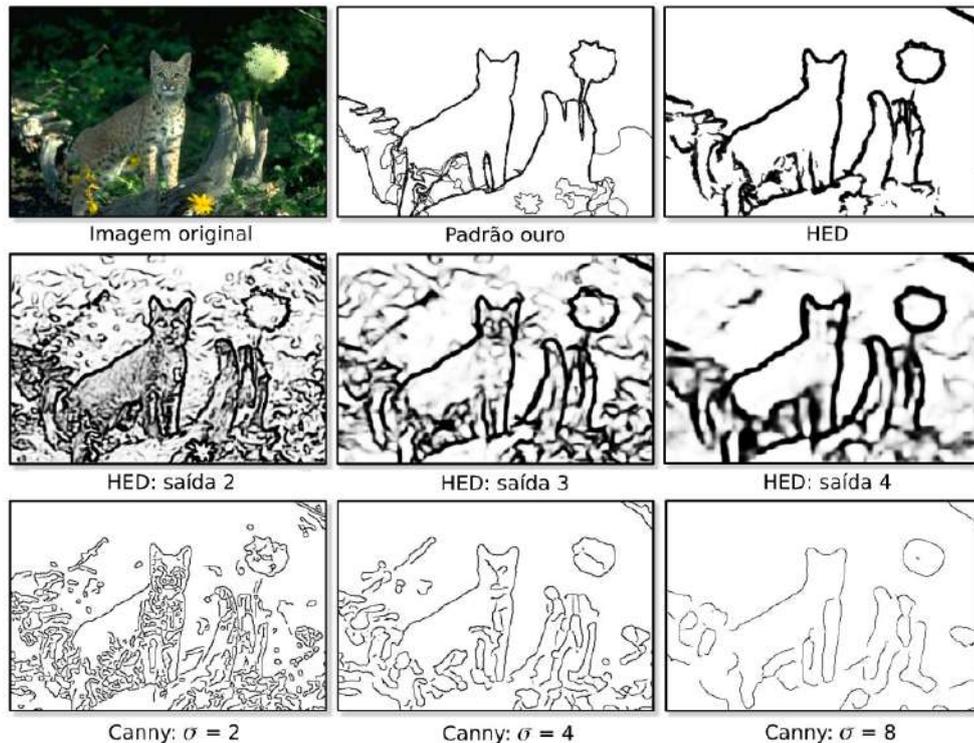
Fonte: Marcos Canêjo (2022)

2.5 XIE E TU (2016)

Com a popularização das *Deep Neural Network (DNN)*, em específico a *Convolutional Neural Network (CNN)*, várias técnicas de detecção de borda e contorno surgiram com foco em seu uso. Uma das principais características de uma arquitetura *deep learning* que faz uso de camadas convolucional, por exemplo, são as representações básicas (borda, traços, linhas, etc) codificadas nas suas primeiras camadas e representações mais complexas (quadrados, retângulos, etc) nas camadas mais profundas, ou seja, uma análise multiescala é produzida na rede.

A [Holistically-Nested Edge Detection \(HED\)](#) (XIE; TU, 2015) tem como principal ideia juntar as saídas de várias camadas de convolução em uma única imagem, algo parecido com a abordagem multiescala, exemplificado na Figura 16 com o algoritmo de Canny.

Figura 16 – Proposta multiescala da [HED](#).



Fonte: [XIE; TU \(2015\)](#)

Visando obter uma rede com várias camadas de convolução, os autores fizeram uso da VGGNet ([SIMONYAN; ZISSERMAN, 2014](#)), porém com algumas modificações. Na primeira, as camadas de convolução conv1, conv2, conv3, conv4 e conv5 foram conectadas às saídas 2, 2, 3, 3, 3, respectivamente, e com as camadas de saída possuindo os mesmos parâmetros. A segunda modificação consiste em um corte na arquitetura, ou seja, remoção da última camada totalmente conectada e da camada de *pooling*, visto que uma nova camada é criada resumindo as informações de todas as outras saídas. A arquitetura final da [HED](#) é formada por 5 camadas de convolução com valores de *strides* 1, 2, 4, 8 e 16, respectivamente.

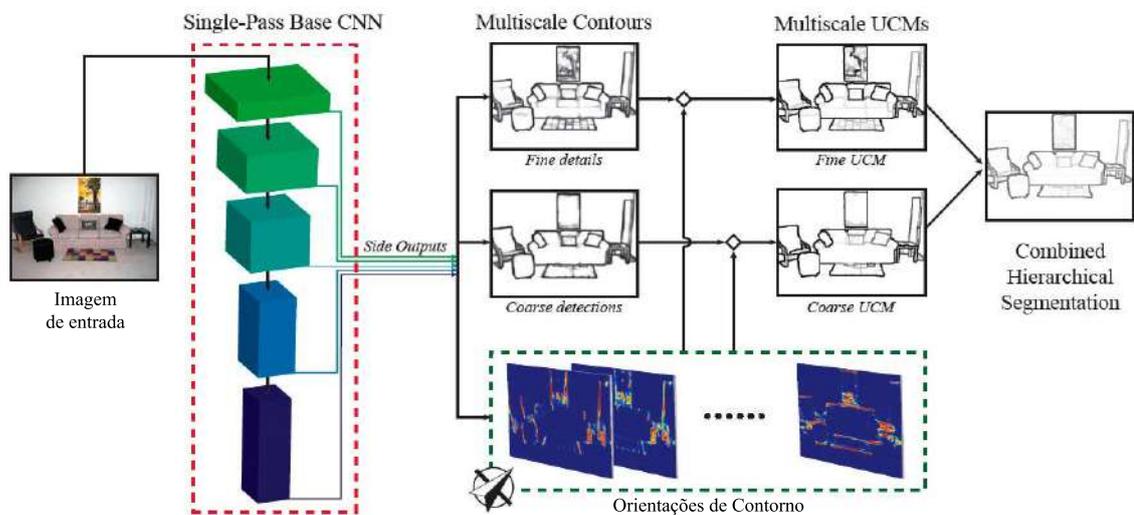
Na fase de treinamento, cada camada de saída é então submetida a uma comparação com o padrão ouro da imagem de entrada, ou seja, um classificador com função de perda, definida pelos autores, é criado. Por fim, na última camada, uma função ponderada, também proposta pelos autores, compõe a imagem final.

2.6 MANINIS ET AL. (2017)

Uma das principais características das abordagens na detecção de contorno é a obtenção da informação de orientação do contorno. Isto é, dado um sinal de contorno (resposta a um

filtro, por exemplo) em certa direção, o objetivo é determinar sua “força” ou probabilidade de existência. No trabalho de Maninis *et al* (MANINIS et al., 2017), utiliza-se uma CNN, denominada *Convolutional Oriented Boundaries (COB)*, para obter essa informação em um esquema multiescala. A Figura 17 apresenta a arquitetura proposta dividida, de forma resumida, em duas grandes partes. Na primeira parte, o objetivo é mais geral, ou seja, detectar contornos que sejam finos ou espessos em diferentes escalas. Na segunda parte, o resultado de cada escala é processado para cada orientação e, então, resumido em uma imagem de contorno.

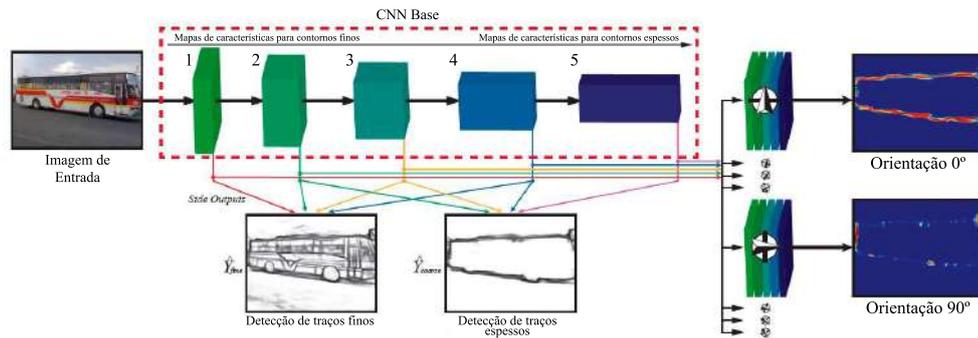
Figura 17 – Arquitetura proposta por Maninis *et al*.



Fonte: MANINIS et al. (2017)

A primeira parte, chamada de base CNN, consiste em uma rede composta em sua maior parte por pares de camadas de convolução e função de ativação ReLu criadas a partir de uma simplificação da ResNet com seus pesos, mas sem as camadas totalmente conectadas e a normalização de *batch*, reunidas em 5 estágios distintos. Cada um dos estágios é responsável pela análise da imagem de entrada em diferentes escalas, dispondo em sua saída uma camada de *max pooling* reduzindo a dimensionalidade dos mapas de características pela metade no qual são processados na segunda parte da rede. Essa primeira parte da arquitetura é simples, porém bastante intuitiva em se tratando de redes neurais profundas aplicadas na detecção de contorno. Isto é, nos estágios mais “rasos”, contornos finos e locais são detectados ao mesmo tempo em que os estágios mais “profundos” detectam contornos espessos e globais, comportamento particular de uma DNN. Na Figura 18 é possível notar que os estágios 1, 2, 3 e 4 são responsáveis pelos traços finos e o último estágio, ligado com os estágios 2, 3 e 4, formam os traços mais espessos do contorno. A saída de cada estágio é então supervisionada e comparada com o padrão ouro.

Em seguida, na segunda parte da arquitetura ou extensão da base CNN, o objetivo

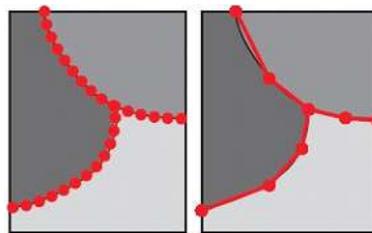
Figura 18 – Arquitetura proposta por Maninis *et al.*

Fonte: MANINIS et al. (2017)

é classificar os pixels dos contornos detectados em relação a uma orientação. Para isso, K diferentes sub-redes com M camadas de convolução, processadas de forma paralela, recebem como entrada as 5 saídas da base CNN. Para realizar essa classificação, os autores tiveram que construir um novo padrão ouro no qual cada pixel presente na imagem padrão ouro possui suas K orientações.

Na Figura 19, é possível notar que o contorno da imagem padrão ouro foi aproximado a um polígono (direita da imagem). Assim, cada pixel do contorno recebe sua orientação a partir do seu segmento de polígono. Cada sub-rede tem como saída uma resposta de ativação para cada pixel do contorno, caso duas orientações (duas redes) resultem em ativações “fortes”, a orientação definida é a média. Caso não haja ativação em nenhuma das sub-redes, então a orientação do pixel é atribuída a um valor aleatório entre 0 e π . A Figura 20 apresenta dois exemplos de resultados para as orientações de ângulos 0° e 90° e a imagem final resultante das K diferentes sub-redes.

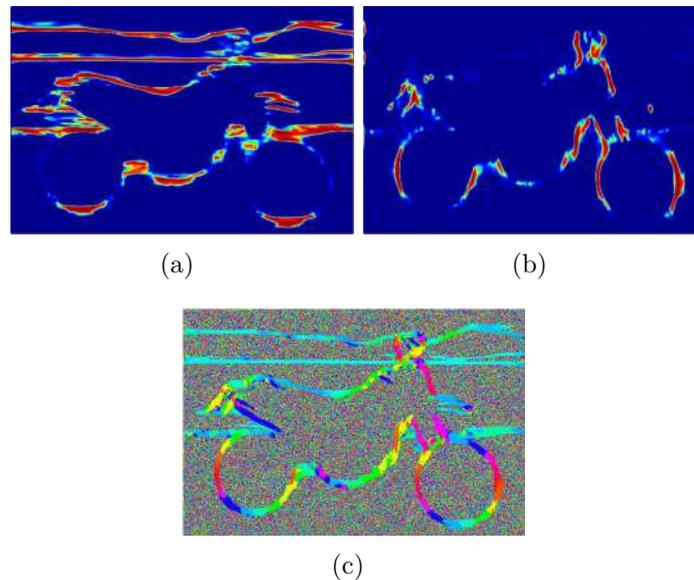
Figura 19 – Exemplo de simplificação dos vários pontos do contorno.



Fonte: MANINIS et al. (2017)

Pode-se dizer que a COB possui dois treinamentos: o primeiro na base CNN, que consiste em um ajuste fino da ResNet, e o segundo das sub-redes para cada orientação com novas imagens padrão ouro com os pixels rotulados em orientações. Os autores propõem 8 orientações ($K = 8$) e 5 camadas de convolução para cada sub-rede ($M = 5$), treinadas com o conjunto *trainval* do PASCAL VOC 2010 (EVERINGHAM et al., 2010).

Figura 20 – Exemplo bordas detectadas para (a) 0°, (b) 90° e (c) todas orientações aprendidas da imagem.



Fonte: MANINIS et al. (2017)

2.7 KELM ET AL. (2019)

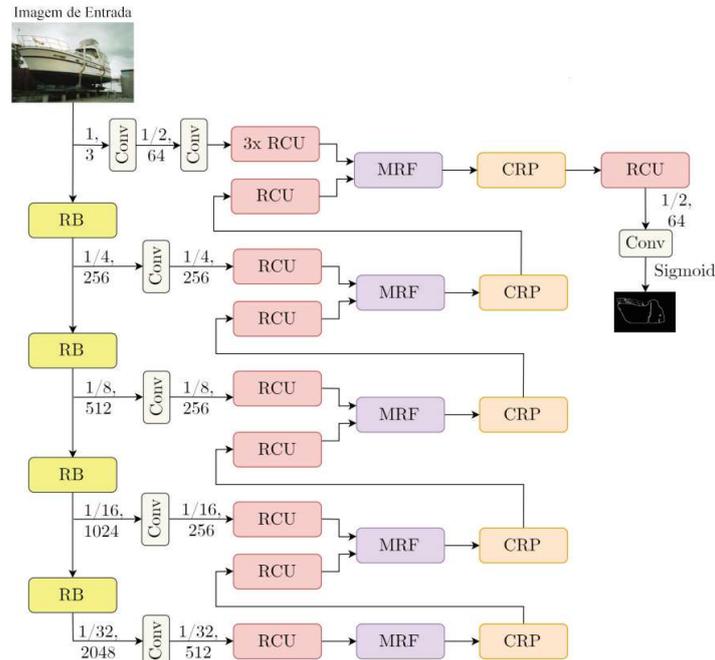
A *Refine Contour Net (RCN)* proposta por Kelm *et al.* (KELM; RAO; ZÖLZER, 2019) é uma CNN no qual os autores focam nas características altas, médias e baixas, agrupadas nesta ordem específica, para detecção de bordas. A rede foi treinada no banco PASCAL VOC 2012 e em seguida no BSDS500 com a técnica de ajuste fino.

Utilizar apenas características de baixo nível faz com que a rede detecte uma grande quantidade de bordas para qualquer tipo de imagem. Então, visando detectar apenas as bordas com significado, um contexto semântico se faz necessário. Para isso, uma rede tem suas camadas mais profundas conectadas com as camadas mais rasas da rede, tornando o processo baseado em um refinamento.

O trabalho é baseado na rede RefineNet de Lin *et al.* (LIN et al., 2017) devido à necessidade de uma imagem de *input* de grande resolução para processamento do seu contexto. Na Figura 21, que apresenta a arquitetura adotada, é possível identificar a ideia do processo de refinamento das camadas mais profundas para as camadas mais rasas da rede. Os blocos *Residual Blocks (RB)*, parte da RefineNet, e os novos *Residual Convolution Unit (RCU)*, *Multi-Resolution Fusion (MRF)* e *Chained Residual Pooling (CRP)* são apresentados na Figura 22.

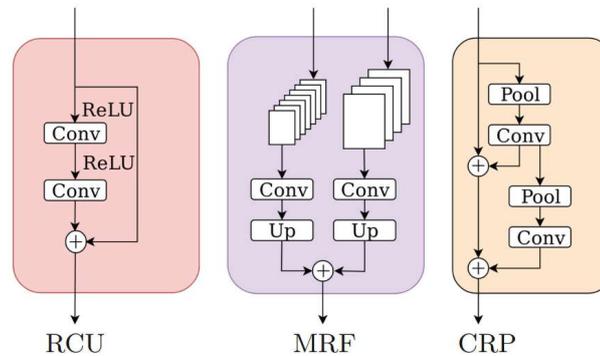
O bloco RCU apenas aumenta o número de convoluções propostas pelo bloco RB seguido do MRF para adaptar as dimensões da imagem de entrada do espaço das características de alto nível com as de baixo nível por meio do *upsample*. Por fim, o bloco CRP obtém mais informação de contexto dos mapas de características do que camadas de *max pooling*.

Figura 21 – Arquitetura proposta por Kelm *et al* que busca tirar proveito do processo de refinamento das camadas mais profundas de uma **CNN**.



Fonte: **KELM; RAO; ZÖLZER (2019)**

Figura 22 – Novos blocos na RefineNet propostos pelos autores.



Fonte: **KELM; RAO; ZÖLZER (2019)**

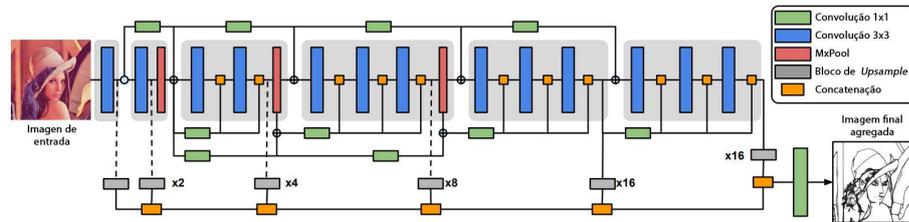
2.8 SORIA ET AL. (2020)

Os autores da **Dense Extreme Inception Network for Edge Detection (DexiNed)** (**SORIA; RIBA; SAPPÀ, 2020**) fazem uso de uma **DNN** sem treinamento ou ajuste fino em bancos de dados clássicos da área de detecção de borda. Eles afirmam que algumas imagens padrão ouro não possuem bordas devidamente anotadas, prejudicando o processo de treino e, conseqüentemente, sua detecção. Assim, tais imagens não foram utilizados no treinamento, no entanto um novo conjunto de imagens é estabelecido.

A Figura **23** apresenta a arquitetura composta por 6 blocos principais (com fundo na cor cinza), conectados por blocos de convolução 1×1 . Cada bloco é formado por sub-blocos

com camadas de convolução 3×3 interconectados com a saída das camadas anteriores. Por fim, uma concatenação (exibido em laranja) após a última camada de convolução de cada bloco, precedendo uma operação de *upsample*, reúne todas as informações para dar a uma única imagem.

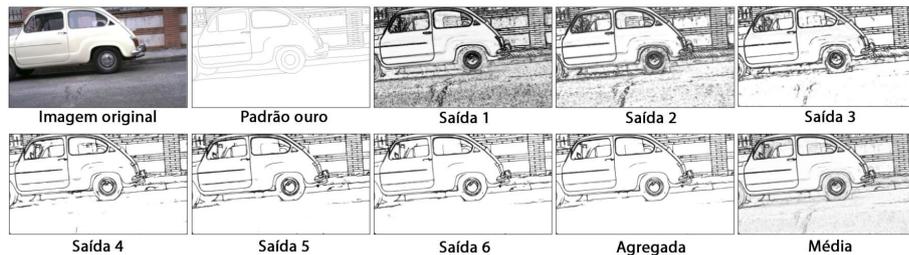
Figura 23 – Arquitetura da DexiNed.



Fonte: SORIA; RIBA; SAPPA (2020)

As 6 saídas apresentam diferentes resultados, como mostra a Figura 24. Em seguida, uma média de todos os resultados é feita, gerando a imagem final submetida às avaliações.

Figura 24 – Exemplo das imagens nas 6 saídas após a operação de *upsample*, seguidas da imagem agregada e da imagem da média.



Fonte: SORIA; RIBA; SAPPA (2020)

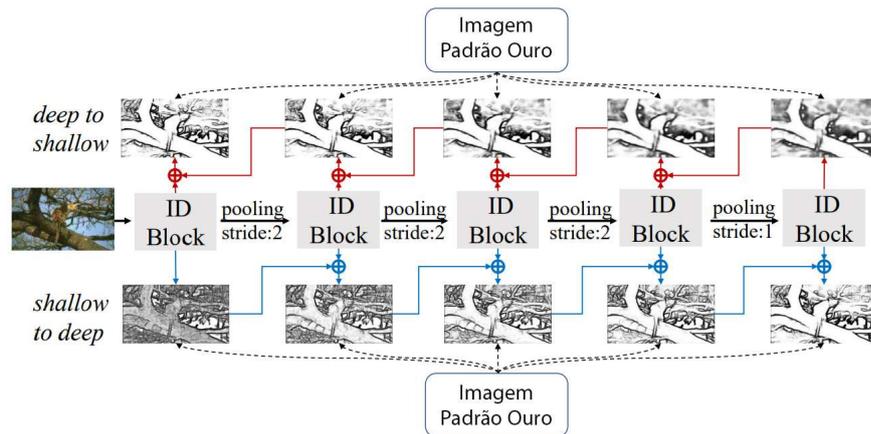
Outra contribuição do trabalho é um banco de dados, nomeado *Barcelona Images for Perceptual Edge Detection* (BIPED), formado por 250 imagens de tamanho 1280×720 pixels, no qual 50 foram selecionadas de forma aleatória para compor o conjunto de teste e as outras 200 para validação e treino, simultaneamente com a técnica de *data augmentation*, totalizando 288 imagens.

2.9 HE ET AL. (2020)

Seguindo a ideia da HED, a *Bi-Directional Cascade Network for Perceptual Edge Detection* (BDCN) (HE et al., 2020) também é baseada na VGGNet com várias camadas de saídas, porém seus autores sugerem que cada camada de saída receba um treinamento específico. Outra variedade é a criação de um módulo, denominado *Scale Enhancement Module* (SEM), para melhorar a detecção de borda em cada escala, que é inspirado na *dilated convolution* (CHEN et al., 2014).

A Figura 25 apresenta, de forma geral, a arquitetura da rede proposta. Para simplificar a imagem, os *ID Block* denotam os blocos básicos da rede, ou seja, as camadas de convolução, juntamente com o SEM. A figura também mostra que cada camada de saída recebe a imagem padrão ouro, durante o treinamento, em diferentes escalas representadas pela ideia de *deep to shallow* e *shallow to deep*, comumente tratadas em CNN.

Figura 25 – Arquitetura da BDCN



Fonte: HE et al. (2020)

A BDCN foi treinada com as imagens do banco de dados BSDS500. Para os parâmetros do SEM, os autores realizaram vários testes e chegaram aos valores 3 e 4 para a quantidade e taxa de dilatações, respectivamente.

As técnicas apresentadas possuem diferentes formas de realizar a detecção de contorno de uma imagem, sendo o gPb utilizada como inspiração no criação do método e todas as outras comparadas com os resultados obtidos apresentados no Capítulo 5.

3 HIPÓTESE

Neste capítulo é apresentado o conceito do **SDC** de forma detalhada e a hipótese de utilizá-lo como nova proposta para solução de problemas de segmentação, como detecção de contornos em cenas naturais e segmentação em imagens de plantas baixas e mapas.

3.1 *SPEED DRAWING CHALLENGE*

A habilidade de desenhar é, sem dúvidas, uma das formas de expressões mais fascinantes no campo artístico. Embora o ato de desenhar esteja presente em nossas vidas desde a infância, desenhar profissionalmente requer um alto grau de domínio sobre o processo de ilustração, criatividade e observação por parte dos artistas. Assim, visando avaliar ou demonstrar suas próprias habilidades, é bastante comum encontrar, nas redes sociais, competições onde o objetivo é produzir ilustrações, porém em tempos pré-definidos e submetê-los a avaliações de seus seguidores. Frequentemente, os desafios propõem o uso de três tempos: o de 10 minutos executado primeiro para servir como treino; o de 1 minuto o segundo; e no final o de 10 segundos. De forma geral, as regras para construção destes desenhos são:

- Um período de tempo suficiente para que a ilustração possua seus detalhes minuciosos, podendo ser visto como um desenho original, a ser referência para os outros dois;
- Outro período intermediário que permita que a ilustração apresente algum nível de detalhamento em comparação com o desenho original;
- E, por último, poucos segundos que irão realmente desafiar o artista em reproduzir com poucos traços, mas ainda assim com alguma clareza, o desenho original. A habilidade do artista destaca-se nesta ocasião.

A Figura **26** descreve bem a ideia do desafio. Nela, é possível verificar o impacto na qualidade dos detalhes ao passo que o tempo disponível para construção do desenho limita-se em segundos. Em 20 minutos, pode-se dizer que o desenho encontra-se completo, ou seja, com vários detalhes representados por traços de vários tamanhos, finos ou não, e bem distribuídos. O desenho de dois segundos é exatamente a representação mais simples possível do desenho original, entretanto contendo ainda informações que o tornem minimamente semelhante ao de 20 minutos (por exemplo, percebe-se que é um olho humano). O desenho de 2 minutos pode ser considerado como um meio termo entre os outros dois.

Ainda sobre a Figura **26**, a ilustração referente ao tempo de dois segundos, possui poucos traços, porém longos, sem muita precisão e composto apenas por formas geométricas simples, ou seja, círculos e retas, algo bastante característico em desenhos de

esboços (*Sketch*). Ainda assim, apesar de ser um desenho bastante elementar, observa-se uma certa identidade ao desenho original, justamente o que se espera de um desenho esboço: pouquíssimos detalhes, mas preservando a intenção do que se deseja realmente representar.

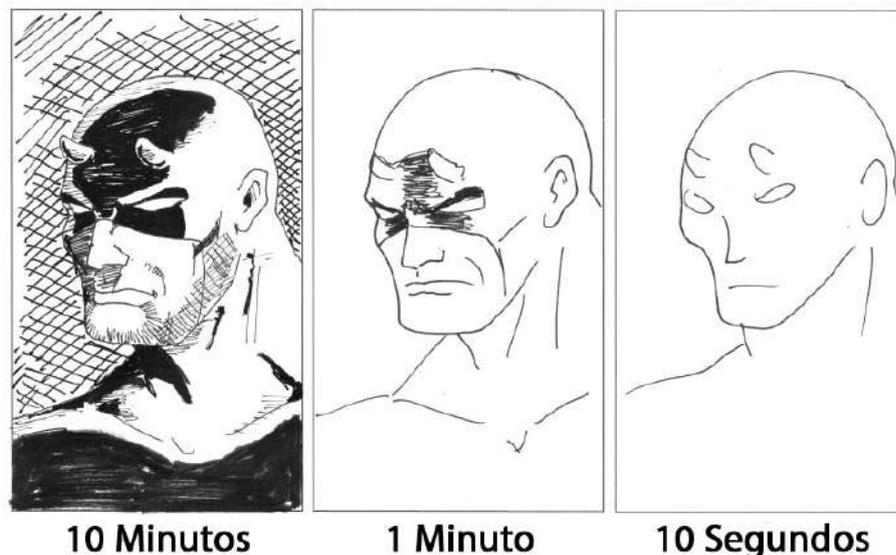
Figura 26 – SDC de 20 minutos, 2 minutos e 2 segundos.



Fonte: INSTAGRAM (2018)

Outro exemplo do SDC é o da Figura 27. Apesar da diferença de tempos, os desenhos apresentam-se com as mesmas características descritas no SDC da Figura 26, isto é, o tempo está totalmente relacionado aos detalhes oferecidos, tornando-se um fator determinante no detalhamento do desenho. Percebe-se também que, apesar do pouco tempo, a imagem de 10 segundos tem referências claras à imagem de 10 minutos, como o contorno do rosto e dos olhos.

Figura 27 – Exemplo de SDC de 10 minutos, 1 minuto e 10 segundos.



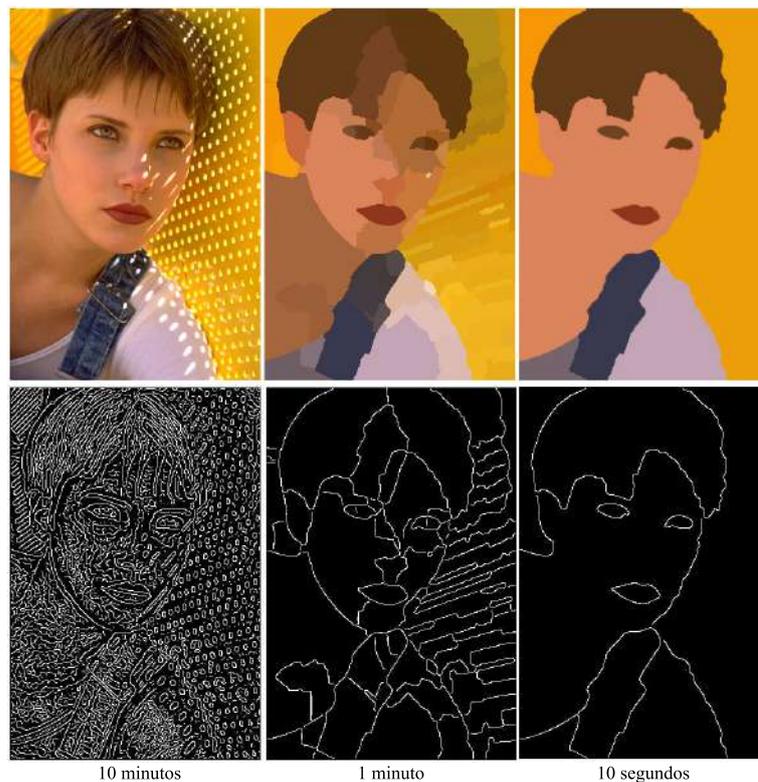
Fonte: Carlos Alexandre (2022)

3.2 SDC APLICADO PARA PROBLEMAS DE DETECÇÃO DE CONTORNO

A detecção de contorno em uma imagem pode ser entendida como resultado da seleção de bordas que envolvem os objetos. No **SDC**, a ilustração de menor tempo corresponde a esta ideia: com pouco tempo disponível, o artista se preocupa em ilustrar apenas os traços com uma maior representatividade, ocasionando em um desenho formado apenas pelo contorno ou algo como um esboço do desenho de 10 minutos.

Partindo do princípio de que uma imagem de 10 minutos representa uma imagem de borda produzida por um algoritmo de detecção de bordas, como Canny, por exemplo, a imagem de 10 segundos, ou seja, de contorno, deve conter apenas os traços referentes ao objeto principal da cena. Para o computador, 10 minutos ou até mesmo 10 segundos, pode representar uma grande quantidade de tempo disponível para realizar uma determinada tarefa. Assim, visando representar a restrição de tempo no **SDC**, a modelagem do método proposto faz uso do conceito de *superpixel*. Isto é, à medida que o tempo se torna menor, o artista foca em detalhes que realmente representem a imagem, ou seja, detalhes que não agregam informação ao objeto da cena são removidos, como mostra a Figura 28. Então, o tempo do **SDC** foi modelado através da perda de detalhes da imagem por meio de um algoritmo de *superpixel*, onde a quantidade de *superpixels* (agrupamento dos pixels) podem produzir tanto o desenho de 1 minuto, quanto o de 10 segundos.

Figura 28 – Exemplo do **SDC** modelado para o problema de detecção de contorno através do conceito de *superpixel*.



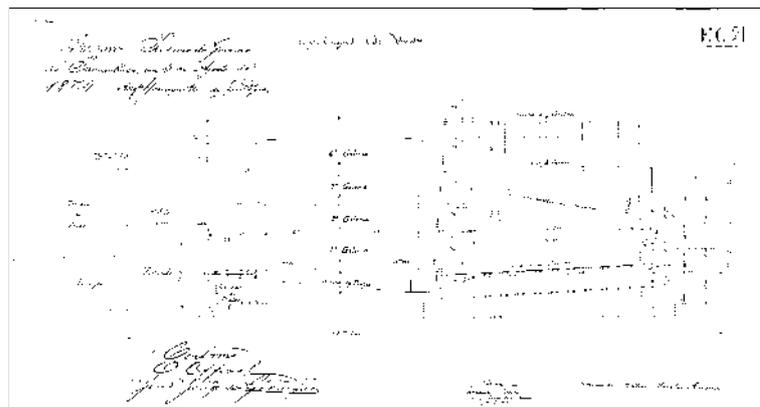
Fonte: Marcos Canêjo (2022)

3.3 SDC APLICADO PARA PROBLEMAS DE SEGMENTAÇÃO EM MAPAS E PLANTAS BAIXAS

Imagens de mapas e, principalmente, plantas baixas, possuem uma grande quantidade de informações gráficas que podem ser entendidas como a essência ou parte principal da imagem, enquanto que a parte textual, mais presente em imagens de mapas, como os detalhes. Então, pode-se dizer que a parte textual presente nestas imagens representam os detalhes nos desenhos do SDC que são comprometidos de acordo com a diminuição do tempo disponível para a criação das ilustrações.

Na modelagem elaborada, a imagem de tempo máximo representa a imagem completa sem qualquer segmentação, ou seja, uma imagem binarizada pré-processada, contendo texto, linhas ou simplesmente o “desenho em sua forma original”. A Figura 29 representa este conceito.

Figura 29 – Imagem que pode ser interpretada como o resultado do SDC de tempo máximo.



Fonte: Marcos Canêjo (2022)

Como visto, a figura de 1 ou 2 minutos do SDC deve ser entendida como o meio termo entre o tempo máximo e o tempo mínimo disponível. Avançando com esta observação, a Figura 29 deve conservar mais informações relevantes do desenho (o que na modelagem significa as informações gráficas da imagem) do que os detalhes (texto). A Figura 30 exibe um possível resultado para esta concepção.

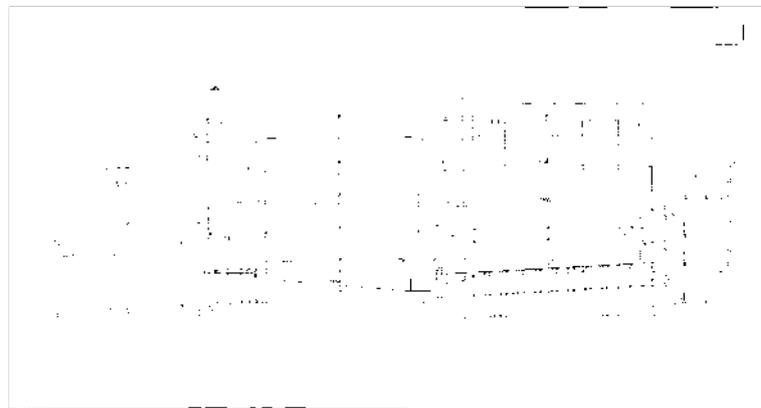
Na Figura 30, devido ao pouco tempo disponível, o “artista” se concentrou em ilustrar a parte gráfica do desenho original e apenas representou os detalhes mais próximos devido à conectividade entre eles. Este critério de conectividade ou proximidade é totalmente subjetivo e varia de artista para artista. Por exemplo, na Figura 26, para o desenho intermediário de 2 minutos, apenas os cílios superiores foram representados em contrapartida aos inferiores que não tiveram qualquer presença.

Outro exemplo a respeito do critério de exclusão dos detalhes nas imagens de tempo intermediário do desafio, encontra-se na Figura 27, na qual, na imagem de 1 minuto, todo *background* que não pertence ao desenho principal foi excluído.

Por fim, a ilustração de poucos segundos é aquela que representa um esboço do desenho original. Na imagem original de exemplo desta seção (Figura 29), que é uma imagem de planta baixa, o seu esboço seria algo contendo apenas linhas longas com representações bastante inacabadas e sem precisão, sem conter qualquer texto, pois, neste exemplo, são considerados os detalhes da imagem. O resultado de todo esse conceito é apresentado na Figura 31.

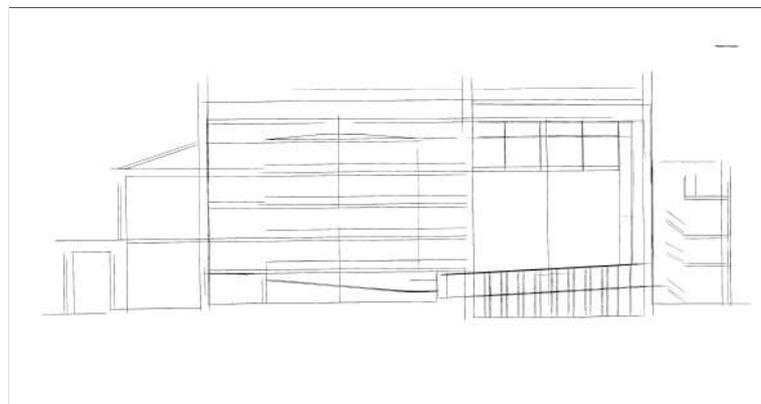
Assim, temos as três representações do desafio definidas e exemplificadas. O resultado apresentado na Figura 30 logo pode ser percebido como uma imagem submetida a uma segmentação. A maioria dos elementos textuais foram excluídos, no entanto, o cenário de sobreposição entre textos e elementos gráficos ainda estão presentes. Por fim, a Figura 31 representa a ideia dos 10 segundos.

Figura 30 – Imagem representando o resultado de um desafio de 1 minuto.



Fonte: Marcos Canêjo (2022)

Figura 31 – Imagem esboço, representando o desenho feito em 10 segundos no SDC.



Fonte: Marcos Canêjo (2022)

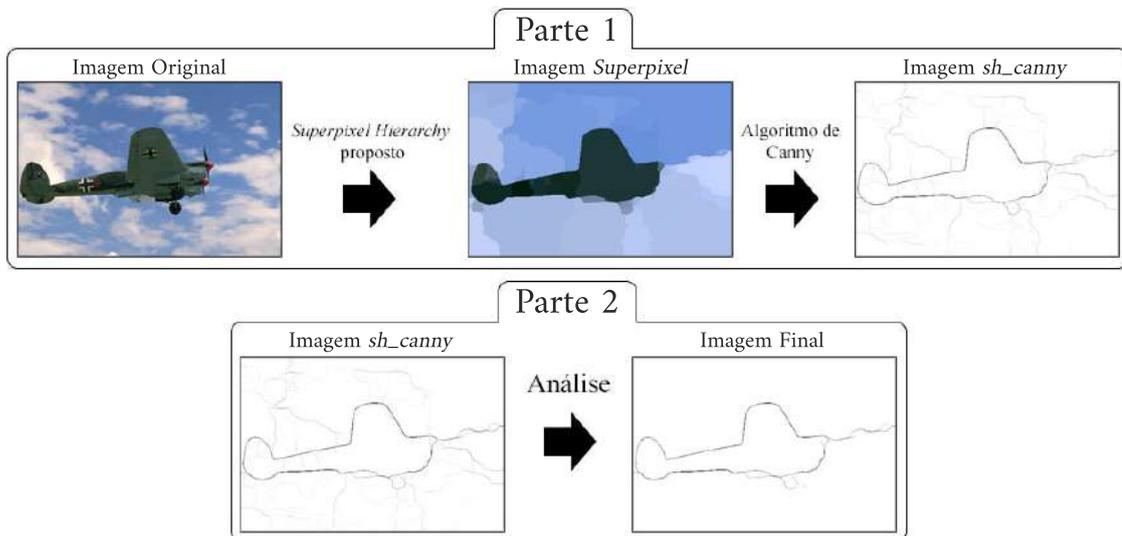
A modelagem do *Speed Drawing Challenge* corresponde a criar três representações gráficas diferentes, com base em observações nos efeitos causados pela restrição de tempo disponível para os artistas. Porém, a nossa abordagem está focada, principalmente, na

simulação do resultado de menor tempo, gerando uma imagem com remoção dos elementos textuais conectados a partes gráficas.

4 MÉTODO PROPOSTO PARA SEGMENTAÇÃO DE IMAGENS DE CENAS NATURAIS

O objetivo do método proposto é gerar uma imagem de contorno através da inspiração do [SDC](#). O método foi dividido em duas grandes partes: na primeira, o objetivo é obter um valor de *superpixels* que represente a imagem original a partir da ideia do desenho de 10 segundos. Ou seja, através de um algoritmo de *superpixels*, com modificações propostas, um valor de *superpixels* é definido para cada imagem de entrada. Com o número de *superpixels* definido, uma imagem de borda é gerada; na segunda, uma análise é feita para remover traços que não fazem parte do conceito de contorno aqui aplicado. A Figura [32](#) apresenta um *overview* do método com suas principais saídas para cada passo.

Figura 32 – Apresentação do método proposto dividido em duas partes.



Fonte: Marcos Canêjo (2022)

4.1 PARTE 1 - AGRUPAMENTO POR SUPERPIXELS

Transformar a imagem original na imagem de *superpixels* envolve a ideia de diminuir a quantidade de detalhes da imagem, isto é, conforme o artista é submetido a realizar o desenho em menos tempo, a tendência é que a quantidade de detalhes diminua. Para isso, então, o algoritmo [Superpixel Hierarchy \(SH\)](#) ([WEI et al., 2018](#)) é empregado.

O *Superpixel Hierarchy*, como o próprio nome diz, faz parte de uma categoria de trabalhos que abordam algoritmos que realizam um agrupamento de pixels para uma representação menos detalhada da imagem: os chamados *superpixels*. Por exemplo, na Figura [33](#), a imagem teve seus detalhes resumidos a partir de um determinado número de *superpixels*. O agrupamento dos pixels, nesta situação, pode facilitar uma possível detecção de partes da estrutura, dado que o telhado e os outros objetos foram agrupados

de formas distintas para 16 *superpixels*. Os algoritmos de *superpixels* também possuem aplicações em segmentação de imagens médicas (WU et al., 2019) além da detecção de contorno (LEVINSHTEIN; SMINCHISESCU; DICKINSON, 2010).

De acordo com Wei *et al.*, quatro propriedades são essenciais em um algoritmo de *superpixel*:

- Precisão na segmentação: cada *superpixel* deve compor apenas um objeto. Isto é, os limites dos objetos devem ser respeitados a exemplo da Figura 33 onde, para 16 *superpixels*, os pixels que representam o céu, as construções e o rio foram simplificados de acordo com a imagem original;
- Eficiência computacional: a imagem resultante do agrupamento é vastamente utilizada como pré-processamento em outras áreas, deste modo, o algoritmo deve ter um comportamento linear independentemente do tamanho da imagem;
- Segmentação hierárquica: esta propriedade está ligada à modelagem do sistema visual humano e sua percepção hierárquica das informações exibidas em uma imagem. O ser humano facilmente consegue distinguir objetos em uma cena e os algoritmos devem respeitar essa percepção. Não faria sentido, por exemplo, reunir os pixels da construção com os do seu reflexo no rio;
- Preservação topológica: os *superpixels* devem respeitar a uma topologia simples, de modo que a relação entre as vizinhanças seja mantida. Esta propriedade se evidencia no uso de grafos descritos posteriormente.

Então, Wei *et al.* propuseram o SH que contém todas propriedades citadas e analisadas de forma independentes por *benchmarks* específicos que não fazem parte do escopo deste documento.

Um dos meios de processar o conceito de *superpixel* em uma imagem é modelando sua estrutura a partir de um grafo. Isto é, seja $G = (V, E)$ um grafo não direcionado com vértices $v \in V$ e arestas $e \in E$, cada pixel é associado a um vértice e localmente conectado a 4 vizinhos com suas arestas $e_{ij} = (v_i, v_j)$ ponderadas por um valor normalmente não negativo representando a dissimilaridade entre os dois vértices. Na segmentação por *superpixels*, k denota o número de *superpixels* a serem formados, ou seja, um segmento S do grafo G de uma partição de V com k componentes.

O SH faz uso do algoritmo de Borůvka (WEST et al., 1996) devido a sua capacidade de paralelismo, informação multiescala (a cada iteração os valores das arestas são atualizados formando um novo agrupamento) e a formação hierárquica durante o agrupamento para qualquer valor de *superpixels*. Porém, pode-se dizer que o SH é uma versão do algoritmo de Borůvka modificado visando contemplar as quatro propriedades propostas pelos autores.

A Figura 34 ilustra os principais passos do SH. Primeiramente, cada vértice representa um *cluster* de pixels conectado a outro vértice por uma aresta ponderada, como mostra

Figura 33 – Exemplo de imagem segmentada com 16, 256, 4096 e 65536 *superpixels*.



Fonte: WEI et al. (2018)

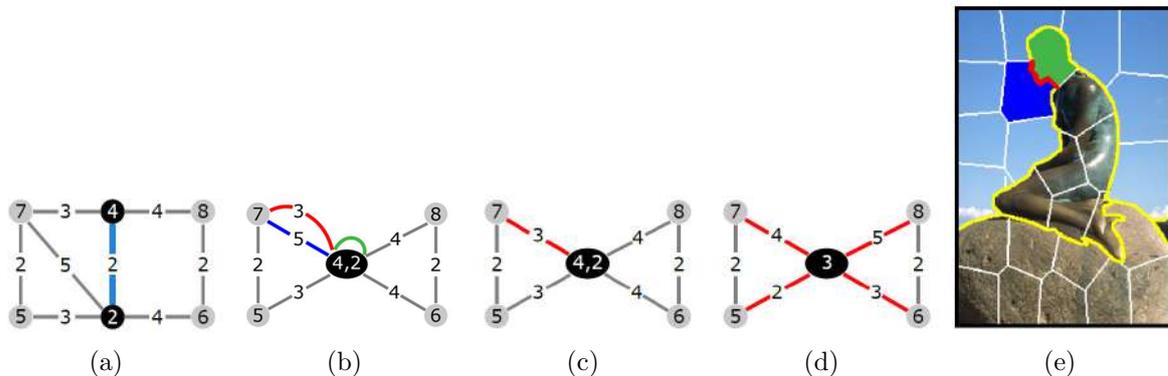
a Figura 35(a). Os vértices 4 e 2 são então combinados resultando em um super vértice (Figura 34(b)), que posteriormente irá assumir o valor médio 3. Em seguida, a aresta em *loop* de cor verde é removida e as arestas paralelas de cor azul e vermelha são substituídas pela aresta de menor valor, ou seja, 3 (Figura 35(c)). Todo esse processo consiste na execução comum do algoritmo de Borůvka. Após esse agrupamento, o SH propõe uma atualização de todas as arestas conectadas ao vértice recém-criado, baseada nas cores e informações de bordas geradas pelo algoritmo SFE (DOLLÁR; ZITNICK, 2013). Os novos valores das arestas são dados por

$$D(C_1, C_2) = d_c \times d_e, \quad (4.1)$$

sendo C_1 e C_2 os vértices em análise e d_c a distância qui-quadrado (X^2) do histograma de cores e d_e o valor médio de borda entre as regiões demonstrada pela linha vermelha na Figura 35(e). Todo esse processo é repetido até que o número de *superpixels* (vértices) previamente definido seja atingido.

Algoritmos de *superpixels* servem muitas vezes como um pré-processamento para algoritmos que tratam do problema de detecção de contorno. No *benchmark* apresentado pelos autores para este tipo de problema, é ressaltado que a combinação das informações de cor e principalmente de borda tornaram seus resultados mais relevantes, porém para um número fixo de 600 *superpixels* para todas as imagens testadas do BSDS500.

Figura 34 – (a-c) Execução do algoritmo de West *et al.*, (d) ajuste de todas arestas proposta pela nova abordagem (e) com base nas cores e bordas da imagem.



Fonte: WEI *et al.* (2018)

4.1.1 Definindo o Número de *Superpixel* de Forma Automática

Duas modificações são feitas no algoritmo original de *Superpixel Hierarchy*. A primeira modificação diz respeito à quantidade de *superpixels* que o algoritmo irá criar. No artigo original, este número é pré-definido, porém, no método proposto o objetivo é obter o menor número que melhor preserva as bordas dos principais objetos da cena. Para isso, o SH é aplicado para um intervalo de *superpixels* e submetido a duas métricas.

Inicialmente, para as imagens de validação e treino do banco de dados BSDS500, o número de *superpixel* foi definido em 100, porém várias bordas foram geradas. Então, em seguida, o número 50 foi escolhido visando observar os resultados. Por fim, os números 10, 15, 20, 25, 35 e 45 foram testados e observou-se um melhor desempenho para o número 30 após uma análise visual.

O número mínimo do intervalo deve representar o cenário mais simples encontrando em uma imagem, ou seja, uma imagem de *background* e *foreground* bem definidos. Portanto, o número mínimo foi definido em 2. Para o número máximo, o valor definido foi 30, pois foi entendido após uma vasta análise nas imagens de validação e treino do BSDS500 que valores altos retornava imagens de borda com vários traços não relacionados aos principais objetos da cena, fugindo da ideia principal da imagem de 10 segundos. Assim, para cada valor inteiro compreendido entre 2 e 30, o SH é submetido para cada imagem de entrada e aplicada as duas métricas.

A primeira métrica é Davies-Bouldin (DB) (DAVIES; BOULDIN, 1979), que avalia a qualidade do agrupamento com base nas cores. O valor de DB é definido

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} D_{i,j}, \quad (4.2)$$

sendo $D_{i,j}$ a relação de distância entre os *clusters* i e j definida

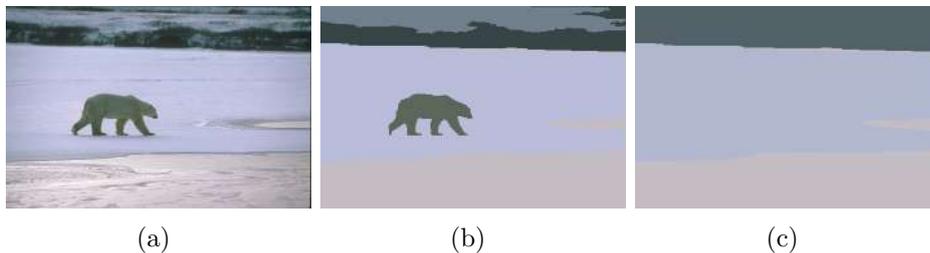
$$D_{i,j} = \frac{(\bar{d}_i + \bar{d}_j)}{d_{i,j}}, \quad (4.3)$$

considerando \bar{d}_i e \bar{d}_j o valor médio da distância entre seus pontos e centroide, $d_{i,j}$ a **Distância Euclidiana (DE)** entre os centroides de cada *cluster* i e j .

O valor máximo de $D_{i,j}$ representa o pior caso para o *cluster* i , sendo o menor valor o resultado da **DB**. Então, para cada canal vermelho, verde e azul da imagem de *superpixel*, seus valores são normalizados e submetidos ao **DB** utilizando o algoritmo *k-Means* com o número de classes igual ao número de *superpixel* em teste. A escolha do *k-Means* se deu devido ao seu baixo custo computacional.

A **DB** avalia apenas a qualidade do agrupamento realizado pelo *k-Means*, deixando de lado as fronteiras entre os objetos na cena. Por exemplo, na Figura 35, dois resultados de 5 e 3 *superpixels* obtidos pelo **SH** são apresentados. Na Figura 36(c), retornado pela **DB** como o melhor, o urso é totalmente suprimido. Assim, visando evitar este problema, as imagens de borda da imagem original e da imagem de *superpixels* são comparadas por meio da **DE**.

Figura 35 – Exemplo da (a) imagem original, suas versões segmentadas pelo **SH** com (b) 5 e (c) 3 (com o urso totalmente suprimido).



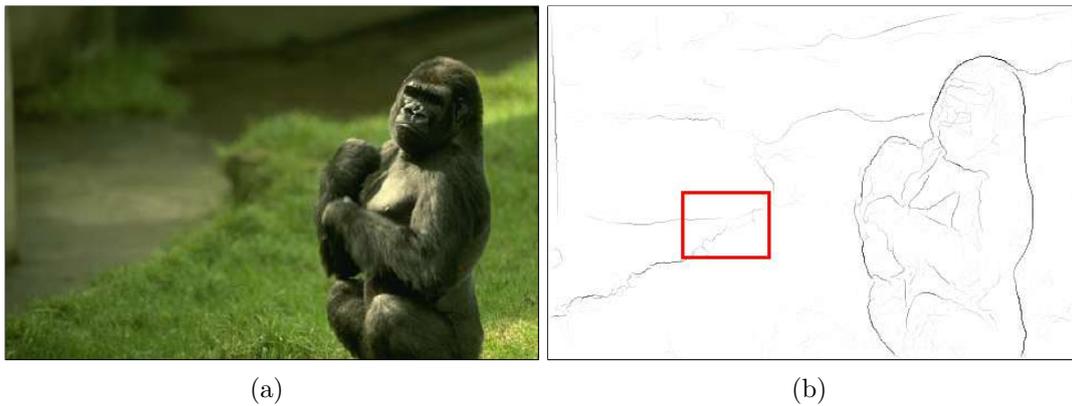
Fonte: Marcos Canêjo (2022)

Portanto, o objeto principal da cena da imagem original é mantido na imagem de *superpixels* criada, preservando suas fronteiras. O número de *superpixels* que obtiver o menor valor a partir da soma de **DB** e **DE** é escolhido.

4.1.2 Orientação de Borda

Além do número de *superpixels*, as informações de borda da imagem original também fazem parte dos parâmetros de entrada do **SH**, sendo o detector de borda **SFE** utilizado na versão original do artigo. Porém, identificar bordas em regiões onde o *background* e *foreground* da imagem são similares é uma tarefa difícil, mesmo para o SFE. Por exemplo, a Figura 36 apresenta uma detecção fraca na fronteira entre estes dois planos da imagem, destacados em vermelho na Figura 37(b). Assim, as 8 imagens de borda para diferentes orientações geradas pelo **gPb** são usadas.

Figura 36 – (a) Imagem original e (b) suas bordas detectadas pelo SFE.



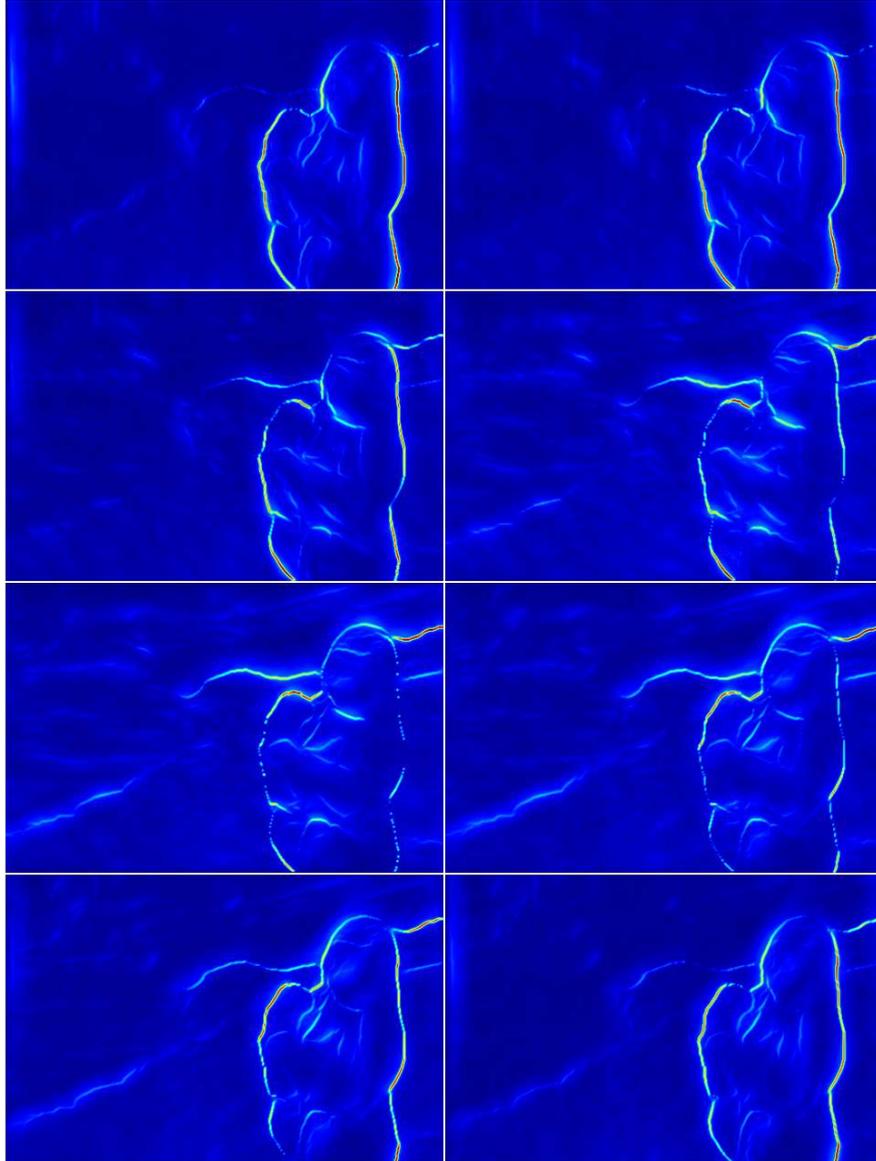
Fonte: Marcos Canêjo (2022)

A Figura 37, exibida em um esquema de cores diferentes para melhor compreensão com os tons mais claros representando uma detecção forte, é possível identificar que as diferentes orientações fornecem informações que podem ser úteis. Por exemplo, nas duas imagens da última e penúltima linha, as bordas mal detectadas pelo SFE estão presentes, mesmo que ainda de forma fraca.

Como consequência do uso de 8 imagens de borda do gPb, o SH produz 8 imagens de *superpixel* para cada imagem de entrada, sendo o próximo passo a detecção de suas bordas. Para isso, o algoritmo de Canny é aplicado, uma vez que as imagens de *superpixel* possuem poucos detalhes, tornando fácil a detecção de suas bordas e também seu rápido processamento. Assim, no final do processo, 8 imagens de borda de Canny, como mostra a Figura 38, são geradas para cada uma das imagens de *superpixel* do SH proposto.

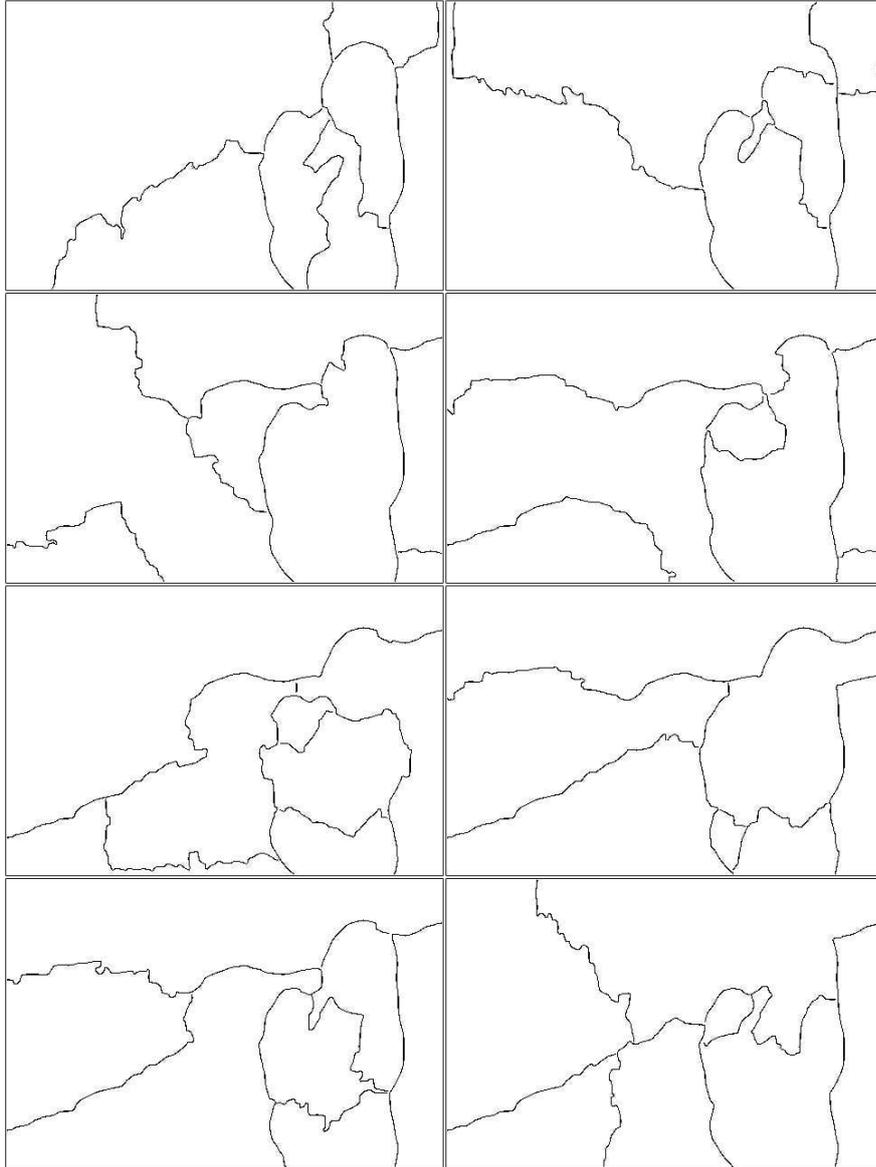
Em seguida, uma única imagem de borda é criada, denominada *sh_canny*, a partir da soma de todas as imagens de Canny geradas anteriormente. Esta soma pode ser entendida como uma definição de relevância das bordas. Isto é, as imagens binarizadas de Canny, ou seja, com a borda representada pelo valor 1 e o *background* por 0, quando somadas, tornam as bordas com maior frequência identificadas por valores altos. Assim, a relevância de uma borda é dada pelo seu alto valor na imagem *sh_canny*, exemplificada na Figura 39.

Figura 37 – Detecção de borda para 8 diferentes orientações geradas pelo gPb.



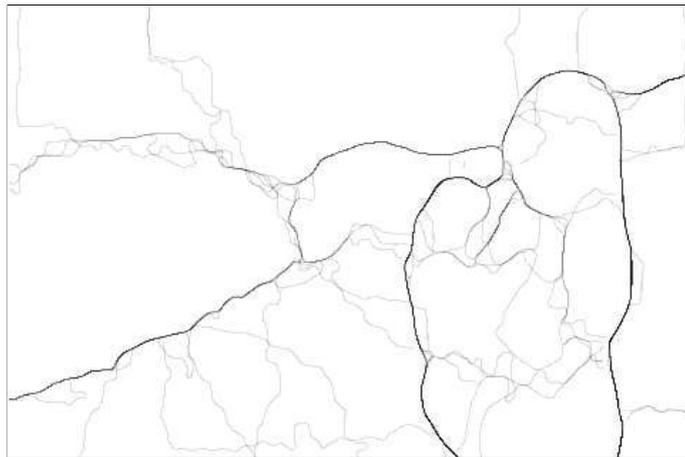
Fonte: Marcos Canêjo (2022)

Figura 38 – Imagens de borda geradas pelo algoritmo de Canny aplicado as imagens de *super-pixel* criadas a partir das modificações propostas.



Fonte: Marcos Canêjo (2022)

Figura 39 – Exemplo da imagem *sh_canny* da Figura 37(a).



Fonte: Marcos Canêjo (2022)

4.2 PARTE 2 - ANÁLISE DE TRAÇOS VIA MAPA DE SALIÊNCIA

A Parte 2 do algoritmo é composta por 2 passos de análise dos traços presentes na imagem *sh_canny*. Nesta parte, os algoritmos de saliência simulam a percepção do artista em relação a quais traços devem ser mantidos ou removidos.

A detecção de saliência também pode ser entendida como uma modelagem da percepção visual humana em identificar regiões distintas ou objetos de uma cena que possuem maior importância ou partes mais informativas. Os mapas de saliência, resultado da detecção de saliência em uma imagem, são utilizados em problemas de segmentação (SHEN et al., 2017), compressão (ZÜND et al., 2013) e diversas outras áreas (CRAYE; FILLIAT; GOUDOU, 2016; JERRIPOTHULA et al., 2017; MUSTAFA; HILTON, 2017).

O trabalho de Zhang *et al.* (ZHANG et al., 2015) explora a qualidade e velocidade de geração dos mapas de saliência. Com a grande demanda atual, um bom algoritmo de detecção de saliência de objetos deve, além de gerar mapas de saliência proveitosos, ser rápido em termos computacionais.

Uma das formas de realizar a distinção entre *background* e *foreground* em uma imagem é avaliando a conectividade dos pixels em relação a borda da imagem (ZHU et al., 2014; WEI et al., 2012; ZHANG; SCLAROFF, 2013; YANG et al., 2013) através, por exemplo, da métrica *Minimum Barrier Distance (MBD)* (STRAND et al., 2013; CIESIELSKI et al., 2014), porém seu uso torna a detecção de saliência lenta (ZHU et al., 2014; YANG et al., 2013). Assim, visando avaliar a conectividade dos pixels com a borda de forma rápida, Zhang *et al.* propõe uma versão modificada da *MBD*, chamada de *FastMBD*, combinada com a *appearance-based backgroundness cue* (JIANG et al., 2013), tornando a geração do mapa rápida e com qualidade.

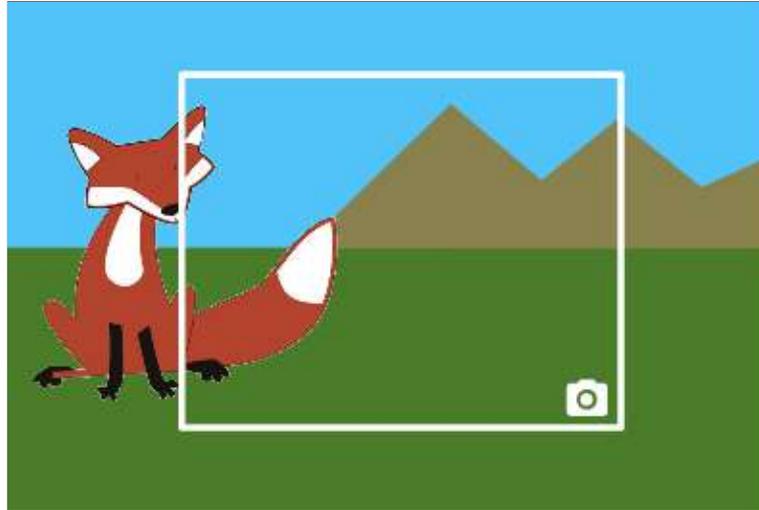
A *MBD* é uma *Distance Transform (DT)* (ROSENFELD; PFALTZ, 1968; BORGEFORS, 1984; BORGEFORS, 1996; DANIELSSON, 1980) que serve como uma ferramenta importante na análise morfológica e geométrica de objetos (HILDEBRAND; RÜEGSEGGER, 1997; SAHA et al., 2010) que, basicamente, calcula um mapa de distância entre os pixels da imagem. Diferentemente, da *MBD*, a *FastMBD* faz uso de mapas auxiliares que tornam seu desempenho mais rápido, porém, no trabalho de Zhang *et al.*, a principal contribuição está na adição da *appearance-based backgroundness cue*.

Por exemplo, na Figura 40 a raposa representa o *foreground* da imagem e todo o resto o *background*. No momento da aquisição, representada pela borda branca, apenas a cauda da raposa e outros detalhes do animal foram capturados. Esta situação retrata o foco da *appearance-based backgroundness cue*. Ou seja, o método será capaz de relacionar os pixels pertencentes a cauda do animal parcialmente removidos no momento da aquisição, com os pixels presentes na borda.

A *appearance-based backgroundness cue* consiste no cálculo de um mapa de fronteira, considerando os quatro limites de borda da imagem (cima, baixo, esquerda e direita), onde para cada limite, os valores de cor média e covariância de cores é feito, equivalente

ao trabalho de Rosenholtz (ROSENHOLTZ, 2001).

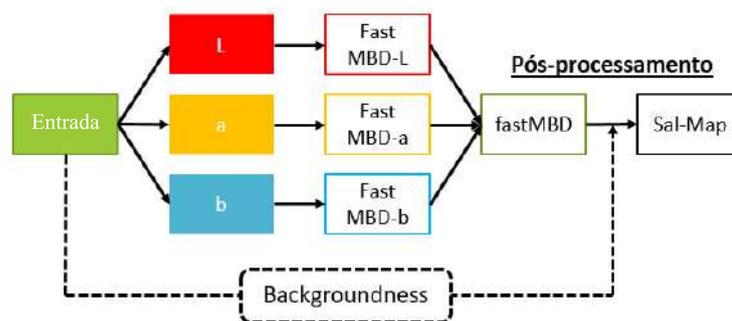
Figura 40 – Exemplo de cena natural no qual o objeto principal (representado pela raposa) tem parte dos seus pixels não capturados, mas relacionados com a borda da imagem.



Fonte: MBD... (2019)

A Figura 41 apresenta uma visão geral do algoritmo proposto. Nela, é possível verificar que a *fastMBD* é aplicada para cada canal do espaço de cor CIELAB da imagem. Isto é, os pixels de borda da imagem se tornam os alvos e a métrica *fastMBD* é então aplicada para cada canal. Em seguida, os mapas para cada canal de cor são combinados e seus pixels dimensionados de uma forma que o valor máximo seja 1, representando uma estimativa do pixel pertencer ao *foreground* da imagem.

Figura 41 – Visão geral do algoritmo *FastMBD*. Adaptada de (MBD..., 2019).



Fonte: MBD... (2019)

4.2.1 Detecção de Bordas Fracas

Embora a imagem *sh_canny* esteja próxima de um resultado final, ainda é possível observar a presença de bordas fracas, ou seja, que foram detectadas em apenas uma, duas ou três imagens Canny. No entanto, existe a possibilidade de algumas delas serem relevantes.

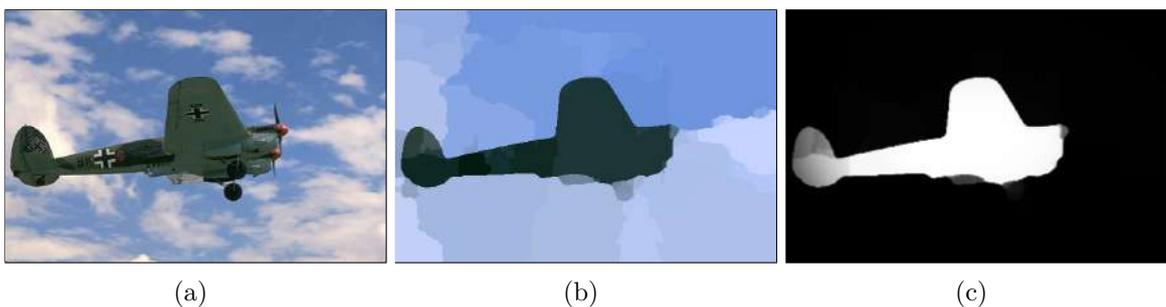
Como primeiro passo da remoção, baseado na ideia proposta por Yang e Min (YANG; MIN, 2019), na qual os autores propõem a utilização de uma imagem gradiente e uma imagem de saliência com o objetivo de obter uma imagem contendo os pixels mais expressivos (Imp), de acordo com a seguinte expressão:

$$Imp(x, y) = d \cdot \hat{S}(x, y) + (1 - d) \cdot \hat{G}(x, y), \quad (4.4)$$

sendo $\hat{S}(x, y)$ e $\hat{G}(x, y)$ pontos de coordenadas (x, y) na imagem de saliência e imagem gradiente, respectivamente; d uma variável de controle para o grau de relevância definida pelos autores em 0,5, tornando ambas as imagens igualmente ponderadas. No método proposto neste trabalho, as imagens de saliência e gradiente foram substituídas pela sh_canny e imagem de borda SFE da imagem original. Assim, após aplicação da Equação 4.4, a imagem $Imp(x, y)$ é limiarizada dando origem a sh_imp ; a limiarização é explicada em seguida.

Além das 8 imagens de Canny geradas, o método também produz 8 imagens de *superpixel*, relatado na seção 4.1.2. Então, assim como a ideia executada nas 8 imagens de Canny, de forma equivalente, as 8 imagens de *superpixel* são reunidas e transformadas em uma única imagem denominada $sh_colored$. Para cada canal RGB, seus valores são somados e normalizados no final. Em seguida, com a imagem $sh_colored$ definida, um mapa de saliência é criado, por meio do MBD+, para detectar quais regiões os principais objetos da cena estão presentes. A Figura 42 mostra a imagem $sh_colored$ e seu mapa de saliência para um exemplo do BSDS500. Como se vê, o mapa criado destaca apenas o objeto principal da cena, *i.e.*, o avião.

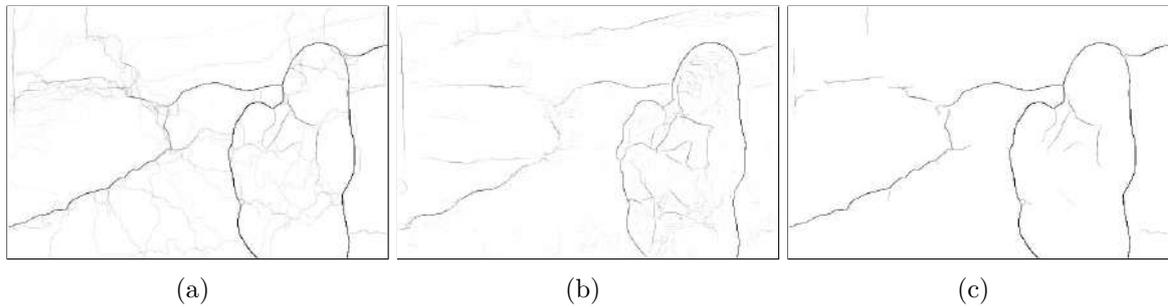
Figura 42 – (a) Imagem original, (b) sua versão das 8 imagens de *superpixel* resumida ($sh_colored$) e (c) mapa de saliência detectado pelo MBD+.



Fonte: Marcos Canêjo (2022)

Em seguida, uma máscara é criada a partir da binarização do mapa de saliência visando torna-ló mais preciso. Como o mapa é formado por valores entre 0 e 255, sendo 255 a região branca em destaque, valores abaixo de 10 são removidos. Em seguida, um valor médio dos pixels presentes na imagem sh_canny limitados a máscara é definido. Com isso, todos os pixels abaixo deste valor são removidos na imagem $Imp(x, y)$ completando a criação da imagem sh_imp . A Figura 43 mostra o resultado deste passo.

Figura 43 – Exemplo de bordas fracas removidas da (a) imagem *sh_canny* com ajuda da (b) imagem de borda SFE resultando na (c) imagem *sh_imp*.



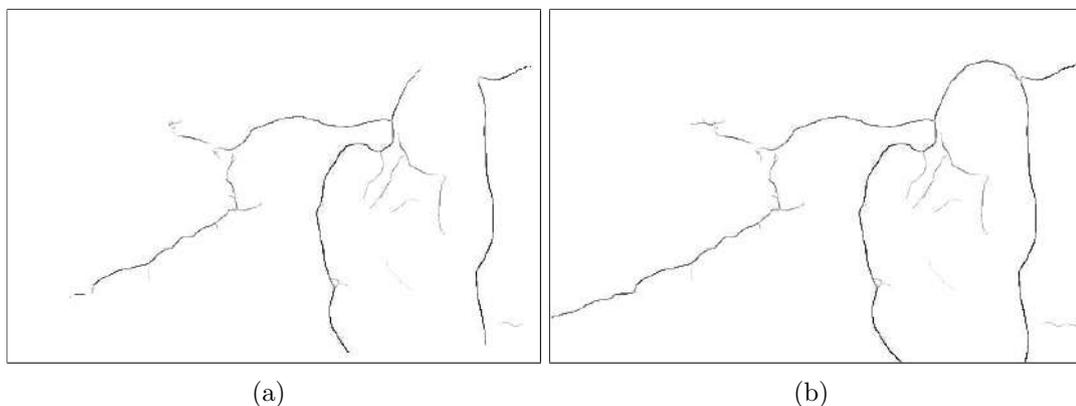
Fonte: Marcos Canêjo (2022)

4.2.2 Remoção de Traços Fora da Cena Principal

No passo anterior, a maioria das bordas fracas foram removidas. Assim, com uma imagem quase livre de bordas fracas, o próximo passo consiste em remover bordas que não fazem parte da cena principal, mas se mantiveram. Desse modo, o mapa de saliência criado a partir da imagem *sh_colored* é usado e todas as bordas que não estão presentes na região em destaque são removidas. Porém, em alguns casos, o mapa de saliência pode não incluir todo o objeto principal da cena, exigindo um passo de recuperação de bordas removidas incorretamente.

Um passo para recuperar é então executado, no qual as bordas removidas devem satisfazer os seguintes critérios: (i) elas devem estar conectadas as bordas que permaneceram, ou seja, fazem parte da região destacada pelo mapa de saliência; e (ii) o valor médio de seus pixels (em escala de cinza) deve ser maior ou igual as bordas restantes. Por fim, a imagem *sh_final* é criada contendo apenas as bordas do objeto principal da cena de sua imagem original. A Figura 44 exemplifica este passo.

Figura 44 – Exemplo das imagens com (a) as bordas removidas imprecisamente e (b) passo final de recuperação.



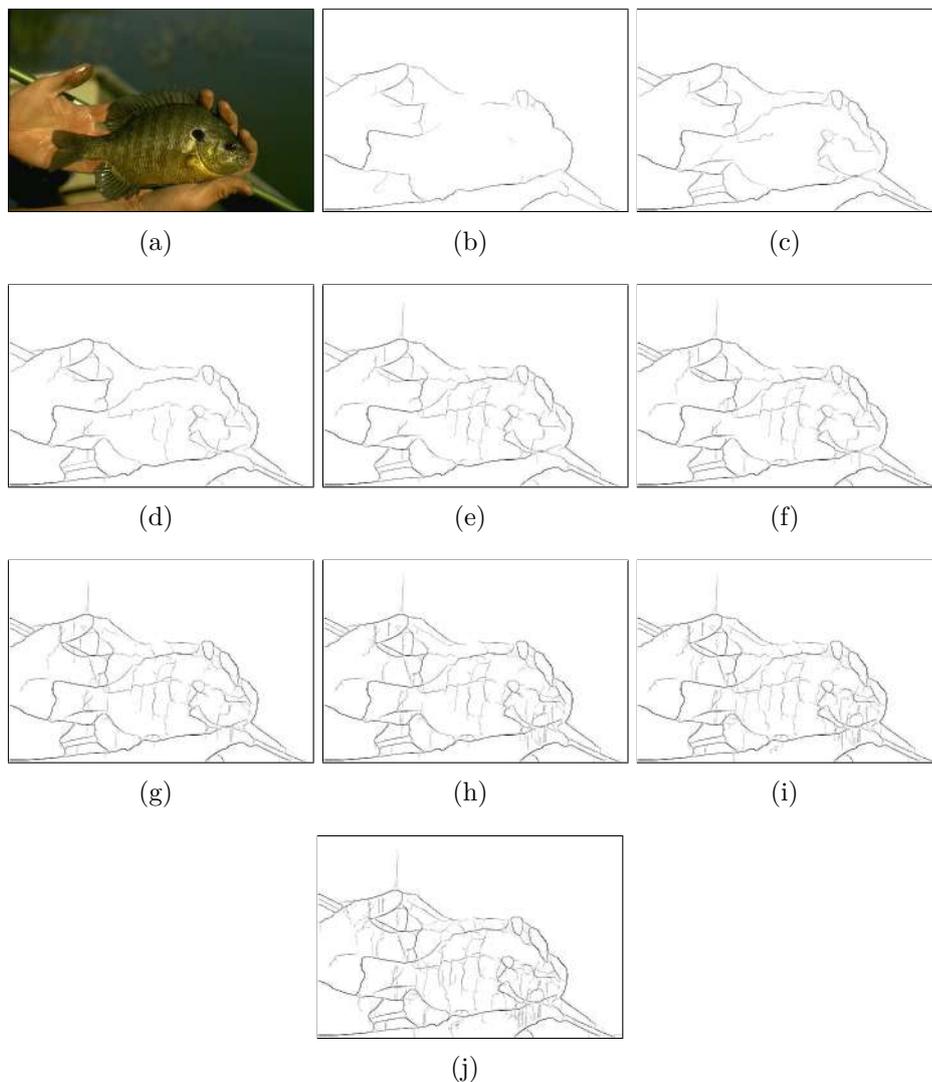
Fonte: Marcos Canêjo (2022)

4.3 MODELANDO IMAGEM DE 1 MINUTO E 10 MINUTOS

Para a imagem de 10 segundos do desafio, o algoritmo é o descrito na seção anterior. Para criar as imagens de 1 minuto e 10 minutos, o número de *superpixel* definido no primeiro passo é multiplicado por 5 (para imagem de 1 minuto) e 10 (para imagem de 10 minutos).

Na Figura 45, é possível observar a influência do número de *superpixels* na detecção de bordas. À medida que o valor aumenta, através do fator de multiplicação, é possível observar que a quantidade de bordas detectadas também segue aumentando, mas sempre relacionadas ao objeto principal da cena. Por exemplo, a quantidade de bordas encontradas na Figura 46(j) é muito maior que a da Figura 46(b), porém sempre com bordas relacionadas ao peixe e à mão, ou seja, aos objetos principais da cena.

Figura 45 – Exemplo de bordas detectadas para (a) imagem original com diferentes números de *superpixels* e (b) imagem de 10 segundos para comparação. O fator de multiplicação é: (c) 3, (d) 5, (e) 7, (f) 10, (g) 13, (h) 15, (i) 17 e (j) 20.



Fonte: Marcos Canêjo (2022)

É possível notar a variação do número de bordas detectadas pelo método proposto de acordo com o valor do fator de multiplicação, assim como acontece no **SDC** em relação ao tempo e sua relação com os detalhes presentes no desenho. Diferentes valores do fator podem simular diferentes tempos determinados no desafio.

5 RESULTADOS

Neste capítulo são apresentados os resultados obtidos para o problema de detecção de contorno e borda a partir da modelagem do *Speed Drawing Challenge* e analisados de forma qualitativa e quantitativa com uma metodologia específica aplicada nas métricas *Optimal Dataset Scale (ODS)* (limiar fixo para todas as imagens do banco de dados), *Optimal Image Scale (OIS)* (limiar por imagem) e *Average Precision (AP)* (média de precisão em relação aos limiares) para 30 limiares. As imagens utilizadas fazem parte do conjunto de teste do BSDS500 com 200 imagens de dimensões 321×481 e 481×321 .

5.1 DETECÇÃO DE BORDAS RELEVANTES

A Tabela 1 exibe os resultados obtidos para os tempos 10 segundos, 1 minuto e 10 minutos (SDC10s, SDC1m e SDC10m) comparados aos principais algoritmos, agrupados em duas classes: não baseados em CNN (Pb, UCM, gPb, ST, SFE); métodos baseados em CNN (DexiNed, COB, HED, BDCN, RCN). É importante notar que o foco da imagem gerada pelo SDC10s é nas bordas mais importantes. Isto é, bordas detectadas por seres humanos no *background* da cena, por exemplo, provavelmente, não estão presentes no SDC10s e SDC1m, diminuindo o *score* em comparação as outras abordagens.

Tabela 1 – Comparação entre os resultados do SDC proposto com outros algoritmos para o BSDS500.

		ODS	OIS	AP
Humano	-	0,80	0,80	-
não baseados em CNN	Pb	0,67	0,70	0,65
	ucm	0,68	0,75	0,58
	gPb	0,71	0,73	0,72
	ST	0,73	0,75	0,77
	SFE	0,74	0,76	0,77
CNN	DexiNed	0,68	0,70	0,51
	COB	0,78	0,80	0,64
	HED	0,79	0,81	0,84
	BDCN	0,80	0,81	0,50
	RCN	0,82	0,84	0,62
Proposto	SDC10s	0,67	0,67	0,52
	SDC1m	0,67	0,69	0,68
	SDC10m	0,66	0,67	0,67

Fonte: Marcos Canêjo (2022)

A ideia da imagem de 10 segundos consiste em detectar apenas as bordas que possuem uma alta relevância na cena da imagem original, removendo todas as outras. Assim, para demonstrar a influência desta detecção, um *benchmark* clássico da área para detecção de borda é feito, porém avaliando o impacto da remoção destas bordas nas imagens geradas por outros algoritmos. Esta forma de avaliação é proposta, dado que a forma clássica não demonstra os resultados alcançados. No *benchmark*, as imagens padrão ouro foram criadas por seres humanos em um cenário completamente diferente do **SDC**, onde o tempo, por exemplo, não é limitado, tornando a quantidade de bordas alta.

Como mostra a Tabela 2, adicionando as bordas do SDC10s nas bordas do **gPb** (coluna gPb+SDC10s na tabela) nenhuma relevância é notada. Isso significa que a maior parte das bordas do **gPb** também aparecem no SDC10s. Na remoção, entretanto, os valores sofrem uma exagerada perda (coluna gPb-SDC10s na tabela).

Tabela 2 – Valores obtidos no *benchmark* após adição (gPb+SDC10s) ou remoção (gPb-SDC10s) das bordas detectadas pelo SDC10s no **gPb**.

	gPb	gPb+SDC10s	gPb-SDC10s
ODS	0,711	0,707	0,497
OIS	0,730	0,731	0,548
AP	0,718	0,724	0,404

Fonte: Marcos Canêjo (2022)

Na Tabela 3 é possível observar um impacto similar para o SFE que possui melhores resultados na detecção de borda. Como na tabela anterior, a coluna SFE+SDC10s representa o resultado obtido da adição das bordas do SDC10s no SFE e SFE-SDC10s a remoção. Novamente, adicionar bordas detectadas pelo SDC10s não ofereceu melhoras nos resultados, porém na remoção, por exemplo, para o valor de **ODS**, o impacto se torna evidente.

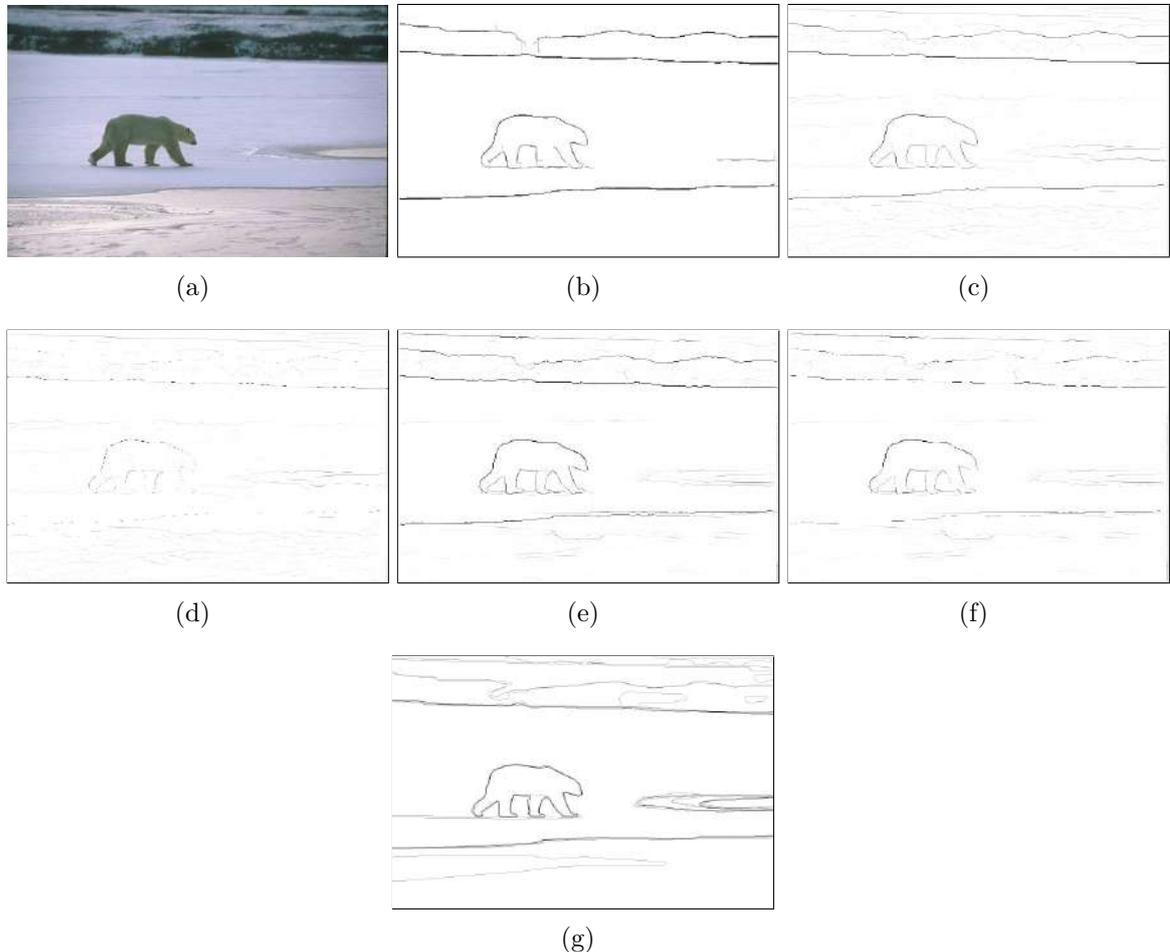
Tabela 3 – Valores obtidos no *benchmark* após adição (SFE+SDC10s) ou remoção (SFE-SDC10s) das bordas detectadas pelo SDC10s no **SFE**.

	SFE	SFE+SDC10s	SFE-SDC10s
ODS	0,735	0,739	0,686
OIS	0,758	0,761	0,711
AP	0,774	0,778	0,701

Fonte: Marcos Canêjo (2022)

A Figura 46 expõe um exemplo do dano causado na imagem **gPb** e **SFE** após a remoção das bordas, como proposto anteriormente. Em ambos os algoritmos, a imagem final expressa uma grande perda nas bordas.

Figura 46 – Exemplo da metodologia aplicada na análise quantitativa para (a) imagem original, sendo (b) a imagem de 10 segundos, (c) gPb, (d) gPb sem as bordas detectadas pela imagem de 10 segundos, (e) SFE, (f) SFE também sem as bordas presentes na imagem de 10 segundos e (g) o padrão ouro.



Fonte: Marcos Canêjo (2022)

5.2 RESULTADOS QUALITATIVOS

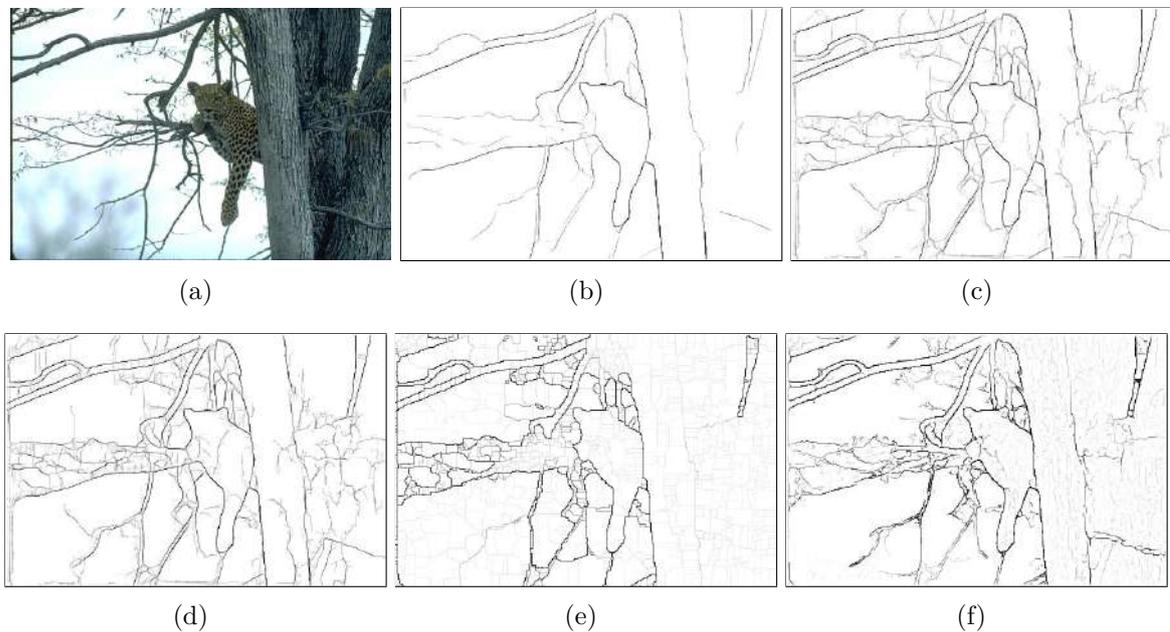
A principal qualidade oferecida pelo método proposto é a possibilidade de gerar imagens com diferentes níveis de bordas relevantes com uma simples modificação. As imagens de 1 minuto e 10 minutos, quando comparadas à imagem de 10 segundos, possuem muito mais bordas detectadas, tornando nítida a diferença entre elas. Entretanto, como mostra a Figura 47, quando comparadas a outros métodos, é possível identificar uma relação das bordas com características da imagem original. Por exemplo, mesmo na imagem de mais tempo, ou seja, 10 minutos (Figura 48(d)), as bordas detectadas estão mais relacionadas ao *foreground*.

Um dos problemas observados na abordagem é a dificuldade em diferenciar bordas causadas por texturas. Nas Figuras 48(e) e 48(f), uma quantidade considerável de bordas que correspondem às texturas da árvore e da onça ficam nítidas. Apesar de a proposta não

ter um passo específico para análise de textura, as 8 imagens **gPb** usadas para supressão de detalhes minimizou a ocorrência deste tipo de borda.

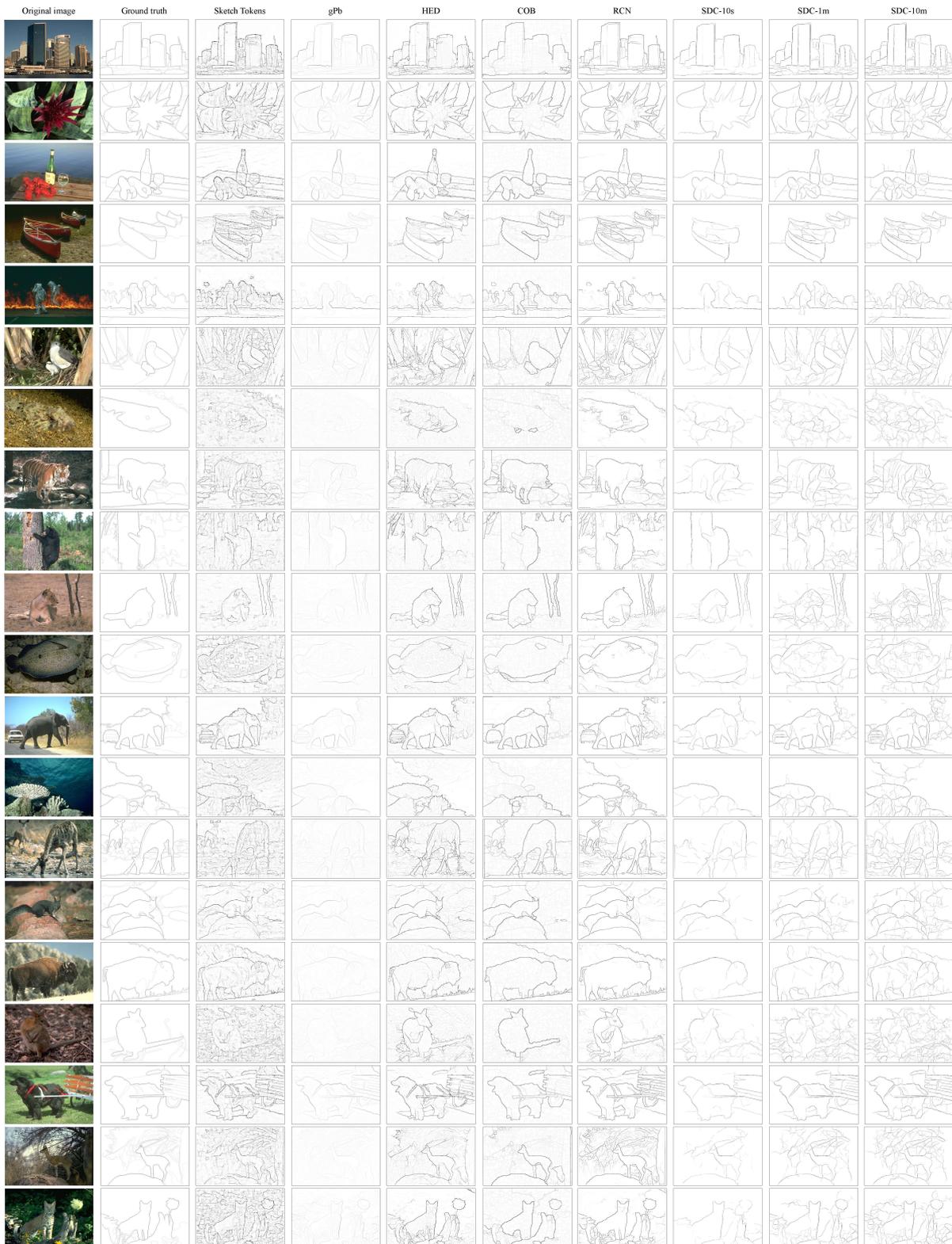
Os resultados qualitativos são os que melhor expressam os resultados obtidos. Na Figura **48**, 20 imagens são apresentadas com as imagens geradas pelo SDC10s, SDC1m e SDC10m, juntamente com vários outros métodos. As bordas detectadas para o tempo de 10 segundos representam bem a ideia original do **SDC**, *i.e.*, poucas bordas, mas suficiente para entendimento do desenho. A ideia se destaca quando comparada a outros métodos analisados, principalmente para a imagem de 10 segundos, e imagens rotuladas por humanos do BSDS500.

Figura 47 – (a) Imagem original e bordas detectadas por (b) SDC10s, (c) SDC1m, (d) SDC10M, (e) COB e (f) HED.



Fonte: Marcos Canêjo (2022)

Figura 48 – Resultado de 20 imagens criadas por diferentes algoritmos de detecção de borda e contorno, juntamente com as imagens do método proposto (para 10 segundos, 1 e 10 minutos).



Fonte: Marcos Canêjo (2022)

6 APLICAÇÃO: SEGMENTAÇÃO DE MAPAS E PLANTAS BAIXAS

A modelagem do **SDC** para a segmentação de mapas e plantas baixas tem como principal função separar as informações textuais sobrepostas a elementos gráficos. Esta aplicação demonstra como a inspiração do **SDC** pode ser aplicada para problemas de outros domínios de forma a produzir resultados satisfatórios.

A segmentação de informações gráficas e textuais para o cenário de mapas de plantas baixas são, principalmente, úteis para catalogações digitais, ajudando, por exemplo, a busca por um documento através do seu título. Outra importante aplicação são as contribuições nas áreas de arquitetura e engenharia civil. Por exemplo, para arquitetura, a imagem contendo apenas as informações gráficas pode ajudar o processo de identificação do estilo utilizado em uma planta baixa (preservação histórica). Para engenharia civil, as escalas, medições e outros valores segmentados, se tornam essenciais para, por exemplo, análise da estrutura. Para mapas, a segmentação pode ajudar, por exemplo, no processo histórico de criação de cidades de uma determinada região.

6.1 TRABALHOS RELACIONADOS

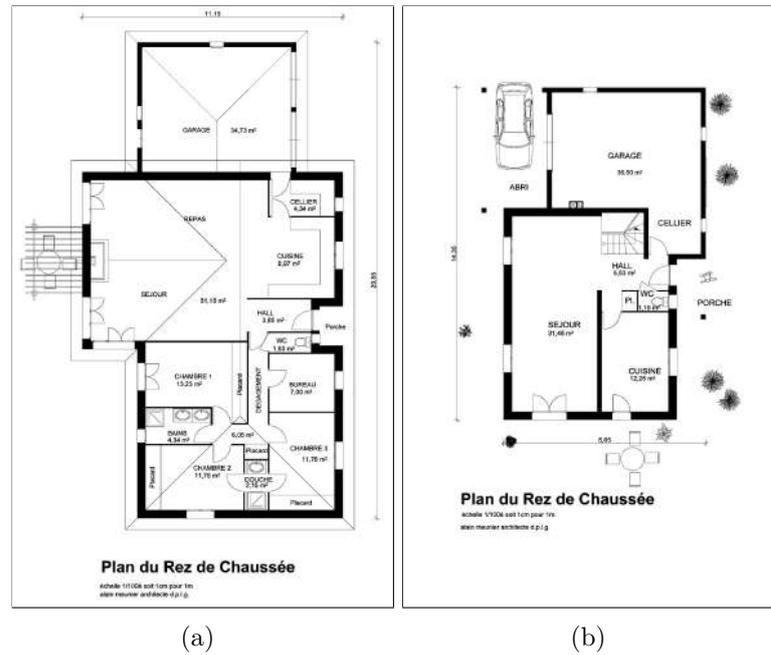
Este capítulo apresenta uma breve explicação dos trabalhos relacionados ao problema de segmentação de mapas e plantas baixas utilizados para demonstrar uma aplicação da ideia baseada no **SDC**.

6.1.1 Ahmed *et al.* (2011)

Em (AHMED M. WEBER; DENGEL, pp. 734-738, 2011), a segmentação do texto é feita após uma análise da planta baixa, convertendo-a em duas imagens: uma contendo apenas as informações de texto e a outra, os elementos gráficos. A Figura 49 apresenta as duas imagens originais utilizadas nos testes.

Antes da segmentação propriamente dita, é preciso retirar todos os elementos gráficos que representam as paredes externas, pois elas podem ser erroneamente rotuladas como texto durante o processo. A retirada destes elementos é feita aplicando três vezes a operação morfológica de erosão binária, seguida pela operação de dilatação com uma máscara de 3×3 aplicada quatro vezes, fazendo com que apenas os elementos de traços largos (as paredes) permaneçam na imagem. A imagem resultante deste processo é, então, comparada à imagem original e os elementos presentes em ambas são descartados, eliminando-se as paredes externas. Como mostra a Figura 51(a), os elementos de mesma espessura também podem ser eliminados neste processo como, por exemplo, o título da planta baixa, porém, nas etapas seguintes, esses casos são resolvidos.

Figura 49 – Exemplos de imagens utilizadas nos testes.



Fonte: HERAS et al. (2015a)

Para a remoção de linhas sobrepostas a textos, novamente, operações morfológicas binárias são aplicadas. A operação de abertura retira, primeiramente, linhas finas com uma máscara horizontal e, logo em seguida, uma vertical. Ao final, a imagem gerada, apresentada na Figura 51(b), é submetida ao processo de extração de elementos textuais.

A parte textual não é simplesmente definida por elementos pretos conectados encontrados na imagem, mas sim por uma avaliação que verifica o tamanho/área (quantidade de pixels), as densidades ($\frac{area}{largura \times altura}$), as dimensões, os *bounding boxes*, a largura, a altura e os centroides dos componentes conectados que representam a parte textual, uma vez que algumas linhas finas ainda estão presentes na imagem.

Após uma verificação da área de maior ocorrência (A_{amp}) e o valor médio (A_{avg}) dos elementos presentes na imagem, os limiares T_1 (limiar de tamanho máximo) e T_2 (limiar de alongamento máximo) são calculados:

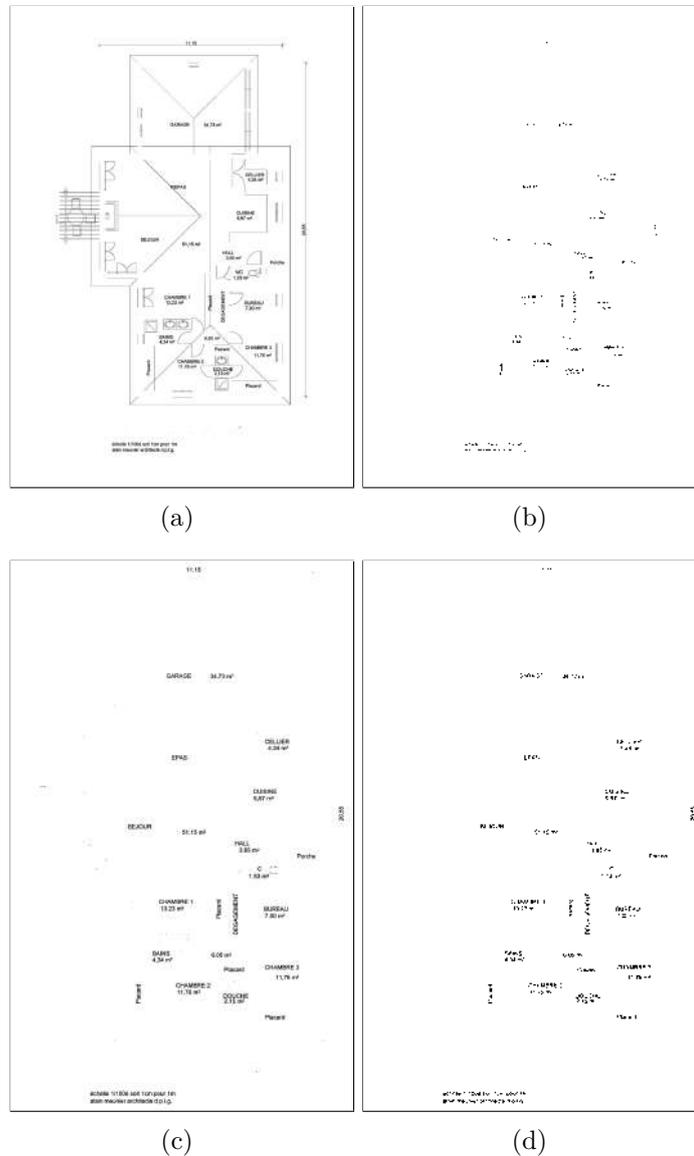
$$T_1 = N \times \max(A_{amp}, A_{avg}), \quad (6.1)$$

$$T_2 = \max(avg_{altura}, avg_{largura}), \quad (6.2)$$

onde $N = 5$ é um fator de tamanho escolhido de forma empírica pelos autores do trabalho, $avg_{largura}$ e avg_{altura} são a largura média e altura média dos componentes conectados, respectivamente. Assim, os componentes conectados que satisfaçam os seguintes critérios são classificados como texto:

- Área menor que o limiar T_1 ;

Figura 50 – (a) Imagem resultante do processo de remoção das paredes de traço largo, (b) remoção de linhas finas, (c) classificação dos componentes conectados em texto e (d) remoção de ruídos deixados pelos processos anteriores.



Fonte: Marcos Canêjo (2022)

- $\frac{\text{altura}}{\text{largura}}$ deve estar no intervalo $[\frac{1}{T_2}, T_2]$;
- Largura e altura são menores que $\sqrt{T_1}$.

A Figura 51(c) mostra que, de fato, algumas linhas restantes foram retiradas da imagem, porém, por exemplo, a letra “W” da palavra “WC” foi equivocadamente excluída.

O próximo passo do algoritmo é a avaliação dos elementos conectados restantes na imagem intermediária gerada pelo passo anterior. Esta avaliação representa a remoção dos elementos que são pequenos demais para serem texto e que são considerados ruídos deixados pelos passos anteriores. Inicialmente, um “borramento” horizontal e vertical na imagem intermediária é executado pretendendo unir os elementos que estão próximos uns

dos outros. No artigo original, não há informações de como esse “borramento” deve ser feito, então o adotado na implementação foi o filtro Gaussiano com σ igual a 2 e janela com tamanho 8 e 5 para o borramento na horizontal e vertical, respectivamente.

Os elementos são excluídos caso satisfaçam qualquer um dos critérios a partir da imagem borrada:

- Sua largura ou altura for menor que a média;
- Densidade menor que o limiar denotado por T_d ;
- E sua área for muito pequena.

Os autores não deixaram claro o que seria uma área pequena. Sendo assim, a partir de avaliações baseadas nas imagens testadas, elementos conectados com áreas inferiores a 100 pixels foram consideradas pequenas. Para o limiar T_d , o adotado foi o mesmo que o do artigo original, 30%. O resultado deste passo pode ser visto na Figura 51(d).

Como último passo, a imagem contendo apenas as paredes externas resultante do primeiro passo, como já citado, contém o título que deve ser recuperado. A sua recuperação é feita após uma inspeção da densidade e largura dos elementos conectados após o mesmo processo de “borramento” do passo anterior. Assim, os elementos com densidade menor que 70%, estabelecido empiricamente pelos autores, e com largura fora do intervalo [15 – 100], são excluídos. De acordo com os autores, o intervalo para análise da largura é de [200 – 600], contudo, devido à ausência de informações sobre o processo de “borramento” da imagem, tais valores precisaram ser adaptados ao que foi implementado, resultando em uma imagem diferente da apresentada no artigo como pode ser visto na Figura 51.

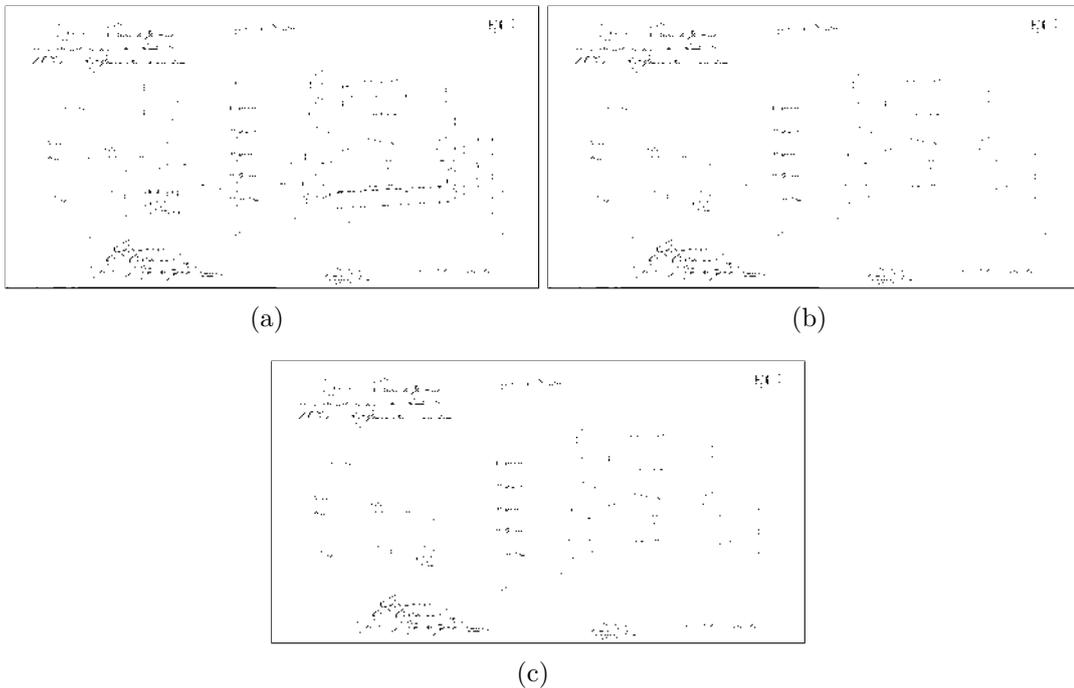
O artigo apresentado possui várias questões em aberto que tornam difícil a implementação do método. A Figura 53(a) mostra que o parâmetro $N = 5$ para a Figura 50(b) produziu uma imagem com alguns elementos textuais perdidos e impossíveis de serem recuperados. A Figura 53(b), que representa a imagem final, contém apenas o título de elemento textual e vários ruídos.

6.1.2 Mello e Machado (2014).

Plantas baixas e mapas apresentam suas informações em forma de textos e desenhos que se relacionam com o objetivo de representarem uma forma visual do espaço em questão. Em (MELLO; MACHADO, pp. 3476-3480, 2014), um novo algoritmo é proposto para a segmentação de texto em imagens de plantas baixas e mapas topográficos. A segmentação é dividida em: remoção do *background*, segmentação do texto, remoção de linhas e restauração.

O primeiro passo é apenas executado se o *background* do documento for classificado como texturizado a partir de uma verificação estatística dos pixels com valores acima

Figura 53 – (a) Imagem binarizada, (b) remoção de pequenos e (c) grandes elementos do resultado final do passo de segmentação de texto.



Fonte: Marcos Canêjo (2022)

documento é o que se mantém. Esta simulação de percepção de um objeto à distância é empregada por dois operadores morfológicos de fechamento com discos como elementos estruturantes onde seu raio é definido de acordo com a largura do traço do texto presente no documento. A imagem tem o seu tamanho reduzido até que o texto não possa ser visto e recuperada para o seu tamanho original para avaliação do que pertence ao plano de fundo. A imagem é, então, equalizada e binarizada pela combinação do método de Otsu e o algoritmo *k-Means* com três classes: texto, *background* e valores intermediários. A Figura 54(a) expõe o resultado.

A etapa de segmentação do texto se inicia com a remoção de pequenas áreas (menos que 20 pixels), que provavelmente são ruídos, e áreas grandes que apresentam mais que 10% do tamanho da maior área rotulada da etapa anterior. Isso resulta na imagem que pode ser vista na Figura 54(b). Alguns outros elementos que ainda não foram removidos são eliminados a partir da comparação de duas imagens geradas por operações morfológicas de abertura e fechamento, com um elemento estruturante em forma de disco com raio igual a 2. Frequentemente, é possível encontrar bordas neste tipo de documento que passou por algum tipo de degradação. Então, pixels pretos conectados à borda da imagem são deletados. Como nenhuma informação *a priori* acerca da forma do texto foi analisada, alguns elementos não textuais podem ainda não terem sido descartados, mas eles podem ser facilmente reconhecidos por sua distribuição não uniforme ao longo da imagem e então retirados. Por fim, a segmentação do texto propriamente dita é feita por uma

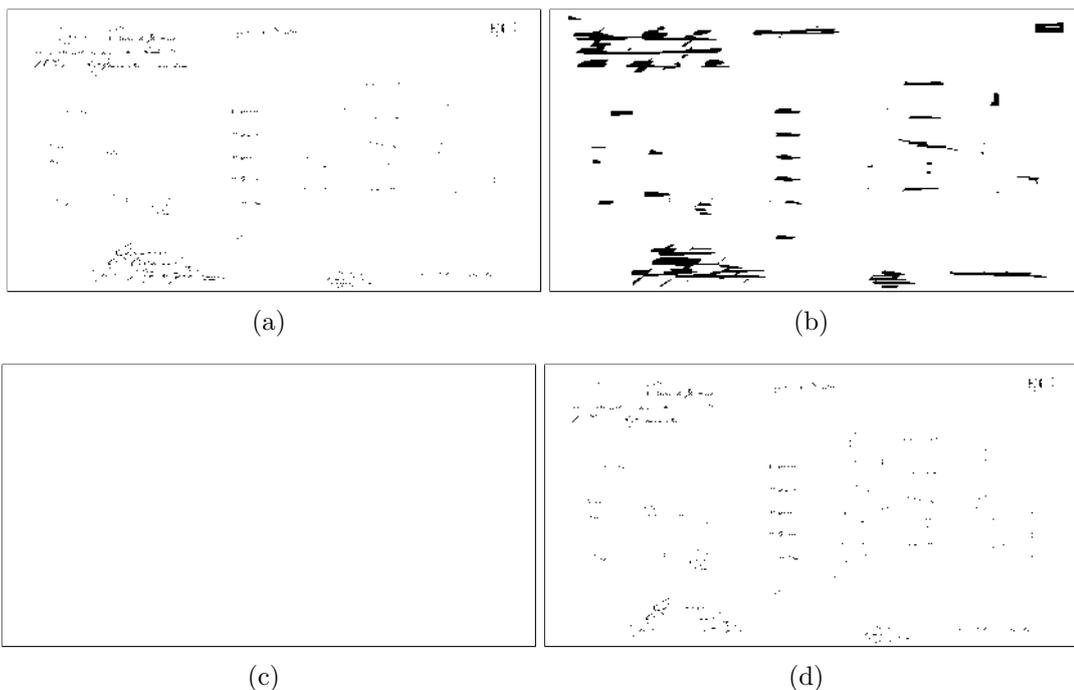
simples comparação entre os centroides dos caracteres. Centroides longe uns dos outros significa que os possíveis caracteres não formam uma palavra, ou seja, são desconsiderados. Na Figura 54(c), é possível verificar que, apesar de a imagem conter em sua maioria informações textuais, algumas linhas ainda permanecem. Elas são removidas no passo seguinte.

Com o intuito de remover as linhas de grade do papel, quando presentes, duas imagens com componentes maiores que 400 pixels e outra com os componentes menores que 400 pixels são criadas. Na primeira imagem (Figura 55(a)), a operação morfológica de fechamento (Figura 55(b)) com uma linha na horizontal com 200 pixels como elemento estruturante é aplicada juntamente com a operação de afinamento (Figura 55(c)). As regiões que perderam mais de 80% de sua área são potencialmente consideradas texto e, obviamente, não são removidas. Logo em seguida, as coordenadas dos outros componentes são utilizadas para calcular a distância (erro relativo que pode ser interpretado como o erro entre a linha projetada e a linha real) entre as linhas normais e as que foram afinadas. Esse erro é calculado

$$\text{distância} = \text{abs}(-y + m.x + b/\sqrt{1 + m^2}), \quad (6.3)$$

onde m representa o declive da linha e b o ponto em que a linha cruza o eixo y . A imagem resultante (Figura 55(d)) é chamada de *image_text* e deve apenas conter texto.

Figura 54 – (a) Imagem com os componentes maiores que 400 pixels, (b) operação de fechamento, (c) afinamento e (d) resultado final.

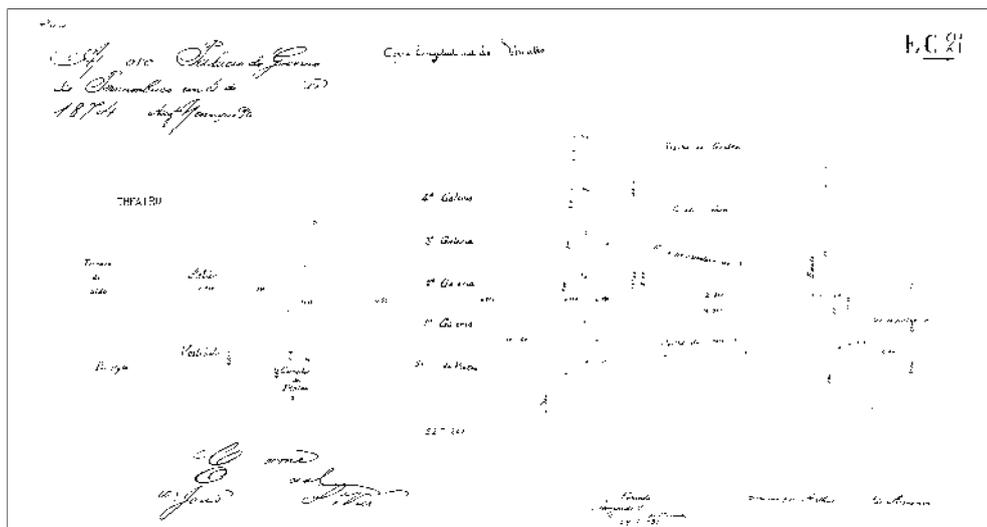


Por fim, alguns caracteres podem ter sido removidos erroneamente e sua recuperação é feita em dois passos.

No primeiro, caracteres que estavam entre outros caracteres são recuperados a partir de uma verificação de distância e comparação de suas alturas. Caso o elemento analisado esteja a uma distância máxima de 300 pixels entre dois outros caracteres e a diferença de altura não ultrapasse os 20 pixels, o elemento é considerado um caractere e volta a pertencer à imagem.

O segundo passo é mais complexo porque tenta recuperar caracteres que estavam conectados com elementos gráficos. Aqueles elementos maiores que seus vizinhos são escolhidos como possíveis candidatos. Então, um algoritmo de afinamento é aplicado em todos os elementos (analisado e seus vizinhos) e tem os seus traços isolados para avaliação. Os traços do elemento analisado que forem menores que a soma da média e desvio padrão do comprimento dos seus vizinhos são considerados texto e, portanto, recuperados. A Figura 55 exibe a imagem final com o máximo possível de texto e o mínimo de elementos gráficos.

Figura 55 – Resultado do passo final de recuperação de caracteres.



Fonte: Marcos Canêjo (2022)

6.1.3 Machado e Mello. (2015)

Diferente dos outros métodos apresentados para segmentação de texto (AHMED M. WEBER; DENGEL, pp. 734-738, 2011; MELLO; MACHADO, pp. 3476-3480, 2014) neste trabalho, a proposta apresentada em (MACHADO; MELLO, pp. 1-8, 2015) faz uso da técnica de inteligência computacional *Support Vector Data Description (SVDD)* (TAX; DUIN, 2004) para atingir uma melhor classificação dos elementos textuais presentes na imagem.

O SVDD é um método bastante útil para problemas de classificação de classe única e como o problema de segmentação de texto abordado neste estudo possui apenas uma única classe (elementos textuais), o uso do SVDD se torna oportuno.

O algoritmo proposto apresenta seis etapas: pré-processamento, remoção de elementos gráficos, detecção de caracteres, separação de pequenos elementos alongados, detecção de palavras (ou sequência de caracteres conectados) e recuperação.

O pré-processamento consiste na remoção do *background* através da técnica (MES-QUITA; V.ALMEIDA, vol. 21, pp. 133-146, Novembro 2014) que simula a percepção de objetos à distância. Isto é, à medida que um objeto se afasta, a percepção de detalhes diminui, tornando, por exemplo, em um documento, mais evidente os caracteres do seu *background* e diminuindo os detalhes da escrita. Esta simulação é executada por meio da operação morfológica de fechamento com dois discos de raio 3 e 5, como elementos estruturantes, e uma redução no tamanho de altura para 40 pixels da imagem. Os discos provocam o arredondamento de elementos de borda, assim como acontece no sistema visual humano; e a redução do tamanho simula o afastamento do objeto em relação ao observador. A imagem é então redimensionada ao seu tamanho original e subtraída a imagem de entrada. A Figura 56 apresenta um exemplo deste processo.

Após o pré-processamento da imagem de entrada, a remoção de elementos gráficos é feita por meio de uma avaliação da área de cada componente conectado. Aqueles que tiverem uma área maior que 0,2% da área total da imagem e uma área menor que 13 pixels são removidos. Logo depois, a operação morfológica de abertura com um elemento estruturante em forma de cruz de tamanho 3×3 é aplicada para remover possíveis elementos com 1 pixel de largura. Aqueles que não foram eliminados, mas sofreram perda de pixels, são recuperados à sua forma original. Na Figura 57, que representa a imagem resultante deste processo, é possível visualizar que vários elementos gráficos foram removidos, porém, de forma errônea, alguns caracteres também.

Através das características para detecção de textos definidas em (KARAOGLU; FERNANDO; TREMEAU, 2010; SHI et al., 2013) por Karaoglu *et al.* e Shi *et al.*, respectivamente, o SVDD é aplicado para decidir quais elementos são textos ou não. O *Aspect Ratio*, *Occupy Ratio* e quantidade de “buracos” nos caracteres, são características robustas a transformações de tamanho e, por isso, foram as escolhidas nesta abordagem, uma vez que o tamanho dos textos encontrados em mapas e plantas baixas podem variar.

O *aspect ratio* (Equação 6.4) representa a espessura do Elemento Conectado (EC) e o *OccupyRatio* (Equação 6.5) representa o quanto ele ocupa em termos de área do *bounding box*.

$$\text{AspectRatio}(\text{EC}) = \frac{\text{menorComprimentoDoEixo}(\text{EC})}{\text{maiorComprimentoDoEixo}(\text{EC})}, \quad (6.4)$$

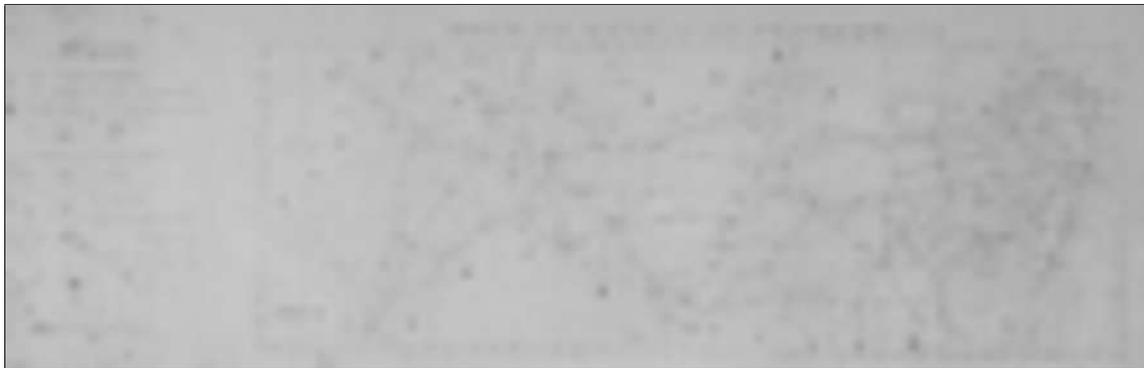
$$\text{OccupyRatio}(\text{EC}) = \frac{\text{Área}(\text{EC})}{\text{BoundingBoxÁrea}(\text{EC})}. \quad (6.5)$$

A quantidade de “buracos” pode ser interpretada como o número de espaços em branco cercados por pixels do elemento. Esse valor pode ser obtido pela diferença entre uma imagem preenchida e a imagem original.

Figura 56 – (a) Imagem de entrada com os detalhes realçados, (b) imagem desfocada pela simulação de afastamento do objeto em relação ao observador e (c) imagem final.



(a)



(b)



(c)

Fonte: Marcos Canêjo (2022)

As características *Compactness* (Equação 6.6) e *Regularity* (Equação 6.7) descrevem a complexidade da forma dos elementos. A primeira é calculada pela relação entre a área e o quadrado do perímetro. A *Regularity* divide a área da esqueletização do elemento pelo seu perímetro. Por fim, a *Corner Based Interpolated Feature* (CBIF) (KARAOGLU; FERNANDO; TREMEAU, 2010) é adicionada ao conjunto de características no qual o SVDD utiliza e classifica todos os elementos conectados em texto ou não.

Figura 57 – Imagem após o passo de remoção de elementos gráficos.



Fonte: Marcos Canêjo (2022)

$$Compactness(EC) = \frac{\text{Área}(EC)}{\text{Perímetro}(EC)} \quad (6.6)$$

$$Regularity(EC) = \frac{\text{Área}(\text{Esqueleto}(EC))}{\text{Perímetro}(EC)} \quad (6.7)$$

Um pequeno passo entre a classificação e detecção dos caracteres é a eliminação de linhas e traços baseados em suas áreas e comparadas com as dos elementos textuais detectados no passo anterior.

No último passo para a detecção de texto, o *Deformable Histogram of Oriented Gradients (Deformable HOG)* (J.ALMAZAN; E.VALVENY, pp. 1022-1026, 2013) é utilizado para a detecção de sequência de caracteres que, diferente do HOG clássico, propõe um *grid* adaptável baseado no centro de massa da imagem. Novamente, o *SVDD*, com os valores obtidos pelo *Deformable HOG*, é então aplicado agora na classificação das sequências.

Então, no final de todo o processo, um passo de restauração é feita para recuperar elementos que foram erroneamente classificados. A imagem contendo o texto é criada a partir dos elementos obtidos após esse passo como pode ser vista na Figura 58.

Figura 58 – Imagem final após a recuperação de caracteres.



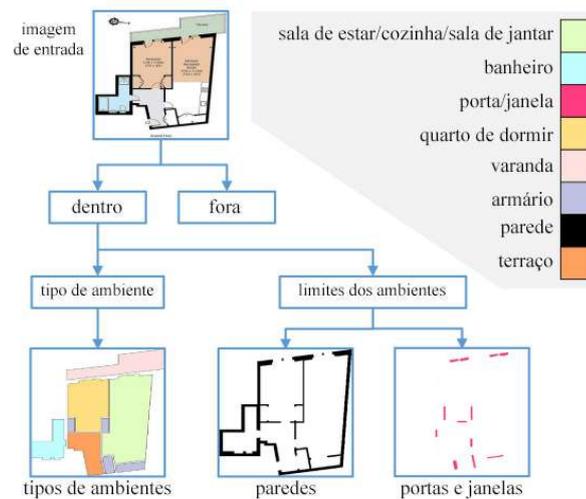
Fonte: Marcos Canêjo (2022)

6.1.4 Zeng *et al.* (2019)

Um importante aspecto nos trabalhos que envolvem imagens de plantas baixas é a detecção de paredes, salas, portas etc. Para Zeng *et al.* (ZENG *et al.*, 2019), estes elementos possuem uma relação semântica estrutural. O problema é então modelado a partir de uma rotulação semântica de forma hierárquica através de uma *deep multi-task neural network*. Dada uma imagem de entrada, a rede aprende as características iniciais e vai refinando a detecção até classificação dos elementos, como mostra a Figura 59.

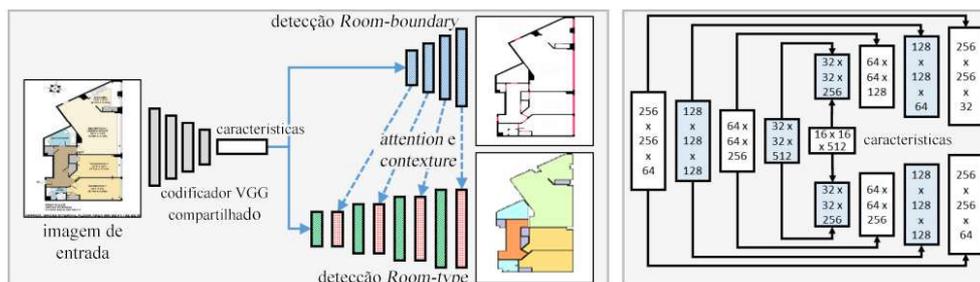
A Figura 60 ilustra o processo realizado pela rede. Um codificador VGG extrai as características da imagem de entrada que são compartilhadas para duas diferentes tarefas: detecção dos *Room-boundary*, ou seja, paredes, portas, janelas etc.; detecção do *Room-type* apresentados na Figura 59. Como pode ser visto, as tarefas possuem decodificadores diferentes, na primeira (*Room-boundary*), os autores usaram a ideia de *attention* (ilustrado por setas azuis) que ajuda o decodificador nas características *contextual* (caixas em vermelho) para detecção dos ambientes.

Figura 59 – Detecção hierárquica realizada pela rede proposta.



Fonte: ZENG *et al.* (2019)

Figura 60 – Ilustração da *deep multi-task neural network* proposta.

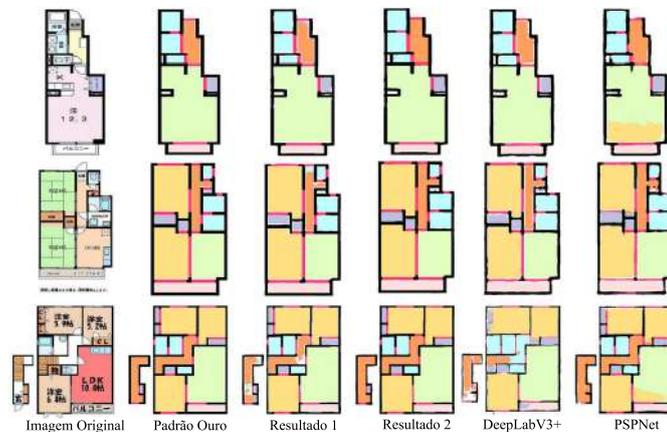


Fonte: ZENG *et al.* (2019)

Devido à falta de dados rotulados, os autores prepararam dois bancos de dados, denominados R2V e R3D. O primeiro possui 815 imagens; o segundo, 214 imagens de (LIU et al., 2015) mais 18 imagens adicionadas pelos autores. No treinamento, 715 imagens do R2V serviram para o treino e 100 para o teste. Já para o R3D, 179 imagens foram usadas para treino e 53 para teste.

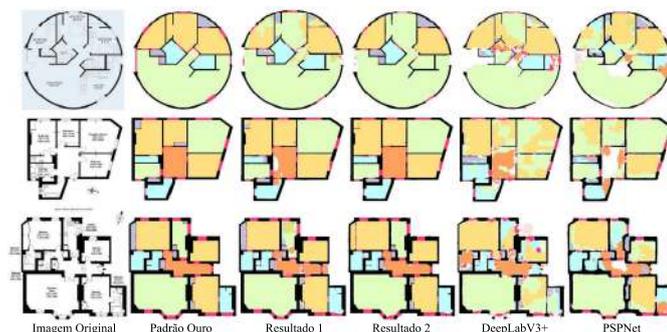
A Figura 61 apresenta os resultados obtidos para o banco R2V: o primeiro (Resultado 1) descrito acima; o segundo (Resultado 2) com um pós-processamento que conecta regiões que tiveram seus limites removidos erroneamente, comparados com as técnicas DeepLabV3+ (CHEN et al., 2018) e PSPNet (ZHAO et al., 2017), como também a Figura 62 para o banco R3D seguindo o mesmo conceito.

Figura 61 – Resultados do método de Zeng *et al* para o banco R2V em relação a outras técnicas.



Fonte: ZENG et al. (2019)

Figura 62 – Resultados obtidos por Zeng *et al* para o banco R3D comparados a outras técnicas.



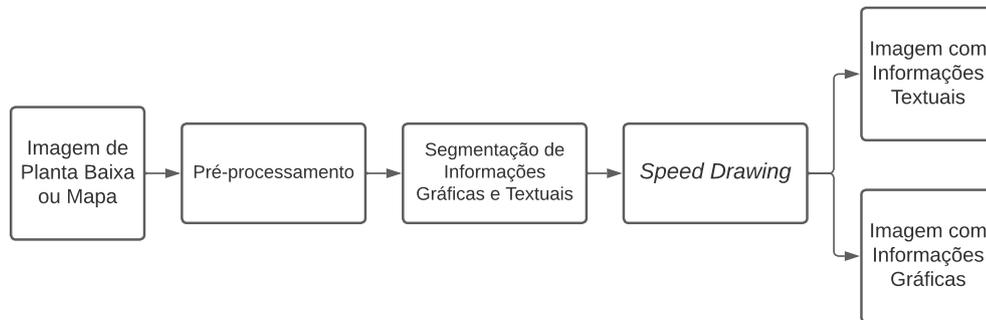
Fonte: ZENG et al. (2019)

6.2 MÉTODO DE SEGMENTAÇÃO POR SDC

Esta seção apresenta o método proposto baseado no SDC para o problema de segmentação de mapas e plantas baixas. O método possui 3 passos visando separar as informações

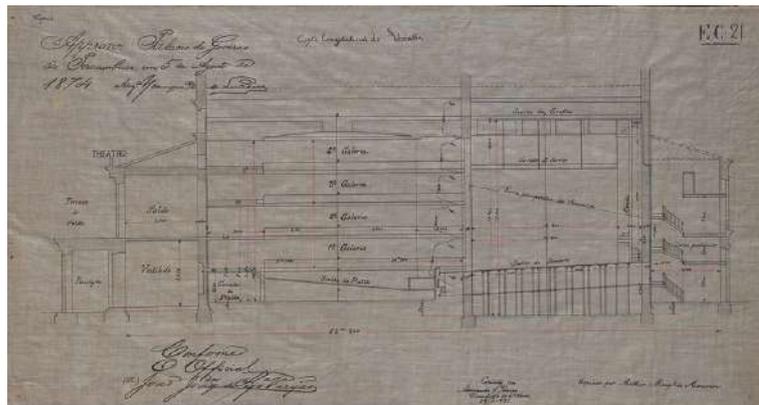
textuais e gráficas. A Figura 63 apresenta os passos do método e a Figura 64 a imagem original usada como exemplo.

Figura 63 – Diagrama de blocos do método para segmentação de informações textuais e gráficas em imagens de mapas e plantas baixas.



Fonte: Marcos Canêjo (2022)

Figura 64 – Imagem original de planta baixa usada como exemplo.

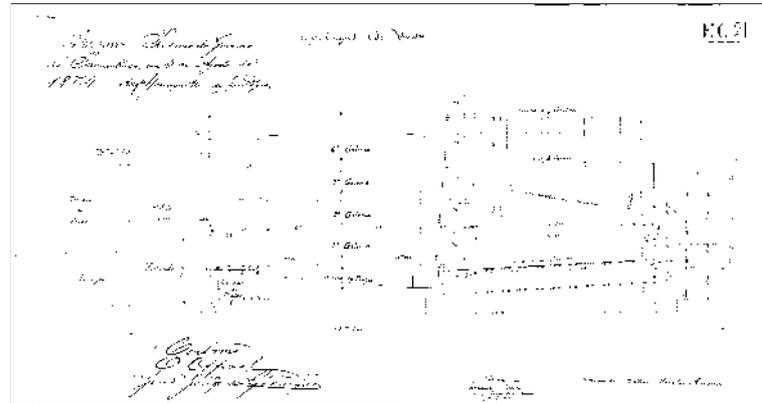


Fonte: Marcos Canêjo (2022)

6.2.1 Pré-processamento

Documentos históricos possuem degradações geradas por fatores internos (composição do papel, tipo de pigmentação etc.) e, principalmente, por fatores externos (modo de armazenamento, manuseio etc.), gerando um problema para sua conservação, além de dificultar o processamento de suas versões digitais. Assim, o primeiro passo do método consiste em um pré-processamento para remoção de partes dos elementos da imagem que não são relevantes para a aplicação (o *background*). Para tanto, usa-se a ideia proposta em (MACHADO; MELLO, pp. 1-8, 2015), no qual simula-se um afastamento da imagem de forma a perder a percepção de detalhes dela. Assim, o *background* da imagem é convertido para branco. Um exemplo deste passo é apresentado na Figura 65.

Figura 65 – Resultado do pré-processamento para planta baixa.



Fonte: Marcos Canêjo (2022)

6.2.2 Segmentação de informações gráficas e textuais

Com o *background* da imagem removido, o próximo passo é realizar a divisão entre as informações gráficas e textuais. Em ambos os documentos é possível identificar que as informações textuais estão dispostas de formas totalmente distintas. Por exemplo, no mapa o título possui um tamanho próprio, as letras referentes às palavras dos estados de fronteira estão distantes umas das outras, e etc. Assim, a divisão é feita a partir de uma análise para cada componente conectado.

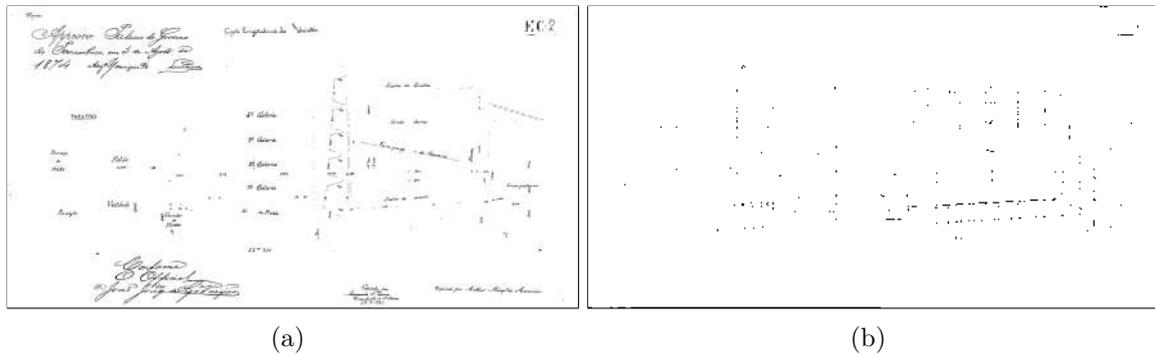
Como proposto por Ahmed *et al.* (AHMED M. WEBER; DENGEL, pp. 734-738, 2011), os componentes conectados são analisados a partir de estatísticas simples como tamanho de área, tamanho médio do *bouding box*, largura e altura. O método sugere a criação de uma imagem com informações textuais e as gráficas divididas em outras duas com os elementos classificados como pequenos e grandes. Porém, para o problema de segmentação de mapas e plantas baixas, dividir as informações em textuais e gráficas é suficiente.

Com isso, duas novas imagens são criadas: *text_layer*, contendo os textos detectados; *graphics_layer*, com informações gráficas. A Figura 66 apresenta um exemplo do resultado deste passo. Diferentemente do trabalho original, os valores de n e T_2 variaram para o tipo de documento em análise, ou seja, para mapas os valores assumidos foram $n = 40$ e $T_2 = 8$, bem como para plantas baixas $n = 5$ e $T = 3$. Estes valores foram definidos a partir de uma análise visual dos resultados no qual a maioria das imagens apresentam respostas consideradas satisfatórias.

6.2.3 Speed Drawing

A Figura 67(a) deveria conter apenas informações textuais, porém vários traços pertencentes ao desenho da planta permanecem. Por outro lado, a outra parte da imagem, isto é, a imagem com as informações gráficas, exemplificada na Figura 67(b), contém informações textuais erroneamente classificadas devido à sobreposição entre texto e desenho.

Figura 66 – Exemplo do passo de divisão entre as informações textuais e gráficas, dando origem as imagens (a) *text_layer* e (b) *graphics_layer*.



Fonte: Marcos Canêjo (2022)

Então, visando recuperar os textos, o esboço da imagem com as informações gráficas é produzido baseado na ideia dos 10 segundos do [SDC](#).

A imagem *graphics_layer* é submetida à ideia de ser reproduzida em poucos segundos, dando origem a uma imagem esboço (*sketch*). Como considerado anteriormente, o “artista”, percebendo que a imagem possui mais informação gráfica do que textual, irá reproduzir um desenho sem os detalhes, retratados aqui como sendo os caracteres que permaneceram. A imagem esboço é criada a partir do conceito de *smearing* ([CAO et al., 2007](#); [KAUR; JINDAL; KUMAR, 2021](#)), usado na análise de documentos.

O *smearing* usado neste passo é realizado por uma sequência de convoluções aplicadas na imagem *graphics_layer* através de um filtro h criado da seguinte forma:

1. Construa um segmento de linha de tamanho len e angulo $theta$, centrado no meio de h ;
2. Para cada ponto de h calcular a menor distância mais próxima;
3. $h = \max(1 - \text{menordistância}, 0)$;
4. Normalizar h .

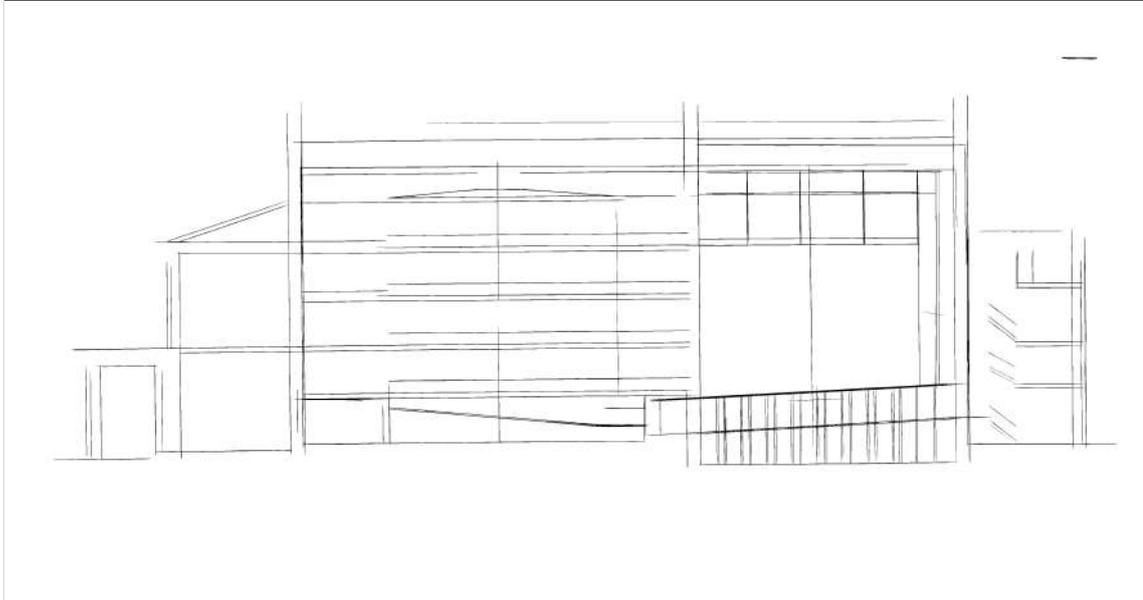
Por exemplo, para um valor de $len = 5$ e $theta = 45$ o filtro construído é:

0	0	0	0,050	0,030
0	0	0,051	0,177	0,050
0	0,051	0,177	0,051	0
0,050	0,177	0,051	0	0
0,030	0,050	0	0	0

O *smearing* é então obtido aplicando um filtro h para valores de $theta$ variando de 0 até 360. O valor de len depende do tamanho da imagem. Por exemplo, a Figura [67](#), que

representa o resultado do *smearing*, ou seja, a imagem *sketch*, possui dimensões 7885×4194 pixels. Nos testes realizados, o valor de *len* foi fixado em 500.

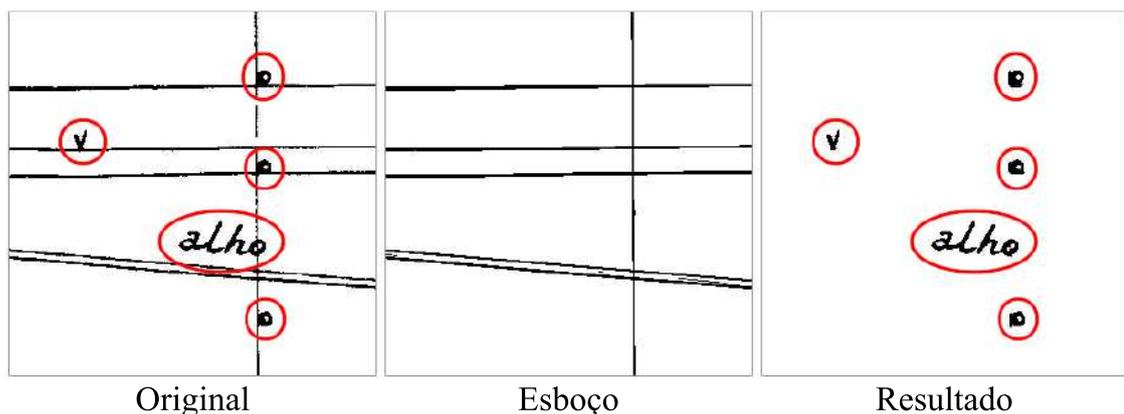
Figura 67 – Imagem sketch criada a partir da ideia de *smearing*.



Fonte: Marcos Canêjo (2022)

Com a imagem *sketch* produzida, o objetivo é utilizá-la como uma máscara para remover os traços sobrepostos, como mostra a Figura 68. Em seguida, com os textos isolados dos traços, o passo 6.2.2 é novamente aplicado, segmentando totalmente as informações.

Figura 68 – Resultado dos elementos textuais separados.



Fonte: Marcos Canêjo (2022)

6.3 RESULTADOS

Nesta seção são apresentados os resultados para o problema de segmentação de mapas e plantas baixas de forma qualitativa, apenas como uma motivação da aplicação da mode-

lagem do [SDC](#) em um outro domínio.

As plantas baixas utilizadas fazem parte do banco de dados CVC-FP ([HERAS et al., 2015b](#)) com 122 imagens, porém com características semelhantes, divididas em 4 categorias. Assim, após uma seleção em cada categoria, 10 imagens foram escolhidas de forma aleatória para os testes. Porém, observou-se um comportamento do algoritmo similar entre os resultados. Então, duas imagens são exploradas na seção seguinte.

As imagens de mapas fazem parte do banco de dados disponível por Weinman *et al*, juntamente com imagens obtidas pelos autores deste trabalho. Novamente, o algoritmo proposto apresentou um comportamento semelhante para diferentes imagens, então, novamente, duas imagens são exploradas na análise.

6.3.1 Segmentação de plantas baixas

As Figuras [69](#) e [70](#) apresentam dois exemplos obtidos como aplicação em um outro domínio da ideia do [SDC](#). Para cada imagem original, as imagens esboço e as versões finais contendo informações textuais e gráficas também são expostas.

Na Figura [69](#) a imagem *sketch* produzida possui uma semelhança com sua imagem original, porém sem os detalhes, característica principal na imagem de 10 segundos do [SDC](#). Além de uma falta de detalhes nos traços reproduzidos, todo o texto presente na imagem original foi completamente removido, correspondendo ao melhor cenário, uma vez que a imagem *sketch* é utilizada como uma máscara para isolar as informações, como já citado na seção [6.2.3](#). Porém, na Figura [70\(c\)](#), é possível observar que alguns elementos gráficos, como por exemplo traços referentes a partes do banheiro e portas, se mantiveram.

A Figura [70](#) apresenta outro resultado, mas agora para uma imagem com um grau de degradação. Exatamente devido à degradação do documento, a região com o carimbo se torna o pior cenário, pois, além da quantidade de informação textual e gráfica sobreposta, a tinta perdeu sua real cor. Na Figura [71\(d\)](#) fica mais claro observar como esta região ficou prejudicada na segmentação. Os pequenos pixels presentes não contêm nenhuma informação gráfica, sendo semelhantes a ruídos.

A Figura [64](#), apresentada anteriormente, é um exemplo claro de imagem de planta baixa contendo os problemas discutidos; informações textuais com tamanhos desiguais, em locais diversos etc. Diante disso, a Figura [71](#) apresenta a solução dos métodos de ([MACHADO; MELLO, pp. 1-8, 2015](#)) e o proposto para efeito de comparação. A quantidade de texto detectado pelo método proposto é evidente, principalmente nas que estão sobrepostas por linhas. A Figura [72](#) apresenta outro resultado obtido.

6.3.2 Segmentação de mapas

As imagens de mapas testadas possuem um maior grau de informações em relação às plantas baixas. Por exemplo, nas Figuras [73](#) e [74](#) é possível facilmente identificar a grande

Figura 69 – Exemplo do resultado obtido para uma (a) planta baixa, (b) sua imagem *sketch* motivada pela ideia do **SDC** e imagens segmentadas para as informações (c) textuais e (d) gráficas.



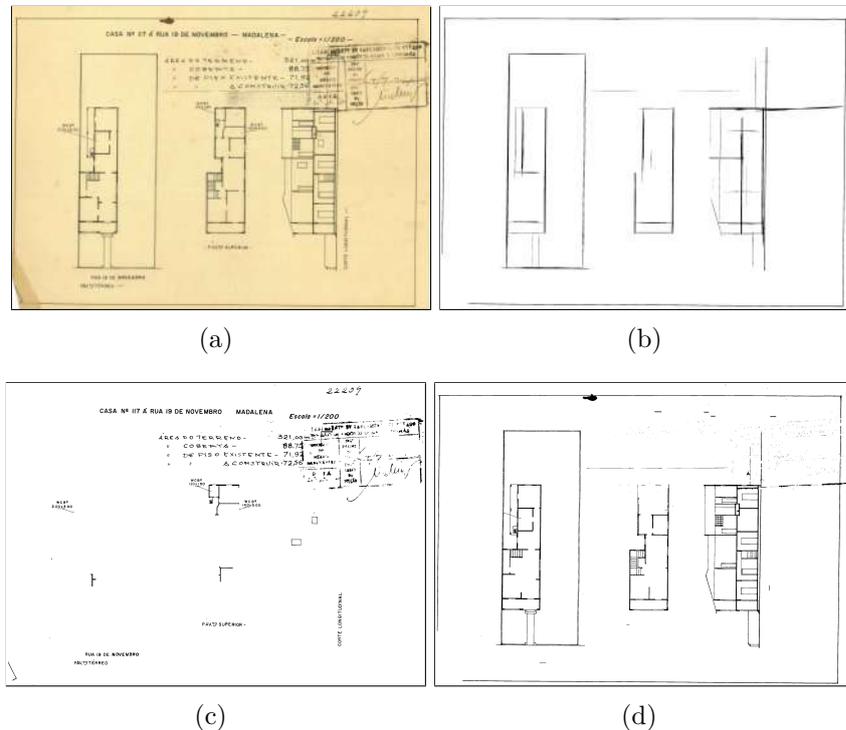
Fonte: Marcos Canêjo (2022)

quantidade de informação, dificultando ainda mais a segmentação. Por exemplo, na Figura 73 os traços representam demarcações de cidades, rios e vias férreas. Outro grande problema neste tipo de documento são linhas traçadas que podem ser facilmente confundidas por letras ou números. A escolha destas duas imagens para análise se deu justamente pela dificuldade na segmentação.

Como proposto, a Figura 74(b) representa a imagem de 10 segundos, porém as regiões do mapa com uma alta densidade de traços foram completamente ignoradas. Ainda assim, devido ao uso do esboço como máscara, outras informações textuais sobrepostas às gráficas puderam ser corretamente segmentadas, como mostra a Figura 74(c). Na Figura 74(d), é possível notar que grande parte do texto não recuperado foi ocasionado exatamente pela sobreposição da quantidade exagerada de traços existentes.

A Figura 74 é outro exemplo de um mapa com uma ampla variedade de traços, nomes e números, dispostos de diferentes formas. Na imagem esboço (Figura 75(b)) observa-se novamente que linhas pontilhadas foram representadas por traços contínuos e sem precisão. Além disso, os traços da região *LAKE ERIE* foram preservados na imagem que deveria conter apenas texto (Figura 75(c)), porém também verifica-se a variedade de texto detectado, da mesma maneira que a Figura 75(d) mantém as informações gráficas.

Figura 70 – Exemplo de segmentação do método proposto para (a) imagem de planta baixa degradada, (b) imagem *sketch* e resultados obtidos para as informações (c) textuais e (d) gráficas.



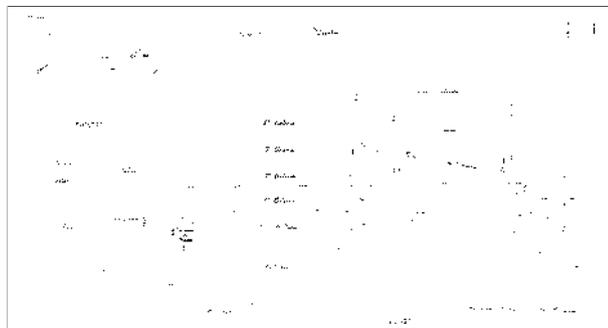
Fonte: Marcos Canêjo (2022)

Comparando o método proposto com a técnica de (MACHADO; MELLO, pp. 1-8, 2015), mas agora para imagens de mapas, novamente, a ideia do SDC, enfatiza os resultados alcançados. Na Figura 75 vários nomes de cidades, principalmente na região mais densa do mapa, foram recuperados. A Figura 76 apresenta outro resultado.

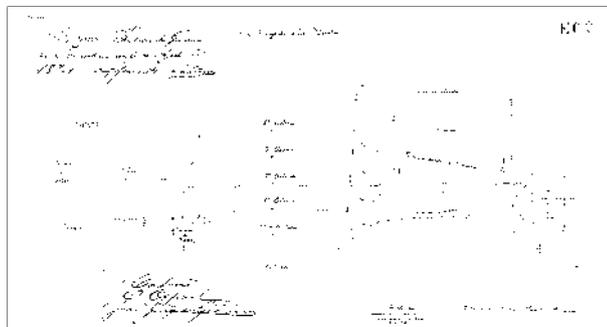
Imagens de plantas baixas e mapas possuem características semelhantes que tornam difícil a segmentação de suas informações textuais e gráficas. Embora a ideia baseada no SDC ter sido primariamente aplicada a detecção de contorno e borda, este capítulo apresenta uma nova aplicação em um outro domínio com resultados consideráveis para ambos os tipos de imagens sem a demanda de algum tipo de treinamento ou técnica de aprendizado.

Nos resultados apresentados é possível identificar um impasse no algoritmo na detecção de traços que representam o numeral um e delimitadores de regiões, por exemplo. Como trabalhos futuros, especificar um passo para diferenciar seu conteúdo se torna essencial para avanço na aplicação.

Figura 71 – Imagem final contendo apenas texto obtido pelo método de (b) (MACHADO; MELLO, pp. 1-8, 2015) comparado com o (b) proposto.



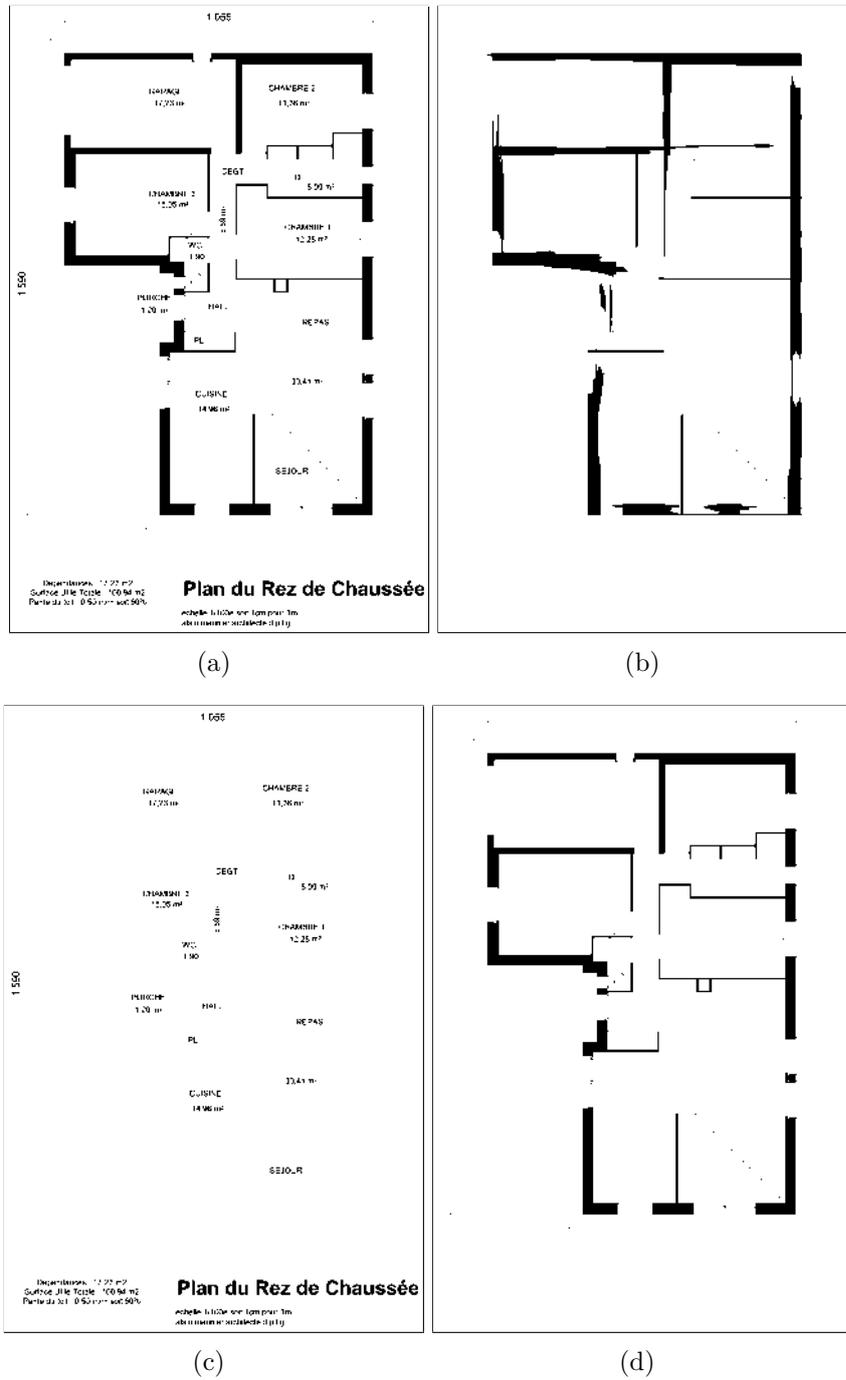
(a)



(b)

Fonte: Marcos Canêjo (2022)

Figura 72 – Exemplo de segmentação do método proposto para (a) imagem de planta baixa, (b) imagem *sketch* e resultados obtidos para as informações (c) textuais e (d) gráficas.

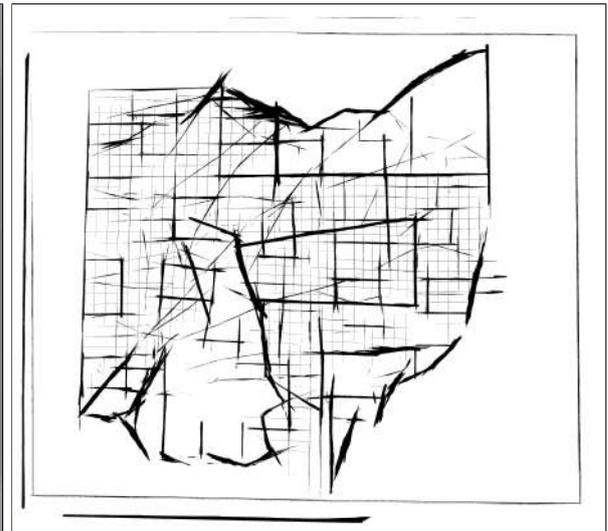


Fonte: Marcos Canêjo (2022)

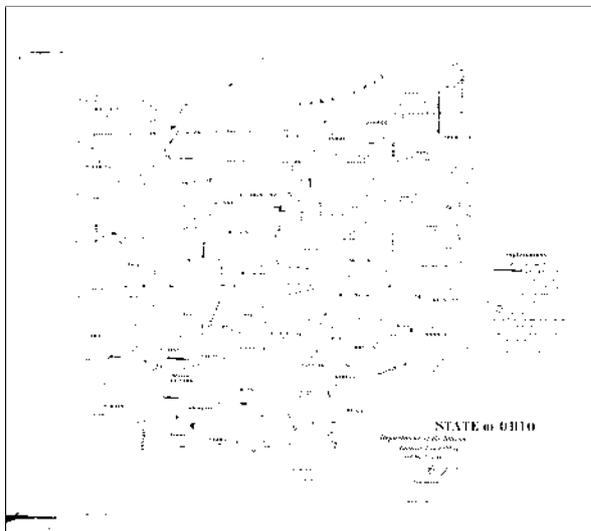
Figura 74 – Exemplo de mapa com muita informação gráfica e textual, sendo (a) mapa original e suas imagens (b) *sketch*, (c) textual e (d) gráfica.



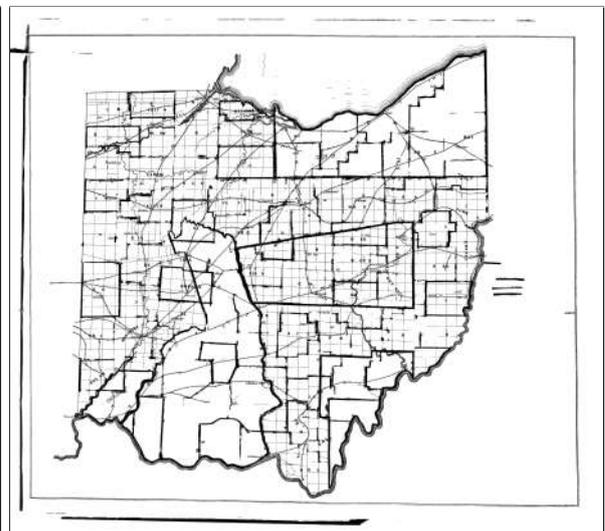
(a)



(b)



(c)



(d)

Fonte: Marcos Canêjo (2022)

Figura 75 – (a) Resultado da técnica concedida por (MACHADO; MELLO, pp. 1-8, 2015) relacionada com (b) o SDC proposto como solução em outro domínio.



(a)



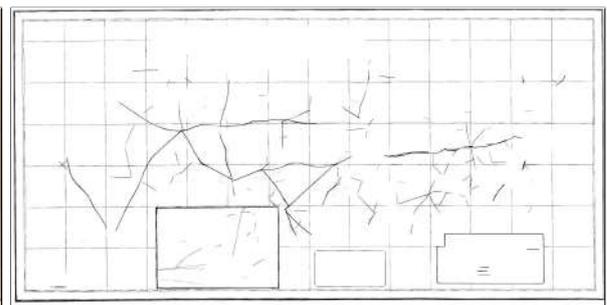
(b)

Fonte: Marcos Canêjo (2022)

Figura 76 – Segmentação de (a) um antigo mapa de Pernambuco, (b) imagem de 10 segundos e suas informações (c) textuais e (d) gráficas.



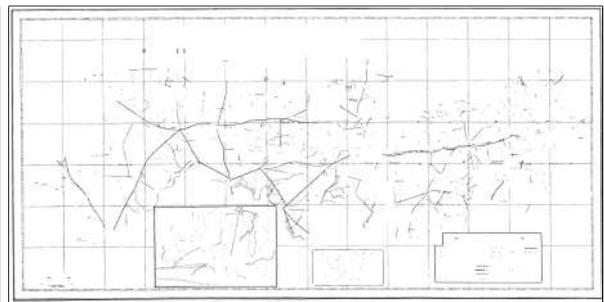
(a)



(b)



(c)



(d)

Fonte: Marcos Canêjo (2022)

7 CONCLUSÃO

Esta seção tem como objetivo apresentar as considerações finais sobre os principais tópicos abordados nesta tese, incluindo as contribuições alcançadas e indicações para trabalhos futuros.

7.1 CONSIDERAÇÕES FINAIS

Neste trabalho é proposto um método para detecção de bordas em imagens de cenas naturais. Por meio do ajuste do número de *superpixel* é possível a geração de imagens contendo apenas bordas com o principal objeto da cena ou imagens de borda semelhantes a algoritmos baseados em máscaras. Os principais passos do método proposto são: (i) geração de bordas pela supressão de detalhes mediante uma variação do algoritmo *Superpixel Hierarchy* usado na imagem original e (ii) análise de borda visando checar sua relação com o objeto principal da cena, removendo aquelas que não possuem relevância. Por fim, um passo de pós-processamento para recuperação de bordas erroneamente excluídas pelos passos anteriores.

é importante mencionar que o desempenho do algoritmo está totalmente relacionado ao tamanho da imagem de entrada, além dos valores de *superpixels* escolhidos como base para criação das imagens de 10 minutos, 1 minuto e 10 segundos. Isso significa que um tamanho diferente de imagem (maior ou menor que as testadas), terá um impacto direto nos resultados.

O método foi avaliado de forma quantitativa e qualitativa. Na análise quantitativa, a avaliação clássica não foi capaz de reproduzir os resultados obtidos. Por isso, a avaliação é feita com a remoção das bordas detectadas pelo método proposto em outras imagens de borda para expressar a sua relevância na cena. Ou seja, diferentemente do método clássico de análise, juntamente com a análise qualitativa, é possível determinar os resultados obtidos. O trabalho também apresenta uma aplicação para segmentação de mapas e plantas baixas focada na divisão entre as informações textuais e gráficas relatadas de maneiras distintas de forma quase única para cada documento analisado. Inicialmente a imagem de mapa ou planta baixa é submetida a uma etapa para remoção de marcas de degradação geradas pelo tempo, comumente encontradas em documentos antigos. Em seguida, considerações são realizadas a partir de valores estatísticos para divisão das informações em imagens distintas. Porém alguns textos sobrepostos ao desenho são erroneamente classificados permanecendo na imagem que deveria apenas conter informações gráficas. Então, a ideia de *smearing* é submetida, visando criar uma imagem esboço (inspirada na imagem de 10 segundos do *Speed Drawing Challenge*) para ser usada como uma máscara, efetuando a divisão de textos sobrepostos a linhas pertencentes ao desenho da planta baixa,

por exemplo. Por fim, os elementos separados são analisados para verificação de sua semelhança com os textos da imagem original, formando, finalmente, a imagem final. Uma análise qualitativa evidencia os resultados alcançados.

7.2 PRINCIPAIS CONTRIBUIÇÕES

O *Superpixel Hierarchy* original possui dois parâmetros: a imagem de borda, que guia o processo de agrupamento dos pixels e o número de *superpixels*. Como principais contribuições deste trabalho, podemos destacar o aprimoramento da construção da imagem de *superpixels* através de modificações em ambos os parâmetros.

No primeiro, o algoritmo original faz uso de uma única imagem de borda gerada pelo *Structured Forest Edges*. Visando aprimorar o processo, é proposta a utilização de 8 imagens de bordas orientadas criada pelo *Globalized Probability of Boundary*. Ou seja, para cada imagem de entrada, 8 novas imagens de *superpixel* são criadas para cada orientação de borda e em seguida são resumidas dando origem a imagem *sh_colored*. Além disso, como já descrito nesta Tese, o número de *superpixel*, se mal definido, pode acabar removendo o objeto principal da cena. Por isso, a determinação do número de *superpixels* foi automatizada através de técnicas de agrupamento, juntamente com métricas para verificar seu desempenho, evitando a exclusão.

Na submissão realizada ao *International Symposium on Visual Computing* (CANÊJO; MELLO, 2020), que compreende uma análise quantitativa e qualitativa dos mapas de saliência gerados por diferentes abordagens para a imagem *sh_colored*, notou-se uma melhora significativa na detecção do objeto principal da cena. A Figura 77 apresenta o desenvolvimento do mapa de saliência, sendo a Figura 78(d) a primeira versão do aprimoramento proposto apresentada na submissão e a Figura 78(f) a abordagem final. A Tabela 4 compara a forma antiga (CANÊJO; MELLO, 2020) da nova proposta neste trabalho.

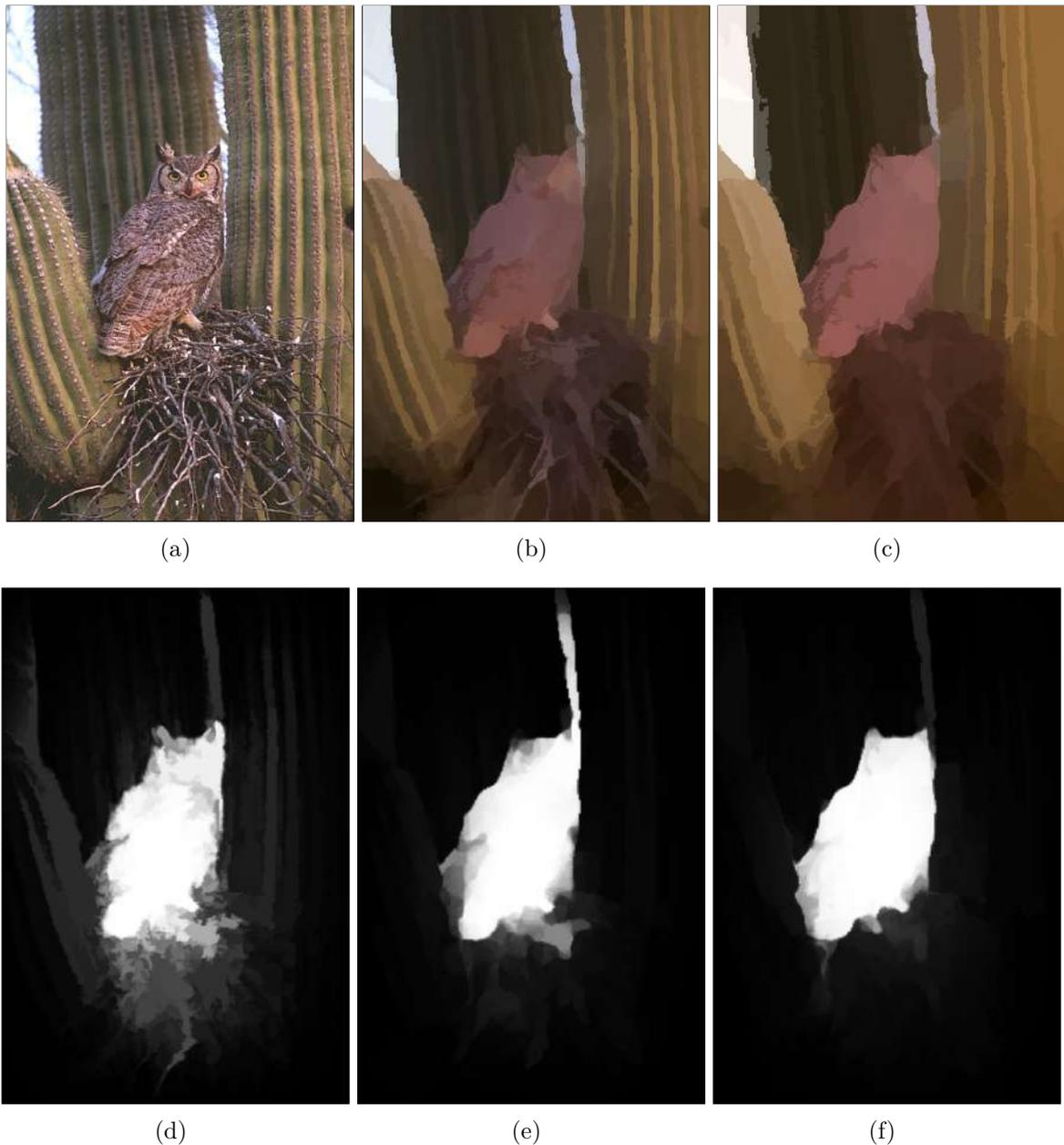
Tabela 4 – Valores de Weighted- F_β para a nova abordagem adotada para aprimoramento do *Superpixel Hierarchy* em relação a antiga.

	Antigo	Novo
MBS+	0,561	0,577
LPS	0,473	0,492
EQCUT	0,523	0,5457
RBD	0,547	0,5689

Fonte: Marcos Canêjo (2022)

Uma contribuição adicional da pesquisa se refere a uma observação sobre o método clássico de avaliação das imagens de contorno. Por exemplo, a Figura 78 demonstra resultados alcançados com valores baixos para o *benchmark*, porém a partir de uma inspeção visual é possível notar a diferença entre os traços detectados pelos diferentes métodos,

Figura 77 – (a) Imagem original, (b) imagem *sh_colored* definida na forma antiga e (c) a usada neste trabalho, junta com os (d-f) mapas de saliência, respectivamente.



Fonte: Marcos Canêjo (2022)

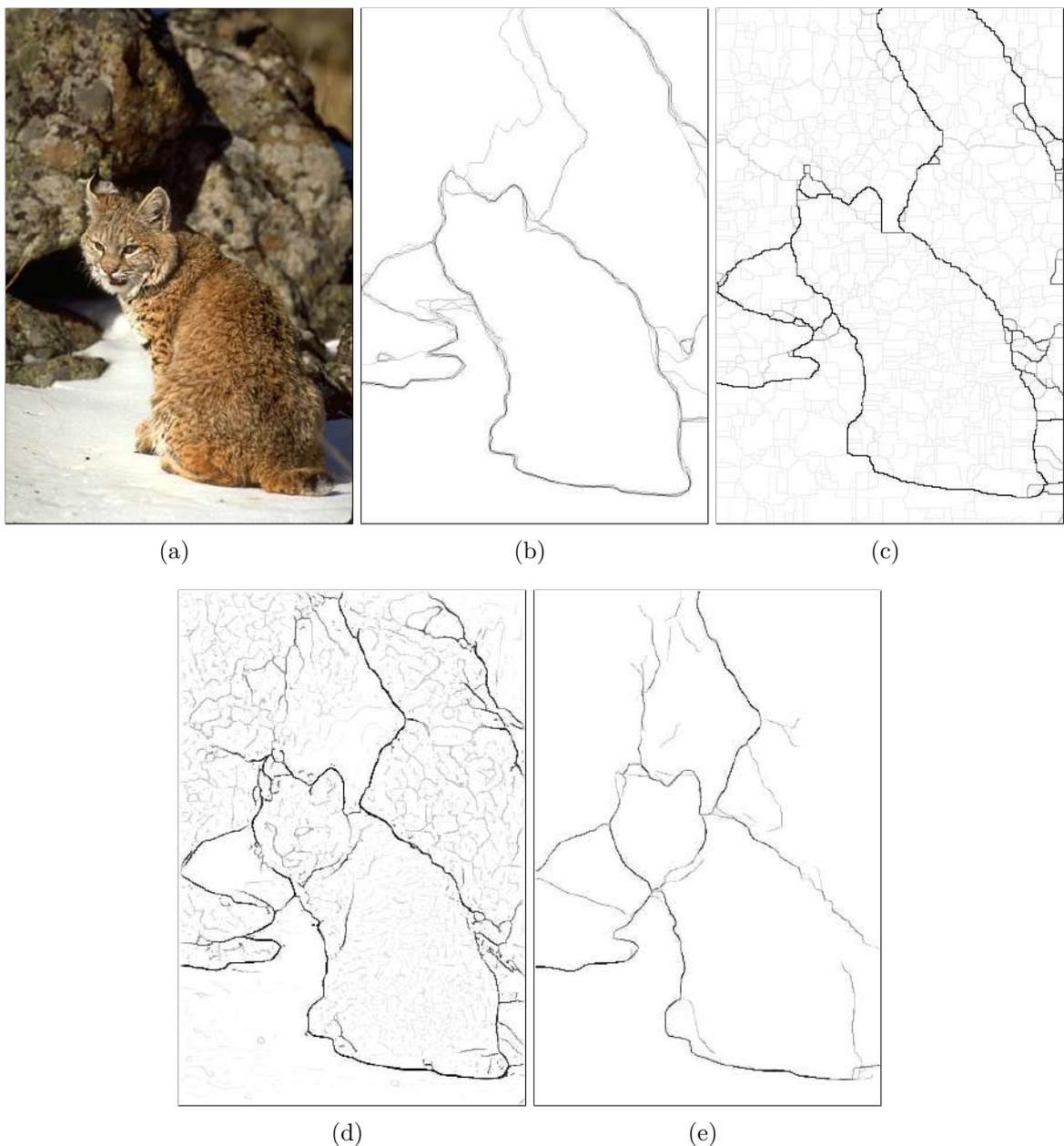
principalmente quando comparados com o SDC10s (Figura 79(e)). Os valores de *Optimal Image Scale* obtidos para os métodos COB, HED e SDC10s foram 0,924, 0,859 e 0,717, respectivamente, ou seja, o *benchmark* não é apreciado como a melhor referência sobre os resultados obtidos.

No Capítulo 7, a ideia do *Speed Drawing Challenge* foi aplicada para o domínio de segmentação de imagens de mapas e plantas baixas, ressaltando que o método proposto pode ser aplicável a outros domínios. Ademais, apesar de não ter sido explorado, a publicação (CANÊJO; MELLO, 2021) exemplifica o uso da ideia para o problema de *Non-photorealistic*

Rendering.

Para o problema de detecção de contorno e borda, o método baseado no *Speed Drawing Challenge*, descrito neste trabalho, apresenta resultados expressivos quanto a outras abordagens. Além de gerar imagens contendo a ideia dos tempos de 10 segundos, 1 minuto e 10 minutos, preservando bordas detectadas preliminares, qualquer outro “tempo” pode ser dado de forma simples por meio do fator de multiplicação, significando uma abrangência do método em representar traços relevantes semelhantes à capacidade humana em escolher os que mais representam uma cena natural, a traços associados a ideia de borda.

Figura 78 – (a) Imagem original, (b) padrão ouro e imagens de contorno obtidas pelos algoritmos (c) COB, (d) HED e (e) SDC10s.



Fonte: Marcos Canêjo (2022)

7.3 TRABALHOS FUTUROS

Para trabalhos futuros, um método baseado em redes neurais profundas deve ser explorado. Em particular, a *Generative Adversarial Network* que pode fazer uso dos resultados obtidos para a etapa de treinamento devido a sua forte capacidade de replicar o conjunto de treino para outras imagens, além das características inerentes ao método como *transfer learning*, *data augmentation*, etc. As imagens de 10 segundos, por exemplo, podem servir como dados sintéticos para o treinamento, uma vez que a obtenção em grande escala de imagens rotuladas por seres humanos é uma tarefa difícil.

As métricas **DB** e **DE** são utilizadas neste trabalho para avaliar a qualidade do agrupamento feito pelo **SH** na geração do valor de superpixel de forma automática, porém testes feitos de forma prévia indicam que apenas o uso da **DE** seja suficiente. Com isso, fica como trabalhos futuros a análise das métricas utilizadas, considerando apenas a **DE** como forma de avaliação. Além disso, para gerar as imagens de 1 e 10 minutos, um fator de multiplicação é utilizado definido de forma empírica. Assim, formas de melhor relacionar o fator de multiplicação com o tempo do desafio devem ser exploradas.

Domínio de aplicação inspirados na ideia do *Speed Drawing Challenge* são explorados neste trabalho, mas outros domínios, como o *Non-photorealistic Rendering*, podem adaptar-se como uma possível solução. Isto é, como acontece com o *Superpixel Hierarchy*, que usa uma imagem de borda para guiar o processo de agrupamento dos pixels, a imagem de *superpixel* e contorno gerada pelo método proposto pode corresponder a uma solução na renderização de fotos não realísticas. Ou seja, o objetivo é recriar imagens reais com algum estilo artístico como um desenho animado, por exemplo. A fusão dos resultados de diferentes etapas do método pode gerar imagens foto não realísticas, como apresenta a Figura 79, onde resultados preliminares são apresentados. Algumas abordagens são construídas a partir de informações visuais com poucos traços (MARKOSIAN et al., 1997).

Figura 79 – Dois exemplos de *Non-photorealistic Rendering* como possibilidade de aplicações futuras do trabalho.



Fonte: Marcos Canêjo (2022)

7.4 PUBLICAÇÕES

Como resultado desta tese, duas publicações foram realizadas:

- CANÊJO, M. J.; MELLO, C. A. Improvements on the superpixel hierarchy algorithm with applications to image segmentation and saliency detection. In: SPRINGER. International Symposium on Visual Computing. [S.l.], 2020. p. 182–193.
- CANÊJO, M. J.; MELLO, C. A. B. de. Edge detection in natural scenes inspired by the speed drawing challenge. International Journal of Image and Graphics, World Scientific, p. 2350009, 2021.

REFERÊNCIAS

- AHMED M. WEBER, M. L. S.; DENGEL, A. Text/Graphics Segmentation in Architectural Floor Plans. In: *11th International Conference on Document Analysis and Recognition*. [S.l.: s.n.], pp. 734–738, 2011.
- AKININA, A. V.; NIKIFOROV, M. B.; SAVIN, A. V. Multiscale image segmentation using normalized cuts in image recognition on satellite images. In: *2018 7th Mediterranean Conference on Embedded Computing (MECO)*. [S.l.: s.n.], 2018. p. 1–3.
- AKTER, L. A.; KWON, G. Integration of Contourlet Transform and Canny Edge Detector for Brain Image Segmentation. In: *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. [S.l.: s.n.], 2018. p. 798–800. ISSN 2165-8536.
- ARBELAEZ, P. Boundary extraction in natural images using ultrametric contour maps. In: *IEEE. Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*. [S.l.], 2006. p. 182–182.
- ARBELÁEZ, P.; MAIRE, M.; FOWLKES, C.; MALIK, J. From contours to regions: An empirical evaluation. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. p. 2294–2301. ISSN 1063-6919.
- ARBELÁEZ, P.; MAIRE, M.; FOWLKES, C.; MALIK, J. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 33, n. 5, p. 898–916, 2011.
- ART, M. S. *SPEED CHALLENGE: 10 Minutes | 1 Minute | 10 Seconds - Drawing Spiderman*. 2022. <https://vimeo.com/226508728>. Acessado em 19 de junho de 2022.
- BORGEFORS, G. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, Elsevier, v. 27, n. 3, p. 321–345, 1984.
- BORGEFORS, G. On digital distance transforms in three dimensions. *Computer Vision and Image Understanding*, Elsevier, v. 64, n. 3, p. 368–376, 1996.
- BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.
- CANÊJO, M. J.; MELLO, C. A. Improvements on the superpixel hierarchy algorithm with applications to image segmentation and saliency detection. In: *SPRINGER. International Symposium on Visual Computing*. [S.l.], 2020. p. 182–193.
- CANÊJO, M. J.; MELLO, C. A. B. de. Edge detection in natural scenes inspired by the speed drawing challenge. *International Journal of Image and Graphics*, World Scientific, p. 2350009, 2021.
- CANNY, J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, n. 6, p. 679–698, Novembro 1986. ISSN 0162-8828.
- CAO, H.; PRASAD, R.; NATARAJAN, P.; MACROSTIE, E. Robust page segmentation based on smearing and error correction unifying top-down and bottom-up approaches. In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. [S.l.: s.n.], 2007. v. 1, p. 392–396.

- CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K.; YUILLE, A. L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.
- CHEN, L.-C.; ZHU, Y.; PAPANDREOU, G.; SCHROFF, F.; ADAM, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*. [S.l.: s.n.], 2018. p. 801–818.
- CIESIELSKI, K. C.; STRAND, R.; MALMBERG, F.; SAHA, P. K. Efficient algorithm for finding the exact minimum barrier distance. *Computer Vision and Image Understanding*, Elsevier, v. 123, p. 53–64, 2014.
- CRAYE, C.; FILLIAT, D.; GOUDOU, J. Environment exploration for object-based visual saliency learning. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2016. p. 2303–2309.
- DANIELSSON, P.-E. Euclidean distance mapping. *Computer Graphics and Image Processing*, Elsevier, v. 14, n. 3, p. 227–248, 1980.
- DAVE, R.; VYAS, A.; DESAI, N. P. Face Recognition Techniques: A Survey. *CoRR*, abs/1803.07288, 2018. Withdrawn. Disponível em: <http://arxiv.org/abs/1803.07288>.
- DAVIES, D. L.; BOULDIN, D. W. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1, n. 2, p. 224–227, 1979.
- DAVIES, E. R. *Machine vision: theory, algorithms, practicalities*. [S.l.]: Elsevier, 2004.
- DHARAMPAL; MUTNEJA, V. Methods of Image Edge Detection: A Review. *Journal of Electrical & Electronic Systems*, OMICS International., v. 4, n. 2, p. 1–5, 2015. ISSN 2332-0796. Disponível em: <https://www.omicsonline.org/open-access/methods-of-image-edge-detection-a-review-2332-0796-1000150.php?aid=57249>.
- DICKINSON, S.; PIZLO, Z. *Shape perception in human and computer vision*. [S.l.]: Springer, 2013.
- DOLLÁR, P.; TU, Z.; PERONA, P.; BELONGIE, S. Integral channel features. BMVC Press, 2009.
- DOLLÁR, P.; ZITNICK, C. L. Structured forests for fast edge detection. In: *2013 IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2013. p. 1841–1848.
- EVERINGHAM, M.; GOOL, L. V.; WILLIAMS, C. K. I.; WINN, J.; ZISSERMAN, A. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, v. 88, n. 2, p. 303–338, Junho 2010.
- GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2014. p. 580–587.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 2nd. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001. ISBN 0201180758.
- HE, J.; ZHANG, S.; YANG, M.; SHAN, Y.; HUANG, T. Bdcn: Bi-directional cascade network for perceptual edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2020.

HERAS, L.-P. de las; TERRADES, O.; ROBLES, S.; S'ANCHEZ, G. Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool. *International Journal on Document Analysis and Recognition*, 2015.

HERAS, L.-P. de las; TERRADES, O.; ROBLES, S.; SANCHEZ, G. Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool. *International Journal on Document Analysis and Recognition*, 2015.

HILDEBRAND, T.; RÜEGSEGG, P. A new method for the model-independent assessment of thickness in three-dimensional images. *Journal of microscopy*, Wiley Online Library, v. 185, n. 1, p. 67–75, 1997.

INSTAGRAM. 2018.

<https://www.instagram.com/p/Bn4JmleAzNL/?igshid=1ugymfei0ez2o>.

Acessado em 17 de outubro de 2018.

J.ALMAZAN, A.; E.VALVENY. Deformable HOG-Based Shape Descriptor. In: *12th International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.: s.n.], pp. 1022–1026, 2013.

JERRIPOTHULA, K. R.; CAI, J.; LU, J.; YUAN, J. Object co-skeletonization with co-segmentation. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 3881–3889. ISSN 1063-6919.

JIANG, H.; WANG, J.; YUAN, Z.; WU, Y.; ZHENG, N.; LI, S. Salient object detection: A discriminative regional feature integration approach. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2013. p. 2083–2090.

KARAOGLU, S.; FERNANDO, B.; TREMEAU, A. A Novel Algorithm for Text Detection and Localization in Natural Scene Images. In: *2010 International Conference on Digital Image Computing: Techniques and Applications*. [S.l.: s.n.], 2010. p. 635–642.

KAUR, R. P.; JINDAL, M.; KUMAR, M. Text and graphics segmentation of newspapers printed in gurmukhi script: a hybrid approach. *The Visual Computer*, Springer, v. 37, n. 7, p. 1637–1659, 2021.

KELM, A. P.; RAO, V. S.; ZÖLZER, U. Object contour and edge detection with refinecontournet. In: SPRINGER. *International Conference on Computer Analysis of Images and Patterns*. [S.l.], 2019. p. 246–258.

KONTSCHIEDER, P.; BULO, S. R.; BISCHOF, H.; PELILLO, M. Structured class-labels in random forests for semantic image labelling. In: IEEE. *2011 international conference on computer vision*. [S.l.], 2011. p. 2190–2197.

LEVINSHTEIN, A.; SMINCHISESCU, C.; DICKINSON, S. Optimal contour closure by superpixel grouping. In: SPRINGER. *European Conference on computer vision*. [S.l.], 2010. p. 480–493.

LI, L.; ZHENG, W.; KONG, L.; OZGUNER, U.; HOU, W.; LIAN, J. Real-time Traffic Scene Segmentation Based on Multi-Feature Map and Deep Learning. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2018. p. 7–12. ISSN 1931-0587.

LIM, J.; ZITNICK, C.; DOLLAR, P. Sketch tokens: A learned mid-level representation for contour and object detection. In: . [S.l.: s.n.], 2013. p. 3158–3165.

- LIN, G.; MILAN, A.; SHEN, C.; REID, I. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 1925–1934.
- LIU, C.; SCHWING, A. G.; KUNDU, K.; URTASUN, R.; FIDLER, S. Rent3d: Floor-plan priors for monocular layout estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 3413–3421.
- LUKASHEVICH, M.; SADYKHOV, R. Texture analysis: Algorithm for texture features computation. In: *2012 IV International Conference “Problems of Cybernetics and Informatics” (PCI)*. [S.l.: s.n.], 2012. p. 1–3.
- MACHADO, S.; MELLO, C. Text segmentation in ancient topographic maps and floor plans with support vector data description. In: *International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], pp. 1–8, 2015.
- MAIRE, M.; ARBELAEZ, P.; FOWLKES, C.; MALIK, J. Using contours to detect and localize junctions in natural images. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2008. p. 1–8.
- MANINIS, K.-K.; PONT-TUSET, J.; ARBELAEZ, P.; GOOL, L. V. *Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks*. 2017.
- MARKOSIAN, L.; KOWALSKI, M. A.; GOLDSTEIN, D.; TRYCHIN, S. J.; HUGHES, J. F.; BOURDEV, L. D. Real-time nonphotorealistic rendering. In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. [S.l.: s.n.], 1997. p. 415–420.
- MARTIN, D.; FOWLKES, C.; MALIK, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 5, p. 530–549, 2004.
- MARTIN, D.; FOWLKES, C.; TAL, D.; MALIK, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. [S.l.: s.n.], 2001. v. 2, p. 416–423 vol.2.
- MARTIN, D.; FOWLKES, C.; TAL, D.; MALIK, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. [S.l.: s.n.], 2001. v. 2, p. 416–423 vol.2.
- MBD Oral Presentation. 2019. [Http://cs-people.bu.edu/jmzhang/fastmbd/MBD/oral/presentation.pptx](http://cs-people.bu.edu/jmzhang/fastmbd/MBD/oral/presentation.pptx). Acessado em 26 de agosto de 2019.
- MEHTA, S.; MERCAN, E.; BARTLETT, J.; WEAVER, D.; ELMORE, J.; SHAPIRO, L. Y-Net: Joint Segmentation and Classification for Diagnosis of Breast Biopsy Images. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2018.
- MEHTA, S.; RASTEGARI, M.; CASPI, A.; SHAPIRO, L.; HAJISHIRZI, H. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. *European Conference in Computer Vision (ECCV)*, 2018.

- MELLO, C.; MACHADO, S. Text segmentation in vintage floor plans and maps using visual perception. In: *International Conference on Systems, Man and Cybernetics (SMC)*. [S.l.: s.n.], pp. 3476–3480, 2014.
- MESQUITA, C. M. R.; V.ALMEIDA, L. E. A new thresholding algorithm for document images based on the perception of objects by distance. *Integrated Computer-Aided Engineering*, vol. 21, pp. 133–146, Novembro 2014.
- MUSTAFA, A.; HILTON, A. Semantically coherent co-segmentation and reconstruction of dynamic scenes. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 5583–5592. ISSN 1063-6919.
- NAJMAN, L.; SCHMITT, M. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 18, n. 12, p. 1163–1173, Dezembro 1996. ISSN 0162-8828.
- NOWOZIN, S.; LAMPERT, C. H. et al. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, Now Publishers, Inc., v. 6, n. 3–4, p. 185–365, 2011.
- PANDEY, P. S. K.; KULKARNI, R. Traffic sign detection for advanced driver assistance system. In: *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*. [S.l.: s.n.], 2018. p. 182–185.
- PREWITT, J. M. Object Enhancement and Extraction. *Picture processing and Psychopictorics*, v. 10, n. 1, p. 15–19, 1970.
- QUIROS, A. R. F.; BEDRUZ, R. A.; UY, A. C.; ABAD, A.; BANDALA, A.; DADIOS, E. P.; MANINGO, J. M.; VICERRA, R. R. Localization of license plates using optimized edge and contour detection technique. In: *TENCON 2017 - 2017 IEEE Region 10 Conference*. [S.l.: s.n.], 2017. p. 1075–1080.
- REN, X.; FOWLKES, C. C.; MALIK, J. Figure/ground assignment in natural images. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2006. p. 614–627.
- RIOMOROS, I.; GUIJARRO, M.; PAJARES, G.; HERRERA, P. J.; BURGOS-ARTIZZU, X. P.; RIBEIRO, A. Automatic image segmentation of greenness in crop fields. In: *2010 International Conference of Soft Computing and Pattern Recognition*. [S.l.: s.n.], 2010. p. 462–467.
- ROSENFELD, A.; PFALTZ, J. L. Distance functions on digital pictures. *Pattern recognition*, Elsevier, v. 1, n. 1, p. 33–61, 1968.
- ROSENHOLTZ, R. Search asymmetries? what search asymmetries? *Perception & Psychophysics*, Springer, v. 63, n. 3, p. 476–489, 2001.
- SAHA, P. K.; GAO, Z.; ALFORD, S. K.; SONKA, M.; HOFFMAN, E. A. Topomorphologic separation of fused isointensity objects via multiscale opening: Separating arteries and veins in 3-d pulmonary ct. *IEEE transactions on medical imaging*, IEEE, v. 29, n. 3, p. 840–851, 2010.
- SAVITZKY, A.; GOLAY, M. J. E. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, v. 36, p. 1627–1639, 1964.

SHEN, T.; LIN, G.; LIU, L.; SHEN, C.; REID, I. D. Weakly supervised semantic segmentation based on co-segmentation. *CoRR*, abs/1705.09052, 2017. Disponível em: <http://arxiv.org/abs/1705.09052>.

SHI, C.; WANG, C.; XIAO, B.; ZHANG, Y.; GAO, S. Scene Text Detection Using Graph Model Built Upon Maximally Stable Extremal Regions. *Pattern Recogn. Lett.*, Elsevier Science Inc., Nova York, NY, EUA, v. 34, n. 2, p. 107–116, Janeiro 2013. ISSN 0167-8655. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2012.09.019>.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

SOBEL, I. An Isotropic 3x3 Image Gradient Operator. *Presentation at Stanford A.I. Project 1968*, Fevereiro 2014.

SORIA, X.; RIBA, E.; SAPPA, A. D. *Dense Extreme Inception Network: Towards a Robust CNN Model for Edge Detection*. 2020.

SPEED CHALLENGE: 10 Minutes | 1 Minute | 10 Seconds - Drawing Spiderman. 2018. <https://vimeo.com/226508728>. Acessado em 17 de outubro de 2018.

STRAND, R.; CIESIELSKI, K. C.; MALMBERG, F.; SAHA, P. K. The minimum barrier distance. *Computer Vision and Image Understanding*, Elsevier, v. 117, n. 4, p. 429–437, 2013.

SZELISKI, R. *Computer vision: algorithms and applications*. [S.l.]: Springer Science & Business Media, 2010.

TAX, D. M.; DUIN, R. P. Support Vector Data Description. *Machine Learning*, v. 54, n. 1, p. 45–66, Janeiro 2004. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/B:MACH.0000008084.60811.49>.

WEI, X.; YANG, Q.; GONG, Y.; AHUJA, N.; YANG, M.-H. Superpixel hierarchy. *IEEE Transactions on Image Processing*, v. 27, n. 10, p. 4838–4849, 2018.

WEI, Y.; WEN, F.; ZHU, W.; SUN, J. Geodesic saliency using background priors. In: FITZGIBBON, A.; LAZEBNIK, S.; PERONA, P.; SATO, Y.; SCHMID, C. (Ed.). *Computer Vision – ECCV 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 29–42. ISBN 978-3-642-33712-3.

WEST, D. B. et al. *Introduction to graph theory*. [S.l.]: Prentice hall Upper Saddle River, NJ, 1996. v. 2.

WINDER, S.; HUA, G.; BROWN, M. Picking the best daisy. In: *Computer Vision and Pattern Recognition*. IEEE Computer Society, 2009. Disponível em: <https://www.microsoft.com/en-us/research/publication/picking-the-best-daisy/>.

WU, Y.; ZHAO, Z.; WU, W.; LIN, Y.; WANG, M. Automatic glioma segmentation based on adaptive superpixel. *BMC Medical Imaging*, BioMed Central, v. 19, n. 1, p. 1–14, 2019.

XIE, S.; TU, Z. Holistically-nested edge detection. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2015. p. 1395–1403.

- YANG, C.; ZHANG, L.; LU, H.; RUAN, X.; YANG, M. Saliency detection via graph-based manifold ranking. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2013. p. 3166–3173. ISSN 1063-6919.
- YANG, H.; MIN, K. Importance-based approach for rough drawings. *The Visual Computer*, v. 35, n. 4, p. 609–622, Abril 2019. ISSN 1432-2315.
- ZENG, Z.; LI, X.; YU, Y. K.; FU, C.-W. Deep floor plan recognition using a multi-task network with room-boundary-guided attention. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 9096–9104.
- ZHANG, J.; SCLAROFF, S. Saliency detection: A boolean map approach. In: *2013 IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2013. p. 153–160. ISSN 1550-5499.
- ZHANG, J.; SCLAROFF, S.; LIN, Z.; SHEN, X.; PRICE, B.; MECH, R. Minimum barrier salient object detection at 80 fps. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2015. p. 1404–1412.
- ZHAO, H.; SHI, J.; QI, X.; WANG, X.; JIA, J. Pyramid scene parsing network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 2881–2890.
- ZHU, W.; LIANG, S.; WEI, Y.; SUN, J. Saliency optimization from robust background detection. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 2814–2821. ISSN 1063-6919.
- ZÜND, F.; PRITCH, Y.; SORKINE-HORNUNG, A.; MANGOLD, S.; GROSS, T. Content-aware compression using saliency-driven image retargeting. In: *2013 IEEE International Conference on Image Processing*. [S.l.: s.n.], 2013. p. 1845–1849. ISSN 1522-4880.