



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

METODOLOGIA PARA EXTRAÇÃO DE CONTEÚDO DE MONITORES E TVs

Felipe de Souza Farias

Manaus – Amazonas

Junho de 2016

Felipe de Souza Farias

METODOLOGIA PARA EXTRAÇÃO DE CONTEÚDO DE MONITORES E TVs

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Orientador: Prof. D.Sc. Waldir Sabino da Silva Júnior

Co-orientador: Prof. D.Sc. Eddie Batista de Lima Filho

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

F224m Farias, Felipe de Souza
Metodologia para extração de conteúdo de monitores e TVs /
Felipe de Souza Farias. 2016
85 f.: il. color; 31 cm.

Orientador: Waldir Sabino da Silva Junior
Coorientador: Eddie Batista Lima Filho
Dissertação (Mestrado em Engenharia Elétrica) - Universidade
Federal do Amazonas.

1. processamento digital de imagens. 2. detecção de retângulos.
3. inspeção automática de TVs e monitores. 4. campo aleatório de
Markov. I. Silva Junior, Waldir Sabino da II. Universidade Federal
do Amazonas III. Título

Felipe de Souza Farias

METODOLOGIA PARA EXTRAÇÃO DE CONTEÚDO DE MONITORES E TVs

Banca Examinadora

Prof. D.Sc. Waldir Sabino da Silva Júnior – Presidente e Orientador
Departamento de Eletrônica e Computação – UFAM

Prof. D.Sc. Rogério Caetano
Fundação Paulo Feitoza

Prof. D.Sc. Celso Barbosa Carvalho
Departamento de Eletrônica e Computação – UFAM

Manaus – Amazonas

Junho de 2016

Agradecimentos

- Primeiramente, ao meu pai Carlos e minha mãe Maria Helena, pelo suporte incondicional.
- Às minhas irmãs, Fernanda e Carla, pela inspiração e pela disponibilidade em ajudar na revisão do trabalho.
- Aos meus orientadores, Waldir e Eddie, pelo exemplo, paciência, dedicação e companheirismo.
- Aos colegas do Programa de Pós-Graduação em Engenharia Elétrica, pelo auxílio na caminhada.
- Juscelino e Ismael, pela amizade de longa data. Terminem também pelo amor.
- Jasmine, pelo amor e pelo suporte que eu não sei colocar em palavras aqui.
- À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro.
- À Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) pelo apoio financeiro.
- Parte dos resultados apresentados neste trabalho foram obtidos através do Projeto de Pesquisa e formação de recursos humanos, em nível de graduação e pós-graduação, nas áreas de automação industrial, software para dispositivos móveis e TV Digital, financiado pela Samsung Eletrônica da Amazônia Ltda., no âmbito da Lei no. 8.387 (art. 2º)/91”.

Resumo da Dissertação apresentada à UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

METODOLOGIA PARA EXTRAÇÃO DE CONTEÚDO DE MONITORES E TVS

Felipe de Souza Farias

Orientadores: Waldir Sabino da Silva Júnior

Eddie Batista de Lima Filho

Programa: Pós-Graduação em Engenharia Elétrica

Neste trabalho, apresentamos uma metodologia de extração de conteúdo de TVs e monitores que utiliza um modelo baseado no campo aleatório de Markov (MRF). Duas contribuições foram feitas. Na primeira contribuição, modificamos um método de detecção de objetos quadrangulares em imagens coloridas. Isto é feito adaptando as técnicas de detecção de borda e de retângulos para a detecção e seleção de um único objeto retangular com características de uma tela de TV ou monitor. Além desta contribuição, concebemos duas bases de dados com, respectivamente, 504 e 600 imagens de TV/monitores adquiridas em diferentes resoluções, condições de iluminação e distância entre câmera e tela, assim como imagens de referência com o conteúdo apresentado nos aparelhos no momento da captura. O desempenho da metodologia foi avaliado em um contexto de detecção e avaliação de conteúdo de monitores, utilizando as duas bases de dados concebidas neste trabalho. Para comparação, utilizamos métodos de detecção de objetos retangulares existentes na literatura na mesma aplicação da metodologia proposta. Os experimentos demonstram que o desempenho da metodologia sofre grande influência da complexidade do conteúdo e do *background* da imagem.

Palavras-chave: processamento digital de imagens, detecção de retângulos, inspeção automática de TVs e campo aleatório de Markov.

Abstract of Dissertation presented to UFAM as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

METHODOLOGY FOR TV SCREENS AND MONITORS CONTENT
EXTRACTION

Felipe de Souza Farias

Advisors: Waldir Sabino da Silva Júnior

Eddie Batista de Lima Filho

Department: Postgraduate in Electrical Engineering

In this work we present a TV content extraction methodology using a model based on a Markov random field (MRF). We present two major contributions. For the first contribution, we modified a method for quadrangular object detection in color images, by means of adapting edge and rectangle detection techniques to detect and select a sole rectangular object with features of a TV or monitor screen. Besides this contribution, we conceived two databases consisted of, respectively, 504 and 600 TV and monitor photos, acquired under different sizes, different illumination conditions and different distance between camera and device, as well as reference images with the content presented in the devices in the acquisition moment. The methodology's performance was evaluated in the context of detection and evaluation of monitor content, using the databases conceived in this work. For comparison, we used existing methods for detecting rectangular objects in the same context of the proposed methodology. The experiments demonstrate that the methodology's performance is greatly influenced by the content complexity and the image background.

Keywords: digital image processing, rectangle detection, automatic TV inspection and Markov random field.

Sumário

1	Introdução	1
1.1	Caracterização do Problema	3
1.2	Contribuições do Trabalho	4
1.3	Objetivos	5
1.3.1	Objetivos Específicos	5
1.4	Organização da Dissertação	5
2	Fundamentos Teóricos	7
2.1	Processamento Digital de Imagens	7
2.2	Detecção de Bordas	8
2.3	Classificação e Métodos para Detecção de Borda	8
2.4	Detecção de Linhas Utilizando Transformada de <i>Hough</i>	9
2.5	Campo Aleatório de Markov (MRF)	12
2.6	Trabalhos Relacionados	13
3	Metodologia Proposta	18
3.1	Introdução	18
3.2	Metodologia de Extração de Conteúdo Proposta	20
3.3	Detecção de Bordas	21
3.3.1	Cálculo do Gradiente	21
3.3.2	Seleção de Candidatos	24
3.3.3	Verificação de Candidatos	24
3.4	Detecção de Linhas	25
3.4.1	Detecção de Linhas Utilizando a Transformada de <i>Hough</i>	26
3.4.2	Combinação e Descarte de Linhas	27

3.5	Detecção de Retângulos	28
3.5.1	Sistema de Vizinhanças Utilizando Campo Aleatório de Markov (MRF)	29
3.5.2	Algoritmo de Classificação de Linhas Utilizando Campo Aleatório de Markov (MRF)	31
3.5.3	Escolha do Retângulo que Representa a Tela	34
3.5.4	Formatação	35
3.6	Considerações Finais	37
4	Resultados Experimentais	38
4.1	<i>Setup</i> dos Experimentos	38
4.2	Bases de Dados	38
4.2.1	Base de Dados Sob Condições Não Controladas	39
4.2.2	Base de Dados Sob Condições Controladas	40
4.2.3	Rótulos Manuais das Bases de Dados	41
4.3	Métricas de Desempenho	42
4.4	Resultados Experimentais	44
4.4.1	Primeiro Experimento	44
4.4.2	Segundo Experimento	47
5	Conclusões	52
5.1	Trabalhos Futuros	53
	Referências Bibliográficas	55
A	Artigos Publicados	65
A.1	Artigos Diretamente Relacionados ao Tema	65
A.2	Outros Artigos	65
B	Imagens de Referência	67
B.1	Base de Dados Sob Condições Não-Controladas (<i>base 1</i>)	67
B.2	Base de Dados Sob Condições Controladas (<i>base 2</i>)	68

Lista de Figuras

2.1	Imagem com pontos colineares e sua representação no espaço m, b . Adaptado de [1].	11
2.2	Imagem com dois pontos e a representação paramétrica.	11
3.1	Fluxograma do método de extração de telas de TV e monitores.	20
3.2	Imagem e seu mapa de bordas.	21
3.3	Fluxograma da técnica de detecção de bordas.	22
3.4	Direção de gradiente de um pixel e seus vizinhos. Adaptado de [2].	24
3.5	Fluxograma da técnica de detecção de linhas.	26
3.6	Representação paramétrica de Hough. Adaptado de [3]	26
3.7	Fluxograma da técnica de detecção de retângulos.	29
3.8	Linhas paralelas e perpendiculares. Adaptado de [2].	31
3.9	Transformação da região quadrangular A'B'C'D' no retângulo ABCD. Adaptado de [4].	35
4.1	Imagens da base de dados sob condições não controladas.	40
4.2	Imagens da base de dados sob condições controladas.	41
4.3	Imagem rotulada manualmente.	42
4.4	Processo do NCC-BB para uma imagem adquirida e uma imagem de referência.	43
4.5	Outline do segundo experimento.	48
4.6	Imagens processadas com tela detectada. Nas imagens à esquerda a detecção foi considerada um sucesso. Nas imagens da direita a detecção a detecção não foi considerada satisfatória. Em vermelho temos a área delimitada pelo <i>rótulo manual</i> . Em verde temos a área delimitada pelo <i>rótulo automático</i>	50

B.1	imagem de referência - menu 1.	67
B.2	imagem de referência - menu 2.	67
B.3	imagem de referência - menu 3.	67
B.4	imagem de referência - menu 4.	67
B.5	imagem de referência - video 1.	68
B.6	imagem de referência - video 2.	68
B.7	imagem de referência - grid.	68
B.8	imagem de referência - xadrez.	68
B.9	imagem de referência - colorbar (20% saturação).	68
B.10	imagem de referência - colorbar (60% saturação).	68
B.11	imagem de referência - colorbar (100% saturação).	69
B.12	imagem de referência - vermelho (20% saturação).	69
B.13	imagem de referência - vermelho (60% saturação).	69
B.14	imagem de referência - vermelho (100% saturação).	69
B.15	imagem de referência - verde (20% saturação).	69
B.16	imagem de referência - verde (60% saturação).	69
B.17	imagem de referência - verde (100% saturação).	69
B.18	imagem de referência - azul (20% saturação).	69
B.19	imagem de referência - azul (60% saturação).	70
B.20	imagem de referência - azul (100% saturação).	70
B.21	imagem de referência - amarelo (20% saturação).	70
B.22	imagem de referência - amarelo (60% saturação).	70
B.23	imagem de referência - amarelo (100% saturação).	70
B.24	imagem de referência - ciano (20% saturação).	70
B.25	imagem de referência - ciano (60% saturação).	70
B.26	imagem de referência - ciano (100% saturação).	70
B.27	imagem de referência - magenta (20% saturação).	71
B.28	imagem de referência - magenta (60% saturação).	71
B.29	imagem de referência - magenta (100% saturação).	71
B.30	imagem de referência - foto (casa azul).	71
B.31	imagem de referência - foto (telhados).	71
B.32	imagem de referência - foto (mercadinho).	71

B.33 imagem de referência - foto (ponte).	71
B.34 imagem de referência - foto (barco).	71
B.35 imagem de referência - foto (terraço).	72
B.36 imagem de referência - foto (parquinho).	72
B.37 imagem de referência - foto (palácio).	72
B.38 imagem de referência - foto (jardim).	72
B.39 imagem de referência - foto (templo).	72
B.40 imagem de referência - foto (farinha).	72
B.41 imagem de referência - foto (árvores).	72
B.42 imagem de referência - índio.	72
B.43 imagem de referência - menu 1.	73
B.44 imagem de referência - menu 2.	73
B.45 imagem de referência - menu 3.	73
B.46 imagem de referência - menu 4.	73

Capítulo 1

Introdução

Processamento digital de imagens refere-se ao processamento de imagens através de um computador digital [5]. Este processamento pode ter diversos objetivos, como aprimoramento da imagem, menor uso de dados ou consumo por sistemas computacionais. Podemos encontrar aplicações de processamento digital de imagens em diversas áreas e linhas de pesquisa. Dentre estas aplicações, podemos citar a detecção de objetos retangulares [3,6], detecção de bordas [7,8] e filtragem para eliminação de ruído [9,10].

A detecção de objetos retangulares está entre os problemas mais importantes de processamento digital de imagens devido ao fato de muitos objetos feitos por mão humana terem formato retangular [2]. Entre as técnicas para detecção de retângulos em imagens, podemos citar as que envolvem variações da transformada de *Hough* [3,6], a análise de informações geométricas de linhas [11,12] e algoritmos genéticos [13,14].

Nesta dissertação, propomos uma metodologia de extração de conteúdo de TV e monitores que utiliza um modelo baseado no campo aleatório de Markov (MRF). Esta metodologia fornece uma alternativa para o problema de detecção de objetos quadrangulares em imagens coloridas. A metodologia proposta envolve diversas etapas, tais como: extração de bordas da imagem através de análise de gradiente multi-dimensional; detecção de segmentos de linha presentes no mapa de bordas da imagem utilizando a transformada de *Hough* e classificação de segmentos de linha que são borda de objetos retangulares usando um algoritmo baseado em um modelo MRF.

A metodologia proposta se difere de trabalhos presentes na literatura, como o proposto por Liu *et al.* [2] e o proposto por Kastelan *et al.* [4]. A diferença em relação ao trabalho de Liu *et al.* está no objeto detectado por cada método. Enquanto os pesquisadores buscam encontrar retângulos quaisquer em imagens, nesta dissertação buscamos encontrar a região da TV/monitor que apresenta conteúdo. Uma região retangular única, de orientação e relação de aspecto definidos, que permite que o sistema se especialize em detectar este tipo de região. O trabalho de Katona *et al.*, diferentemente da proposta desta dissertação, utiliza imagens capturadas sob condições extremamente controladas. Devido ao menor grau de controle na captura, a metodologia proposta nesta dissertação utiliza técnicas de detecção mais robustas que as empregadas pelos pesquisadores.

A fim de avaliar o desempenho da metodologia, foram realizados experimentos. Para este fim, foram concebidas duas bases de dados, uma sob condições controladas e outra não, compostas por imagens coloridas de TVs e monitores, e imagens de referência, que são o conteúdo mostrado nas TVs/monitores. A base de dados sob condições não controladas é composta por 6 imagens de referência e 504 imagens adquiridas, retratando 14 aparelhos capturados sob condições diversas de iluminação, distância entre câmera e tela e em resoluções variando de 2048×1152 a 5312×2988 pixels. A base de dados sob condições controladas é composta por 40 imagens de referência e 600 imagens adquiridas, retratando 15 monitores capturados sob condições controladas de ambiente, iluminação, distância entre câmera e tela, na resolução de 5312×2988 pixels. As imagens foram rotuladas manualmente com as coordenadas dos quatro vértices da região que corresponde ao conteúdo apresentado efetivamente pelo aparelho (denominado popularmente como *tela*). Dois experimentos foram executados. No primeiro experimento, avaliamos o desempenho da metodologia em detectar um retângulo considerado correto em cada uma das imagens das bases de dados. No segundo experimento, utilizamos a metodologia em conjunto com métodos de comparação de imagens para classificar imagens segundo o conteúdo.

Os experimentos realizados mostraram que a metodologia, assim como técnicas existentes na literatura [3], têm uma tarefa desafiadora em extrair conteúdo das bases de dados concebidas, com melhor desempenho em imagens cujo *background*

tenha alto contraste em relação ao conteúdo e cujo conteúdo seja formado por imagens simples. Entretanto, as bases de dados concebidas apresentam um desafio a quaisquer metodologias de detecção de conteúdo, como pode-se ver pelos resultados apresentados tanto pela metodologia proposta neste trabalho quanto pelas metodologias existentes.

1.1 Caracterização do Problema

A detecção de falhas em TVs e monitores é um problema abordado por diversos trabalhos e alguns destes utilizam técnicas de campos multidisciplinares, como processamento de sinais, engenharia de software, inteligência artificial e reconhecimento de padrões. Por exemplo, no sistema mostrado em [15], os autores propõem encontrar erros no software utilizado no aparelho, tais como *deadlocks*, uma situação em que duas ou mais tarefas são suspensas devido a aguardar um sinal de aprovação, vazamentos de memória, situação em que uma parte da memória é alocada mas nunca liberada, e problemas na alocação de tarefas. Outro exemplo é apresentado em [16], que usa um sinal de referência misturado com um sinal de erro através de software para encontrar problemas na interação entre hardware e software. Alguns trabalhos de detecção de falhas propõem um sistema de inspeção automática baseado na análise digital da imagem [4, 17–20]. Estes sistemas fornecem um diagnóstico da TV ou monitor inspecionados através da análise da informação de saída da tela. Esta informação permite encontrar falhas de software ou hardware que gerem defeitos na imagem final.

Dentre os métodos de inspeção automática de TVs baseados em análise digital de imagem, destacamos os métodos em que a captura da imagem é feita através de uma câmera [4, 21]. Nestes métodos, é capturada uma imagem do monitor inspecionado funcionando. Devido à variação de tamanho, posição e conteúdo presente na tela dos aparelhos de TV e monitores na ocasião da captura, é necessário um método robusto a tamanho, orientação e cor para detecção de tela e extração do conteúdo da mesma na imagem para o sucesso da inspeção.

A detecção de objetos retangulares é um problema antigo do processamento de imagens e reconhecimento de padrões, encarado por diversas metodologias, como

a transformada de *Hough*, análise de informações de linha e até utilizando algoritmos genéticos. Por exemplo, em [6] os autores desenvolvem uma variação da transformada de *Hough* chamada Transformada Retangular de *Hough* (RHT), que detecta diretamente formas retangulares em uma imagem binária. Esta transformada é utilizada em um sistema de detecção de partículas em imagens microscópicas. Outro exemplo é encontrado em [11], onde segmentos de linha são encontrados e agrupados de forma que a técnica consiga descobrir se fazem parte ou não de objetos retangulares.

Roth *et al.* demonstrou que pode-se encarar o problema de detecção de retângulos como um problema de otimização [13]. Assim, foram desenvolvidas implementações da detecção de retângulos baseadas em algoritmos genéticos, como por exemplo em [22], onde os autores utilizam um algoritmo genético eficiente para encontrar primitivas geométricas complexas. Uma aplicação derivada do uso de algoritmos genéticos é encontrada em [2], onde os autores combinam um modelo MRF a um algoritmo gradiente descendente para identificar linhas pertencentes a um objeto retangular.

1.2 Contribuições do Trabalho

Métodos de inspeção automática de TVs utilizam técnicas simples para extrair conteúdo dos aparelhos, por isso capturam imagens sob condições extremamente controladas [4]. A principal contribuição desta dissertação é a concepção de uma metodologia de extração de conteúdo de aparelhos em imagens coloridas baseada no modelo MRF, capaz de extrair conteúdo de imagens sob condições não controladas. Assim, variações no *background*, tamanho do aparelho ou leves variações na orientação do mesmo não afetam de forma significativa a extração. Para isso, modificamos técnicas de detecção de telas inicialmente utilizadas por Liu *et al.*, adaptando-as com informações específicas de telas de TV/monitores e com características das imagens alvo, assim como técnicas de pré-processamento para detectar e selecionar de uma única região retangular que apresenta o conteúdo do aparelho retratado na imagem.

Além desta contribuição, concebemos duas bases de dados com imagens co-

loridas de TVs e monitores funcionais, assim como imagens de referência com os conteúdos apresentados nos aparelhos. Uma base é composta por 504 imagens de TVs e monitores, capturadas sob condições não controladas de iluminação, posição da câmera e fundo, juntamente com 6 imagens de referência. A segunda base é formada por 600 imagens de TVs e monitores, capturadas em um ambiente com iluminação controlada e posição de câmera e objeto fixos, assim como 40 imagens representando o conteúdo mostrado em tela. Nas duas bases de dados, as imagens adquiridas foram rotuladas manualmente com coordenadas dos pontos que delimitam a tela dos aparelhos.

1.3 Objetivos

O objetivo principal desta dissertação é investigar uma metodologia para extração de conteúdo de TV e monitores usando um modelo baseado no campo aleatório de Markov.

1.3.1 Objetivos Específicos

- Investigar o desempenho da metodologia de extração de conteúdo de TV e monitores sob diferentes condições de iluminação, pose, distância e ambiente.
- Criar duas bases de dados com imagens de TVs e monitores.
- Realizar experimentos para validar a metodologia, utilizando as bases de dados concebidas.

1.4 Organização da Dissertação

Este trabalho será organizado conforme segue:

- No Capítulo 2, revisamos os fundamentos teóricos utilizados neste trabalho. Primeiramente, os conceitos relacionados a processamento digital de imagens são abordados. Após isto, descrevemos em termos gerais alguns métodos populares de detecção de bordas, assim como a detecção de linhas através da

transformada de *Hough* e o campo aleatório de Markov (MRF). Por fim, revisamos alguns trabalhos relacionados com o tema deste trabalho.

- No Capítulo 3, apresentamos a primeira contribuição deste trabalho, que é uma metodologia para detecção e extração de telas de TV e monitores. Primeiramente, contextualizamos o problema de detecção de objetos quadrangulares e de telas de TV. Após isto, descrevemos as técnicas utilizadas nesta metodologia para detecção de borda, detecção de linhas, detecção de retângulos e escolha do retângulo que representa a tela.
- No Capítulo 4, apresentamos os experimentos feitos para validar a metodologia e detalhamos as duas bases de dados concebidas, a segunda contribuição deste trabalho. Mostramos os procedimentos experimentais dos dois experimentos feitos e, por fim, discorremos sobre os resultados encontrados.
- No Capítulo 5, apresentamos as considerações finais do trabalho e sugestões de trabalhos futuros.

Capítulo 2

Fundamentos Teóricos

Neste capítulo, faremos uma revisão dos conceitos e métodos utilizados nesta dissertação. O capítulo é organizado da seguinte forma: inicialmente, apresentamos os conceitos relacionados a técnicas de processamento de sinais utilizados neste trabalho. A seguir, descrevemos métodos populares de processamento digital de imagens que envolvem estas técnicas. Por último, revisamos alguns trabalhos relacionados com o tema da dissertação.

2.1 Processamento Digital de Imagens

Podemos definir uma imagem como uma função $I(x, y)$, onde x e y são coordenadas espaciais, que implica em um valor ou conjunto de valores para cada coordenada. Uma imagem é dita monocromática quando os valores de $I(x, y)$ são escalares, que representam a *intensidade* ou *nível de cinza* da imagem em cada coordenada. Em uma imagem colorida, os valores de $I(x, y)$ são vetores que representam as cores da imagem naquela coordenada. A forma e o tamanho do vetor variam conforme a representação utilizada. A imagem é considerada digital quando tanto x quanto y e os valores de $I(x, y)$ são finitos e discretos. Uma imagem digital é formada por um número finito de elementos, cada um com uma localização no espaço x, y e um valor atribuído. Cada elemento é chamado *pixel* [5].

Processamento digital de imagens refere-se à manipulação ou extração de dados de imagens por meio de um computador digital. É uma área ampla, que se sobrepõe a diversas outras áreas, tais como aprendizado de máquina, visão com-

putacional e reconhecimento de padrões. Embora não seja consenso, as técnicas de processamento digital de imagens geralmente são divididas em três categorias, conforme exposto a seguir:

- **baixo nível:** estas técnicas são caracterizadas por ter imagens como entrada e saída. Entre elas podemos citar controle de contraste [23] e filtragem para diminuir ruído [9] ou para destacar bordas [24];
- **médio nível:** estas técnicas têm como entrada imagens e como saída atributos extraídos das imagens, como bordas, linhas e regiões. Dentre estas técnicas podemos destacar a detecção de bordas [7], detecção de formas [3] e segmentação [25];
- **alto nível:** estas técnicas buscam analisar os atributos das imagens de forma a desempenhar funções cognitivas normalmente associadas à visão, como detecção de pose [26,27] e de ações humanas [28];

2.2 Detecção de Bordas

O termo *borda* refere-se à mudança abrupta ou descontinuidade em uma ou mais propriedades de sinais bidimensionais, como imagens, entre áreas vizinhas [29]. Exemplos de propriedades das imagens são a intensidade, no caso de imagens em preto e branco, e os valores de R, G e B no caso de imagens coloridas. As descontinuidades devem ser destacadas pois podem corresponder à borda do objeto retratado, bem como mudanças na sua profundidade, iluminação ou textura [30]. A detecção de bordas busca, portanto, detectar onde ocorre uma descontinuidade.

Em processamento de imagens, a detecção de bordas possui diversas aplicações, como por exemplo detecção de tumores no cérebro [31], estudos sobre o comportamento dos pombos [32] e navegação espacial [33].

2.3 Classificação e Métodos para Detecção de Borda

A maioria dos métodos de detecção de bordas em imagens descritos na literatura pode ser dividida em dois grupos principais, conforme a informação que extraem da imagem para a detecção [34]:

- **gradiente**, ou primeira derivada: As técnicas que pertencem a este grupo encontram as bordas baseadas na localização de máximos e mínimos na primeira derivada das imagens. Entre seus exemplos mais utilizados estão o detector desenvolvido por Canny [35], o baseado no operador de Prewitt [36], o operador de Sobel [37] e o de Roberts [38].

Este grupo de métodos se aproveita do fato que uma diferença aguda no valor de uma função, no caso de imagens uma borda, gera um valor muito alto ou muito baixo em sua primeira derivada. Os detectores de primeira ordem fornecem o gradiente através do cálculo da convolução da imagem com operadores. As vantagens dos métodos que utilizam a primeira derivada são a simplicidade, possibilidade de obter o ângulo do gradiente, mas têm a desvantagem de ser suscetíveis a ruído [39], portanto fazendo necessário um esquema de suavização das imagens.

- **Laplaciano**, ou segunda derivada: Os métodos do segundo grupo encontram as bordas procurando onde a segunda derivada da imagem passa por zero. Dentre os detectores este grupo podemos destacar o operador de Marr-Hildreth [40] e Haralick [41].

Este segundo grupo de métodos se aproveita do fato que uma borda em uma imagem gera uma passagem pelo valor zero em sua segunda derivada. Assim, os operadores utilizados neste tipo de detector visam encontrar onde a segunda derivada das imagens passa por zero.

Dentre as vantagens deste grupo de detectores está a invariância à direção da borda, mas apresenta como desvantagem uma sensibilidade a ruído ainda maior do que nos detectores de primeira ordem [39].

2.4 Detecção de Linhas Utilizando Transformada de *Hough*

O termo *segmento de linha*, em processamento de imagens, refere-se a um conjunto de pontos colineares e conectados pertencentes a uma imagem [42]. Apesar de segmentos de linha serem essencialmente bordas, esta restrição geométrica permite

que detectores de linha tenham funcionamento diferente do de detectores de borda. Na literatura, podem ser encontrados muitos trabalhos dedicados ao problema da detecção de linhas. Entre os trabalhos mais influentes neste tema, podemos citar os métodos de detecção de linhas que utilizam a transformada de *Radon* [43, 44] e modelos de contorno ativos [45, 46].

Entretanto, desde que surgiu, a transformada de *Hough* tem se tornado um dos processos mais utilizados não só para detectar linhas, mas no processamento de imagens da atualidade [47]. Esta transformada, mais tarde descoberta como um caso especial da transformada de *Radon* [48], deu origem a inúmeras variantes que visam resolver problemas específicos com desempenho melhor que a transformada original [49–52].

A transformada de *Hough* teve por objetivo, inicialmente, detectar padrões complexos em imagens binárias, como elipses e retângulos. Para atingir este objetivo, são determinados parâmetros característicos destes padrões. Assim, o problema de detectar um padrão em uma imagem binária é transformado no problema de detectar um valor no espaço paramétrico [1].

No caso específico de detecção de segmentos de linha, os parâmetros utilizados por Hough para caracterizar uma linha no espaço x, y são a inclinação m e interseção b , portanto a linha $y = mx + b$ é representada como um ponto no espaço m, b . A detecção de uma linha l , em uma imagem, considerando um conjunto de n pontos igual a $N = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, é obtida conforme segue: primeiramente, para cada linha que contém um dos pontos é computado um voto no espaço paramétrico m, b . Assim, a linha com maior número de votos é considerada a linha a qual os pontos pertencem. Na Figura 2.1, apresentamos um exemplo de representação de pontos colineares no espaço paramétrico.

Considerando que a parametrização no espaço m, b não permite representar corretamente linhas verticais ou próximas desta inclinação, os pesquisadores Duda e Hart [53] propuseram uma nova parametrização para a transformada de *Hough*. Nesta parametrização, as retas no espaço x, y são representadas no espaço (ρ, θ) , da seguinte maneira:

$$\rho = x \cos \theta + y \sin \theta \quad (2.1)$$

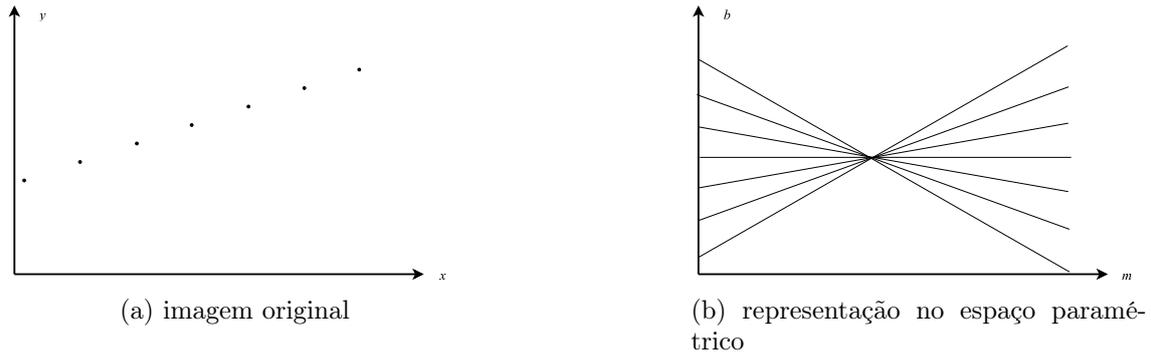


Figura 2.1: Imagem com pontos colineares e sua representação no espaço m, b . Adaptado de [1].

Na equação (2.1), x e y são as coordenadas de um segmento de reta no espaço x, y , ρ é a distância entre a origem da imagem e a linha, pelo seu ponto mais próximo, e θ é o ângulo que a normal à reta faz com o eixo horizontal. Assim, dado um pixel de borda, para cada possível reta que passe por ele, é computado um ponto no espaço (ρ, θ) . O conjunto de linhas possíveis passando por um determinado ponto forma um senoide naquele espaço. Com dois ou mais pontos na imagem pertencendo à mesma linha, as sinusoides que os representam se tocam justamente no ponto que representa a mesma. Por exemplo, na Figura 2.2 apresentamos uma imagem com dois pontos brancos e sua representação no espaço de Hough.

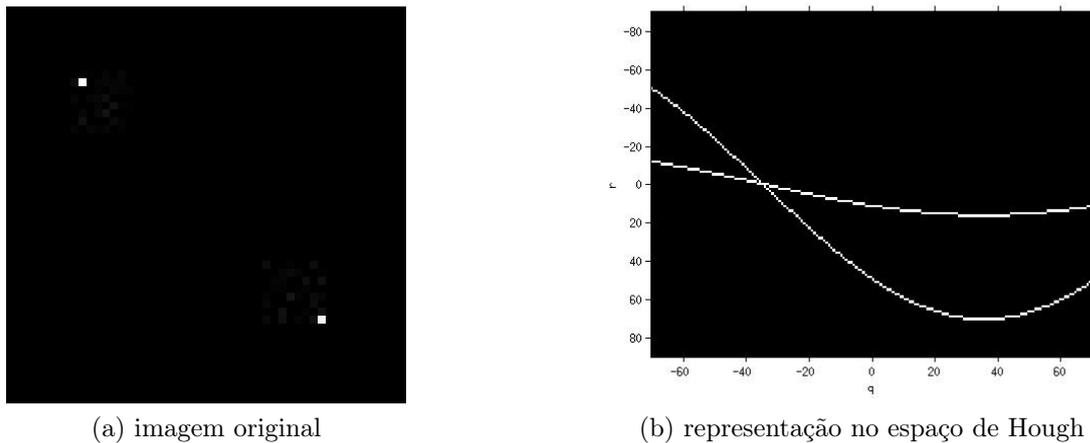


Figura 2.2: Imagem com dois pontos e a representação paramétrica.

2.5 Campo Aleatório de Markov (MRF)

Na literatura, o campo aleatório de Markov (MRF) tem sido muito utilizado em tarefas de processamento digital de imagens [54], como segmentação de imagens [55], restauração de imagens [56] e aprimoramento de imagens [57].

Um Campo Aleatório de Markov (MRF) é um processo estocástico que apresenta propriedade Markoviana. A definição desta propriedade é que o estado futuro de um processo pode ser previsto sabendo apenas o estado atual, ignorando seus estados anteriores [29]. O MRF é utilizado em processamento digital de imagens devido a conseguir estimar a probabilidade de propriedades na imagem toda a partir de uma pequena vizinhança [54].

Dado um conjunto de variáveis aleatórias $F = (F_1, \dots, F_n)$ definidas no conjunto S , em que cada variável F_i tem um valor f_i é representado como $F_i = f_i$. O evento conjunto em que o cada uma das variáveis de F apresenta um conjunto de valores $f = \{f_1, f_2, \dots, f_n\}$ é representado como $(F_1 = f_1, \dots, F_n = f_n)$, ou $F = f$. O conjunto de valores f é chamado de *configuração* de F . A probabilidade deste evento conjunto é representada por $P(F = f)$.

O conjunto F é um MRF se a probabilidade de qualquer evento referente a F depende de sua vizinhança. A vizinhança é representada por $N = \{N(F_i) | \forall i \in d\}$, onde $N(F_i)$ é a vizinhança da variável aleatória F_i e d é o conjunto de índices de F . Portanto, F é considerado um MRF em relação à vizinhança N , se obedece à seguinte condição [29]:

$$P(f_i | f_{S-\{i\}}) = 0, P(f_i | f_{N_{f_i}}) \quad (2.2)$$

Na equação (2.2), $P(f_{S-\{i\}})$ refere-se à probabilidade de todos os elementos de S , menos f_i .

A teoria do MRF permite modelar a probabilidade *a priori* de elementos relacionados entre si. Em conjunto com teorias de estimação, é formulada uma função objetiva para estimar os valores das variáveis aleatórias. Esta função pode ser encaixada em um algoritmo devido a um teorema que equivale o MRF a uma distribuição de Gibbs [58]. Devido a esta equivalência, a probabilidade de um campo aleatório de Markov pode ser calculada seguindo esta distribuição, matematicamente

expressa pela seguinte equação:

$$P(F = f|L) = Z^{-1} \exp \left[\frac{-E(f)}{T} \right] \quad (2.3)$$

Na equação (2.3), T é a temperatura, Z é o fator de normalização e $E(f)$ é a função de energia posterior $E(f) = \frac{U(f)}{T} + U(l|f)$, onde $U(f)$ e $U(l|f)$ são respectivamente a energia anterior e a energia de aproximação.

2.6 Trabalhos Relacionados

Os pesquisadores Liu *et al.* [2] propuseram um sistema de detecção de objetos retangulares em imagens coloridas que combina um método de detecção de linhas com um modelo baseado no campo aleatório de Markov (MRF). O sistema pode ser descrito conforme segue: primeiramente, é utilizado um algoritmo de detecção de bordas em imagens coloridas através de análise de gradiente multidimensional. A seguir, são extraídos segmentos de linha do mapa de bordas através de uma versão da transformada de *Hough* que utiliza a informação de ângulo de gradiente. Segmentos vizinhos e paralelos são combinados e os segmentos resultantes que estão na borda de objetos retangulares são rotulados através de um modelo MRF. Os experimentos utilizaram uma base de dados, que não foi disponibilizada para a comunidade científica, formada por 513 imagens coloridas no formato RGB, com resolução variando entre 73×42 e 3072×2048 pixels. O número, o tamanho, cor e a orientação dos objetos retangulares variava a cada imagem. Os experimentos indicam que o sistema cumpre satisfatoriamente a tarefa de encontrar múltiplos objetos retangulares em imagens coloridas, independente de cor, orientação e tamanho dos objetos.

O trabalho proposto por Jung e Schramm [3] apresenta uma metodologia para detectar objetos retangulares em imagens monocromáticas que utiliza uma variação da transformada de *Hough* chamada *Windowed Hough Transform*. A metodologia pode ser descrita resumidamente nos seguintes passos: inicialmente, é considerado um pixel (x_0, y_0) de uma imagem e uma região circular vizinha a este pixel. Esta região deve grande o suficiente para conter as bordas de um retângulo com centro em (x_0, y_0) e pequena o suficiente para evitar bordas de outros objetos. É calculada a transformada de *Hough* na vizinhança de cada pixel da imagem, procurando

picos relevantes no espaço paramétrico. Em seguida, dentre os picos encontrados, são selecionados quatro picos que atendem a certas condições espaciais particulares de retângulos, como correspondentes a um mesmo objeto retangular presente naquela vizinhança. Os resultados experimentais mostram que a metodologia consegue detectar objetos retangulares de diversas orientações e tamanhos, tanto em imagens sintéticas quanto em imagens naturais. A metodologia foi testada em diversas aplicações, como detecção de placas de carro ou detecção de prédios em imagens aéreas. Entretanto, foi percebido que podem haver falsos positivos se os retângulos detectados estiverem muito próximos uns dos outros.

Os pesquisadores Kastelan *et al.* [4] propuseram um sistema automático em tempo real para verificação de qualidade de imagem de telas de TV. A verificação é composta por três passos: captura da imagem, extração do conteúdo e verificação de qualidade. A captura da imagem da tela é feita através de câmera. O conteúdo é extraído em duas etapas: primeiramente, a tela é detectada através de um esquema de detecção de linhas longas verticais e horizontais combinado com detecção de retângulos. Após isto, a região detectada como tela é transformada para corresponder às dimensões de imagens de referência. A verificação de qualidade da imagem é feita através de três métodos que comparam o conteúdo da tela com os de imagens de referência. As imagens de referência podem ser obtidas capturando a imagem da memória interna da TV ou capturando uma imagem considerada correta através de câmera. O método de menor erro médio (LAE) calcula a diferença entre o conteúdo da tela e a imagem de referência pixel a pixel. A pontuação dada à imagem é o somatório das diferenças. O método da correlação cruzada normalizada (NCC) calcula a semelhança entre o conteúdo e uma imagem de referência calculando a correlação entre as imagens. O método da correlação cruzada normalizada através de blocos (NCC-BB) divide cada imagem em blocos. A pontuação do conteúdo da tela é obtida calculando diferença entre a autocorrelação da imagem de referência e a correlação entre o conteúdo da tela e a imagem de referência. Foram realizados experimentos para verificar a eficácia dos métodos de verificação de qualidade da imagem utilizando diferentes imagens de referência, assim como imagens capturadas sob condições constantes e variáveis de iluminação. As técnicas se mostraram eficazes em identificar corretamente o conteúdo na tela em todos os experimentos,

exceto em imagens sob condições de iluminação variável. O sistema proposto foi utilizado para automatizar o processo de verificação no final de uma linha de produção de TVs. Foi mostrado que a etapa de verificação ficou 5 vezes mais rápida em comparação com a verificação manual.

A metodologia proposta nesta dissertação difere da proposta por Liu *et al.* [2] em dois aspectos fundamentais: o objetivo e escopo. O objetivo da metodologia proposta neste trabalho é detectar uma tela de TV em uma imagem, ao contrário da proposta pelos pesquisadores, que visa encontrar quaisquer retângulos presentes em uma imagem. Aproveitando-se de características particulares da tela de TV, como a relação de aspecto e orientação, modificamos as etapas de detecção de linhas e detecção de retângulos de tal forma que o sistema possa se especializar em detectar retângulos com estas características. Além disso, implementamos uma etapa de seleção do retângulo que representa a tela, etapa que não existe no sistema proposto anteriormente. As imagens que são alvo desta metodologia têm alta resolução, *background* com baixo contraste e alta possibilidade de ruído. Implementamos técnicas de pré-processamento robustas a ruído, buscando aumentar as chances de detecção em imagens com estas características.

As diferenças encontradas entre a metodologia proposta nesta dissertação e a proposta por Jung e Schramm [3] estão em objetivo, escopo e restrições. O objetivo da metodologia de Jung é encontrar todos os retângulos presentes em uma imagem, ao ponto em que a metodologia proposta neste trabalho se propõe a encontrar um único retângulo considerado a tela. Quanto ao escopo, é percebido que a metodologia de Jung tem como alvo imagens monocromáticas de baixa resolução, enquanto nesta dissertação são detectados retângulos em imagens coloridas de alta resolução. Isto pode ser explicado pelo alto tempo de processamento do algoritmo que utiliza a metodologia dos pesquisadores brasileiros. Por fim, percebemos que os retângulos que a metodologia procura devem ter seu tamanho previamente estimado, do contrário a metodologia não é capaz de detectá-los. Esta restrição não existe na metodologia proposta neste trabalho, ainda que usemos restrições de tamanho na etapa de seleção do retângulo que representa a tela.

A diferença fundamental entre a metodologia proposta nesta dissertação e o sistema proposto por Kastelan *et al.* [4] é o escopo. O método proposto nesta

dissertação tem como objetivo encontrar telas em imagens de monitores capturadas sob circunstâncias não-controladas, enquanto o trabalho dos sérvios, mais específico, captura as imagens sob condições extremamente controladas. Por visar imagens mais difíceis, o método proposto utiliza técnicas diferentes para pré-processamento e processamento das imagens capturadas, dentre elas técnicas de detecção de borda no estado da arte e técnicas de detecção de retângulos mais robustas a variações no tamanho, orientação, cor e ruído e um método novo de formatação da região detectada como tela.

	Liu, Ikenaga e Goto [2]	Kastelan et. al. [4]	Jung e Schramm [3]	Proposto
objetivo	detectar retângulos vários objetos várias orientações ambiente não controlado distância variável	detectar telas um objeto próxima a 90° ambiente controlado distância fixa	detectar retângulos vários objetos várias orientações ambiente não controlado tamanho estimado	detectar telas um objeto próxima a 90° ambiente não controlado distância variável
resolução	73 × 42 (min) 3072 × 2048 (max)	- -	- -	2048 × 1152 (min) 5312 × 2988 (max)
pré-processamento	filtragem gaussiana	filtragem gaussiana	variável	correção de gama redimensionamento filtragem gaussiana
detecção de bordas	análise de gradiente	Scharr	variável	análise de gradiente
detecção de linhas	hough modificado	detecção de linhas longas	-	hough
detecção de retângulos	MRF	combinação de linhas	<i>Windowed Hough Transform</i>	MRF

Tabela 2.1: Diferenças entre trabalhos relacionados e metodologia proposta.

Capítulo 3

Metodologia Proposta

Este capítulo apresenta a metodologia de extração de conteúdo de TV e monitores. O capítulo está organizado em seções, que são como segue: a introdução contextualiza o problema de detecção de objetos quadrangulares e telas de TV e monitores, a detecção de bordas aborda as técnicas utilizadas nesta metodologia para detectar bordas, a detecção de linhas detalha como foi feita a detecção de linhas, a detecção de retângulos aborda as técnicas utilizadas para reconhecer as linhas encontradas como parte de retângulos, assim como o processo de escolha do retângulo que representa a tela, área que apresenta o conteúdo do monitor/TV.

3.1 Introdução

O problema da detecção de objetos quadrangulares em imagens [59,60] possui diversas aplicações. Por exemplo, navegação de robôs [61,62] e detecção de prédios em imagens de satélites [63–65]. Para caracterizarmos o problema da detecção de objetos quadrangulares precisamos considerar que um objeto quadrangular [59] em uma imagem é definido como um objeto delimitado por quatro segmentos de reta distintos pertencentes à imagem, conectados por quatro vértices. Em particular, podemos encarar a detecção de telas de TV e monitores como um problema de detecção de objetos quadrangulares [4]. De fato, isto pode ser feito devido à padronização da fabricação de TVs e monitores [66], que recomenda algumas características para as telas. Este cenário favorece a criação de um sistema de inspeção de telas podendo auxiliar em sistemas para inspeção automática [4].

Os métodos de inspeção automática de TVs e monitores por análise digital da imagem podem ser divididos em duas categorias principais. Na primeira, encontramos os métodos em que a captura da imagem é feita diretamente da memória da TV ou monitor. Perceba que a captura é feita antes da exibição da imagem na TV ou monitor [17, 67, 68]. Esta categoria de métodos proporciona a detecção de erros de hardware e software que produzam como resultado uma imagem defeituosa na memória da TV ou monitor. Na segunda, temos os métodos em que a captura da imagem é feita através de uma câmera [4, 21]. Assim, podemos analisar a imagem exibida na TV ou monitor do ponto de vista do usuário. Conseguimos detectar, além das falhas detectadas na primeira categoria, problemas nos circuitos entre a memória interna da TV e a tela. Geralmente, os métodos de inspeção automática de TVs e monitores em que a captura da imagem é feita por uma câmera utilizam uma ou mais das seguintes etapas:

- **pré-processamento:** as imagens capturadas geralmente necessitam de algum pré-processamento, seja para eliminar ruído ou para destacar as características necessárias para que as etapas seguintes possam ser executadas.
- **detecção da região da tela na imagem:** utilizando as informações geradas na fase de pré-processamento, o método deve ser capaz de identificar na imagem a região que corresponde à tela. Isto normalmente é feito através de uma combinação de detecção de retângulos e identificação, se aproveitando de características da tela.
- **extração do conteúdo da tela:** identificada a região, nesta etapa é feita a extração do conteúdo da tela. Ele é extraído para um formato que seja compatível com a etapa de comparação.
- **inspeção:** nesta etapa é feita a inspeção da tela, através da comparação do conteúdo extraído com uma ou mais referências. A comparação pode ser feita de diversas formas, utilizando toda a imagem capturada ou parte dela. O resultado pode ser uma pontuação, uma imagem ou um rótulo (ok ou não ok).

3.2 Metodologia de Extração de Conteúdo Proposta

Nesta dissertação, conforme comentamos, apresentamos uma metodologia de extração de conteúdo que representa uma alternativa para integrar um sistema de inspeção automática de telas de TV e monitores. Devido à crescente necessidade de um sistema rápido e eficiente de inspeção de telas [67], esta metodologia pode ser empregada em linhas de montagem de TVs e monitores. Em relação às contribuições científicas, comentamos que esta metodologia modifica técnicas de detecção de borda através de análise de gradiente multi-dimensional e detecção de retângulo utilizando um modelo baseado no MRF [2] para detectar e selecionar uma região quadrangular que representa a tela de um monitor/TV, fornecendo uma alternativa para o problema de extração de conteúdo apresentado em [4].

A Figura 3.1 ilustra as etapas do método, detalhados nas próximas seções. A metodologia completa possui quatro etapas, descritas em linhas gerais conforme a seguir. Primeiramente^①, a detecção de bordas é feita através de uma combinação da análise de gradiente multi-dimensional [69] e no detector de Canny [35]. Esta é uma técnica para detecção de bordas em imagens coloridas robusta a mudanças na iluminação e cores. Um esquema de detecção de linhas^② baseado na transformada de Hough é utilizado no mapa de bordas resultante da etapa de detecção de bordas. Estas linhas são agrupadas como parte de objetos retangulares^③ através de um modelo baseado no campo aleatório de Markov (MRF) [2]. Por último, os retângulos resultantes são classificados^④, de forma a encontrar o retângulo que representa a tela.

A seguir, conforme comentamos, detalharemos todas as etapas da metodologia de extração de conteúdo proposta, enfatizando as suas particularidades, algoritmos e modelagem.

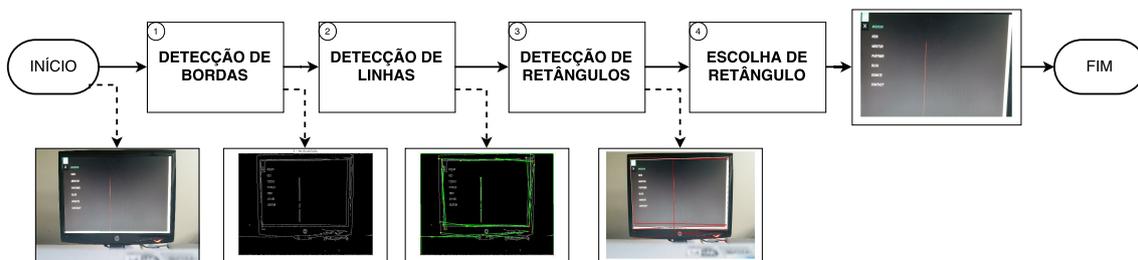


Figura 3.1: Fluxograma do método de extração de telas de TV e monitores.

3.3 Detecção de Bordas

A detecção de bordas é uma técnica de processamento de imagens que identifica descontinuidades em propriedades de uma imagem [7]. Dada uma imagem digital de entrada, a saída de um detector de bordas é um mapa de bordas, uma matriz binária do tamanho da imagem em que cada pixel de valor 1 corresponde a borda. A Figura 3.2 mostra um exemplo de extração de bordas de uma imagem.



Figura 3.2: Imagem e seu mapa de bordas.

O esquema de detecção de bordas utilizado nesta dissertação é baseado na análise de gradiente multi-dimensional [69] e no detector de Canny [35]. Suas etapas, ilustradas na Figura 3.3, serão descritas conforme segue. Primeiramente^①, a imagem é tratada como um campo vetorial e são calculadas a magnitude e direção de seu gradiente. Utilizando esse esquema é possível obter o gradiente de imagens coloridas sem a necessidade de recorrer à conversão para tons de cinza. Após o cálculo do gradiente, são feitas a seleção^② e verificação^③ de candidatos a pixel de borda. Na etapa da seleção, são separados, dentre os pixels, os mais propícios a pertencerem a bordas de objetos. Isto é feito utilizando um esquema simples mostrado na Seção 3.3.2, que tem baixo custo computacional, porém produz muitos falsos positivos. A verificação, mais custosa computacionalmente que a seleção, é feita para descartar, dentre os pixels anotados como candidatos, os que realmente não pertençam a uma borda, conforme será mostrado na Seção 3.3.3.

3.3.1 Cálculo do Gradiente

Nesta seção, apresentamos o cálculo da magnitude V e direção θ do gradiente G de uma imagem colorida, ou seja, formada por canais RGB. Para isso, vamos

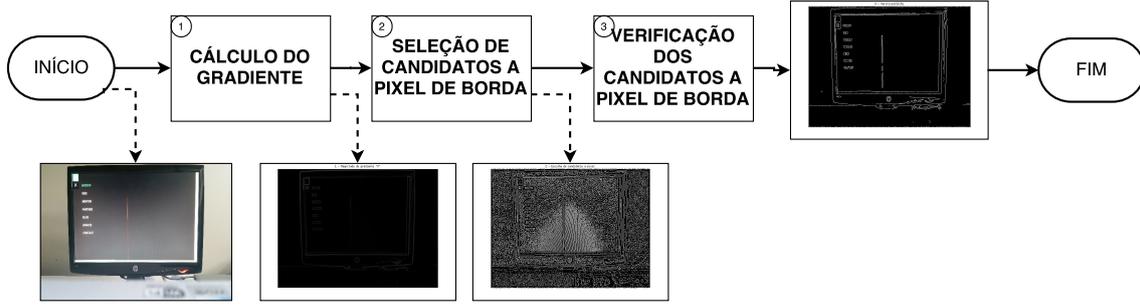


Figura 3.3: Fluxograma da técnica de detecção de bordas.

considerar primeiramente o cálculo do gradiente G de uma imagem em tons de cinza $I(x, y)$, ou seja, de um canal. O gradiente G é um vetor $\left\{ \frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right\}$ em que $\frac{\partial I(x, y)}{\partial x}$ e $\frac{\partial I(x, y)}{\partial y}$ são as derivadas de primeira ordem da imagem nas direções x e y . A magnitude é igual a $|G| = \sqrt{\left(\frac{\partial I(x, y)}{\partial x} \right)^2 + \left(\frac{\partial I(x, y)}{\partial y} \right)^2}$ e a matriz Jacobiana da imagem é dada por $J_b = \begin{bmatrix} \frac{\partial I(x, y)}{\partial x} & \frac{\partial I(x, y)}{\partial y} \end{bmatrix}$. Considere a matriz:

$$M_b = J_b^T J_b = \begin{bmatrix} \left(\frac{\partial I(x, y)}{\partial x} \right)^2 & \left(\frac{\partial I(x, y)}{\partial x} \right) \left(\frac{\partial I(x, y)}{\partial y} \right) \\ \left(\frac{\partial I(x, y)}{\partial x} \right) \left(\frac{\partial I(x, y)}{\partial y} \right) & \left(\frac{\partial I(x, y)}{\partial y} \right)^2 \end{bmatrix} \quad (3.1)$$

Na equação (3.1), o autovalor e autovetor da matriz M_b podem ser utilizados para calcularmos a magnitude e direção do gradiente, respectivamente. O autovalor é o traço da matriz M_b , ou seja, o autovalor é igual a $\left(\left(\frac{\partial I(x, y)}{\partial x} \right)^2 + \left(\frac{\partial I(x, y)}{\partial y} \right)^2 \right)$ e equivale à magnitude do gradiente $|G|$ elevada ao quadrado. O respectivo autovetor é simplesmente o gradiente $G = \left\{ \frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right\}$.

Considerando uma imagem colorida com canais RGB, a informação em cada pixel não é mais um escalar, e sim um vetor com três valores. O vetor diferença DV que representa o deslocamento infinitesimal no plano da imagem na direção $\{dx, dy\}$ é dado por $DV = (dx \ dy) J_c^T$, em que J_c é a matriz Jacobiana da imagem:

$$J_c = \begin{bmatrix} \frac{\partial I_r(x, y)}{\partial x} & \frac{\partial I_r(x, y)}{\partial y} \\ \frac{\partial I_g(x, y)}{\partial x} & \frac{\partial I_g(x, y)}{\partial y} \\ \frac{\partial I_b(x, y)}{\partial x} & \frac{\partial I_b(x, y)}{\partial y} \end{bmatrix} \quad (3.2)$$

Em que $\frac{\partial I_r(x, y)}{\partial x}$, $\frac{\partial I_r(x, y)}{\partial y}$, $\frac{\partial I_g(x, y)}{\partial x}$, $\frac{\partial I_g(x, y)}{\partial y}$, $\frac{\partial I_b(x, y)}{\partial x}$ e $\frac{\partial I_b(x, y)}{\partial y}$ são as derivadas parciais de primeira ordem dos canais RGB da imagem, iguais a I_r , I_g e I_b nas

direções x e y . A magnitude euclidiana quadrada de DV é dada por:

$$DV^2 = (dx \ dy)M_c(dx \ dy)^T \quad (3.3)$$

Em que podemos definir M_c , M_{xx} , M_{xy} e M_{yy} como:

$$M_c = J_c^T J_c = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{xy} & M_{yy} \end{bmatrix} \quad (3.4)$$

$$M_{xx} = \left(\frac{\partial I_r(x, y)}{\partial x} \right)^2 + \left(\frac{\partial I_g(x, y)}{\partial x} \right)^2 + \left(\frac{\partial I_b(x, y)}{\partial x} \right)^2$$

$$M_{xy} = \left(\frac{\partial I_r(x, y)}{\partial x} \right) \left(\frac{\partial I_r(x, y)}{\partial y} \right) + \left(\frac{\partial I_g(x, y)}{\partial x} \right) \left(\frac{\partial I_g(x, y)}{\partial y} \right) + \left(\frac{\partial I_b(x, y)}{\partial x} \right) \left(\frac{\partial I_b(x, y)}{\partial y} \right)$$

$$M_{yy} = \left(\frac{\partial I_r(x, y)}{\partial y} \right)^2 + \left(\frac{\partial I_g(x, y)}{\partial y} \right)^2 + \left(\frac{\partial I_b(x, y)}{\partial y} \right)^2$$

O maior autovalor de M_c é dado por:

$$V = \frac{\sqrt{((M_{xx} + M_{yy})^2 - 4x(M_{xx} \times M_{yy} - (M_{xy})^2)) + M_{xx} + M_{yy}}}{2} \quad (3.5)$$

e seu autovetor correspondente é $\{M_{xy}, V - M_{yy}\}/2$. Portanto, a direção do gradiente é dada por:

$$\theta = \tan^{-1} \left(\frac{V - M_{xx}}{M_{xy}} \right) \quad (3.6)$$

Por fim, para uma imagem $I(x, y)$ colorida composta por três canais RGB, a magnitude V e a direção θ do gradiente G são obtidas através das equações (3.5) e (3.6), respectivamente. Observe que a raiz quadrada do maior autovalor e seu autovetor correspondente da matriz expressa na equação (3.4) equivalem à magnitude V e direção θ do gradiente G em um ponto (x, y) da imagem $I(x, y)$.

3.3.2 Seleção de Candidatos

A seleção de pixels de borda normalmente é executada calculando a magnitude V do gradiente G , conforme descrito na equação (3.5) e utilizando um limiar fixo para controlar a quantidade de pixels de borda. Encontrar este limiar pode se tornar uma tarefa difícil, devido à variação de imagem para imagem [70].

Nesta técnica, buscamos abordar este problema utilizando um limiar variável para seleção e um sistema de verificação para eliminar falsos positivos. O limiar de magnitude de gradiente utilizado para classificar um pixel como candidato a borda depende dos valores de magnitude dos pixels vizinhos. Consideramos candidatos os pixels cuja magnitude é localmente máxima, ou seja, os pixels cuja magnitude é maior que a dos seus dois vizinhos na direção mais próxima à direção do gradiente, como ilustrado pela Figura 3.4.

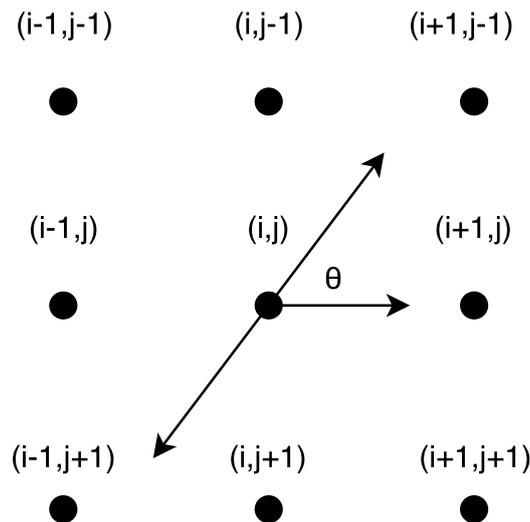


Figura 3.4: Direção de gradiente de um pixel e seus vizinhos. Adaptado de [2].

3.3.3 Verificação de Candidatos

A verificação é um processo necessário para descartar os candidatos falsos positivos escolhidos na etapa de seleção (descrita na Seção 3.3.2). Os falsos positivos podem ser gerados devido a ruído originado na aquisição da imagem, diferenças na iluminação dos objetos, efeitos de luz e sombra ou oclusão de objetos [71]. A verificação é composta por duas operações que fornecem informações espaciais da imagem regionalmente, conforme descritas a seguir:

- Primeiramente, a vizinhança do pixel é analisada. O candidato a pixel de borda deve ter vizinhos candidatos. Se nenhum dos oito pixels vizinhos for um candidato a pixel de borda, aquele pixel é descartado;
- Não há grande variação entre as magnitudes de pixels de borda vizinhos [2]. Portanto, podemos calcular a diferença entre as magnitudes do gradiente dos pixels vizinhos. Caso esta diferença seja grande, o pixel é descartado. Isto pode ser feito conforme a expressão:

$$\left(\sum_0^{NUM(i,j)-1} (|V(i,j) - V(i+p, j+q)|) - \sum_0^{NUM(i,j)-1} (V(i+p, j+q)) \right) > 0 \quad (3.7)$$

Na equação (3.7), $NUM(i, j)$ é o número de pixels vizinhos ao candidato a pixel de borda localizado em (i, j) . Cada vizinho é localizado em $(i + p, j + q)$, em que p e q pertencem ao intervalo $\{-1, 0, 1\}$. Caso a equação (3.7) seja verdadeira, o candidato a pixel de borda é descartado.

3.4 Detecção de Linhas

A detecção de linhas é uma técnica de processamento de imagens que localiza segmentos de linha pertencentes a uma imagem [1]. O objetivo de um detector de linhas é, para uma imagem digital de entrada, produzir uma lista dos segmentos de linha pertencentes a esta imagem.

Nesta dissertação, montamos um esquema de detecção e combinação de linhas baseado na transformada de *Hough*. As etapas, ilustradas na Figura 3.5, serão descritas em termos gerais conforme segue. Primeiramente^①, detectamos os segmentos de linha utilizando a transformada de *Hough*. Após isto, combinamos^② segmentos de linha de orientação e distância da origem semelhante. Por último, eliminamos^③ linhas que têm ângulo muito diferente de 0° e 90° .

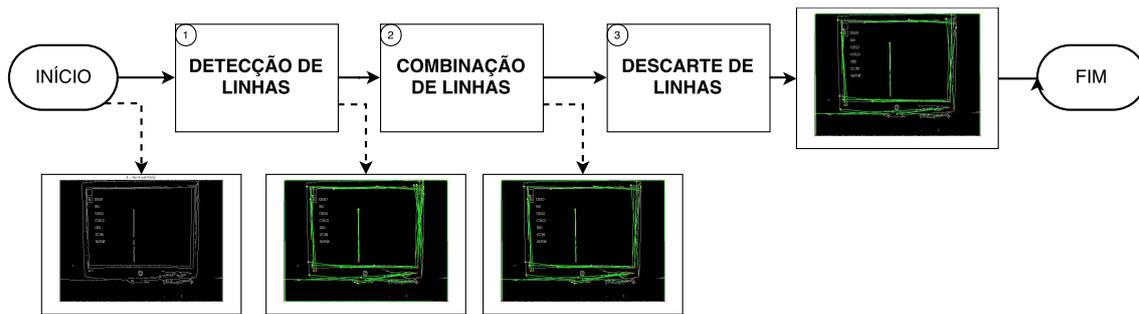


Figura 3.5: Fluxograma da técnica de detecção de linhas.

3.4.1 Detecção de Linhas Utilizando a Transformada de *Hough*

A transformada de *Hough* é uma técnica popular para detecção de linhas e teve por objetivo, inicialmente, a detecção de padrões complexos em imagens binárias, tais como círculos e retângulos [72]. Para este fim, são determinados parâmetros característicos dos padrões e a imagem é transformada para o espaço paramétrico, de forma que estes parâmetros sejam detectados. Para detecção de linhas, utilizamos os parâmetros ρ e θ , que consistem, respectivamente, na distância entre o segmento e a origem da imagem e na orientação do segmento em relação com o eixo horizontal [53]. Na Figura 3.6, temos representadas as duas linhas l_1 e l_2 no espaço x, y , assim como suas coordenadas (ρ_1, θ_1) e (ρ_2, θ_2) no espaço paramétrico ρ, θ .

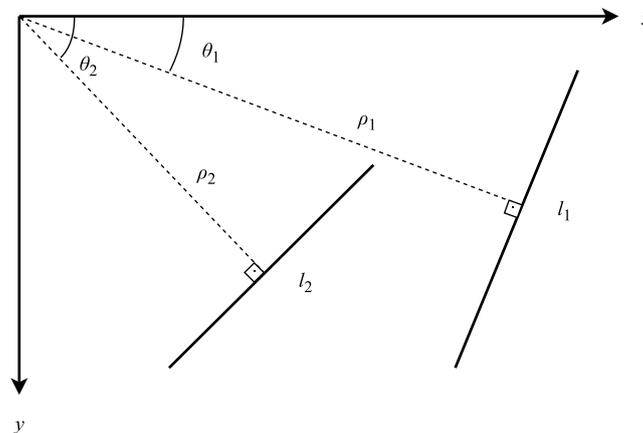


Figura 3.6: Representação paramétrica de Hough. Adaptado de [3]

Nesta dissertação, utilizamos um algoritmo baseado na transformada de *Hough* para detectar linhas. Dado um mapa de bordas de entrada, temos como saída do detector de linhas uma lista com as linhas que compõem este mapa. Cada linha é caracterizada por seu ponto inicial, ponto final, ρ e θ . O algoritmo é composto pelos seguintes passos:

- Primeiramente, escaneamos toda a imagem de borda e computamos, para cada pixel assinalado como borda, seu conjunto de coordenadas no espaço paramétrico (ρ, θ) . Um ponto é acrescentado ao acumulador da coordenada à qual o pixel corresponde, no espaço paramétrico. Assim, quanto mais pixels pertencentes à mesma linha, mais pontos tem o acumulador na coordenada que representa aquela linha;
- Escaneamos o espaço paramétrico para encontrar picos no acumulador, ou seja, as coordenadas que apresentam mais pontos no acumulador. Assinalamos a cada um deles um segmento de linha e guardamos os valores de ρ e θ . A fim de evitar linhas pequenas demais, é estabelecido um limiar mínimo de tamanho das linhas;
- Para cada segmento de linha assinalado, procuramos na imagem de borda os pontos que passam por aquele segmento, para assinalar os pontos inicial e final do mesmo;

Utilizando o algoritmo proposto, obtemos uma lista com os segmentos de linhas pertencentes à imagem, assim como informações sobre os mesmos que são utilizados no processo de combinação de segmentos semelhantes.

3.4.2 Combinação e Descarte de Linhas

Devido a diferenças na iluminação, curvatura da lente e ruído, pode haver erro na detecção de linhas executada pelo algoritmo descrito na Seção 3.4.1. Assim, o mesmo segmento de linha pode ser detectado como dois ou mais segmentos separados e próximos, ou um mesmo segmento pode ser identificado pelo algoritmo como dois segmentos paralelos e próximos. A fim de diminuir a ocorrência deste tipo de problema, propomos um algoritmo que identifica e combina segmentos próximos que possuam ρ e θ semelhantes aproveitando-se das informações obtidas pelo algoritmo de detecção de tela descrito na Seção 3.4.1. O algoritmo combina as linhas que atendam às seguintes condições:

- As linhas devem ter ρ e θ semelhantes. Devido ao ruído e a problemas na quantização, é possível que algumas linhas do mesmo segmento tenham orientação

ou distância da origem um pouco diferentes entre si, portanto estipulamos uma faixa de tolerância para ρ e θ , que compreende 10% a mais ou a menos nos valores de ρ e θ ;

- A distância entre os segmentos de linha não pode ser maior que 5 pixels;
- Segmentos paralelos não podem se sobrepor significativamente quando projetados na direção perpendicular à linha. Neste algoritmo, consideramos uma taxa de sobreposição acima de 60% do tamanho do menor segmento como o critério para eliminar segmentos que se sobrepõem;

Por último, descartamos segmentos de linha que têm ângulos muito diferentes de 0° e 90° . Com isto, diminuimos o número de linhas analisadas nas próximas etapas do método e eliminamos segmentos que podem gerar falsos positivos. Os ângulos foram escolhidos devido as telas (de TVs e monitores) apresentarem ângulos retos. O algoritmo para eliminação busca os segmentos de linha da lista e descarta os segmentos de linha cujo θ não obedece à condição:

$$|\theta| > n(90) + \sigma \quad (3.8)$$

Na equação (3.8), θ é representado em graus, n é um número natural e σ é uma faixa de tolerância. Utilizando os algoritmos descritos nesta seção, podemos obter uma lista com os segmentos de linhas presentes na imagem, assim como informações geométricas importantes sobre os mesmos, tais como orientação, distância da origem, pontos inicial e final.

3.5 Detecção de Retângulos

A detecção de retângulos é um dos problemas mais antigos e importantes do processamento de imagens e detecção de padrões [2]. Um objeto retangular pode ser definido como um objeto delimitado por quatro segmentos de reta distintos e perpendiculares entre si, ligados por quatro vértices. O objetivo de um detector de retângulos é, dada uma imagem digital, fornecer uma lista dos retângulos presentes na mesma.

Nesta dissertação, detectamos objetos retangulares através de um modelo baseado no campo aleatório de Markov (MRF). A teoria do MRF utiliza as relações entre elementos vizinhos para estimar o estado futuro de seus elementos. Portanto, propomos um algoritmo baseado neste modelo para rotular as linhas encontradas como parte de um objeto retangular. As etapas deste algoritmo, ilustrados na Figura 3.7, serão descritos em termos gerais conforme segue. Primeiramente^①, criamos um sistema de vizinhanças para obter informações espaciais das linhas. Em seguida, as informações são fornecidas a um algoritmo de classificação que rotula^② as linhas encontradas como parte de um objeto retangular. Por último, combinamos^③ as informações obtidas no sistema de vizinhanças com os rótulos das linhas para assinalar a qual objeto retangular as linhas pertencem e escolhemos^④ dentre estes retângulos qual representa a tela.

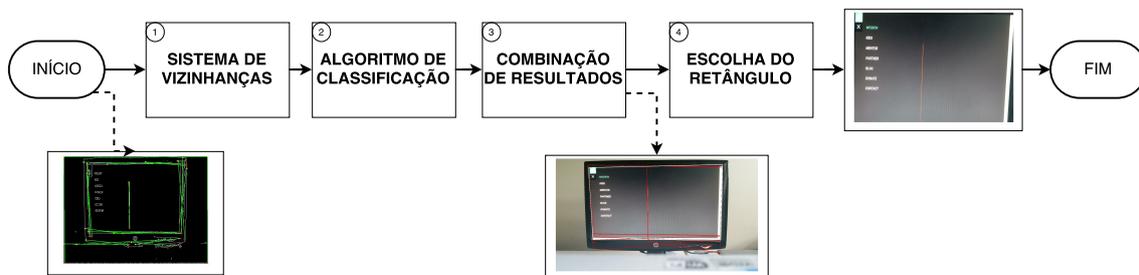


Figura 3.7: Fluxograma da técnica de detecção de retângulos.

3.5.1 Sistema de Vizinhanças Utilizando Campo Aleatório de Markov (MRF)

Conforme mencionamos (Seção 2.5), um campo aleatório de Markov (MRF) é um processo estocástico que apresenta propriedade Markoviana. A definição desta propriedade é que o estado futuro de um processo pode ser previsto sabendo apenas o estado atual, ignorando seus estados anteriores.

Um campo aleatório é considerado um MRF se a probabilidade condicional de um elemento depender de seus elementos vizinhos [73]. O sistema de vizinhanças caracteriza os elementos do campo como vizinhos ou não entre si. O sistema de vizinhança é caracterizado como segue: o conjunto de elementos vizinhos do elemento l_i é representado por $N(l_i)$. Pertencem a $N(l_i)$ elementos que observem as seguintes condições:

- l_i não pertence a $N(l_i)$;
- se l_j pertence a $N(l_i)$, l_i pertence a $N(l_j)$;

No caso específico deste modelo, os elementos são linhas. A vizinhança é caracterizada utilizando suas relações espaciais. Para uma linha l_j ser considerada vizinha de l_i , são necessárias três condições:

- l_j deve ser paralela ou perpendicular a l_i . Uma variável $H(i, j)$ é formada para avaliar as relações espaciais entre as linhas l_i e l_j . Se $H(i, j) = 1$, l_i e l_j são perpendiculares entre si, ao passo que se $H(i, j) = 0$, l_i e l_j são paralelas. Caso os ângulos entre l_i e l_j não pertençam a nenhuma das categorias anteriores, $H(i, j) = -1$;
- A distância $D(i, j)$ entre l_i e l_j não pode ser muito grande ou muito pequena. A distância é computada de duas formas, dependendo de $H(i, j)$. Caso $H(i, j) = 0$, a distância entre l_i e l_j é a menor distância entre as duas linhas, ou seja, o comprimento de uma reta normal às duas que toca as duas linhas. Neste caso, $D(i, j)$ deve ser menor que o *maior tamanho de retângulo possível* (MRS), normalmente igual ao tamanho da imagem, e maior que o *menor retângulo possível* (MIS). Caso $H(i, j) = 1$, a distância é considerada a menor distância entre o fim de um segmento e outro segmento. Neste caso, $D(i, j)$ deve ser menor que a metade do tamanho do menor segmento;
- Por último, se l_i e l_j são paralelos, eles devem se sobrepor em uma porção significativa quando projetados na direção perpendicular às linhas. Se a sobreposição for superior a 60% do tamanho das linhas, então as linhas são consideradas vizinhas.

A Figura 3.8 ilustra as condições explicadas. Nela, l_1 e l_2 são linhas paralelas, l_3 é uma linha perpendicular às duas, $D(1, 2)$ é a distância entre l_1 e l_2 e $D(2, 3)$ é a distância entre l_2 e l_3 . A projeção da sobreposição da linha l_1 na linha l_2 é o segmento AB . Somente se $AB > 0.6 * \max(\text{tam}_{l_1}, \text{tam}_{l_2})$, l_2 será considerado vizinho de l_1 .

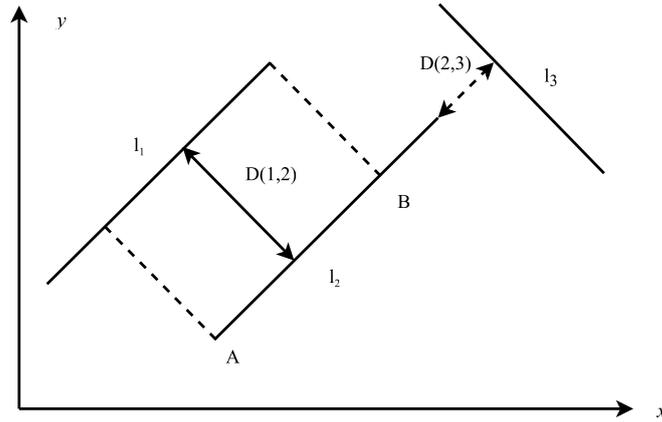


Figura 3.8: Linhas paralelas e perpendiculares. Adaptado de [2].

3.5.2 Algoritmo de Classificação de Linhas Utilizando Campo Aleatório de Markov (MRF)

Detectar objetos retangulares em uma imagem é um problema que pode ser encarado como o problema de classificar as linhas presentes na imagem que pertencem à borda de objetos retangulares. Nesta seção descrevemos um algoritmo que, dada uma lista de linhas pertencentes a uma imagem, rotula as linhas que estão presentes nas bordas de objetos retangulares.

O problema de classificação encarado neste trabalho pode ser caracterizado a seguir. Seja $L = \{l_1, l_2, \dots, l_n\}$ o conjunto de segmentos de linha encontrados pelo algoritmo de detecção de linhas descrito na Seção 3.4, $d = \{1, 2, \dots, n\}$ o conjunto de índices de L e $F = \{F_1, F_2, \dots, F_n\}$ uma família de variáveis aleatórias, em que cada variável F_i toma um valor f_i no intervalo $[0, 1]$. Este valor indica a probabilidade da linha l_i pertencer a um objeto retangular.

O evento da variável F_i ter o valor f_i é representado como $F_i = f_i$. O evento conjunto em que cada uma das variáveis de F apresenta um conjunto de valores $f = \{f_1, f_2, \dots, f_n\}$ é representado como $(F_1 = f_1, \dots, F_n = f_n)$, ou $F = f$. O conjunto de valores f é chamado de *configuração* de F . A probabilidade deste evento conjunto é representada por $P(F = f)$. Sendo F um MRF, a probabilidade de qualquer evento referente a F depende de sua vizinhança. A vizinhança, encontrada através do esquema descrito na Seção 3.5.1, é representada por $N = \{N(l_i) | \forall i \in d\}$, sendo $N(l_i)$ a coleção de segmentos vizinhos a um segmento de linha l_i .

A maior parte dos problemas de visão computacional e processamento de

imagens é tratada como um problema de otimização [29]. Isto se deve ao alto grau de incertezas que permeia qualquer processo que envolva imagens. Ruído, descontinuidades e oclusão de parte dos objetos são geradores de incertezas, portanto é muito difícil obter uma solução exata. Uma solução inexata, porém ótima, é mais facilmente obtida.

A função de energia tem um papel importante nestes tipos de problema. Ela pode ser usada como uma medida da qualidade da solução, assim como facilita a procura por uma solução mínima, ou ótima. Uma função de energia bem projetada tem seu valor mínimo quando a solução é ótima.

Assim, o reconhecimento de lado do retângulo pode ser encarado como o seguinte problema: para um dado L , $\operatorname{argmax} P(f_i|l_i)$ para cada l_i é encontrado por $\operatorname{argmin} E(f)$, ou seja, minimizar a função de energia implica em maximizar a probabilidade definida na equação (2.3). Desta forma, teremos a máxima estimativa posterior de potenciais lados de retângulos na imagem de bordas. A função de energia $E(f)$ é minimizada enquanto cada f_i converge para 0 ou 1. A função $E(f)$ é matematicamente descrita por:

$$E(f) = E_1(f) + E_2(f) + E_3(f) + E_4(f) \quad (3.9)$$

$$E_1(f) = \alpha_1 \sum_{i=1}^n \sum_{j \in N(l_i)} f_i f_j D(i, j) \quad (3.10)$$

$$E_2(f) = \alpha_2 \sum_{i=1}^n \frac{f_i}{LEN_i} \times AVGLLEN \quad (3.11)$$

$$E_3(f) = -\alpha_3 \sum_{i=1}^n (f_i \ln f_i + (1 - f_i) \ln(1 - f_i)) \quad (3.12)$$

$$E_4(f) = \alpha_4 \sum_{i=1}^n \sum_{j \in N(l_i)} f_i f_j \left\{ H(i, j) \times \min_{k \in (N(l_i) \cap N(l_j))} \frac{D(k, i) + D(k, j)}{f_k} + \right. \quad (3.13)$$

$$\left. (1 - H(i, j)) \times \min_{k \in (N(l_i) \cap N(l_j))} \frac{D(k, i) + D(k, j)}{f_k H(k, j)} \right\}$$

Onde α_1 , α_2 , α_3 e α_4 são constantes positivas que controlam as contribuições

de cada termo individual da função de energia. $AVGLEN$ é o comprimento médio do conjunto de segmentos L e é definido por:

$$AVGLEN = \frac{1}{n} \sum_{i=1}^n LEN(i) \quad (3.14)$$

Na equação (3.14), $LEN(i)$ é o comprimento do segmento l_i . Os f_i são todos inicializados em 0,5 [2] e a energia é gradualmente reduzida usando um algoritmo gradiente descendente com uma esquema de aprendizado definido pelas equações:

$$f_{i(t+1)} = f_{i(t)} - \mu(t) \nabla E$$

$$\mu_{(t+1)} = \mu_{(t)} - \beta$$

em que μ é um parâmetro de passo positivo, que influencia na taxa de convergência, β é uma pequena constante que faz o tamanho do passo diminuir a cada iteração, e ∇E é o gradiente de E .

Minimizando a energia, os segmentos de L cujo $f_i \simeq 1$ são selecionados como lados de objetos retangulares desconhecidos. Se faz necessário, então, um sistema que utilize as informações espaciais das linhas escolhidas para agrupar as linhas selecionadas em seus respectivos retângulos.

O sistema de vizinhanças foi construído para prover estas informações espaciais. Após rotulados os segmentos de linha, é feita uma busca em seus segmentos vizinhos a fim de formar retângulos integrados, agrupando segmentos de linha de quatro em quatro seguindo os passos descritos a seguir. Para cada segmento de linha l_i , marcado como pertencente a um retângulo, são selecionados três segmentos que atendam às seguintes condições:

- pertencem à vizinhança de l_i ;
- são rotulados como pertencentes a retângulos;
- um segmento é paralelo e dois são perpendiculares a l_i ;
- os três segmentos estão na vizinhança uns dos outros.

3.5.3 Escolha do Retângulo que Representa a Tela

A escolha do retângulo que representa a tela da TV/monitor utiliza critérios criados através da análise das propriedades geométricas de uma tela e de características da captura das imagens. Classificamos um retângulo como tela caso ele obedeça às seguintes condições:

- **o retângulo não está nas bordas da imagem:** para evitar que alguma parte da tela não seja capturada. Além disso, o diferencial do método aqui explicado é conseguir capturar a tela de um monitor em um ambiente qualquer;
- **o retângulo possui uma razão de aspecto próxima a 16/9:** a relação de aspecto mais comum em monitores e telas de TV é de 16/9, assim como a razão inversa 9/16 é a mais comum em celulares. Isso se deve em parte por ser o formato padrão de imagens em *HDTV*, *Full HD* e transmissão de TV digital e analógica [66]. Assim, a fim de evitar falsos positivos, o método descarta, dentre os objetos encontrados, os que não tiverem uma relação de aspecto próxima dos monitores que estão sendo estudados;
- **o retângulo preenche mais que 30% da imagem:** é considerado que a tela, embora não possa ser o único objeto presente na imagem, precisa ser grande o suficiente para que seja possível extrair informações dela. Para isso, é necessário que a tela ocupe um espaço mínimo da imagem. Assim, a área dos objetos retangulares é calculada e comparada com o tamanho da imagem, a fim de descartar os que têm área abaixo da mínima;

O resultado da seleção deve ser apenas retângulos de tamanho considerável e com a relação de aspecto próxima à de TVs e monitores. Com isto, se faz necessário um critério de verificação para que o retângulo que mais se assemelha à tela seja o escolhido como verdadeiro. Nesta dissertação, utilizamos os seguintes critérios [4]:

- caso não seja encontrado retângulo atendendo às condições, a resposta do método é que nenhum retângulo que representa a tela foi encontrado;
- caso seja encontrado somente um retângulo atendendo às condições, este retângulo é considerado a tela;

- caso seja encontrado mais de um retângulo atendendo às condições, os dois maiores retângulos encontrados ao final do processo representam as bordas da tela e do monitor. Portanto, o segundo maior retângulo é escolhido como tela;

3.5.4 Formatação

O formato e o tamanho da região da imagem escolhida como tela (através do sistema descrito nas Seções 3.5.2 e 3.5.3) são variáveis. O tamanho da TV/monitor, assim como a distância e o ângulo entre câmera e TV/monitor são fatores que influenciam no tamanho e na forma da região encontrada. Para fins de avaliação do conteúdo da tela de TV/monitor, se faz necessário que a região dada como resposta pela metodologia aqui proposta tenha forma e tamanho fixos. Assim, se faz necessário um esquema de formatação desta região, a fim de converter o trecho da imagem original em uma imagem de tamanho e formato predefinidos.

O esquema de formatação aqui proposto se dá em duas fases: a transformação da imagem e o redimensionamento. A transformação, ilustrada na Figura 3.9, visa transmitir o conteúdo de uma região quadrangular qualquer $A'B'C'D'$ para um retângulo perfeitamente regular $ABCD$. O problema é encontrar a qual parte $G'(x',y')$ da região retangular pertence cada parte $G(x,y)$ do retângulo transformado.

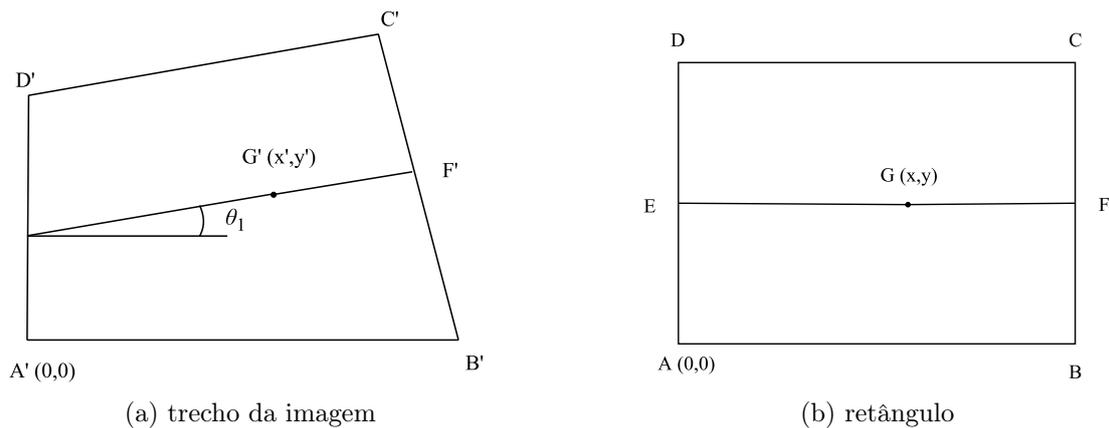


Figura 3.9: Transformação da região quadrangular $A'B'C'D'$ no retângulo $ABCD$. Adaptado de [4].

Dado que o retângulo transformado tem tamanho (col_r, lin_r) , que a região da imagem $A'B'C'D'$ é delimitada pelos pontos $A' = (x_{a'}, y_{a'})$, $B' = (x_{b'}, y_{b'})$, $C' = (x_{c'}, y_{c'})$ e $D' = (x_{d'}, y_{d'})$, podemos definir as variações dependentes de x como

$dx1 = \frac{x_{b'} - x_{a'}}{col_r}$ e $dy1 = \frac{y_{b'} - y_{a'}}{col_r}$. As variações dependentes de y são $dx2 = \frac{x_{c'} - x_{a'}}{lin_r}$ e $dy2 = \frac{y_{c'} - y_{a'}}{lin_r}$.

Assim, um ponto $G(x, y)$ no retângulo transformado tem valores iguais a um ponto $G'(x', y')$ na imagem, segundo as equações:

$$x = y_{a'} + dy1(x') + dy2(y')$$

$$y = x_{a'} + dx1(x') + dx2(y')$$

A etapa de redimensionamento busca dar ao conteúdo da tela um tamanho pré-definido. O redimensionamento é feito através da interpolação bilinear, que é uma técnica de interpolação de funções de duas variáveis, o que a faz muito interessante para interpolar imagens. Utilizando esta técnica, dada uma imagem I , é possível encontrar o valor $I'(x, y)$ de uma imagem interpolada I' através da equação:

$$I'(x, y) = \frac{y_2 - y}{y_2 - y_1} I'(x, y_1) + \frac{y - y_1}{y_2 - y_1} I'(x, y_2) \quad (3.15)$$

Na equação (3.15), $I'(x, y)$ é o valor interpolado da imagem I' na posição (x, y) , y_1, y_2 são valores conhecidos da imagem original e $I'(x, y_1)$ e $I'(x, y_2)$ são aproximados através das equações:

$$I'(x, y_1) = \frac{x_2 - x}{x_2 - x_1} I(x_1, y_1) + \frac{x - x_1}{x_2 - x_1} I(x_2, y_1) \quad (3.16)$$

$$I'(x, y_2) = \frac{x_2 - x}{x_2 - x_1} I(x_1, y_2) + \frac{x - x_1}{x_2 - x_1} I(x_2, y_2) \quad (3.17)$$

Nas equações (3.16) e (3.17), são conhecidos os valores de $I(x_1, y_1)$, $I(x_1, y_2)$, $I(x_2, y_1)$ e $I(x_2, y_2)$ nas posições (x_1, y_1) , (x_1, y_2) , (x_2, y_1) e (x_2, y_2) na imagem original.

Com esta técnica, é possível obter uma aproximação do trecho da tela da TV ou monitor presentes na imagem capturada, do tamanho e formato pré-definidos.

3.6 Considerações Finais

Neste capítulo, apresentamos uma das contribuições da dissertação, que consiste em uma metodologia para extrair conteúdo de TVs e monitores em imagens digitais. Esta metodologia visa localizar a região da imagem digital que corresponde ao conteúdo apresentado em uma TV ou monitor. Para isto, esta metodologia é utilizada um modelo baseado no campo aleatório de Markov, tornando possível localizar a região independente do tamanho, *background*, cor da TV ou monitor, conteúdo ou da distância entre a câmera e a tela.

Esta metodologia é utilizada no contexto de um sistema de inspeção automática de TVs. Em um sistema de inspeção, utilizar a metodologia aqui apresentada permite a detecção de defeitos de hardware e software que gerem defeitos na imagem gerada pelo aparelho, além de problemas nos circuitos entre a memória interna da TV e a tela.

Capítulo 4

Resultados Experimentais

Para avaliar o desempenho do método de extração do conteúdo de TVs e monitores, apresentado no Capítulo 3, elaboramos experimentos que abordam o problema da detecção de telas em TVs e monitores. Em complemento, realizamos um breve experimento para inspeção.

A organização do capítulo é feita como segue: na primeira seção, mostramos o *setup* dos experimentos, seguida por uma seção que detalha as bases de dados utilizadas. Após, apresentamos as métricas de desempenho de cada experimento, finalizando o capítulo com uma seção detalhando os procedimentos experimentais e os resultados, bem como uma análise de desempenho.

4.1 *Setup* dos Experimentos

Os experimentos foram implementados na linguagem MATLAB, na versão R2012a, em ambiente Linux, processador Intel(R) Core(TM) i7-3770 CPU @ 3,4GHz e memória RAM 16GB. Diversos *scripts* foram programados para que fosse possível implementar as simulações propostas.

4.2 Bases de Dados

Para os experimentos descritos neste capítulo, utilizamos duas bases de dados com imagens contendo aparelhos de TV e monitores. Estas bases, concebidas pelo autor, foram criadas com o objetivo de apresentar um conjunto de imagens em alta

resolução e diferentes condições de iluminação, pose e distância.

4.2.1 Base de Dados Sob Condições Não Controladas

Esta base de dados foi criada para fornecer um conjunto de imagens de monitores e TVs sob condições não controladas de iluminação, ambiente, pose e distância. Esta base apresenta imagens com 14 aparelhos, conforme mostrado na Tabela 4.1.

Aparelho	Tipo	Modelo	Tamanho (polegadas)
1	monitor	HP L200b	20
2	notebook	Samsung RV420	14
3	monitor	HP L185b	18,5
4	monitor	Samsung Syncmaster 713N	17
5	TV	Samsung UN46F6800AG	46
6	TV	LG 32LE5500	32
7	TV	Panasonic TC L50ET60B	50
8	notebook	Apple MacBook Air 13	13,3
9	notebook	Dell Inspiron 14z	14
10	notebook	Toshiba Satellite A505-S6980	16
11	notebook	Lenovo L1125	14
12	monitor	Dell P2314H	23
13	monitor	HP Compaq LA2006x	20
14	monitor	Dell P2312H	23

Tabela 4.1: Monitores e TVs utilizados na base de dados sob condições não controladas.

A base de dados possui 6 (seis) imagens de referência, listadas conforme tipo de imagem na Tabela 4.2 e mostradas na seção B.1 do Apêndice B. As imagens de referência são coloridas e têm resolução de 1600×900 pixels. A captura das imagens foi feita nas seguintes condições: cada aparelho foi fotografado ligado, em um ambiente fechado com um misto de iluminação artificial e natural (por exemplo, uma sala com janelas) durante o dia, apresentando uma das imagens de referência em tela cheia. A câmera foi posicionada manualmente pelo operador em frente ao aparelho, a uma distância em que ocupe a maior parte da imagem. Cada aparelho foi capturado nas resoluções de 5312×2988 , 3264×2448 e 2048×1152 pixels, uma vez com a iluminação artificial do ambiente ligada (iluminação alta) e uma com a iluminação desligada (iluminação baixa), totalizando 504 imagens. Devido a esta forma de captura, condições como iluminação do aparelho, distância entre câmera e aparelho, ângulo e *background* são não-controladas. Alguns exemplos de imagens

capturadas são encontrados na Figura 4.1.

Quantidade	Tipo	Resolução (pixeis)
4	menu	1600×900
2	frame de vídeo	1600×900

Tabela 4.2: Imagens de referência da base de dados sob condições não controladas.

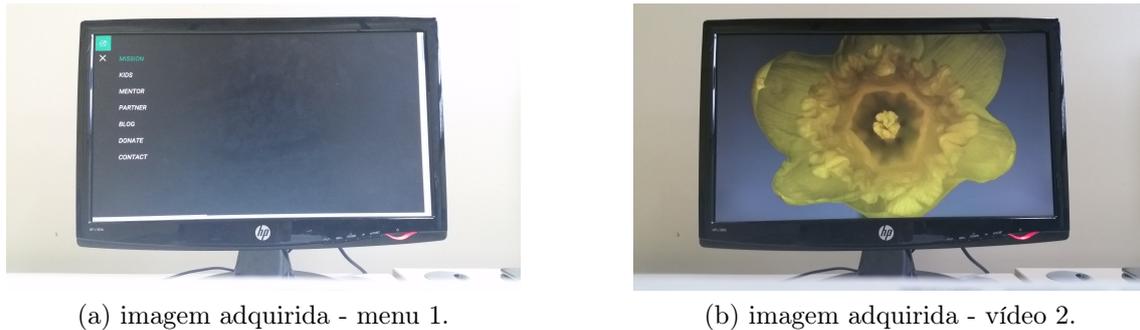


Figura 4.1: Imagens da base de dados sob condições não controladas.

4.2.2 Base de Dados Sob Condições Controladas

Esta base de dados foi criada para fornecer um conjunto de imagens de monitores e TVs sob condições controladas de iluminação, ambiente, pose e distância. Esta base contém imagens de 15 aparelhos, entre monitores e notebooks.

A base de dados possui 40 (quarenta) imagens de referência, listadas conforme tipo de imagem na Tabela 4.3 e mostradas na seção B.2 do Apêndice B. A captura das imagens foi feita nas seguintes condições: uma câmara escura com estrutura cúbica foi confeccionada a fim de garantir um espaço com um mínimo de iluminação externa. Esta câmara foi colocada em um ambiente fechado sem iluminação. A estrutura comporta o celular Samsung Galaxy S5, utilizado para adquirir as imagens, e o aparelho cuja imagem seria adquirida, de forma que tanto o celular quanto o aparelho permaneceram no interior da câmara escura durante a aquisição da imagem. Cada aparelho estava ligado, apresentando uma das imagens de referência em tela cheia. A câmera foi colocada em uma posição fixa dentro da câmara escura e acionada remotamente pelo operador. Cada aparelho foi posicionado de forma que ocupasse a maior parte da imagem. Cada aparelho foi capturado na resolução de 5312×2988 pixeis, totalizando 600 imagens. Alguns exemplos de imagens

capturadas são encontrados na Figura 4.2.

Quantidade	Tipo	Resolução (pixels)
1	grid	1211 × 674
1	xadrez	1211 × 674
1	índio	1024 × 768
3	barra de cores	1600 × 900
18	cores primárias e secundárias	1600 × 900
8	fotos	1600 × 900
4	fotos	2048 × 1152
4	menus	1920 × 1080

Tabela 4.3: Imagens de referência da base de dados sob condições controladas.



(a) imagem adquirida - vermelho.

(b) imagem adquirida - menu 1.

Figura 4.2: Imagens da base de dados sob condições controladas.

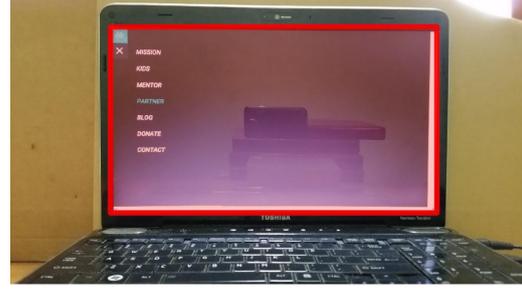
4.2.3 Rótulos Manuais das Bases de Dados

Cada uma das imagens das bases de dados descritas nas Seções 4.2.1 e 4.2.2 foi rotulada manualmente pelo autor. O rótulo corresponde aos quatro vértices (marcações em cruz de cor amarela na imagem da Figura 4.3(a), denominadas por *rótulo manual*) da área que efetivamente apresenta o conteúdo do monitor/TV. Esta área representa popularmente a *tela* do monitor/TV (veja área realçada em vermelho na imagem da Figura 4.3(b)). Desta forma, o *rótulo manual* de cada imagem é considerado *ground truth* e pode ser usado para comparação com as coordenadas encontradas por um sistema de detecção.

Nesta dissertação, consideramos a denominação *rótulo automático* para as coordenadas dos vértices fornecidos como saída pelo algoritmo/sistema que realiza a detecção da tela do monitor/TV. Assim, para cada imagem, temos um rótulo manual e um rótulo automático.



(a) imagem de um notebook com coordenadas marcadas



(b) imagem de um notebook com região da tela destacada

Figura 4.3: Imagem rotulada manualmente.

4.3 Métricas de Desempenho

As métricas de avaliação do desempenho da metodologia foram:

- Taxa de detecção: para cada imagem, a detecção é considerada um sucesso se todos os pontos encontrados pela metodologia, o *rótulo automático*, corresponderem aos pontos do *rótulo manual*, conforme tolerância especificada. Isto pode ser matematicamente representado por:

$$f = \begin{cases} 1, & \text{se } d_i < \alpha * \max(l, c), \forall i \\ 0, & \text{se } d_i \geq \alpha * \max(l, c), \forall i \end{cases} \quad (4.1)$$

Na equação (4.1), f é o valor que indica sucesso ou falha na detecção daquela imagem, l corresponde ao comprimento do retângulo do *rótulo manual*, c corresponde à largura e α é um valor de tolerância. d_i é a distância entre uma coordenada no *rótulo automático* e a coordenada correspondente no *rótulo manual* e é dado pela expressão:

$$d_i = \sqrt{(gt_i(x) - c_i(x))^2 + (gt_i(y) - c_i(y))^2} \quad (4.2)$$

Na equação (4.2), $c_i(x, y)$ é um vetor com as coordenadas do *rótulo automático*, $gt_i(x, y)$ é um vetor com as coordenadas do *rótulo manual* e d_i é a distância entre as coordenadas com $i = [1, 2, 3, 4]$.

- LAE (*least average error*): esta é uma medida da diferença entre duas imagens, calculada pixel a pixel. Considerando duas imagens I_1 e I_2 , o valor do LAE é

dado pela expressão [4]:

$$LAE = \sum_{c=R,G,B} \sum_{x=1}^N \sum_{y=1}^M \left| \frac{I_1(x,y) - \mu_{I_1}}{\sigma_{I_1}} - \frac{I_2(x,y) - \mu_{I_2}}{\sigma_{I_2}} \right| \quad (4.3)$$

Na equação (4.3), c é o canal da imagem (em uma imagem RGB são 3, um para a cor vermelha, um para a azul e um para a verde) μ_{I_1} e μ_{I_2} são as médias, σ_{I_1} e σ_{I_2} são os desvios padrão das imagens I_1 e I_2 , ambas de tamanho $M \times N$.

- **NCC (normalized cross-correlation)**: o método NCC é baseado na correlação cruzada para medir a similaridade entre duas imagens. A similaridade entre imagens I_1 e I_2 é calculada segundo a expressão [4]:

$$NCC = \frac{\sum_{x=1}^N \sum_{y=1}^M (I_1(x,y) - \mu_{I_1})(I_2(x,y) - \mu_{I_2})}{\sqrt{\sum_{x=1}^N \sum_{y=1}^M (I_1(x,y) - \mu_{I_1})^2} \sqrt{\sum_{x=1}^N \sum_{y=1}^M (I_2(x,y) - \mu_{I_2})^2}} \quad (4.4)$$

- **NCC-BB (normalized cross-correlation using blocks)**: Este método é usado para calcular a similaridade relativa entre as imagens, utilizando a equação (4.4). A Figura 4.4 ilustra os passos do método, que serão descritos a seguir [4]:

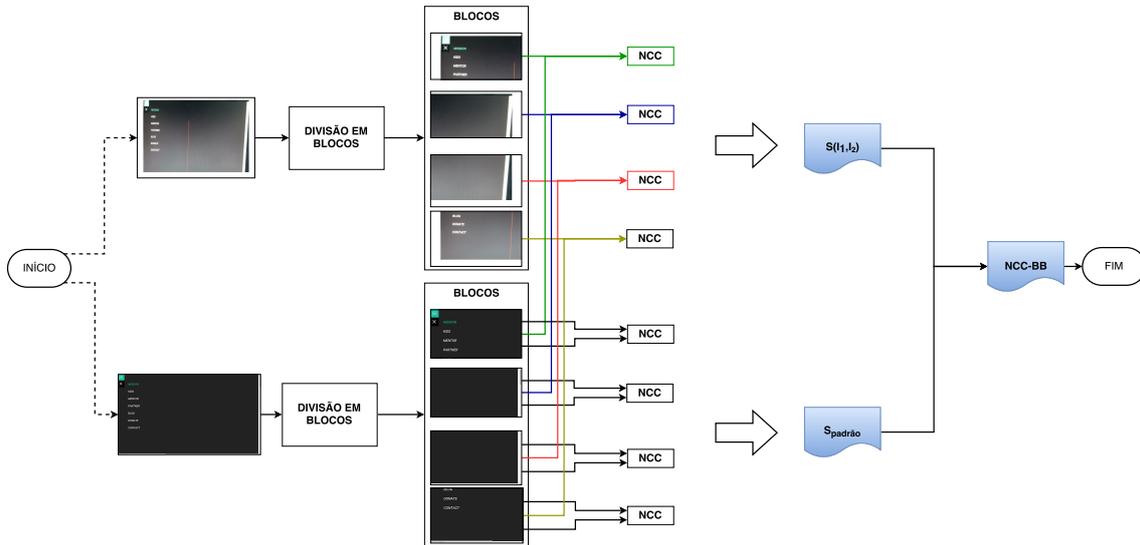


Figura 4.4: Processo do NCC-BB para uma imagem adquirida e uma imagem de referência.

- Dividir as imagens em 16 blocos menores de mesmo tamanho;

- Para a imagem de referência, calcular o NCC de cada um dos blocos com ele mesmo, segundo a equação (4.4). Estes valores de similaridade são então guardados como um conjunto de valores padrão S_{padrao} ;
- Para cada imagem adquirida submetida ao método, calcular o NCC de cada um dos seus blocos com o bloco correspondente na imagem de referência. Estes valores de similaridade são guardados como um conjunto de valores S_{I_1, I_2} ;
- Os valores de similaridade encontrados são comparados bloco a bloco e somados para compor o valor de similaridade relativa. Sendo S_{I_1, I_2} a similaridade entre as imagens I_1 e I_2 , a similaridade relativa pode ser calculada através da equação (4.5);

$$S = \max_{\text{blocos}} |S_{I_1, I_2} - S_{padrao}| \quad (4.5)$$

4.4 Resultados Experimentais

Nesta seção, descrevemos brevemente os experimentos realizados, assim como os resultados obtidos. Ela é organizada da seguinte forma: para cada experimento, inicialmente é feita uma descrição do procedimento experimental. A seguir, são mostrados os resultados obtidos e por último é feita uma análise dos resultados.

4.4.1 Primeiro Experimento

No primeiro experimento, utilizamos a metodologia para detectar um retângulo considerado tela em cada imagem de entrada. Como entrada, utilizamos 333 imagens da base de dados sob condições não controladas (Seção 4.2.1), denominada abreviadamente por base 1, e 230 imagens da base de dados sob condições controladas (Seção 4.2.2), denominada por base 2, totalizando 563 imagens. O objetivo deste experimento é comparar a eficiência da metodologia em detectar retângulos em ambiente controlado e não controlado.

Para cada imagem executamos dois algoritmos. O primeiro dá como resposta um conjunto de coordenadas da região considerada tela em uma determinada imagem colorida. Ele identifica as coordenadas utilizando a metodologia ex-

posta no capítulo 3. O segundo algoritmo compara dois conjuntos de coordenadas, considerando-os *semelhantes* ou *não semelhantes*, segundo a métrica exposta na seção 4.3. Assim, fornecemos ao conjunto de algoritmos uma imagem das bases de dados, junto com as coordenadas do seu rótulo manual. O primeiro algoritmo dá como resposta as coordenadas da tela detectada naquela imagem. O segundo algoritmo compara as coordenadas encontradas pelo primeiro algoritmo com as coordenadas do rótulo manual da imagem, fornecendo como saída a resposta *tela detectada* ou *tela não detectada*. Relacionando cada imagem com um rótulo (*tela detectada* ou *tela não detectada*) podemos obter a taxa de detecção.

Dispomos os resultados da seguinte maneira. A Tabela 4.4 apresenta a taxa de detecção, segundo a definição apresentada na Seção 4.3, das imagens das bases 1 e 2, em comparação com a taxa obtida submetendo as mesmas imagens ao algoritmo de detecção de retângulos descrito em [3]. Exemplos de saídas deste experimento são mostrados na Figura 4.6. As Tabelas 4.5, 4.6 e 4.7 apresentam a taxa de detecção para a base 1, respectivamente discriminada segundo o monitor testado, o conteúdo mostrado e a resolução em que a imagem foi capturada. As Tabelas 4.8 e 4.9 apresentam o resultado para a base 2, discriminado segundo monitor e conteúdo.

Ao final desta seção, apresentamos uma análise dos resultados obtidos no experimento, assim como exemplos de imagens processadas utilizando a metodologia.

Método	Base	Imagens	Telas Detectadas	Taxa de Detecção
Proposto	1	333	42	12,61%
Jung <i>et al.</i> [3]	1	333	0	00,00%
Proposto	2	230	110	47,83%
Jung <i>et al.</i> [3]	2	230	0	00,00%

Tabela 4.4: Taxa de detecção nas duas bases de dados.

4.4.1.1 Análise de resultados

A taxa de detecção obtida neste experimento não pode ser considerada boa, entretanto a análise dos resultados deste experimento demonstra que a metodologia é interessante em algumas situações. Comparando com métodos já existentes na literatura, pode-se inferir que os resultados obtidos nos experimentos são devido à dificuldade apresentada pelas bases de dados concebidas, como pode ser visto na

Monitor	Imagens	Telas Detectadas	Taxa de Detecção
1	36	16	44,44%
2	36	0	00,00%
3	36	15	41,67%
4	36	7	19,44%
5	36	3	8,33%
6	36	0	00,00%
7	36	1	2,78%
8	36	0	00,00%
9	36	0	00,00%
10	9	0	00,00%

Tabela 4.5: Taxa de detecção por monitor na base 1.

Conteúdo	Imagens	Telas Detectadas	Taxa de Detecção
menu	225	36	16,00%
video	108	6	5,56%

Tabela 4.6: Taxa de detecção por conteúdo na base 1.

Resolução	Imagens	Telas Detectadas	Taxa de Detecção
1 (5312 × 2988)	111	15	13,51%
2 (3264 × 2448)	111	15	13,51%
3 (2048 × 1152)	111	12	5,56%

Tabela 4.7: Taxa de detecção por resolução na base 1.

Monitor	Imagens	Telas Detectadas	Taxa de Detecção
1	40	19	47,50%
2	40	20	50,00%
3	40	15	37,50%
4	40	18	45,00%
5	40	22	55,00%
6	30	16	53,33%

Tabela 4.8: Taxa de detecção por monitor na base 2.

Conteúdo	Imagens	Telas Detectadas	Taxa de Detecção
grid	6	0	0,00%
xadrez	6	0	0,00%
indio	6	1	16,67%
colorbar	18	10	55,56%
cor	108	90	83,33%
foto	66	6	9,09%
menu	20	3	15,00%

Tabela 4.9: Taxa de detecção por conteúdo na base 2.

Tabela 4.4.

O experimento mostra um maior sucesso da metodologia em imagens com maior contraste entre o *background* e o conteúdo do aparelho. isto pode ser visto na Tabela 4.5, em que as imagens do monitor 1 e 3 tiveram uma taxa de detecção de mais de 40% enquanto monitores como os 8 e 9 não tiveram sucesso. Este resultado é confirmado pela Tabela 4.3, que mostra as taxas de detecção por monitor da segunda base de dados. Não há grande diferença no *background* em monitores diferentes e não há grande diferença na taxa de detecção entre monitores.

Os experimentos mostram que a metodologia tem maior sucesso na detecção de conteúdo quando ele é composto por imagens monocromáticas, se comparado com conteúdo composto por imagens mais complexas. A Tabela 4.6 mostra a variação da detecção, por conteúdo mostrado. Nela, podemos ver que a metodologia tem mais facilidade de detectar menus que frames de vídeo. Resultado semelhante é mostrado na Tabela 4.9. A taxa de detecção com o conteúdo sendo imagens monocromáticas é maior que com fotos ou menus, que são imagens mais complexas.

4.4.2 Segundo Experimento

No segundo experimento, utilizamos a metodologia para detectar o conteúdo da tela em cada imagem de entrada e classificamos a imagem comparando este conteúdo com imagens de referência. Como entrada, utilizamos 36 imagens adquiridas da primeira base de dados que representam menus, assim como imagens de referência que mostram seleções de menu. O objetivo deste experimento é verificar o desempenho da metodologia em um sistema de comparação de imagens.

Para cada imagem, executamos um sistema de algoritmos cujo funcionamento é ilustrado na Figura 4.5 e descrito em termos gerais a seguir. Após detectar a tela, comparamos a imagem adquirida com diversas imagens de referência e atribuímos pontuações a cada comparação, segundo as métricas LAE, NCC e NCC-BB, definidas na Seção 4.3. Para este experimento, utilizamos quatro classes que correspondem ao conteúdo presente na tela nas imagens adquiridas. Desta forma, temos quatro seleções possíveis de um mesmo menu, mostradas na Figura 4.1. As imagens têm o mesmo tamanho e a mesma cor. A única diferença entre elas é a coloração que corresponde à seleção naquela classe. Para a imagem ser considerada correta, o

sistema deve atribuir à imagem a classe que corresponde à seleção de menu que ela apresenta.

O sistema tem como saída, para cada imagem adquirida, um conjunto de pontuações, conforme descrito na Seção 4.3, referentes à comparação entre o conteúdo da tela e as imagens de referência. Estas pontuações podem ser comparadas a fim de classificar uma imagem conforme a sua similaridade com as imagens.

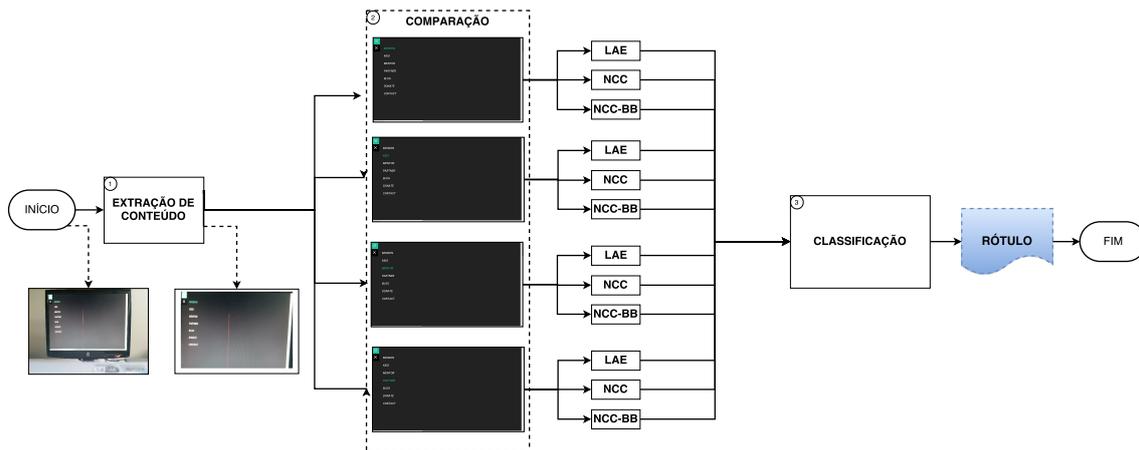


Figura 4.5: Outline do segundo experimento.

Dispomos os resultados da seguinte maneira. A Tabela 4.10 mostra os dados da classificação nas imagens cujo conteúdo foi extraído com sucesso no experimento 1. Nesta tabela temos, em cada coluna o monitor a qual a imagem pertence, o número de imagens daquele monitor cujo conteúdo foi extraído com sucesso, o número de imagens classificadas corretamente, o número de imagens classificadas incorretamente e o número de imagens em que as métricas não foram capazes de escolher uma classe. Na Tabela 4.11 temos os resultados da classificação por métrica utilizada, assim como a classificação final utilizando as três métricas. Na Tabela 4.12 temos os resultados da classificação por classe, assim como a classificação final utilizando as três métricas. A Tabela 4.13 mostra algumas pontuações obtidas comparando uma imagem que representa o menu na primeira seleção com quatro imagens de referência. Em cada coluna, os valores são comparados e o valor que representa a imagem de referência mais semelhante à imagem testada é destacado em azul. A Tabela 4.14 mostra as pontuações da mesma maneira, com a diferença que a imagem testada representa a segunda seleção do menu. Na Tabela 4.15, a imagem representa a terceira seleção, e na Tabela 4.16 a imagem representa a quarta seleção.

Monitor	Imagens	OK	No OK	NC
1	14	6	7	1
3	13	8	4	1
4	5	2	3	0
5	3	2	1	0
7	1	1	0	0

Tabela 4.10: Classificação nas imagens cujo conteúdo foi extraído com sucesso. OK - imagens classificadas corretamente. Não OK - imagens classificadas incorretamente. NC - imagens Não Classificadas

Métrica	Imagens	OK	Taxa de Acerto
LAE	36	17	47,22%
NCC	36	18	50,00%
NCC-BB	36	22	61,11%
Class	36	19	52,78%

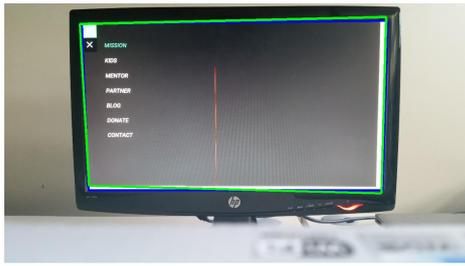
Tabela 4.11: Desempenho da classificação por métrica. OK - Imagens classificadas corretamente

Classe	Imagens	OK	Taxa de Acerto
menu 1	8	6	75,00%
menu 2	9	3	33,33%
menu 3	10	5	50,00%
menu 4	9	5	55,56%

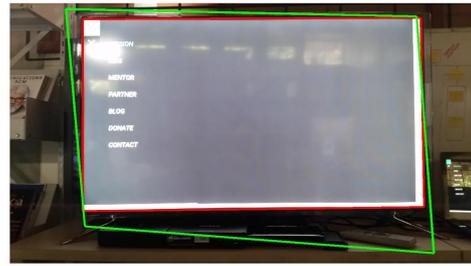
Tabela 4.12: Desempenho da classificação por classe. OK - Imagens classificadas corretamente

4.4.2.1 Análise de Resultado

As tabelas mostram que o experimento de comparação entre imagens obteve uma taxa de classificação correta das imagens de 52,78%, para as imagens cujo conteúdo foi capturado corretamente. Podemos inferir que esta técnica de classificação não é confiável quando as classes correspondem a imagens muito semelhantes. As três métricas têm desempenho individual ligeiramente diferente, com vantagem para o NCC-BB, que tem desempenho 15,79% melhor que o da classificação e 29,41% melhor que a técnica LAE, como pode ser conferido na Tabela 4.11. A Tabela 4.12 mostra há disparidades na classificação entre as diferentes classes, com a classe Menu 1 com o melhor resultado e a classe Menu 2 com o pior. Exemplos destas classificações podem ser vistos nas Tabelas 4.13, 4.14, 4.15 e 4.16.



(a) base 1 monitor 01 - conteúdo detectado.



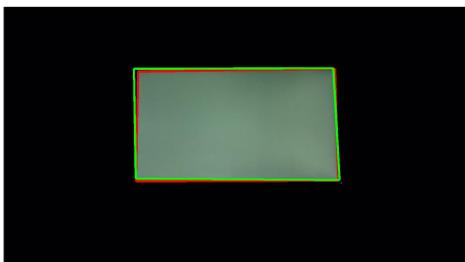
(b) base 1 monitor 05 - conteúdo não detectado.



(c) base 1 monitor 03 - conteúdo detectado.



(d) base 1 monitor 07 - conteúdo não detectado.



(e) base 2 monitor 01 - conteúdo detectado.



(f) base 2 monitor 01 - conteúdo não detectado.



(g) base 2 monitor 02 - conteúdo detectado.



(h) base 2 monitor 02 - conteúdo não detectado.

Figura 4.6: Imagens processadas com tela detectada. Nas imagens à esquerda a detecção foi considerada um sucesso. Nas imagens da direita a detecção não foi considerada satisfatória. Em vermelho temos a área delimitada pelo *rótulo manual*. Em verde temos a área delimitada pelo *rótulo automático*

Imagem de Referência	monitor 5			monitor 6			monitor 9			monitor 10			monitor 12			monitor 13		
	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB
menu 1	4913500	-0.51389	-3.2467	4951300	-0.33074	-3.0283	4171400	-0.13891	-2.9587	4187400	-0.27962	-3.0896	3811900	-0.083533	-3.1228	4462200	0.13217	-2.9202
menu 2	4919100	-0.52371	-3.2617	4953400	-0.33297	-3.0313	4174000	-0.14408	-2.966	4189500	-0.28262	-3.0945	3814600	-0.091609	-3.1316	4464500	0.12588	-2.9258
menu 3	4915800	-0.52313	-3.257	4952200	-0.33342	-3.0322	4171900	-0.14193	-2.9646	4187700	-0.28064	-3.095	3812100	-0.091186	-3.1314	4462400	0.13067	-2.926
menu 4	4914900	-0.522	-3.2544	4951700	-0.33259	-3.031	4171700	-0.14152	-2.964	4187400	-0.28023	-3.0946	3812600	-0.09335	-3.135	4462200	0.13015	-2.9276

Tabela 4.13: Tabela com valores das métricas pra uma imagem representando o menu 1

Imagem de Referência	monitor 5			monitor 6			monitor 9			monitor 10			monitor 12			monitor 13		
	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB
menu 1	4207600	-0.26032	-3.0947	4706300	0.17554	-2.6621	4236000	-0.18188	-3.0081	4239100	-0.29627	-3.0753	4050500	0.074942	-2.9381	4588900	0.070431	-3.0226
menu 2	4207500	-0.25384	-3.0902	4707100	0.17755	-2.6603	4236400	-0.17963	-3.0066	4240000	-0.29468	-3.0748	4049700	0.081405	-2.9336	4590300	0.071505	-3.0214
menu 3	4206000	-0.25644	-3.094	4706000	0.17495	-2.664	4236100	-0.18161	-3.0075	4239300	-0.29632	-3.0762	4048700	0.081003	-2.9359	4589200	0.070226	-3.0235
menu 4	4205500	-0.25557	-3.0927	4705500	0.17592	-2.6626	4236000	-0.1818	-3.0074	4239000	-0.29629	-3.0757	4048400	0.081245	-2.9353	4588600	0.071448	-3.0215

Tabela 4.14: Tabela com valores das métricas pra uma imagem representando o menu 2

Imagem de Referência	monitor 5			monitor 6			monitor 9			monitor 10			monitor 12			monitor 13		
	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB
menu 1	4862600	-0.52917	-3.2141	4443600	0.060699	-2.9702	3984300	0.045629	-2.8268	4222900	-0.26375	-3.0834	3962200	0.68342	-2.5930	4468600	0.14274	-2.9317
menu 2	4864300	-0.52704	-3.2133	4444800	0.060964	-2.9713	3986200	0.042193	-2.8318	4224100	-0.26276	-3.0839	3962400	0.68321	-2.5938	4468500	0.14959	-2.9218
menu 3	4862400	-0.52600	-3.2142	4442800	0.063311	-2.9711	3983900	0.04632	-2.8295	4223100	-0.26387	-3.0859	3961500	0.68344	-2.5958	4466300	0.15658	-2.923
menu 4	4862300	-0.52612	-3.2144	4442700	0.063034	-2.9716	3983900	0.046229	-2.8300	4222900	-0.26392	-3.0856	3961900	0.68144	-2.6004	4466000	0.15633	-2.9243

Tabela 4.15: Tabela com valores das métricas pra uma imagem representando o menu 3

Imagem de Referência	monitor 5			monitor 6			monitor 9			monitor 10			monitor 12			monitor 13		
	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB	LAE	NCC	NCC-BB
menu 1	4383800	-0.45810	-3.1812	5045100	-0.46912	-3.1503	4271600	-0.092546	-2.9698	3973900	-0.022857	-2.9103	4143700	0.011166	-3.0243	4725900	0.28163	-2.9432
menu 2	4383600	-0.45044	-3.1741	5046800	-0.4692	-3.1505	4271800	-0.090072	-2.9681	3974900	-0.020604	-2.9068	4143400	0.017021	-3.0206	4726500	0.28326	-2.9437
menu 3	4382600	-0.45479	-3.1766	5045200	-0.46867	-3.1512	4271700	-0.092234	-2.9691	3973200	-0.017753	-2.9079	4142300	0.01705	-3.0223	4725500	0.2841	-2.9438
menu 4	4381400	-0.45320	-3.1728	5044600	-0.46763	-3.1498	4271400	-0.091729	-2.9678	3973000	-0.016961	-2.9071	4142000	0.017354	-3.0217	4725000	0.28541	-2.9419

Tabela 4.16: Tabela com valores das métricas pra uma imagem representando o menu 4

Capítulo 5

Conclusões

Nesta dissertação, investigamos o desempenho de uma metodologia de extração de conteúdo de TVs e monitores que utiliza um modelo baseado no campo aleatório de Markov (MRF). A metodologia representa um trabalho de base no contexto de detecção de falhas em monitores e TVs, um problema de crescente importância na indústria de eletrônicos. Foram feitas duas contribuições principais.

Na primeira contribuição, descrita no Capítulo 3, modificamos técnicas de detecção de bordas e detecção de retângulos de uma metodologia de detecção de objetos quadrangulares, a fim de criar uma metodologia de extração de conteúdo de monitores e TVs em imagens coloridas. Nesta metodologia, utilizamos um modelo baseado no MRF e características de telas de TV e monitores para detectar a área que corresponde ao conteúdo mostrado no aparelho cuja imagem foi adquirida.

Como segunda contribuição, apresentada no Capítulo 4, concebemos duas bases de dados com imagens de monitores e TVs, uma sob condições controladas e outra não, compostas por imagens coloridas de TVs/monitores, assim como imagens de referência com o conteúdo apresentado nas TVs/monitores no momento da aquisição das imagens. A base de dados sob condições não controladas é composta por 510 imagens, sendo 6 imagens de referência com o conteúdo mostrado nos aparelhos durante a aquisição das 504 imagens. A base de dados sob condições controladas é composta por 640 imagens, sendo 40 imagens de referência e 600 imagens adquiridas. Todas as imagens foram rotuladas manualmente com as coordenadas dos quatro vértices da região que corresponde à tela de cada aparelho.

O desempenho da metodologia para extração de conteúdo de TV e monito-

res proposta foi avaliado através de dois experimentos. No primeiro, utilizamos a metodologia para extrair o conteúdo em imagens das duas bases de dados concebidas. No segundo experimento, comparamos o conteúdo extraído através de métricas descritas no Capítulo 4. Os experimentos realizados demonstraram uma baixa taxa de detecção de algoritmos utilizando a metodologia, ainda que melhores do que utilizando outras metodologias de detecção de retângulos. Analisando os dados, foi notado um melhor desempenho da metodologia com imagens com alto contraste entre o conteúdo da tela e o *background*. Outro dado interessante é o melhor desempenho da metodologia em imagens cujo conteúdo da TV/monitor é composto por imagens mais simples, o que pode ser um indicativo de que imagens utilizar para testes no caso do uso desta metodologia em sistemas de inspeção automática. As métricas utilizadas para comparação de imagens tiveram resultados semelhantes, não sendo possível definir a partir dos experimentos qual a métrica mais adequada para comparação entre conteúdos.

Por fim, podemos comentar que a metodologia proposta foi aplicada em um problema de extração e comparação de conteúdo de TVs e monitores. Embora os resultados apresentados não sejam competitivos, esta metodologia pode ser encarada como uma base para o estudo e o desenvolvimento de técnicas sobre este problema.

5.1 Trabalhos Futuros

- As equações de energia do modelo MRF, assim como o algoritmo gradiente descendente, têm muitos parâmetros que podem influenciar diretamente na taxa de detecção. Encontrar a configuração que produz melhores resultados pode ter grande impacto sobre o custo temporal e a eficácia de um sistema de inspeção automática. Assim, propomos como trabalho futuro investigar estratégias para busca de parâmetros, utilizando técnicas de *grid search* [74,75] que resultem em melhor desempenho da metodologia;
- Foi investigado o desempenho da metodologia com imagens de objetos capturados em um ambiente totalmente escuro. Seria interessante investigar o desempenho da metodologia capturando imagens dos objetos em um ambiente neutro sob diversos níveis de iluminação, para isso criando uma nova base de

dados.

Referências Bibliográficas

- [1] ILLINGWORTH, J., KITTLER, J., “A survey of the Hough transform”, *Computer vision, graphics, and image processing*, v. 44, n. 1, pp. 87–116, 1988.
- [2] LIU, Y., IKENAGA, T., GOTO, S., “An MRF model-based approach to the detection of rectangular shape objects in color images”, *Signal Processing*, v. 87, n. 11, pp. 2649–2658, 2007.
- [3] JUNG, C. R., SCHRAMM, R., “Rectangle detection based on a windowed Hough transform”. In: *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pp. 113–120, 2004.
- [4] KASTELAN, I., KATONA, M., MARIJAN, D., et al., “Automated optical inspection system for digital TV sets”, *EURASIP Journal on Advances in Signal Processing*, v. 2011, n. 1, pp. 1–17, 2011.
- [5] GONZALEZ, R. C., WOODS, R. E., *Digital image processing (3rd Edition)*. Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2006.
- [6] ZHU, Y., CARRAGHER, B., MOUCHE, F., et al., “Automatic particle detection through efficient Hough transforms”, *IEEE Transactions on Medical Imaging*, v. 22, n. 9, pp. 1053–1062, 2003.
- [7] JAIN, R., KASTURI, R., SCHUNCK, B. G., *Machine vision*. v. 5. McGraw-Hill New York, 1995.
- [8] RUSSO, F., LAZZARI, A., “Color edge detection in presence of Gaussian noise using nonlinear prefiltering”, *IEEE Transactions on Instrumentation and Measurement*, v. 54, n. 1, pp. 352–358, 2005.

- [9] DENG, G., CAHILL, L. W., “An adaptive Gaussian filter for noise reduction and edge detection”. In: *Proceedings of the IEEE International Conference on Record on Nuclear Science Symposium and Medical Imaging Conference*, pp. 1615–1619 vol.3, Oct 1993.
- [10] LAINE, A. F., CHANG, C.-M., “De-noising via wavelet transforms using steerable filters”. In: *IEEE International Symposium on Circuits and Systems*, v. 3, pp. 1956–1959 vol.3, Apr 1995.
- [11] LAGUNOVSKY, D., ABLAMEYKO, S., “Straight-line-based primitive extraction in grey-scale object recognition”, *Pattern Recognition Letters*, v. 20, n. 10, pp. 1005–1014, 1999.
- [12] WEN-BING, T., JIN-WEN, T., JIAN, L., “A new approach to extract rectangular building from aerial urban images”. In: *Proceedings of the International Conference on Signal Processing*, v. 1, pp. 143–146, 2002.
- [13] ROTH, G., LEVINE, M. D., “Extracting geometric primitives”, *CVGIP: Image Understanding*, v. 58, n. 1, pp. 1–22, 1993.
- [14] ROTH, G., LEVINE, M. D., “Geometric primitive extraction using a genetic algorithm”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 16, n. 9, pp. 901–905, 1994.
- [15] WANG, G., MIAO, F., ZHANG, W., et al., “A dynamic and interactive diagnosing and testing method for development of digital TV receiver system”. In: *Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, v. 3, pp. 1107–1112, 2007.
- [16] SUNG, A., CHOI, B., “An interaction testing technique between hardware and software in embedded systems”. In: *Proceedings of the Software Engineering Conference*, pp. 457–464, 2002.
- [17] RAU, A. N., “Automated test system for digital TV receivers”. In: *Proceedings of the International Conference on Consumer Electronic*, pp. 228–229, 2000.

- [18] RAMA, A., ALUJAS, R., TARRES, F., “Fast and Robust Graphic Character Verification System for TV Sets”. In: *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 19–19, June 2007.
- [19] VERCAMMEN, N., STAELENS, N., ROMBAUT, A., et al., “Extensive video quality evaluation: A scalable video testing platform”. In: *Proceedings of the International Conference on Computer and Information Technology*, pp. 91–97, Dec 2008.
- [20] MARIJAN, D., ZLOKOLICA, V., TESLIC, N., et al., “Automatic functional TV set failure detection system”, *IEEE Transactions on Consumer Electronics*, v. 56, n. 1, pp. 125–133, 2010.
- [21] KATONA, M., KASTELAN, I., PEKOVIC, V., et al., “Automatic black box testing of television systems on the final production line”, *IEEE Transactions on Consumer Electronics*, v. 57, n. 1, pp. 224–231, 2011.
- [22] LUTTON, E., MARTINEZ, P., “A genetic algorithm for the detection of 2D geometric primitives in images”. In: *Proceedings of the International Conference on Pattern Recognition, Computer Vision and Image Processing*, v. 1, pp. 526–528, 1994.
- [23] CHO, H.-H., CHOI, C.-H., KWON, B.-H., et al., “A design of contrast controller for image improvement of multi-gray scale image”. In: *Proceedings of the IEEE Asia Pacific Conference on ASIC*, pp. 131–133, 2000.
- [24] WOJCIK, Z. M., “Application of rough sets for edge enhancing image filters”. In: *Proceedings of the IEEE International Conference on Image Processing*, v. 2, pp. 525–529 vol.2, Nov 1994.
- [25] CHOONG, M. Y., KOW, W. Y., CHIN, Y. K., et al., “Image segmentation via normalised cuts and clustering algorithm”. In: *Proceedings of the IEEE International Conference on Control System, Computing and Engineering*, pp. 430–435, Nov 2012.

- [26] KIM, T., CHO, S., YOON, J., et al., “Pose Robust Human Detection in Depth Image Using Four Directional 2D Elliptical Filters”. In: *Proceedings of the IEEE International Symposium on Multimedia*, pp. 130–135, Dec 2009.
- [27] SAAD, M. H. M., HUSSAIN, A., KONG, W., et al., “SESRG-InViSS: Image and video data set for human pose, action, activity and behaviour detection”. In: *Proceedings of the IEEE International Conference on System Engineering and Technology*, pp. 37–42, Aug 2013.
- [28] LIU, Y., WANG, Q., “A new simple human abnormal action detection based on static images”. In: *Proceedings of the IEEE International Conference on Computer Science and Automation Engineering*, v. 1, pp. 578–581, June 2011.
- [29] LI, S. Z., *Markov random field modeling in image analysis*. Springer Science & Business Media, 2009.
- [30] BARROW, H. G., TENENBAUM, J. M., “Interpreting line drawings as three-dimensional surfaces”, *Artificial intelligence*, v. 17, n. 1, pp. 75–116, 1981.
- [31] JOSEPH, R. P., SINGH, C. S., MANIKANDAN, M., “Brain tumor MRI image segmentation and detection in image processing”, *International Journal of Research in Engineering and Technology*, v. 3, pp. 1–5, 2014.
- [32] LAU, K.-K., ROBERTS, S., BIRO, D., et al., “An edge-detection approach to investigating pigeon navigation”, *Journal of theoretical biology*, v. 239, n. 1, pp. 71–78, 2006.
- [33] CHENG, Y., JOHNSON, A. E., MATTHIES, L. H., et al., “Optical landmark detection for spacecraft navigation”, *Advances in the Astronautical Sciences*, v. 114, n. 3, pp. 1785–1803, 2003.
- [34] SHRIVAKSHAN, G., CHANDRASEKAR, C., “A comparison of various edge detection techniques used in image processing”, *International Journal of Computer Science Issues*, v. 9, n. 5, pp. 272–276, 2012.
- [35] CANNY, J., “A computational approach to edge detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , n. 6, pp. 679–698, 1986.

- [36] PREWITT, J. M., “Object enhancement and extraction”, *Picture processing and Psychopictorics*, v. 10, n. 1, pp. 15–19, 1970.
- [37] VINCENT, O., FOLORUNSO, O., “A descriptive algorithm for sobel image edge detection”. In: *Proceedings of the Informing Science & IT Education Conference*, v. 40, pp. 97–107, 2009.
- [38] ROBERTS, L. G., *Machine perception of three-dimensional soups*, Ph.D. Thesis, Massachusetts Institute of Technology, 1963.
- [39] MAINI, R., AGGARWAL, H., “Study and comparison of various image edge detection techniques”, *International journal of image processing*, v. 3, n. 1, pp. 1–11, 2009.
- [40] MARR, D., HILDRETH, E., “Theory of edge detection”, *Proceedings of the Royal Society of London B: Biological Sciences*, v. 207, n. 1167, pp. 187–217, 1980.
- [41] HARALICK, R. M., “Digital step edges from zero crossing of second directional derivatives”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , n. 1, pp. 58–68, 1984.
- [42] BONGIOVANNI, G., LUCCIO, F., ZORAT, A., “The Discrete Equation of the Straight Line”, *IEEE Transactions on Computers*, v. C-24, n. 3, pp. 310–313, March 1975.
- [43] COPELAND, A. C., RAVICHANDRAN, G., TRIVEDI, M. M., “Localized Radon transform-based detection of ship wakes in SAR images”, *IEEE Transactions on Geoscience and Remote Sensing*, v. 33, n. 1, pp. 35–45, 1995.
- [44] ZHANG, Q., COULOIGNER, I., “Accurate centerline detection and line width estimation of thick lines using the radon transform”, *IEEE Transactions on Image Processing*, v. 16, n. 2, pp. 310–316, 2007.
- [45] STEGER, C., “An Unbiased Detector of Curvilinear Structures”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 2, pp. 113–125, 1998.

- [46] KASS, M., WITKIN, A., TERZOPOULOS, D., “Snakes: Active contour models”, *International journal of computer vision*, v. 1, n. 4, pp. 321–331, 1988.
- [47] HART, P. E., “How the Hough transform was invented [DSP History]”, *Signal Processing Magazine, IEEE*, v. 26, n. 6, pp. 18–22, 2009.
- [48] DEANS, S. R., “Hough transform from the Radon transform”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , n. 2, pp. 185–188, 1981.
- [49] O’GORMAN, F., CLOWES, M., “Finding picture edges through collinearity of feature points”, *IEEE Transactions on Computers*, v. 100, n. 4, pp. 449–456, 1976.
- [50] LAM, L. T., LAM, W. C., LEUNG, D. N., “A knowledge-based boundary convergence algorithm for line detection”, *Pattern recognition letters*, v. 15, n. 4, pp. 383–392, 1994.
- [51] KANG, W.-J., DING, X.-M., CUI, J.-W., et al., “Fast straight-line extraction algorithm based on improved Hough transform [J]”, *Opto-Electronic Engineering*, v. 3, pp. 023, 2007.
- [52] LI, C., WANG, Z., LI, L., “An improved ht algorithm on straight line detection based on freeman chain code”. In: *Proceedings of the IEEE International Congress on Image and Signal Processing*, pp. 1–4, 2009.
- [53] DUDA, R. O., HART, P. E., “Use of the Hough transformation to detect lines and curves in pictures”, *Communications of the ACM*, v. 15, n. 1, pp. 11–15, 1972.
- [54] RANGARAJAN, A., CHELLAPPA, R., “Markov random field models in image processing”, 1995.
- [55] MEAS-YEDID, V., TILIE, S., OLIVO-MARIN, J. C., “Color image segmentation based on Markov random field clustering for histological image analysis”. In: *Proceedings of the IEEE International Conference on Pattern Recognition*, v. 1, pp. 796–799 vol.1, 2002.

- [56] JEFFS, B. D., GARDINER, A. H., “Markov random field image prior models for map reconstruction of magnetoencephalogram images”. In: *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, v. 1, pp. 314–318 vol.1, Nov 1998.
- [57] HE, H., KONDI, L. P., “MAP based resolution enhancement of video sequences using a Huber-Markov random field image prior model”. In: *Proceedings of the IEEE International Conference on Image Processing*, v. 2, pp. II–933–6 vol.3, Sept 2003.
- [58] HAMMERSLEY, J. M., CLIFFORD, P. E., “Markov random fields on finite graphs and lattices”, Unpublished manuscript, 1971.
- [59] LI, L., WANG, Y., TANG, Z., et al., “Automatic comic page segmentation based on polygon detection”, *Multimedia Tools and Applications*, v. 69, n. 1, pp. 171–197, 2014.
- [60] LEE, K. K., YU, Y. K., WONG, K. H., “Multiple quadrilateral detection for projector-camera system applications”. In: *Proceedings of the IEEE Conference on Industrial Electronics and Applications*, pp. 1949–1954, 2015.
- [61] HAYET, J.-B., LERASLE, F., DEVY, M., “A visual landmark framework for mobile robot navigation”. v. 25, n. 8, pp. 1341–1351, Elsevier, 2007.
- [62] ARANA-AREJOLALEIBA, N., LERASLE, F., BRIOT, M., et al., “A smart sensor based visual landmarks detection for indoor robot navigation”. In: *Proceedings of the IEEE International Conference on Pattern Recognition*, v. 2, pp. 848–851, 2002.
- [63] SAEEDI, P., ZWICK, H., “Automatic building detection in aerial and satellite images”. In: *Proceedings of the IEEE International Conference on Control, Automation, Robotics and Vision*, pp. 623–629, 2008.
- [64] BENEDEK, C., DESCOMBES, X., ZERUBIA, J., “Building detection in a single remotely sensed image with a point process of rectangles”. In: *Proceedings of the IEEE International Conference on Pattern Recognition*, pp. 1417–1420, 2010.

- [65] NORONHA, S., NEVATIA, R., “Detection and modeling of buildings from multiple aerial images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, n. 5, pp. 501–518, 2001.
- [66] (ETSI), E. T. S. I., “Digital Video Broadcasting (DVB) Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream”, 2012.
- [67] MARIJAN, D., TESLIC, N., TEMERINAC, M., et al., “On the Effectiveness of the System Validation Based on the Black Box Testing Methodology”. In: *Proceedings of the IEEE Circuits and Systems International Conference on Testing and Diagnosis*, pp. 1–4, April 2009.
- [68] KASTELAN, I., KUKOLJ, S., PEKOVIC, V., “Functional verification of digital TV receivers using text region extraction”. In: *Proceedings of the IEEE International Conference on Computer as a Tool*, pp. 2017–2020, July 2013.
- [69] LEE, H.-C., COK, D. R., “Detecting boundaries in a vector field”, *IEEE Transactions on Signal Processing*, v. 39, n. 5, pp. 1181–1194, 1991.
- [70] LIU, Y., “Reliable classification system for diverse traffic sign patterns”, 2007.
- [71] FAROOQUE, M. A., ROHANKAR, J. S., “Survey on various noises and techniques for denoising the color image”, *International Journal of Application or Innovation in Engineering & Management*, v. 2, n. 11, pp. 217–221, 2013.
- [72] VC, H. P., “Method and means for recognizing complex patterns”, Dec. 18 1962, US Patent 3,069,654.
- [73] WANG, X., WANG, H., “Evolutionary optimization with Markov random field prior”, *IEEE Transactions on Evolutionary Computation*, v. 8, n. 6, pp. 567–579, 2004.
- [74] CHANG, C., LIN, C., “LIBSVM: A Library for Support Vector Machines”, *CM Transactions on Intelligent Systems and Technology*, v. 2, n. 3, 2011.

- [75] JOACHIMS, T., “Training Linear SVMs in Linear Time”. pp. 217–226, New York, NY, USA, 2006.
- [76] FUNG, H. K., WONG, K. H., “Quadrangle detection based on a robust line tracker using multiple kalman models”, *Journal of ICT Research and Applications*, v. 7, n. 2, pp. 137–150, 2013.
- [77] VON GIOI, R. G., JAKUBOWICZ, J., MOREL, J.-M., et al., “LSD: a line segment detector”, *Image Processing On Line*, v. 2, pp. 35–55, 2012.
- [78] VON GIOI, R. G., JAKUBOWICZ, J., MOREL, J.-M., et al., “On straight line segment detection”, *Journal of Mathematical Imaging and Vision*, v. 32, n. 3, pp. 313–347, 2008.
- [79] FRÄNTI, P., AGEENKO, E. I., KÄLVIÄINEN, H., et al., “Compression of line drawing images using Hough transform for exploiting global dependencies”.
- [80] GEMAN, D., JEDYNAK, B., “An active testing model for tracking roads in satellite images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 18, n. 1, pp. 1–14, 1996.
- [81] MAGLI, E., OLMO, G., PRESTI, L. L., “On-board selection of relevant images: an application to linear feature recognition”, *IEEE Transactions on Image Processing*, v. 10, n. 4, pp. 543–553, 2001.
- [82] DOI, J., SHIMIZU, K., SATO, W., “High-Density Scanning Combined with Digital Image of Real Objects for Complete Reproduction of Cultural Artifacts”. In: *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2006.
- [83] NAJAFABADI, M. J. E., POURGHASSEM, H., “A Novel Method for Improving Edge Detection Using Negative and Gamma Correction Functions”. In: *Proceedings of the International Conference on Intelligent Computation and Bio-Medical Instrumentation*, pp. 60–63, 2011.
- [84] LIU, M., DING, Y., WANG, X., et al., “Gamma correction with revised piecewise curve and edge directed error diffusion”. In: *Proceedings of the*

International Conference on Wireless Communications & Signal Processing, pp. 1–4, 2009.

- [85] RAU, A. N., “System and method for automated testing of digital television receivers”, May 25 2004, US Patent 6,741,277.
- [86] RAU, A. N., “Automated test system for digital TV receivers”. In: *Proceedings of the IEEE International Conference on Consumer Electronics*, pp. 228–229, 2000.
- [87] RAMA, A., ALUJAS, R., TARRÉS, F., “Fast and robust graphic character verification system for TV sets”. In: *Proceedings of the IEEE International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 19–19, 2007.
- [88] ZHAO, B., PARRA, A., DELP, E. J., “Mobile-based hazmat sign detection and recognition”. In: *Proceedings of the Global Conference on Signal and Information Processing*, pp. 735–738, 2013.
- [89] MOUTARDE, F., BARGETON, A., HERBIN, A., et al., “Robust on-vehicle real-time visual detection of American and European speed limit signs, with a modular Traffic Signs Recognition system”. In: *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 1122–1126, 2007.
- [90] YU, Z., BAJAJ, C., “Detecting circular and rectangular particles based on geometric feature detection in electron micrographs”, *Journal of Structural Biology*, v. 145, n. 1, pp. 168–180, 2004.
- [91] BALLARD, D., “Generalizing the Hough transform to detect arbitrary shapes”, *Pattern Recognition*, v. 13, n. 2, pp. 183–194, 1991.

Apêndice A

Artigos Publicados

Neste apêndice, apresentamos os artigos desenvolvidos no decorrer desta dissertação.

A.1 Artigos Diretamente Relacionados ao Tema

1. F. d. S. FARIAS, S. S. WALDIR, E. B. DE LIMA FILHO and W. C. MELO, "Automated content detection on TVs and computer monitors" In: *Proceedings of the IEEE Global Conference on Consumer Electronics (GCCE)*, Osaka, 2015, pp. 177-178.
2. (submetido) F. d. S. FARIAS, L. E. S. E SILVA, W. C. Melo, E. B. DE LIMA FILHO, W. S. S. JUNIOR, "Detecção e Análise de Conteúdo de Telas de TV e Monitores", In: *Anais do Simpósio Brasileiro de Telecomunicações (SBrT)*, Santarém, 2016.

A.2 Outros Artigos

1. K. V. DOS SANTOS, S. S. WALDIR, L. E. S. e. SILVA, F. S. FARIAS, W. C. DE MELO and E. B. DE LIMA FILHO, "Discriminative filtering with principal component analysis and adaptive alignment" In: *Proceedings of the IEEE Global Conference on Consumer Electronics*, Osaka, 2015, pp. 521-522.
2. V. d. S. KLEHM, W. C. MELO, F. S. FARIAS, K. V. SANTOS and S. S. WALDIR, "Iris recognition using minimum average correlation energy and

principal component" In: *Proceedings of the IEEE Global Conference on Consumer Electronics*, Osaka, 2015, pp. 213-214.

Apêndice B

Imagens de Referência

Neste apêndice, apresentamos as imagens de referência utilizadas nas bases de dados concebidas neste trabalho.

B.1 Base de Dados Sob Condições Não-Controladas (*base 1*)

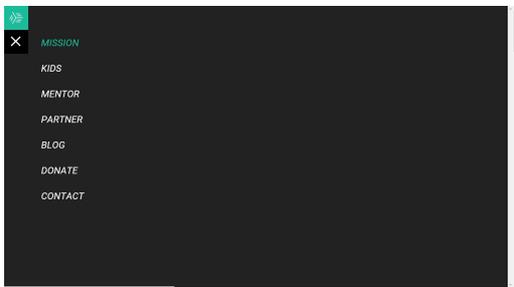


Figura B.1: imagem de referência - menu 1.

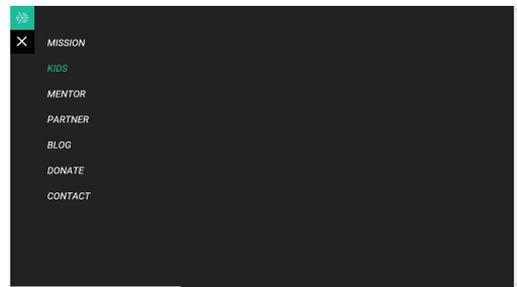


Figura B.2: imagem de referência - menu 2.

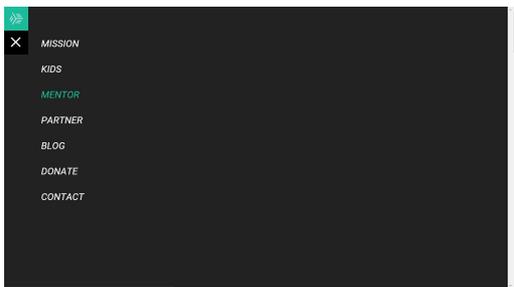


Figura B.3: imagem de referência - menu 3.

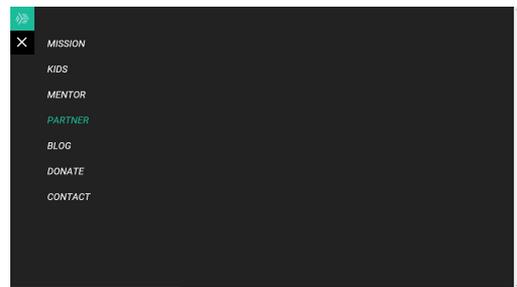


Figura B.4: imagem de referência - menu 4.

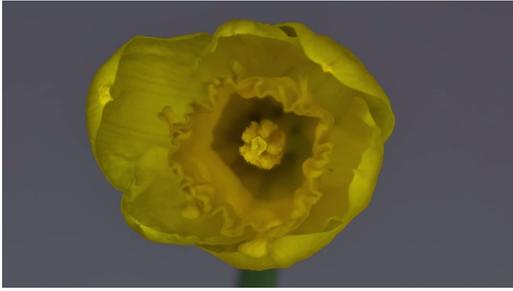


Figura B.5: imagem de referência - vídeo 1.



Figura B.6: imagem de referência - vídeo 2.

B.2 Base de Dados Sob Condições Controladas (*base 2*)

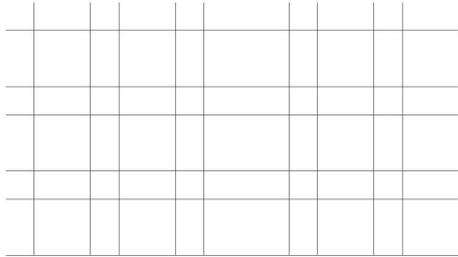


Figura B.7: imagem de referência - grid.

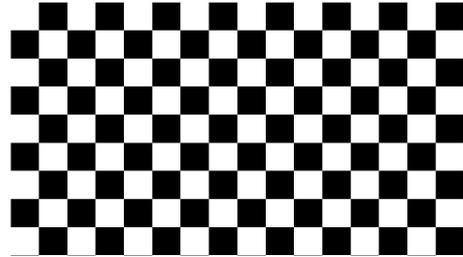


Figura B.8: imagem de referência - xadrez.

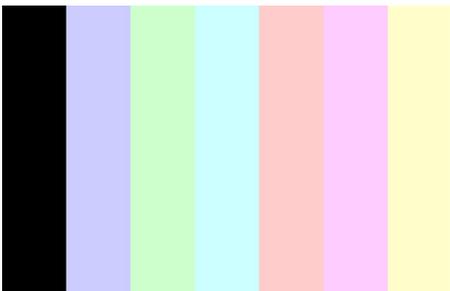


Figura B.9: imagem de referência - colorbar (20% saturação).

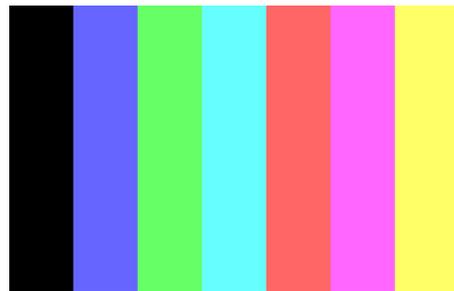


Figura B.10: imagem de referência - colorbar (60% saturação).

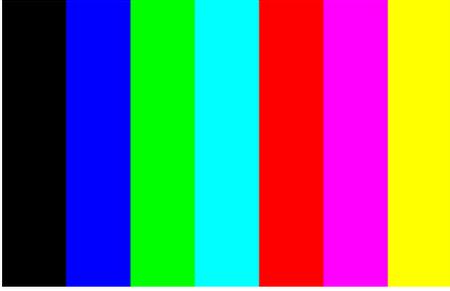


Figura B.11: imagem de referência - colorbar (100% saturação).



Figura B.12: imagem de referência - vermelho (20% saturação).



Figura B.13: imagem de referência - vermelho (60% saturação).

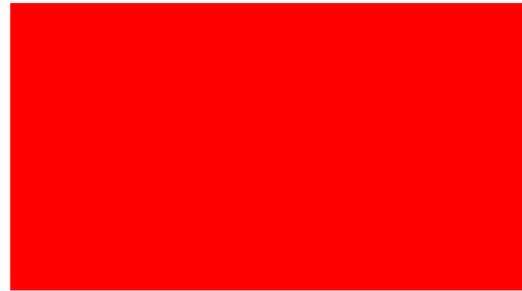


Figura B.14: imagem de referência - vermelho (100% saturação).

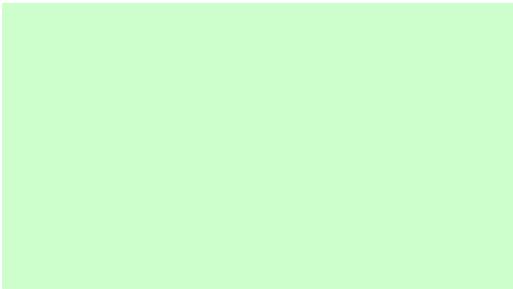


Figura B.15: imagem de referência - verde (20% saturação).



Figura B.16: imagem de referência - verde (60% saturação).



Figura B.17: imagem de referência - verde (100% saturação).



Figura B.18: imagem de referência - azul (20% saturação).



Figura B.19: imagem de referência - azul (60% saturação).



Figura B.20: imagem de referência - azul (100% saturação).



Figura B.21: imagem de referência - amarelo (20% saturação).



Figura B.22: imagem de referência - amarelo (60% saturação).



Figura B.23: imagem de referência - amarelo (100% saturação).

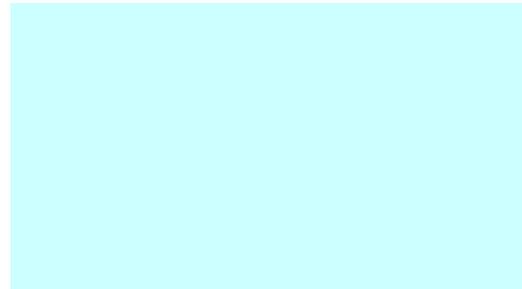


Figura B.24: imagem de referência - ciano (20% saturação).



Figura B.25: imagem de referência - ciano (60% saturação).



Figura B.26: imagem de referência - ciano (100% saturação).



Figura B.27: imagem de referência - magenta (20% saturação).

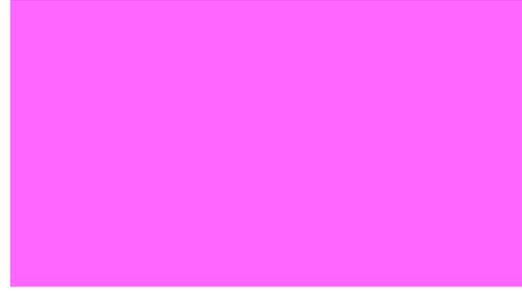


Figura B.28: imagem de referência - magenta (60% saturação).



Figura B.29: imagem de referência - magenta (100% saturação).



Figura B.30: imagem de referência - foto (casa azul).



Figura B.31: imagem de referência - foto (telhados).



Figura B.32: imagem de referência - foto (mercadinho).



Figura B.33: imagem de referência - foto (ponte).



Figura B.34: imagem de referência - foto (barco).



Figura B.35: imagem de referência - foto (terraço).



Figura B.36: imagem de referência - foto (parquinho).



Figura B.37: imagem de referência - foto (palácio).



Figura B.38: imagem de referência - foto (jardim).



Figura B.39: imagem de referência - foto (templo).



Figura B.40: imagem de referência - foto (farinha).



Figura B.41: imagem de referência - foto (árvores).

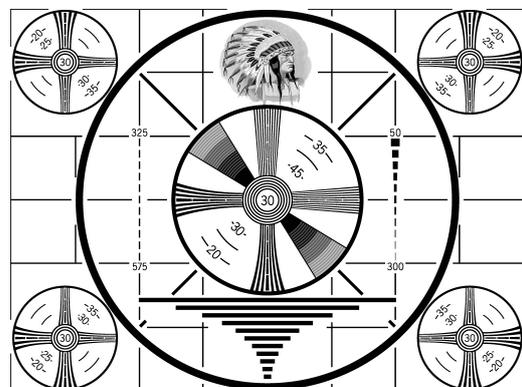


Figura B.42: imagem de referência - índio.



Figura B.43: imagem de referência - menu 1.



Figura B.44: imagem de referência - menu 2.



Figura B.45: imagem de referência - menu 3.

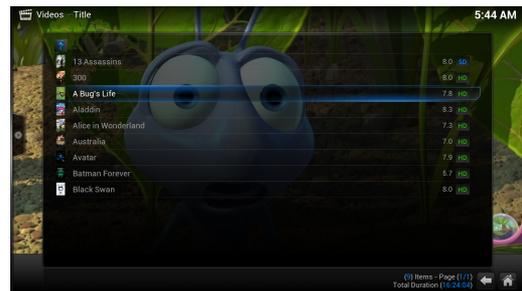


Figura B.46: imagem de referência - menu 4.