UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE ENGENHARIA
CAMPUS DE ILHA SOLTEIRA

PEDRO HENRIQUE PEREIRA

UM MÉTODO PARA CORREÇÃO AUTOMÁTICA DE CARTÃO DE RESPOSTA UTILIZANDO PROCESSAMENTO DIGITAL DE IMAGEM

Ilha Solteira – SP

2018

PEDRO HENRIQUE PEREIRA

UM MÉTODO PARA CORREÇÃO AUTOMÁTICA DE CARTÃO DE RESPOSTA UTILIZANDO PROCESSAMENTO DIGITAL DE IMAGEM

Dissertação apresentada à Faculdade de Engenharia de Ilha Solteira – UNESP como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Área de Conhecimento: Automação.

Prof. Dr. Jozué Vieira Filho

Orientador

Ilha Solteira – SP

2018

FICHA CATALOGRÁFICA
Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

Pereira, Pedro Henrique.
P436m Um método para correção automática de cartão de resposta utilizando processamento digital de imagem / Pedro Henrique Pereira. -- Ilha Solteira: [s.n.], 2018
78 f. : il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2018

Orientador: Jozué Vieira Filho
Inclui bibliografia

1. Reading template. 2. Visão computacional. 3. Correlação cruzada normalizada. 4. Transformada de Hough.

João Josué Barbosa
Serviço Técnico de Biblioteca e Documentação
Diretor Técnico
CRB 8-5642

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Ilha Solteira

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: UM MÉTODO PARA CORREÇÃO AUTOMÁTICA DE CARTÃO DE RESPOSTA UTILIZANDO PROCESSAMENTO DIGITAL DE IMAGEM

AUTOR: PEDRO HENRIQUE PEREIRA
ORIENTADOR: JOZUE VIEIRA FILHO

Aprovado como parte das exigências para obtenção do Título de Mestre em ENGENHARIA ELÉTRICA, área: AUTOMAÇÃO pela Comissão Examinadora:

Prof. Dr. JOZUE VIEIRA FILHO
Coordenadoria Executiva / UNESP - Câmpus de São João da Boa Vista

Profa. Dra. SUELY CUNHA AMARO MANTOVANI
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. MARCO APARECIDO QUEIROZ DUARTE
Departamento de Matemática / Universidade Estadual de Mato Grosso do Sul

Ilha Solteira, 27 de abril de 2018

DEDICO

À minha esposa Flávia e a minha filha Larissa que são o que tenho de mais importante na vida. Sem vocês nenhuma conquista valeria a pena.

AGRADECIMENTOS

À Deus por ser minha fonte de força e paz nos momentos de inquietude.

À minha mãe Lovalina que em sua simplicidade ensinou que a educação seria a melhor maneira de transformar a vida.

Aos meus orientadores, Professor Dr. Jozué Vieira Filho e ao Professor Dr. Nobuo Oki por terem me acompanhado durante todo esse percurso. Ensinando, motivando e por me fazerem enxergar as possibilidades além do que eu podia imaginar.

A Professora Dra. Suely Cunha Amaro Mantovani e ao Professor Dr. Carlos Antônio Alves que fizeram ótimas colaborações durante a fase da qualificação.

A Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro para a apresentação dos resultados do trabalho em congresso.

A todos os familiares e amigos que de alguma forma me incentivaram para a conclusão desse trabalho.

Meus sinceros agradecimentos,

Pedro Henrique Pereira

“Não sei o que possa parecer aos olhos do mundo, mas aos meus pareço apenas ter sido como um menino brincando à beira-mar, divertindo-me com o fato de encontrar de vez em quando um seixo mais liso ou uma concha mais bonita que o normal, enquanto o grande oceano da verdade permanece completamente por descobrir à minha frente.”

Isaac Newton

RESUMO

Neste trabalho propõe-se um método que envolve diferentes técnicas de processamento digital de imagem para a correção automática de cartão de respostas utilizados em avaliações de múltipla- escolha, vestibulares, concursos ou processos de seleção. O método tem como base um modelo de referência onde são identificados os seguintes elementos do cartão: campos de marcação das opções, número de opção de cada questão e uma imagem de marcação para identificar as extremidades da folha. No trabalho foi aplicada a técnica de Correlação Cruzada Normalizada para identificação da imagem na extremidade da folha. A Transformada de Hough foi utilizada para identificação das áreas onde se realiza a marcação das opções. Para identificar qual a questão que foi marcada utilizou-se a contagem de *pixel*. Foram realizados testes com 1.154 gabaritos contendo 40 questões, preenchidos no vestibular de uma faculdade particular. O tempo médio para correção de cada cartão foi de 1,39 segundos. A precisão do método foi avaliada utilizando-se 5% dos cartões digitalizados e processados. Basicamente, a avaliação foi realizada conferindo-se visualmente cada cartão e comparando os resultados com os gerados pela correção automática. O índice de acerto com as amostras utilizadas foi de 100%, o que comprova a eficiência do método proposto.

Palavras-chave: *Reading template*. Visão computacional. Correlação cruzada normalizada. Transformada de *Hough*.

ABSTRACT

This work proposes a method that involves different techniques of digital image processing for automatic correction of card responses used in multiple-choice evaluations, “vestibular”, contests or selection processes. The method is based on a reference model where the following elements of the card are identified: marking fields of the options, the option number of each question and a marking image to identify the edges of the sheet. In this work, the Normalized Cross-Correlation technique was applied to identify the image at the leaf end. The Hough Transform was used to identify the areas where the options are marked. To identify which question was selected, the pixel count was used. We performed tests with 1.154 templates containing 40 questions that were completed in the entrance examination of a private college. The average time to correct each card was 1.39 seconds. Samples were collected on 5% of the scanned and processed cards by the method to verify their accuracy. The total sampled was visually checked with the results presented by the application of the method and it was verified the percentage of 100% of correct answers in the reading of the markings.

Keywords: Reading template. Computer vision. Standardized cross-correlation. Hough transformation.

LISTA DE FIGURAS

Figura 1 - Processo de transformação de uma cena real em uma imagem digital. ---------- 21
Figura 2 - Princípio do funcionamento de um sensor CCD. ---------- 22
Figura 3 - Diagrama básico de um sistema CCD para aquisição de imagem. ---------- 23
Figura 4 - (a) Imagem contínua de uma cena real, (b) amostragem e quantização. ---------- 24
Figura 5 - Representação da matriz *MxN* de uma imagem monocromática. ---------- 25
Figura 6 - Qualidade da imagem para as diferentes resoluções de intensidade, sendo: (a) 256 níveis, (b) 16 níveis, (c) 4 níveis e (d) 2 níveis. ---------- 27
Figura 7 - Cubo representando o espaço de cor RGB. ---------- 28
Figura 8 - Círculos do sistema aditivo RGB. ---------- 29
Figura 9 - (a) Imagem original, (b) Matriz com os valores dos *pixels* RGB, (c) Matriz de conversão da imagem RGB para níveis de cinza. ---------- 30
Figura 10 - Operação pontual em imagem. ---------- 32
Figura 11 - Operação local em imagem. ---------- 33
Figura 12 - Operação global em imagem. ---------- 33
Figura 13 - (a) Máscara de convolução par, (b) máscara de convolução ímpar. ---------- 34
Figura 14 - Operação de convolução com máscara 2x2. ---------- 35
Figura 15 - Convolução periódica para o cálculo de *g(0,0)*. ---------- 36
Figura 16 - Convolução periódica para o cálculo de *g(0,1)*. ---------- 37
Figura 17 - Resultado da convolução periódica. ---------- 37
Figura 18 - Convolução aperiódica para o cálculo de *g(0,0)* para o caso 1. ---------- 38
Figura 19 - Convolução aperiódica para o cálculo de *g(0,1)* para o caso 1. ---------- 38
Figura 20 - Resultado da convolução aperiódica para o caso 1. ---------- 39
Figura 21 - Convolução aperiódica para o cálculo de *g(0,0)* para o caso 2. ---------- 39
Figura 22 - Convolução aperiódica para o cálculo de *g(0,1)* para o caso 2. ---------- 40
Figura 23 - Resultado da convolução aperiódica para o caso 2. ---------- 40
Figura 24 - Máscara genérica 3x3 de representação da convolução. ---------- 41
Figura 25 - Matriz genérica 3x3 de representação dos tons de cinza. ---------- 41
Figura 26 - Máscara usada para detecção de pontos isolados. ---------- 42
Figura 27 - (a)Imagem original, (b) Efeito da aplicação da máscara para detecção de pontos. ---------- 42
Figura 28 - Máscara para detecção de linhas: Horizontal, +45º, vertical e -45º. ---------- 43
Figura 29 - (a)Imagem original, (b) Máscara horizontal, (c) Máscara vertical. ---------- 43

Figura 30 - (a)Imagem original, (b) Máscara +45º, (c) Máscara -45º. ---------- 43
Figura 31 - Representação de uma borda e o sinal de respostas da 1ª e 2ª derivadas. --------- 44
Figura 32 - Representação do cálculo do gradiente. ---------- 46
Figura 33 - Máscaras diagonais para o cálculo do operador gradiente de Roberts. ---------- 46
Figura 34 - Detecção de bordas utilizando operadores gradientes diagonais de Roberts. ----- 47
Figura 35 - Máscara dos operadores de Prewitt.---------- 48
Figura 36 - Detecção de bordas utilizando operadores de Prewitt. ---------- 49
Figura 37 - Máscara dos operadores de Sobel. ---------- 49
Figura 38 - Detecção de bordas utilizando operadores de Sobel. ---------- 50
Figura 39 - (a) Plano xy, (b) Espaço de parâmetros.---------- 51
Figura 40 - Representação da reta na forma polar.---------- 52
Figura 41 - Fluxograma do algoritmo da TH para retas utilizando o espaço de parâmetros. - 52
Figura 42 - Algoritmo para detecção dos parâmetros do círculo.---------- 54
Figura 43 - (a) Imagem original, (b) Imagem após a aplicação da TH. ---------- 54
Figura 44 - Etapas de um sistema de reconhecimento de padrões. ---------- 55
Figura 45 - Resultado da correlação cruzada.---------- 58
Figura 46 - Operação de correlação normalizada.---------- 58
Figura 47 - Correlação Cruzada Normalizada - (a) Imagem a ser analisada, (b) Padrão a ser procurado, (c) Mapa de Correlação.---------- 59
Figura 48 - Áreas de interesse: (a) marcador de borda da página, (b) áreas para marcar questões. ---------- 61
Figura 49 - Diagrama para criação do *template*.---------- 62
Figura 50 - Arquivo XML contendo os parâmetros do *template*.---------- 62
Figura 51 - Diagrama com as etapas do processo de correção de um cartão de respostas.---- 63
Figura 52 - Identificação do marcador de borda e da área de marcação de questões.---------- 64
Figura 53 - Resultado da aplicação da transformada de *Hough* nas áreas selecionadas para processamento. ---------- 65
Figura 54 - Algoritmo para a contagem de *pixels*. ---------- 66
Figura 55 - Algoritmo para identificar a questão que foi marcada.---------- 67
Figura 56 - Ilustração da contagem de *pixels*.---------- 67
Figura 57 - Algoritmo para identificar se a questão está em branco.---------- 70
Figura 58 - Algoritmo para identificar se a questão está com marcações múltiplas.---------- 71
Figura 59 - Preenchimento insuficiente.---------- 73
Figura 60 - Modelo com opções ao centro.---------- 74

Figura 61 - Modelo do gabarito utilizado nos testes reais. ---------- 75

LISTA DE TABELAS

Tabela 1 - Especificações do *template*. ---------- 60
Tabela 2 - Especificações para a digitalização do *template*. ---------- 61
Tabela 3 - Contagem de *pixel* referente ao cartão de resposta. ---------- 68
Tabela 4 - Contagem de *pixel* normalizados pelas Equações (30) e (31). ---------- 69
Tabela 5 - Teste de valores para *Lm* realizados na Figura 59. ---------- 73
Tabela 6 - Teste de valores para *Lm* realizados na Figura 60. ---------- 74

LISTA DE ABREVIATURAS E SIGLAS

CCD	Charge Coupled Device
DPI	Dots Per Inch
HT	Hough Transform
PIXEL	Picture Element
RAM	Random Access Memory
RGB	Red, Blue and Green
TH	Transformada de *Hough*
TW	Transformada de *Wavelet*
WT	Wavelet Transform
XML	eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO 18

1.1 OBJETIVOS 19

1.2 TRABALHOS RELACIONADOS 19

1.3 ORGANIZAÇÃO DO TEXTO 20

2 REFERENCIAL TEÓRICO 21

2.1 AQUISIÇÃO DE IMAGEM 21

2.2 SENSOR DE IMAGEM CCD 21

2.3 AMOSTRAGEM E QUANTIZAÇÃO 23

2.4 REPRESENTAÇÃO MATEMÁTICA DE IMAGEM DIGITAL 24

2.5 RESOLUÇÃO ESPACIAL E RESOLUÇÃO DE INTENSIDADE 25

2.6 FUNDAMENTO DO ESPAÇO DE CORES 27

2.6.1 Espaço de cor RGB 27

2.6.2 Conversão de cores RGB para níveis de cinza 29

2.7 PROPRIEDADES DA IMAGEM DIGITAL 30

2.7.1 Vizinhança 30

2.7.2 Vizinhança $N_4(p)$ 30

2.7.3 Vizinhança $N_D(p)$ 31

2.7.4 Vizinhança $N_8(p)$ 31

2.7.5 Conectividade 31

2.7.6 Adjacência 31

2.7.7 Operações em imagem 32

2.7.8 Operação pontual 32

2.7.9 Operação local 33

2.7.10 Operação global 33

2.8 MÁSCARA DE CONVOLUÇÃO 34
2.8.1 Convolução periódica 36
2.8.2 Convolução aperiódica 37
2.9 SEGMENTAÇÃO DE IMAGEM 40
2.9.1 Detecção de descontinuidades em imagem 41
2.9.2 Detecção de pontos 42
2.9.3 Detecção de linhas 43
2.9.4 Detecção de bordas 44
2.9.5 Operador gradiente 45
2.9.6 Operador de Roberts 46
2.9.7 Operador de Prewitt 48
2.9.8 Operador de Sobel 49
2.9.9 Transformada de *Hough* 50
2.9.10 Transformada de *Hough* para detecção de retas 51
2.9.11 Transformada de *Hough* para detecção de círculos 53
3 RECONHECIMENTO DE PADRÕES 55
3.1 SISTEMAS DE RECONHECIMENTO DE PADRÃO 55
3.1.1 Aquisição de imagem 56
3.1.2 Pré-processamento 56
3.1.3 Segmentação 56
3.1.4 Extração de características 56
3.1.5 Classificação 56
3.2 CORRESPONDÊNCIA DE MODELO 57
3.3 CORRELAÇÃO CRUZADA NORMALIZADA 57
4 METODOLOGIA PROPOSTA 60
4.1 DETECTAR O MARCADOR DE BORDA DA PÁGINA 63

4.2 DETECTAR CAMPOS PARA MARCAÇÃO DAS QUESTÕES 65
4.3 DETECÇÃO DAS QUESTÃO MARCADA 65
4.4 IDENTIFICAR QUESTÃO SEM MARCAÇÃO 69
4.5 IDENTIFICAR QUESTÃO COM MARCAÇÕES MÚLTIPLAS 70
5 RESULTADOS 72
5.1 CAUSAS DE INCONSISTÊNCIAS NA CORREÇÃO 72
5.2 TESTES COM GABARITOS REAIS 75
6 CONCLUSÕES 77
REFERÊNCIAS 78

1 INTRODUÇÃO

A aplicação de testes de múltipla-escolha é um processo tradicional no meio acadêmico para a aplicação de exames multidisciplinares e também para a realização de concursos, pesquisas de mercado ou avaliação de candidato em processo de contratação. Ao final dessas avaliações os candidatos encontram um espaço disponível para marcar as opções selecionadas que comumente é conhecido como cartão de resposta ou gabarito.

Tradicionalmente, existem no mercado sistemas que realizam a verificação de gabaritos utilizando a leitura ótica das marcações com base em uma tecnologia conhecida como *Optical Mark Recognition* (OMR). Tal tecnologia foi aplicada nos Estados Unidos da América pela IBM em 1938 no equipamento IBM 805 *Test Scoring Machine*, que tinha capacidade para realizar a correção de até 750 posições de respostas. Baseado nesse equipamento, encontram-se disponíveis no mercado sistemas que utilizam *scanners* dedicados ao reconhecimento óptico das marcações e que podem processar até 2890 formulários por hora. Além disso, com o avanço dos sistemas computacionais, também surgiram sistemas baseados em *softwares* que realizam essa tarefa utilizando a digitalização das folhas dos gabaritos e a utilização do processamento das marcações.

Comparando o sistema dedicado que utiliza o reconhecimento óptico e o baseado em *software*, verifica-se que um sistema baseado em *software* apresenta um desempenho um pouco inferior quando se compara a quantidade de folhas processadas por hora. Por outro lado, o sistema baseado em *software* tem algumas vantagens como: baixo custo de manutenção, maior facilidade para ser usado em diferentes locais e a opção de utilizar gabaritos customizados pelo usuário, visto que o sistema dedicado, geralmente, tem a restrição da obrigatoriedade de se utilizar alguns modelos especificados pelos fabricantes dos equipamentos.

Sem a utilização de um sistema capaz de realizar a verificação das marcações de forma automática, resta a opção de utilizar um método de verificação manual, o que poder tornar sua utilização inviável para uma quantidade grande de cartões, tendo em vista a demora na realização da correção e a possibilidade real de erros.

1.1 OBJETIVOS

O objetivo deste trabalho é propor um método para correção automática de cartões de respostas utilizando processamento digital de imagens.

A aplicação do método apresentado propicia condições para a elaboração de um sistema computacional capaz de realizar todas as tarefas envolvidas no processo de correção de cartões de respostas.

1.2 TRABALHOS RELACIONADOS

Não há na literatura um número elevado de trabalhos relacionados diretamente ao tema. Porém, como parte importante no desenvolvimento deste trabalho, foi possível identificar e estudar algumas referências.

No trabalho de Chai (2016) é apresentado um método para alinhamento dos campos das questões, que faz uso de uma borda retangular em volta de todo o caderno de questões. Essa borda auxilia a identificação do posicionamento dos campos e facilita a identificação das marcações.

Em Bayar (2016), o autor utiliza a transformada de *Hough* para detectar os círculos onde são realizadas as marcações das questões.

No trabalho "*Efficient and reliable camera based multiple-choice test grading system*" (NGUYEN, T. et al, 2011), a transformada de *Hough* é utilizada para a detecção de linha com o objetivo de identificar a borda da folha na imagem e efetuar a correção da inclinação do cartão considerando que a imagem registrada pela câmera fotográfica pode ser digitalizada com alguma inclinação.

Já em Deng (2008), na região onde são realizadas as marcações das opções de respostas, os autores utilizaram marcações nas laterais do cartão para identificar as coordenadas das linhas e colunas. A intersecção dessas coordenadas identifica os campos de marcação. O autor considera como questão marcada aquela cujo campo esteja com 40% de cobertura dessa área. Se mais de um campo estiver marcado, o campo com maior cobertura será considerado como a opção assinalada.

1.3 ORGANIZAÇÃO DO TEXTO

O texto desta dissertação está organizado como segue: no capítulo 2 é apresentado o referencial teórico sobre imagem digital, que inclui a conversão de uma cena real para uma imagem digital, sua representação matemática, as representações das cores, o processo de conversão de uma imagem colorida para níveis de cinza, as operações básicas que podem ser realizadas sobre uma imagem, o processo de segmentação com a descrição sobre a detecção de bordas e a transformada de *Hough*. Esse capítulo é a base para entender como foi possível aplicar os demais métodos para atingir os objetivos do trabalho. O capítulo 3 trata do reconhecimento de padrões e nele são apresentados os conceitos iniciais sobre reconhecimento de padrão em imagem digital e são descritas algumas técnicas utilizadas na composição deste trabalho, como a técnica de correspondência de modelo. O capítulo 4 apresenta a metodologia proposta, descrevendo-se como as diferentes técnicas foram utilizadas para compor a solução, definindo as propriedades e os algoritmos para aplicação da metodologia. No capítulo 5 são analisados os resultados obtidos com aplicação do método proposto e discutidos os problemas encontrados. No capítulo 6 apresenta-se a conclusão do trabalho e algumas sugestões para possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

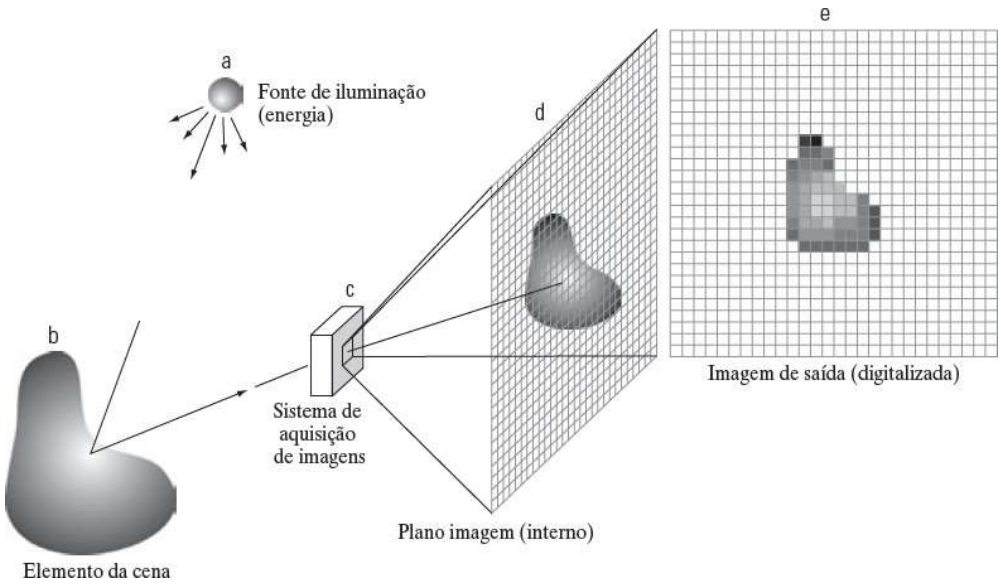
Nesse capítulo são apresentados os principais conceitos sobre imagem digital e discutidas as técnicas básicas de processamento, que dão subsídios para compreender o processo da formação e processamento de uma imagem.

2.1 AQUISIÇÃO DE IMAGEM

A aquisição de uma imagem é a etapa inicial do processo para conversão de uma cena real em uma imagem digital (MARQUES; VIEIRA, 1999).

Na Figura 1 (c) ilustra-se a utilização de um sensor de imagem matricial CCD (*Charge Coupled Device*), que é o principal sensor utilizado em câmeras fotográficas (MARQUES; VIEIRA, 1999).

Figura 1 - Processo de transformação de uma cena real em uma imagem digital.



Fonte: Gonzalez e Woods (2010).

2.2 SENSOR DE IMAGEM CCD

A tecnologia de sensores CCD foi proposta por Willard Boyle e George Smith no laboratório da Bell em 1970. Tal tecnologia é utilizada em grande escala em equipamentos que realizam a captação de imagem, tais como, câmeras digitais, celulares, digitalizadores e fotocopiadoras (BOYLE; SMITH, 1970).

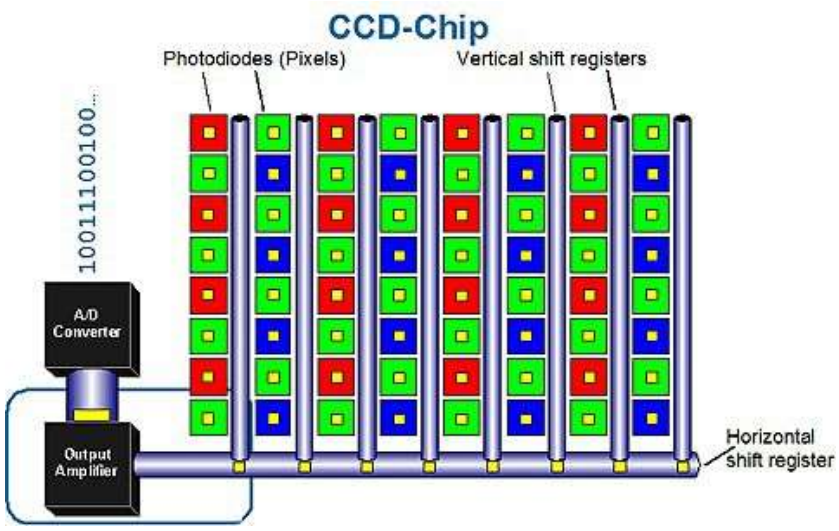
O sensor CCD é composto de milhares de fotodiodos com filtros sensíveis a luz em suas tonalidades vermelho, verde e azul. O sensor também possui registradores de mudanças de estado verticais e horizontais.

Ao capturar um novo elemento de cena do ambiente real, os fotodiodos transferem para os registradores verticais suas leituras. Estes são deslocados para o registrador horizontal que transferem para um amplificador de sinal e, posteriormente, para um conversor Analógico Digital.

Computacionalmente essa conversão torna-se uma matriz contendo os valores das cores vermelho, verde e azul da cena real.

Na Figura 2 a seguir ilustra-se o princípio de funcionamento do sensor CCD.

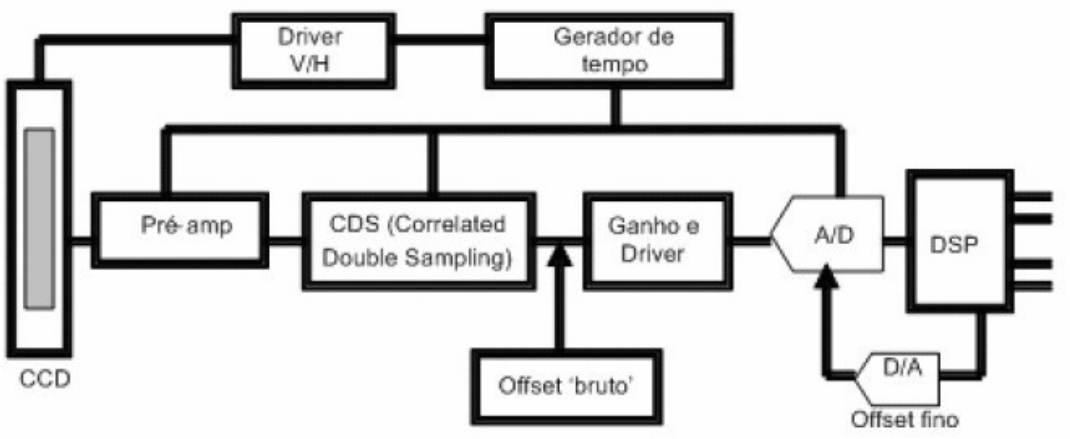
Figura 2 - Princípio do funcionamento de um sensor CCD.



Fonte: Fuji, (2016).

Segundo Lima (2006), a tecnologia CCD não permite a integração de circuitos para o processamento da carga obtida da matriz de *pixels* num único substrato e por esse motivo precisa conectar componentes externos ao sensor. Os circuitos normalmente implementados são os conversores ADC (*Analogic to Digital Converter*), geradores de sinais de sincronismo, controle de ganho entre outros conforme o diagrama de blocos ilustrado na Figura 3.

Figura 3 - Diagrama básico de um sistema CCD para aquisição de imagem.



Fonte: Lima (2006).

2.3 AMOSTRAGEM E QUANTIZAÇÃO

Para a criação de uma imagem digital é necessário converter os dados contínuos dos sensores que estão captando a cena de uma imagem real para o formato digital.

O processo de conversão da cena real para o formato digital envolve dois processos principais: a amostragem e a quantização;

Segundo Marques e Vieira (1999), a amostragem converte a imagem analógica em uma matriz de *MxN* pontos, cada qual denominado *pixel* ou elemento de imagem.

Para Gonzalez e Woods (2010), a digitalização dos valores de coordenada é chamada de amostragem e a digitalização dos valores de amplitude é chamada de quantização.

Pode-se resumir essas duas etapas como:

Amostragem:

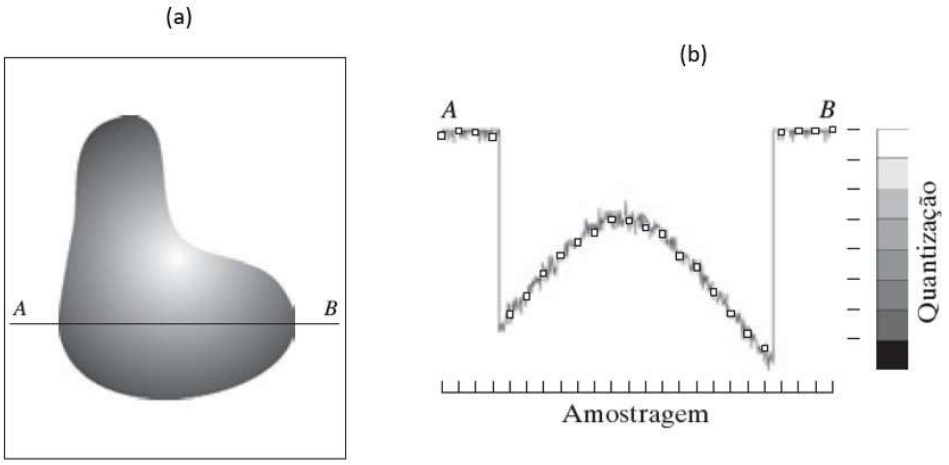
- Discretização espacial;
- Efeito: uma matriz de pontos;

Quantização:

- Discretização de amplitude;
- Efeito: faixa de valores que um *pixel* pode ser representado;

A Figura 4 ilustra a representação das etapas de amostragem e quantização.

Figura 4 - (a) Imagem contínua de uma cena real, (b) amostragem e quantização.



Fonte: Adaptado de Gonzalez e Woods (2010).

2.4 REPRESENTAÇÃO MATEMÁTICA DE IMAGEM DIGITAL

Para uma imagem digital colorida, as etapas de amostragem e quantização geram uma matriz do tipo *MxN* contendo as informações de cores registradas pelo sensor. É possível representar matematicamente uma imagem digital colorida na forma computacional como segue:

$$f([R,G,B]de\ x,y) = \begin{vmatrix} f([R,G,B]de\ 0,0) & f([R,G,B]de\ 0,1) & \cdots & f([R,G,B]de\ 0,N-1) \\ f([R,G,B]de\ 1,0) & f([R,G,B]de\ 1,1) & \cdots & f([R,G,B]de\ 1,N-1) \\ \vdots & \vdots & & \vdots \\ f([R,G,B]de\ M-1,0) & f([R,G,B]de\ M-1,1) & \cdots & f([R,G,B]de\ M-1,N-1) \end{vmatrix}. \quad (1)$$

Considere, por exemplo, uma imagem digital de tamanho 3x3 *pixels* e que o valor máximo para quantizar e representar cada *pixel* seja 8 *bits*; considere ainda que a primeira linha da imagem é inteira de cor vermelha, a segunda linha é inteira de cor verde e a terceira linha é inteira de cor azul. Assim, sua representação matricial pode ser expressa como apresentada na Equação (2).

$$f([R,G,B]de\ x,y) = \begin{vmatrix} 255,0,0 & 255,0,0 & 255,0,0 \\ 0,255,0 & 0,255,0 & 0,255,0 \\ 0,0,255 & 0,0,255 & 0,0,255 \end{vmatrix}. \quad (2)$$

A representação de uma imagem digital monocromática pode ser descrita matematicamente como uma função *f(x,y)* de intensidade luminosa, sendo seu valor, em qualquer ponto de coordenadas espaciais *(x,y)*, proporcional ao brilho ou nível de cinza da imagem naquele ponto (MARQUES; VIEIRA, 1999).

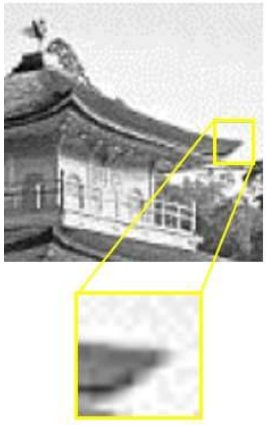
A Equação (3) demonstra a representação de uma imagem monocromática.

$$f(x,y) = \begin{vmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots f(M-1,N-1) \end{vmatrix}. \quad (3)$$

Na Figura 5 têm-se os detalhes de uma matriz *MxN* que representa uma imagem monocromática. Em (a) tem-se a imagem original em níveis de cinza, em (b) tem-se o detalhe de uma área de interesse de 20x20 *pixels*, e em (c) tem-se a representação da matriz com os valores quantizados da área em detalhe.

Figura 5 - Representação da matriz *MxN* de uma imagem monocromática.

(a)



(b)

246	248	244	247	246	238	247	247	244	248	243	240	255	243	238	238	247	244	238	238
246	239	255	244	238	238	238	245	255	239	238	240	255	243	238	238	238	238	238	238
255	253	239	249	250	238	251	248	239	238	238	238	239	249	250	238	238	245	254	238
246	246	247	246	246	247	246	246	247	238	243	246	238	244	246	247	239	241	246	247
240	241	255	245	243	255	242	245	253	240	248	253	238	239	243	253	242	239	238	255
255	255	255	255	255	255	255	248	238	254	255	253	238	250	250	238	252	248	238	255
249	255	255	255	255	255	255	253	249	255	255	254	249	245	245	249	244	246	249	244
215	222	226	226	228	234	238	240	242	242	242	243	247	238	240	255	244	246	251	238
119	120	136	136	144	170	184	194	204	204	204	206	221	221	230	255	255	248	238	238
112	106	122	122	120	114	117	118	120	120	128	133	123	153	178	213	237	246	252	252
120	116	126	121	117	109	109	109	109	109	111	112	109	149	180	217	240	249	255	255
136	136	136	125	119	119	119	119	119	119	110	104	119	166	201	238	252	255	255	255
136	135	120	130	135	135	135	123	103	118	137	157	201	237	253	254	241	239	239	239
141	144	128	133	134	127	112	99	84	136	166	195	232	241	248	255	241	242	247	247
136	142	126	126	123	113	93	86	82	160	197	229	255	243	243	255	242	246	255	253
68	67	51	51	55	68	110	146	187	236	247	255	255	243	243	255	255	255	255	238
139	136	110	117	128	150	182	207	234	249	253	255	255	243	239	243	253	250	243	250
168	168	150	167	181	201	224	238	250	255	253	251	255	246	243	243	249	247	243	255
162	171	187	221	238	238	252	248	238	254	246	240	255	255	255	255	241	245	255	255
140	142	144	140	145	167	215	225	224	254	254	253	255	245	245	255	253	253	255	226

(c)

Fonte: Adaptado de Marques e Vieira (1999).

2.5 RESOLUÇÃO ESPACIAL E RESOLUÇÃO DE INTENSIDADE

A resolução espacial é uma medida do menor detalhe discernível em uma imagem. Quantitativamente, a resolução espacial pode ser expressa de diferentes formas, mas a mais comum é expressa em pontos (*pixels* - *picture element*) por unidade de distância. Assim, considera-se que a resolução espacial está relacionada a quantidade de *pixels* ao longo dos eixos

x e *y*. Esse parâmetro está associado ao espaçamento físico entre as amostras (GONZALEZ; WOODS, 2010).

Considerando uma imagem contida em uma região de 15 cm no eixo *x* e 10 cm no eixo *y* e que tenha sido obtida de maneira uniforme, para cada amostra de 1 mm nos eixos *x* e *y*, tem-se uma matriz *MxN* com M=150 e N=100. Isto significa que foram coletadas 150 amostras no eixo *x* e 100 amostras no eixo *y*, resultando em um total de 15.000 amostras. Nesse caso, diz-se que a imagem tem resolução espacial de 150x100 *pixels*.

A resolução de intensidade refere-se à menor variação discernível de nível de intensidade na imagem e está relacionada ao nível da tonalidade que cada *pixel* pode assumir durante o processo de quantização (GONZALEZ; WOODS, 2010).

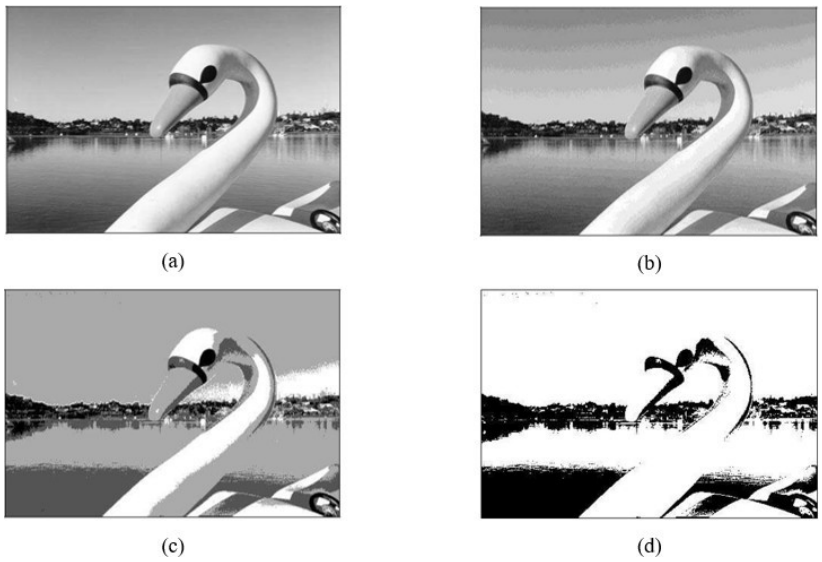
O número de níveis de intensidade é dado por 2^n, sendo *n* um número inteiro. Se o nível de tonalidade for definido como 256, por exemplo, então cada *pixel* pode ser associado a um valor entre 0 e 255, o que exige 8 *bits* para cada *pixel* ser armazenado na memória do computador. Diz-se, então, que a profundidade da imagem é de 8 *bits* por *pixel* (GONZALEZ; WOODS, 2010).

Uma imagem com 320x240 *pixels* de tamanho e 8 *bits* de profundidade gera, para seu armazenamento, um total de 320x240x8 *bits*, ou seja, 614.400 *bits*.

Boa parte dos sistemas de processamento digital de imagem utilizam como padrão de resolução de intensidade 8 *bits* por *pixel*. Porém, em sistemas que precisam realçar determinadas faixas, uma resolução de intensidade com 16 *bits* é mais adequada, apesar do aumento da quantidade de memória necessária para o armazenamento da imagem. Já quantizações utilizando 32 *bits* são mais raras, sendo utilizadas para aplicações muito específicas (GONZALEZ; WOODS, 2010).

Na Figura 6 é ilustrado o efeito na imagem das diferentes resoluções de intensidade aplicado à uma imagem em tons de cinza com 442 x 299 *pixels*.

Figura 6 - Qualidade da imagem para as diferentes resoluções de intensidade, sendo: (a) 256 níveis, (b) 16 níveis, (c) 4 níveis e (d) 2 níveis.



Fonte: Adaptado de Marques e Vieira (1999).

2.6 FUNDAMENTO DO ESPAÇO DE CORES

Em sistemas de visão computacional, algumas aplicações fazem o uso da identificação de cores juntamente com outras características para mapear áreas de interesse na imagem. Segundo Gonzalez e Woods (2010), os modelos de cores descrevem como uma cor pode ser representada.

São encontrados na literatura diversos modelos de cores para representação de imagens digitais, sendo que estes são adotados de acordo com a aplicação. Alguns espaços de cores existentes são: RGB (*red, green, blue*), CMY (*cyan, magenta, yellow*), CMYK (*cyan, magenta, yellow, black*), HSI (*hue, saturation, intensity*) e outros (GONZALEZ; WOODS, 2010).

Este trabalho restringe-se ao modelo de cor RGB e ao processo para conversão de uma imagem colorida no modelo RGB para uma imagem monocromática contendo apenas níveis de cinza.

2.6.1 Espaço de cor RGB

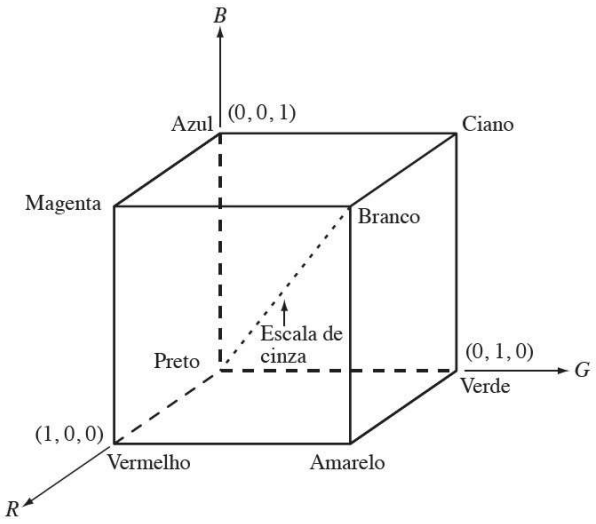
O objetivo de um modelo de cor é facilitar a especificação das cores em alguma forma padronizada e amplamente aceita. Essencialmente, um modelo de cor é uma especificação de um sistema de coordenadas e um subespaço dentro desse sistema no qual cada cor é

representada por um único ponto. A maioria dos modelos de cor utilizado atualmente é orientada ou em direção aos hardwares ou em direção as aplicações envolvendo a manipulação de cores (GONZALEZ; WOODS, 2010).

No modelo de espaço de cor RGB, cada cor é definida pela quantidade de vermelho (*red*), verde (*green*) e azul (*blue*) que a compõem.

O espaço de cor RGB é representado através de uma figura geométrica denominada de cubo das cores, apresentado na Figura 7.

Figura 7 - Cubo representando o espaço de cor RGB.



Fonte: Gonzalez e Woods (2010).

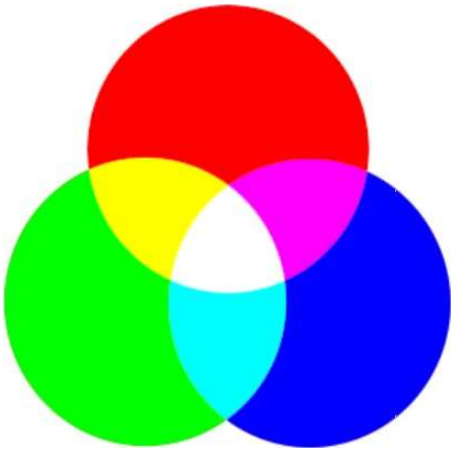
No modelo apresentado na Figura 7, o valor 1 representa o valor máximo da quantização para o componente da cor, assim como o valor 0 está representando a ausência de tal cor.

O modelo de espaço de cor RGB é o mais usado para equipamentos digitais, como câmeras, monitores e televisores. Cada componente de cor pode ser representado por 8 *bits*, dessa forma os valores de cada componente de cor podem variar entre 0 e 255 (MARQUES; VIEIRA, 1999).

A teoria do espaço de RGB foi proposta por Thomas Young (1773) e seu princípio se baseia nos diversos efeitos cromáticos obtidos pela projeção da luz branca através dos filtros vermelho, verde e azul, além da superposição de círculos nas cores projetadas (YOUNG, 1773).

Na Figura 8 ilustra-se o sistema aditivo de cores RGB, que representam as cores primárias e secundárias. Percebe-se que as interseções indicam a cor resultante da adição entre as cores dos círculos correspondentes.

Figura 8 - Círculos do sistema aditivo RGB.



Fonte: Adaptado de Gonzalez e Woods (2010).

2.6.2 Conversão de cores RGB para níveis de cinza

O processamento de uma imagem colorida é realizado preferencialmente quando as informações de cores são relevantes para o trabalho devido ao alto custo computacional para processamento das matrizes de composição das cores. É o caso, por exemplo, de imagens obtidas via satélite, tais como imagens de florestas, a partir das quais podem ser analisadas pastagens, queimadas, desmatamentos, etc.

O custo computacional do processamento de uma imagem colorida é maior porque na imagem em espaço de cor RGB é usual que cada *pixel* colorido seja representado por 24 *bits*, pois cada componente da cor utiliza 8 *bits*.

Em Padmavathi e Thangadurai (2016) é proposta uma equação para converter os componentes da cor RGB para níveis de cinza, onde cada *pixel* da imagem colorida passa a ser representada por apenas 8 *bits* ao invés de 24 *bits*, ou seja:

$$Y = 0.2989 * R + 0,5870 * G + 0,1140 * B. \quad (4)$$

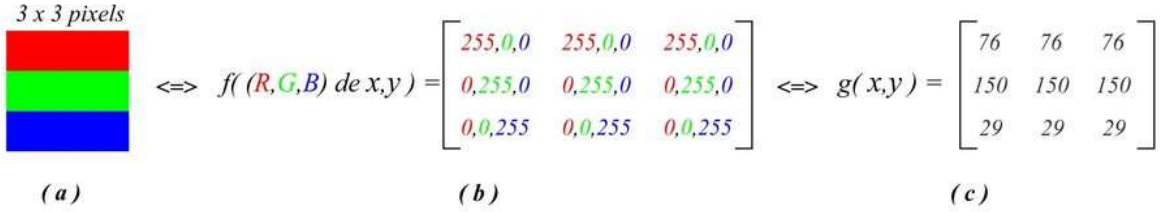
Basicamente, a Equação (4) é uma recomendação da ITU-R BT.601, mais conhecida por Rec. 601 ou BT.601 que é um padrão originalmente emitido em 1982 pela ITU (*Internacional Telecommunication Union*), com o objetivo de codificar sinais de vídeo analógico entrelaçado em vídeo digital (ITU-R, 2011).

A Equação (4) pode ser adaptada para uma função matemática que represente sua aplicação em cada elemento da imagem conforme a Equação (5).

$$g(x,y) = 0{,}2989 * R + 0{,}5870 * G + 0{,}1140 * B, \forall (x,y). \quad (5)$$

Na Figura 9 ilustra-se, através da representação matricial, a conversão de uma imagem colorida para uma imagem em tons de cinza aplicando-se a Equação (5).

Figura 9 - (a) Imagem original, (b) Matriz com os valores dos *pixels* RGB, (c) Matriz de conversão da imagem RGB para níveis de cinza.



Fonte: Elaborado pelo autor.

2.7 PROPRIEDADES DA IMAGEM DIGITAL

Como já discutido, uma imagem digital monocromática pode ser representada por uma função *f(x,y)* discretizada nos eixos *x* e *y*. Sendo assim, a imagem digital é representada em forma de matriz onde cada elemento representa um ponto na imagem e seu valor corresponde ao nível de cinza.

Com base nesses conceitos, serão descritas algumas propriedades básicas, mas essenciais para tratar uma imagem digital.

2.7.1 Vizinhança

O conceito de vizinhança de um *pixel p* qualquer está relacionado aos demais elementos que estão próximos a ele em um dos modelos de vizinhança apresentados a seguir.

2.7.2 Vizinhança $N_4(p)$

Um *pixel p* de coordenadas *(x, y)* tem 2 vizinhos horizontais, com coordenada *(x - 1, y)* e *(x + 1, y)* e dois vizinhos verticais com coordenadas *(x, y - 1)* e *(x, y + 1)*. Estes quatro *pixels*

formam a vizinhança N4(*p*).

2.7.3 Vizinhança N_D(p)

O *pixel p* tem as seguintes coordenadas na vizinhança na diagonal: *(x + 1, y + 1)*, *(x + 1, y - 1)*, *(x - 1, y + 1)* e *(x - 1, y - 1)*, que formam a vizinhança $N_D(p)$.

2.7.4 Vizinhança N_8(p)

Vizinhança de oito é dada pelo *pixel* em questão e seus vizinhos horizontais, verticais e diagonais, sendo a união da vizinhança N_4 e N_D.

2.7.5 Conectividade

A conectividade entre *pixels* é um importante conceito usado para estabelecer limites de objetos e componentes de regiões em uma imagem (MARQUES; VIEIRA, 1999).

Diz-se que os *pixels p* e *q* estão conectados se eles são adjacentes e se seus níveis de cinza satisfazem um critério de similaridade (MARQUES; VIEIRA, 1999).

Fazendo uso do conceito de vizinhança e dado um conjunto *V* qualquer de nível de cinza, a conectividade é definida por:

1) Conectividade – 4: dois *pixels p* e *q* com valores em nível de cinza contidos em *V* se $q \in N_4(p)$;

2) Conectividade – 8: dois *pixels p* e *q* com valores em nível de cinza contidos em *V* se $q \in N_8(p)$;

3) Conectividade – Mista (M): dois *pixels p* e *q* com valores em nível de cinza contidos em *V* se:

   i) $q \in N_4(p)$ ou,

   ii) $q \in N_D(p)$ e $N_4(p) \cap N_4(q) = 0$

2.7.6 Adjacência

Segundo Marques e Vieira (1999), um *pixel p* é adjacente a um *pixel q* se eles forem

conectados. Considerando dois subconjuntos de imagens, *S1* e *S2*,diz-se que eles são adjacentes se algum *pixel* em *S1* é adjacente a algum *pixel* em *S2*.

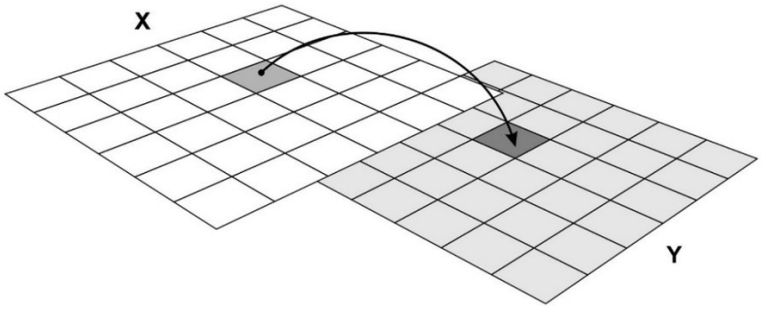
2.7.7 Operações em imagem

A transformação de uma imagem *X[m,n]* em uma imagem *Y[m,n]* podem ser realizadas através de operação pontual, local ou global.

2.7.8 Operação pontual

A operação pontual existe quando o *pixel* na posição *(x,y)* da imagem *Y* depende apenas do *pixel* na imagem *X* na posição *(x,y)* conforme ilustrado na Figura 10.

Figura 10 - Operação pontual em imagem.



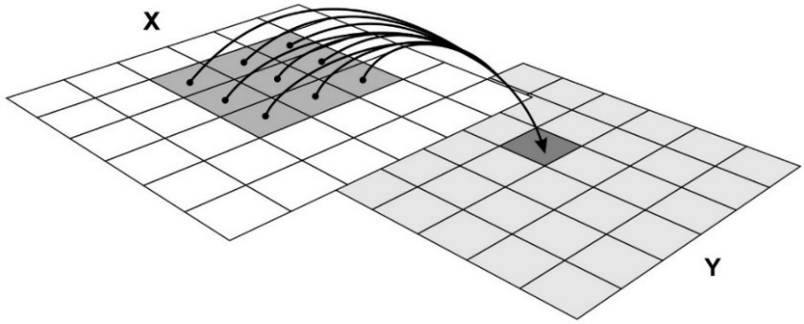
Fonte: Elaborado pelo autor.

Segundo Scuri (2002), as operações pontuais em sua maioria alteram características de cor e luminância, tais como, brilho, contraste, nível de branco e nível de preto e etc.

2.7.9 Operação local

A operação local existe quando o *pixel* na posição *(x, y)* da imagem *Y* depende de uma operação que considera as vizinhanças do *pixel* na imagem *X* conforme ilustrado na Figura 11.

Figura 11 - Operação local em imagem.



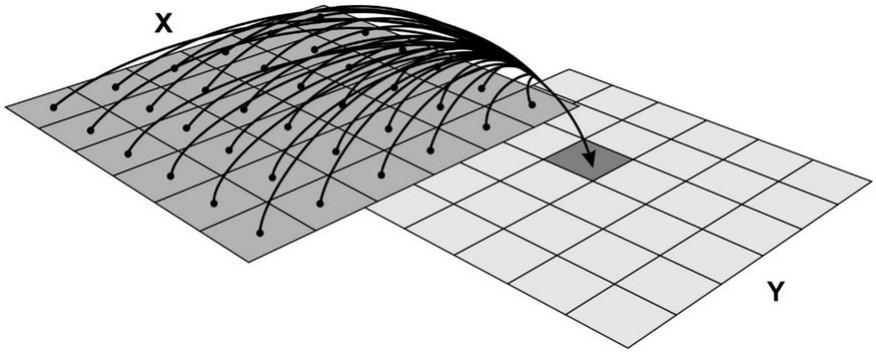
Fonte: Elaborado pelo autor.

As operações locais mais comuns são as de filtragem que usam uma convolução com máscara de dimensão *nxn*, onde *n* é tipicamente de tamanho 3, 5, 7 ou 9, mas podem assumir qualquer valor. Tais operações são muito usadas em processos de detecção de bordas (SCURI, 2002).

2.7.10 Operação global

A operação global existe quando o *pixel* na posição *(x, y)* da imagem *Y* depende de uma operação considerando todos os *pixels* da imagem *X* conforme ilustrado na Figura 12.

Figura 12 - Operação global em imagem.



Fonte: Elaborado pelo autor.

Segundo Scuri (2002), neste grupo de operações estão as transformadas de domínio,

como a Transformada de *Fourier*, a Transformada de *Wavelets* e a Transformada de *Hough*.

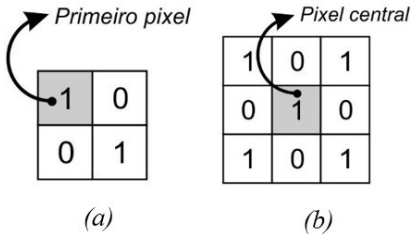
2.8 MÁSCARA DE CONVOLUÇÃO

Em processamento de imagens, convolução é uma operação entre duas matrizes, geralmente bidimensionais, onde uma delas representa a imagem original e a outra representa uma matriz que contém modificações desejadas para imagem. Em termos de sistema, isto representa, de maneira genérica, uma filtragem. É comum denominar a matriz que contém as modificações desejadas de matriz de convolução, elemento estruturante ou máscara.

Uma matriz de convolução *MxN* pode ser representada por uma função matemática qualquer. Essa matriz é aplicada sobre cada *pixel* da imagem original nas posições *I(x,y)* e sua vizinhança imediata. A aplicação dessa máscara resulta em uma outra imagem *O(x,y)* que reflete a relação da imagem *I* com a função matemática utilizada no processamento (GONZALEZ; WOODS, 2010).

As máscaras de convolução podem ter sua formação par (2x2, 4×4 e etc), ou podem ter sua formação ímpar (3x3, 5×5 e etc). Nas máscaras com formação par o resultado da convolução é colocado sobre o primeiro *pixel* conforme ilustrado na Figura 13 (a). Nas máscaras com formação ímpar o resultado da convolução é colocado sobre o *pixel* central, como ilustrado na Figura 13 (b).

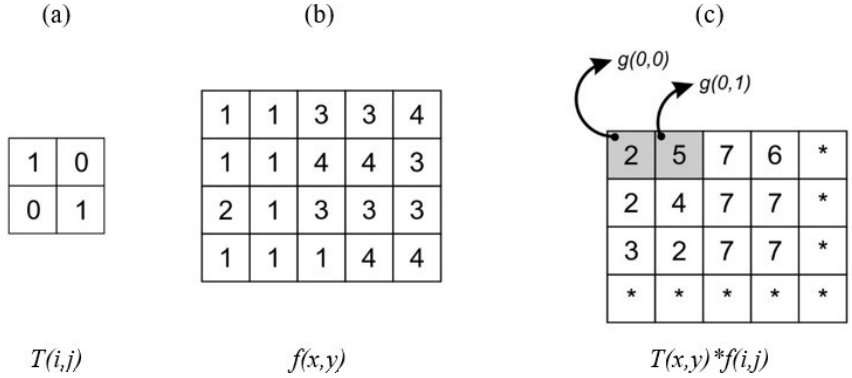
Figura 13 - (a) Máscara de convolução par, (b) máscara de convolução ímpar.



Fonte: Elaborado pelo autor.

Uma operação de multiplicação utilizando uma máscara de convolução de tamanho 2x2, Figura 14 (a), sobre uma imagem *f(x,y)*, Figura 14 (b), será dada por *T(x,y)* * *f(i,j)* e resultará na matriz apresentada na Figura 14 (c).

Figura 14 - Operação de convolução com máscara 2x2.



Fonte: Elaborado pelo autor.

A operação para o cálculo da convolução de *g(0,0)* e *g(0,1)* é exemplificada a seguir:

$$g(0,0) = T(0,0) * f(0,0) + T(0,1) * f(0,1) + T(1,0) * f(1,0) + T(1,1) * f(1,1)$$

$$g(0,0) = 1*1 + 0*1 + 0*1 + 1*1$$

$$g(0,0) = 1 + 0 + 0 + 1$$

$$g(0,0) = 2$$

$$g(0,1) = T(0,1) * f(0,1) + T(0,2) * f(0,2) + T(1,1) * f(1,1) + T(2,1) * f(2,2)$$

$$g(0,1) = 1*1 + 0*3 + 0*1 + 1*4$$

$$g(0,1) = 1 + 0 + 0 + 4$$

$$g(0,1) = 5$$

Conforme ilustra a Figura 14 (c), a imagem gerada pela convolução é menor que a imagem original, Figura 14 (b). Os valores marcados com * não puderam ser calculados.

Para esse caso podem ser aplicadas duas técnicas de convolução:

- Convolução periódica;
- Convolução aperiódica.

2.8.1 Convolução periódica

A convolução periódica é realizada com o deslocamento da máscara sobre todos os *pixels* da imagem original como se as bordas externas fossem conectadas.

O cálculo de *g(0,0)* e *g(0,1)* através da convolução periódica é exemplificada a seguir:

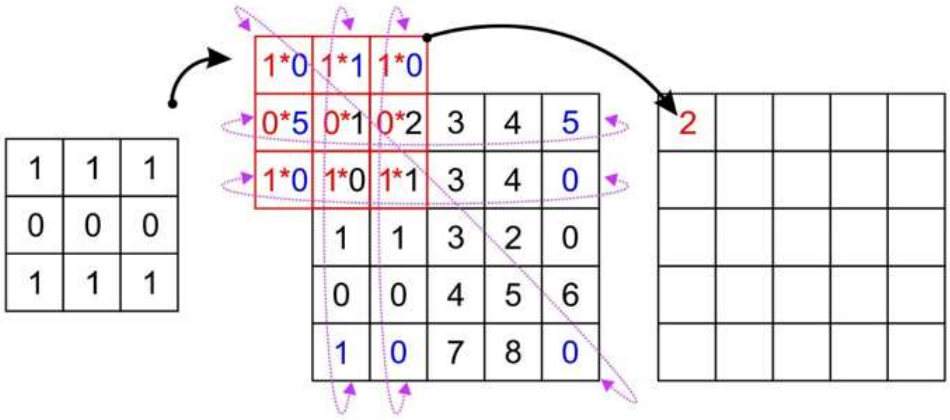
g (0 ,0) = (1x0) + (1x1) + (1 x0)
+ (0x5) + (0x1) + (0 x2)
+ (1x0) + (1x0) + (1 x1)
g (0 ,0) = 2

---

g (0 ,1) = (1 x1) + (1 x0) + (1 x7)
+ (0x1) + (0 x2) + (0 x3)
+ (1x0) + (1 x1) + (1 x3)
g (0 ,1) = 12

Na Figura 15 apresenta-se os passos para o cálculo de *g(0,0)* utilizando a convolução periódica.

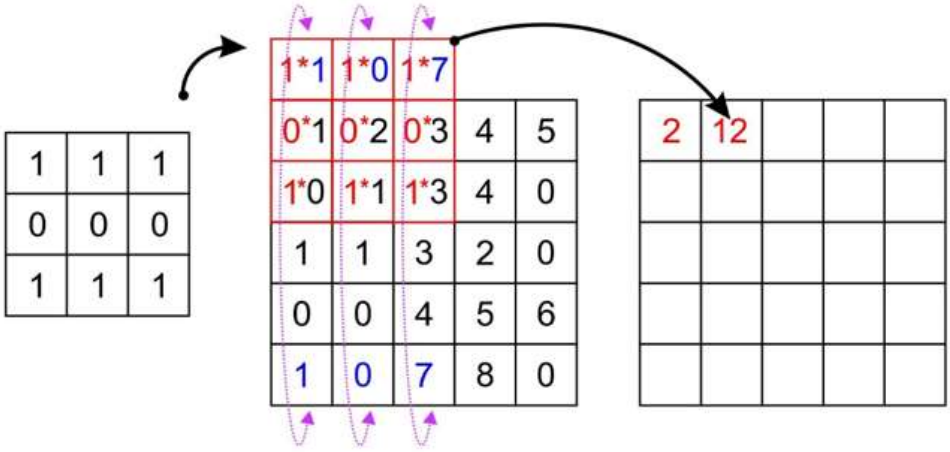
Figura 15 - Convolução periódica para o cálculo de *g(0,0)*.



Fonte: Elaborado pelo autor.

Na Figura 16 ilustra-se os passos para o cálculo de *g(0,1)* utilizando a convolução periódica.

Figura 16 - Convolução periódica para o cálculo de *g(0,1)*.



Fonte: Elaborado pelo autor.

Na Figura 17 mostra-se a matriz resultante da convolução periódica sobre a imagem original.

Figura 17 - Resultado da convolução periódica.

2	12	23	22	13
10	11	15	17	13
7	8	17	22	15
3	13	21	20	12
14	10	18	27	21

Fonte: Elaborado pelo autor.

2.8.2 Convolução aperiódica

Para o cálculo da convolução aperiódica pode-se utilizar um dos casos a seguir:

1) Atribui-se o valor zero aos *pixels* da imagem resultante que não podem ser calculados, ou;

2) Coloca-se a máscara centralizada com o primeiro *pixel* da imagem original, faz-se o cálculo e atribui o valor zero aos valores inexistentes na imagem resultante.

Para o caso 1 o cálculo de *g(0,0)* e *g(0,1)* através da convolução aperiódica é exemplificado a seguir:

$$g(0,0) = (1x1) + (1x2) + (1x3) + (0x0) + (0x1) + (0x3) + (1x1) + (1x1) + (1x3)$$

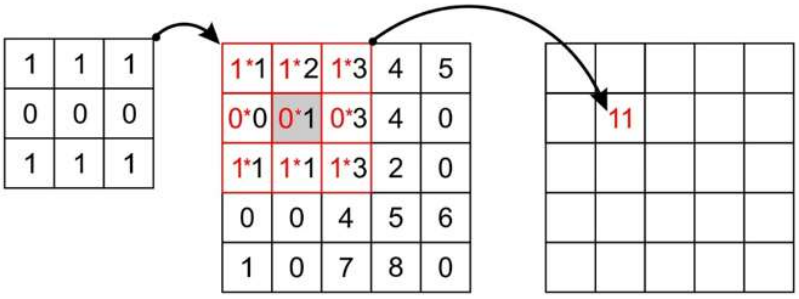
$$g(0,0) = 11$$

$$g(0,1) = (1x2) + (1x3) + (1x4) + (0x1) + (0x3) + (0x4) + (1x1) + (1x3) + (1x2)$$

$$g(0,1) = 15$$

Na Figura 18 ilustra-se os passos para o cálculo de *g(0,0)* utilizando a convolução aperiódica para o caso 1.

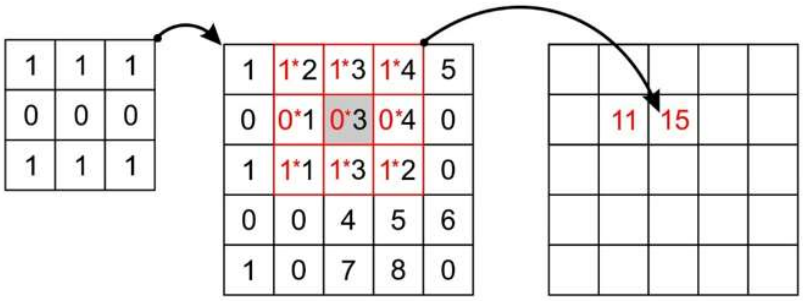
Figura 18 - Convolução aperiódica para o cálculo de *g(0,0)* para o caso 1.



Fonte: Elaborado pelo autor.

Na Figura 19 ilustra-se os passos para o cálculo de *g(0,1)* utilizando a convolução aperiódica para o caso 1.

Figura 19 - Convolução aperiódica para o cálculo de *g(0,1)* para o caso 1.



Fonte: Elaborado pelo autor.

Na Figura 20 apresenta-se a matriz resultante da convolução aperiódica sobre a imagem original para o caso 1.

Figura 20 - Resultado da convolução aperiódica para o caso 1.

0	0	0	0	0
0	11	15	17	0
0	8	17	22	0
0	13	21	20	0
0	0	0	0	0

Fonte: Elaborado pelo autor.

Para o caso 2 o cálculo de *g(0,0)* e *g(0,1)* através da convolução aperiódica é exemplificado a seguir:

$$g(0,0) = (0x0) + (0x0) + (0x0) + (0x0) + (0x1) + (0x2) + (0x0) + (1x0) + (1x1)$$

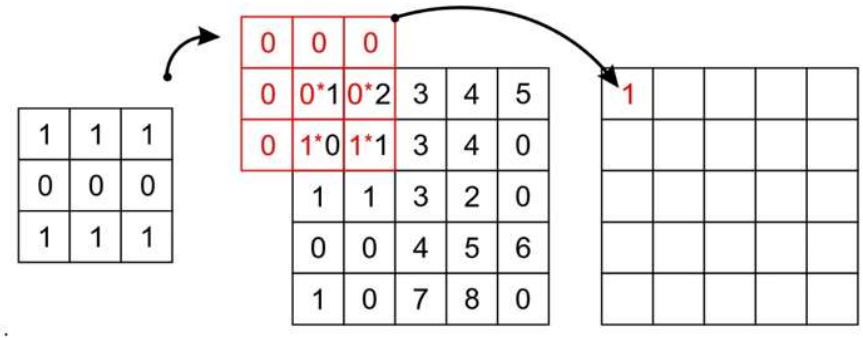
$$g(0,0) = 1$$

$$g(0,1) = (0x0) + (0x0) + (0x0) + (0x1) + (0x2) + (0x3) + (1x0) + (1x1) + (1x3)$$

$$g(0,1) = 4$$

Na Figura 21 ilustra-se os passos para o cálculo de *g(0,0)* utilizando a convolução aperiódica para o caso 2:

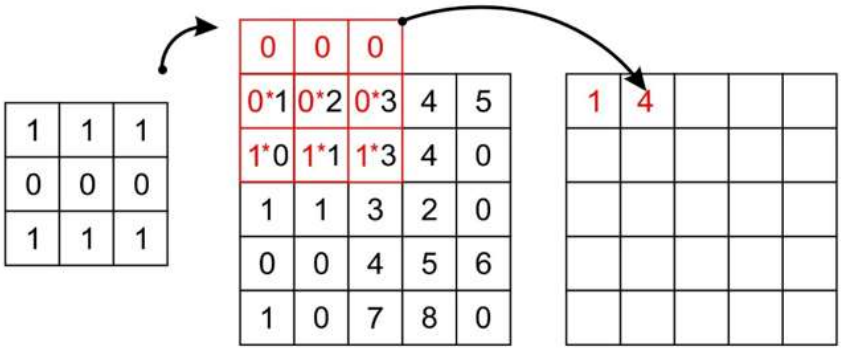
Figura 21 - Convolução aperiódica para o cálculo de *g(0,0)* para o caso 2.



Fonte: Elaborado pelo autor.

Na Figura 22 ilustra-se os passos para o cálculo de *g(0,1)* utilizando a convolução aperiódica para o caso 2:

Figura 22 - Convolução aperiódica para o cálculo de *g(0,1)* para o caso 2.



Fonte: Elaborado pelo autor.

Na Figura 23 é apresentada a matriz resultante da convolução aperiódica sobre a imagem original para o caso 2:

Figura 23 - Resultado da convolução aperiódica para o caso 2.

1	4	8	7	4
4	11	15	17	11
1	8	17	22	15
3	13	21	20	10
0	4	9	15	11

Fonte: Elaborado pelo autor.

2.9 SEGMENTAÇÃO DE IMAGEM

Segmentar uma imagem é subdividi-la em regiões ou objetos que constituem a cena, sendo que cada segmento possui um conjunto de *pixels* idênticos em termos de propriedades ou atributos. As propriedades podem ser referentes a textura do objeto e os atributos podem ser referentes aos níveis de contrastes do segmento (FARIA, 2010).

Os algoritmos utilizados nas técnicas de segmentação são baseados nas propriedades dos *pixels*, considerando sua descontinuidade ou similaridade. Na descontinuidade, a segmentação é baseada nas mudanças bruscas dos valores de intensidade dos *pixels* de uma

determinada região da imagem. Na similaridade, a segmentação é baseada em regiões que mantêm semelhanças entres os *pixels* de acordo com um conjunto de critérios predefinidos.

2.9.1 Detecção de descontinuidades em imagem

As técnicas para detecção de descontinuidades em imagem envolvem a detecção de pontos, linhas e bordas. As segmentações baseadas nessas características são realizadas através da varredura da imagem por uma máscara de convolução definida previamente. O uso de máscaras para esse fim é chamado de filtragem espacial (GONZALEZ; WOODS, 2010).

Na Figura 24 tem-se uma máscara genérica de tamanho 3x3 composta pelos pesos (w1 + w2 + w3 + w4 + w5 + w6 + w7 + w8 + w9).

Figura 24 - Máscara genérica 3x3 de representação da convolução.

$$\begin{bmatrix} W1 & W2 & W3 \\ W4 & W5 & W6 \\ W7 & W8 & W9 \end{bmatrix}$$

Fonte: Adaptado de Marques e Vieira (1999).

A área da imagem que será processada pela máscara é representada por uma matriz *Z*, onde seus valores representam os tons de cinza de cada *pixel*. Na Figura 25 ilustra-se a distribuição dos elementos que representa os níveis de cinza da imagem que será processada pela máscara de convolução *W*.

Figura 25 - Matriz genérica 3x3 de representação dos tons de cinza.

$$\begin{bmatrix} Z1 & Z2 & Z3 \\ Z4 & Z5 & Z6 \\ Z7 & Z8 & Z9 \end{bmatrix}$$

Fonte: Adaptado de Marques e Vieira (1999).

A máscara de convolução *W* percorrerá toda a imagem a ser processada baseado na matriz *Z*. Tal processamento iniciará na parte superior esquerda e finalizará na parte inferior direita da imagem (MARQUES; VIEIRA, 1999).

O cálculo do novo valor do *pixel* é dado pela expressão (6).

$$Z = \sum_{i=1}^{9} Wi.Zi. \quad (6)$$

Nas seções a seguir são apresentadas as máscaras mais utilizadas para a segmentação de objetos de interesse em imagens mais comuns.

2.9.2 Detecção de pontos

Para a detecção de um ponto isolado na imagem utiliza-se a máscara a seguir:

Figura 26 - Máscara usada para detecção de pontos isolados.

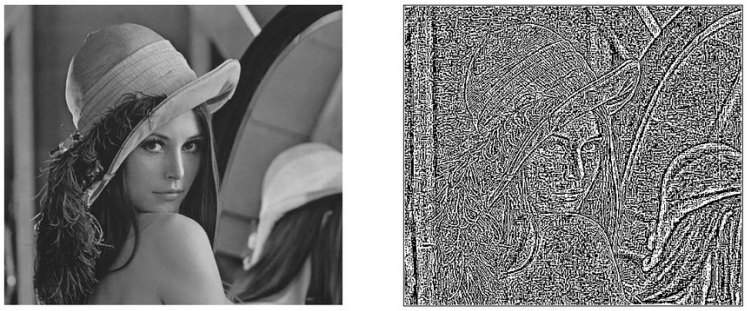
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Fonte: Adaptado de Marques e Vieira (1999).

A máscara para detecção de pontos calcula a diferença entre o ponto central e os seus vizinhos, pois considera-se que o nível de cinza de um ponto isolado é diferenciado em relação ao nível de cinza de seus vizinhos.

Na Figura 27 tem-se um exemplo da aplicação da máscara ilustrada na Figura 26 convolucionada sobre a imagem original da Figura 27 (a) para detecção de pontos.

Figura 27 - (a)Imagem original, (b) Efeito da aplicação da máscara para detecção de pontos.



(a) (b)

Fonte: Adaptado de Gonzalez e Woods (2010).

2.9.3 Detecção de linhas

Para a detecção de linhas são necessárias máscaras específicas para realçar a inclinação da reta. Para o caso da primeira máscara apresentada na Figura 28, que faz a detecção de linhas horizontais, a convolução sobre uma imagem original tem sua resposta máxima quando a linha passa pela linha média da máscara.

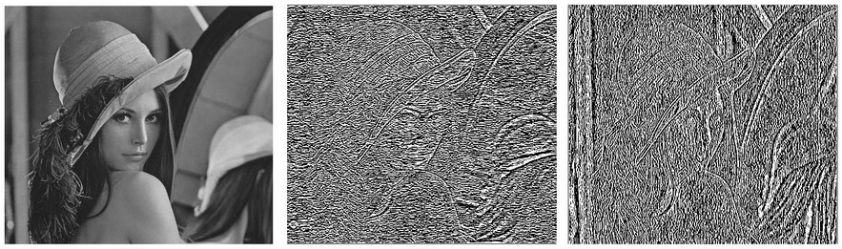
Figura 28 - Máscara para detecção de linhas: Horizontal, +45º, vertical e -45º.

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

Fonte: Adaptado de Marques e Vieira (1999).

Na Figura 29 ilustra-se a aplicação das máscaras para detecção de linhas horizontais - Figura 29 (b) - e linhas verticais - Figura 29 c).

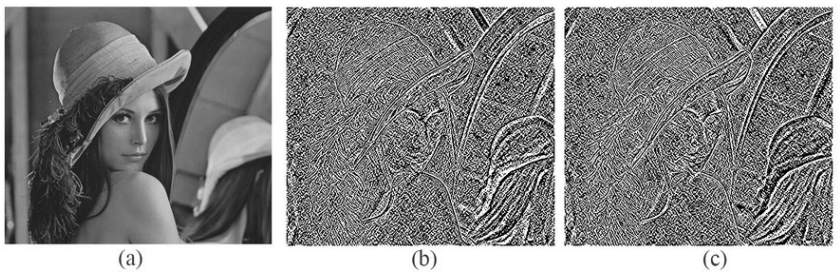
Figura 29 - (a)Imagem original, (b) Máscara horizontal, (c) Máscara vertical.



Fonte: Adaptado de Gonzalez e Woods (2010).

Na Figura 30 tem-se a aplicação das máscaras para detecção de linhas com inclinação a +45º, Figura 30 (b) e linhas com inclinação a -45º, Figura 30 (c).

Figura 30 - (a)Imagem original, (b) Máscara +45º, (c) Máscara -45º.



Fonte: Adaptado de Gonzalez e Woods (2010).

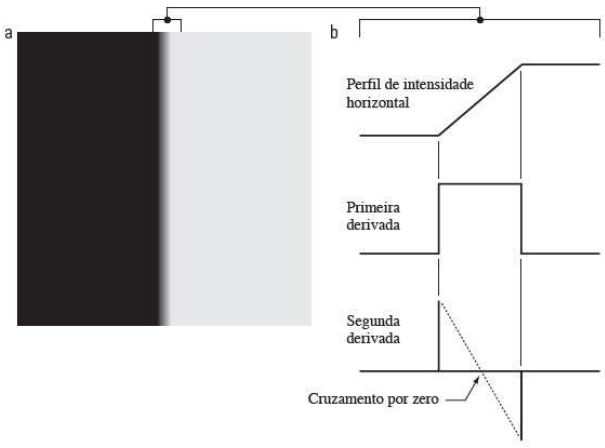
2.9.4 Detecção de bordas

Embora a detecção de pontos e linhas sejam ferramentas importantes para a segmentação, as técnicas para detecção de bordas são utilizadas com mais frequência, pois a detecção de pontos e linhas isoladas não é uma ocorrência comum na maioria das aplicações práticas. Por outro lado, a detecção de bordas é fundamental na maioria dos processos de segmentação (GONZALEZ; WOODS, 2010).

Uma borda pode ser definida como a mudança brusca em nível de cinza entre duas regiões relativamente homogêneas (MARQUES; VIEIRA, 1999).

Na Figura 31 apresenta uma imagem com duas regiões de intensidade constante que são separadas por uma borda vertical. Ao lado da imagem é apresentado o perfil de intensidade de níveis de cinza horizontal na representação das duas regiões da imagem. Observa-se que a magnitude da primeira derivada do perfil de níveis de cinza pode ser usada para a detecção da presença de uma borda e o sinal de resposta da segunda derivada pode ser usado para determinar se um *pixel* da borda se localiza no lado escuro ou claro da imagem (GONZALEZ; WOODS, 2010).

Figura 31 - Representação de uma borda e o sinal de respostas da 1ª e 2ª derivadas.



Fonte: Adaptado de Gonzalez e Woods (2010).

A primeira derivada em um determinado ponto da imagem é obtida usando a magnitude do gradiente do ponto de interesse e a segunda derivada é obtida utilizando-se um operador diferencial de segunda ordem.

2.9.5 Operador gradiente

Segundo Gonzalez e Woods (2010), a ferramenta ideal para encontrar a intensidade e a direção de uma borda é o gradiente dado pela expressão (7).

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}. \quad (7)$$

Esse vetor tem a propriedade geométrica de apontar no sentido da maior taxa de variação de f no local de *(x, y)* e com magnitude expressiva, o que é importante para a detecção de bordas.

A magnitude do vetor f é denotada por *M(x,y),* que representa o valor da taxa de variação na direção do vetor gradiente e é dada pela expressão (8).

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g^2x + g^2y}\,. \quad (8)$$

Dessa forma, utilizando as Equações (7) e (8), o cálculo do gradiente de uma imagem é resultado das derivadas parciais $\frac{\delta f}{\delta x}$ e $\frac{\delta f}{\delta y}$ na posição *(x, y)* para cada *pixel*.

A Equação (8) pode ser aproximada para estimar o gradiente com valores absolutos utilizando a Equação (9).

$$\nabla f = |G_x| + |G_y|. \quad (9)$$

Assim, a Equação (9) resulta em um menor tempo de processamento do que o tempo necessário com a Equação (8).

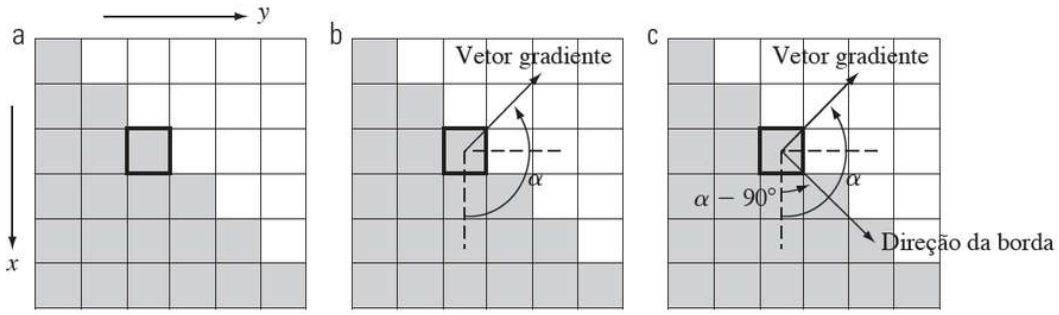
A direção do vetor do gradiente α(x, y), que representa o ângulo da direção do vetor f na posição *(x,y)*, pode ser calculada conforme a Equação (10):

$$\propto (x, y) = tan^{-1}\left(\frac{Gy}{Gx}\right). \quad (10)$$

Na Figura 32 mostra-se o uso do gradiente para determinar a intensidade e a direção da borda em um ponto na imagem. É possível notar na ilustração que a borda é perpendicular à

direção do vetor gradiente no ponto onde o gradiente é calculado.

Figura 32 - Representação do cálculo do gradiente.



Fonte: Gonzalez e Woods (2010).

Segundo Marques e Vieira (1999), a filtragem utilizando gradiente ou laplaciano pode ser aproximada por máscaras de convolução ou operadores 3x3, detalhados a seguir.

2.9.6 Operador de Roberts

Proposto por Lawrence Gilman Roberts (1965), esse é o operador gradiente mais tradicional no processamento digital de imagens e o mais simples de ser implementado, pois utiliza duas máscaras de convolução de tamanho 2x2, resultando em um valor de gradiente para cada diagonal da máscara (ROBERTS, 1965).

Na Figura 33 ilustra-se o padrão das máscaras proposto por Roberts.

Figura 33 - Máscaras diagonais para o cálculo do operador gradiente de Roberts.

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

(a) (b)

Fonte: Elaborado pelo autor.

As definições propostas por Roberts baseiam-se no cálculo das diferenças diagonais definidas pelas Equações (11) e (12):

$$g_x = (Z_9 - Z_5). \tag{11}$$

$$g_y = (Z_8 - Z_6). \tag{12}$$

Sendo Z5, Z6, Z8 e Z9 representados na matriz de uma imagem genérica na Figura 25.

O cálculo do gradiente utilizando o operador de Roberts é baseado nas Equações (8) e (12), e dados pela Equação (13).

$$M(x,y) = [\,(Z_9 - Z_5)^2 + (Z_8 - Z_6)^2]^{\frac{1}{2}}. \tag{13}$$

Porém, devido ao custo computacional para calcular as operações de potência, a equação (13) pode ser aproximada pela Equação (14).

$$M(x,y) \approx |(Z_9 - Z_5) + (Z_8 - Z_6)|. \tag{14}$$

Na Figura 34 ilustra-se o efeito produzido na imagem para a detecção de bordas na imagem utilizando os operadores de Roberts.

Figura 34 - Detecção de bordas utilizando operadores gradientes diagonais de Roberts.



(a) (b)

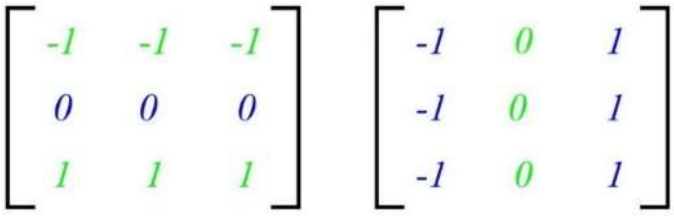
Fonte: Adaptado de Gonzalez e Woods (2010).

2.9.7 Operador de Prewitt

A detecção de borda de Prewitt foi proposta por Judith M. S. Prewitt (1970) para estimar a magnitude e a orientação da borda da imagem (PREWITT, 1970).

O operador de Prewitt utiliza máscaras de tamanho 3x3 como filtros para cálculo de gradiente. As máscaras são representadas na Figura 35:

Figura 35 - Máscara dos operadores de Prewitt.



Fonte: Adaptado de Marques e Vieira (1999).

Assim como nos operadores de Roberts, onde o gradiente foi aproximado pela equação (14), o cálculo dos operadores de Prewitt pode ser aproximado pelas Equações (15) e (16).

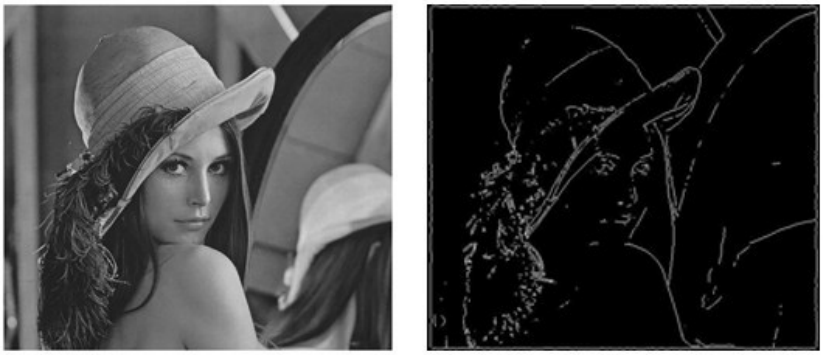
$$g_x = (Z_7 + Z_8 + Z_9) - (Z_1 + Z_2 + Z_3). \quad (15)$$

$$g_y = (Z_3 + Z_6 + Z_9) - (Z_1 + Z_4 + Z_7). \quad (16)$$

Sendo Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9 representados na matriz de uma imagem genérica na Figura 25.

Na Figura 36 ilustra-se o efeito produzido na imagem para a detecção de bordas na imagem utilizando os operadores de Prewitt.

Figura 36 - Detecção de bordas utilizando operadores de Prewitt.



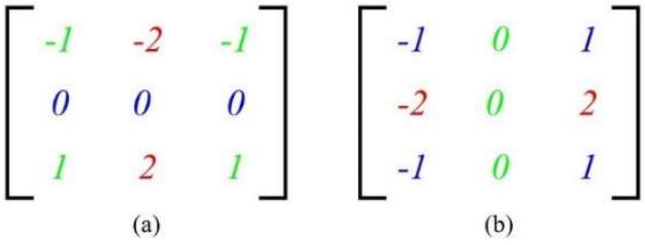
Fonte: Adaptado de Gonzalez e Woods (2010).

2.9.8 Operador de Sobel

O operador de Sobel foi proposto por Irwin Sobel em 1970 e tem a característica de ressaltar as bordas e também de suavizar a imagem, pois a derivação aumenta os ruídos presente na imagem em processamento (SOBEL, 1970)

O operador de Sobel utiliza máscaras de tamanho 3x3 como filtro para cálculo de gradiente. As máscaras são representadas pela Figura 37.

Figura 37 - Máscara dos operadores de Sobel.



Fonte: Adaptado de Marques e Vieira (1999).

Os cálculos dos operadores de Sobel podem ser aproximados pelas seguintes Equações (17) e (18).

$$g_x = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3). \quad (17)$$

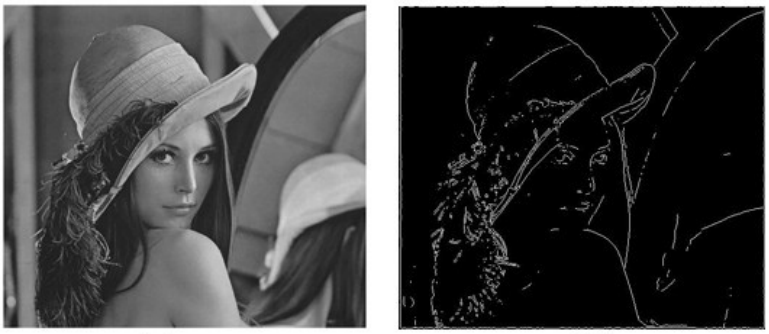
$$g_y = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7). \quad (18)$$

Sendo Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9 representados na matriz de uma imagem genérica na Figura 25.

O operador de Sobel introduziu o valor 2 nas Equações (15) e (16) como peso no centro do coeficiente para produzir a suavização da imagem.

Na Figura 38 é mostrado o efeito produzido na imagem para a detecção de bordas na imagem utilizando os operadores de Sobel.

Figura 38 - Detecção de bordas utilizando operadores de Sobel.



Fonte: Adaptado de Marques e Vieira (1999).

2.9.9 Transformada de *Hough*

A Transformada de *Hough* (TH) foi proposta por Paul Hough (1962) e foi elaborada para detectar características analiticamente representáveis em imagens binarizadas contendo apenas dois níveis de cinza, assim como linhas, círculos e elipses, mas tornou-se uma ferramenta de uso comum na visão artificial para o reconhecimento destas características (HOUGH, 1962).

2.9.10 Transformada de *Hough* para detecção de retas

Para o caso da reta, considerando um ponto específico localizado em (x_i, y_i) na imagem, a equação geral na forma inclinação-interseção, é dada pela Equação (19).

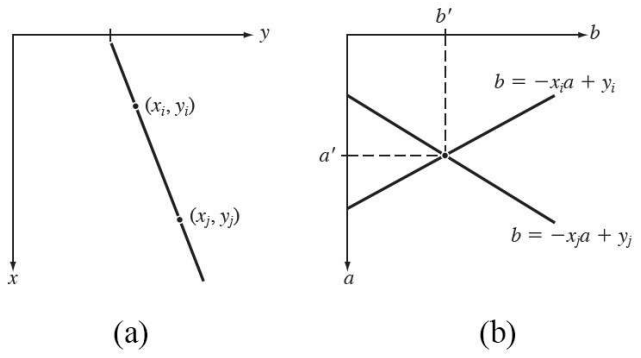
$$y_i = ax_i + b. \tag{19}$$

Embora infinitas retas passem por (x_i, y_i) todas elas satisfazem a Equação (19) para diferentes valores de *a* e *b.* Pode-se definir um espaço de parâmetros que produza uma única reta para pares fixos de $f(x_i, y_i)$, dado pela Equação (20):

$$b = -x_i a + y_i. \tag{20}$$

Caso seja inserido outro ponto específico na mesma imagem localizado em (x_j, y_j) todos os pontos no espaço de parâmetros se cruzam nos pontos *a'* e *b'* conforme ilustra a Figura 39.

Figura 39 - (a) Plano xy, (b) Espaço de parâmetros.



Fonte: Adaptado de Gonzalez e Woods (2010).

As Equações (19) e (20) apresentam uma dificuldade em sua representação, pois a inclinação de uma reta se aproxima do infinito a medida que reta se aproxima da direção vertical. Por esse motivo, utiliza-se a representação da reta em coordenadas polares, dada pela Equação (21).

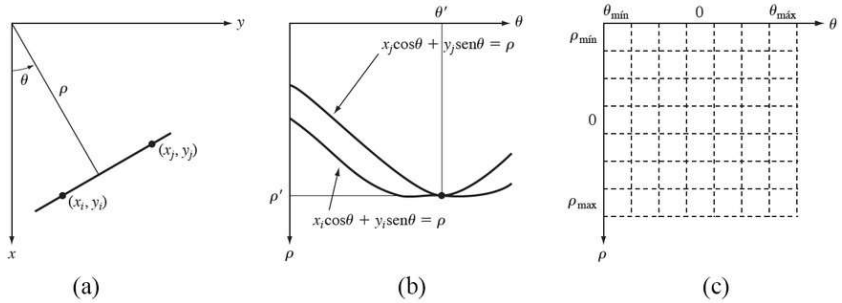
$$\rho = x\cos\theta + y\,sen\theta. \tag{21}$$

Sendo ρ a distância da reta até a origem e θ o ângulo entre o eixo *x* e a reta entre a origem e ponto mais próximo da imagem.

A Figura 40 (a) ilustra a representação geométrica dos parâmetros ρ e θ no plano *(x, y).* As senoides da Figura 40 (b) são a representação de retas que passam por um ponto (x_k, y_k) no plano *(x,y)* e o ponto de interseção ilustrado nessa mesma imagem corresponde a reta que passa por (x_i, y_i) e (x_j, y_j) no plano (*x,y)*. A Figura 40 (c) ilustra as células acumuladoras que poderão ser representadas computacionalmente por uma matriz *MxN*.

Cada célula tem um contador que é incrementado cada vez que um *pixel* for mapeado para aquela posição e, assim, as células que tiverem o maior valor no contador indicam maior possiblidade de ter a forma procurada, no caso, uma reta.

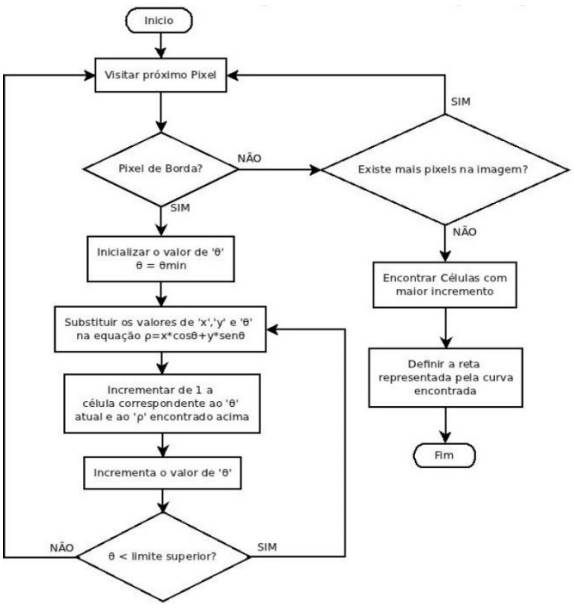
Figura 40 - Representação da reta na forma polar.



Fonte: Adaptado de Gonzalez e Woods (2010).

Um fluxograma para a elaboração de um algoritmo para implementação da TH utilizando o espaço de parâmetros ρ e θ foi proposto por Gonçalves (2011).

Figura 41 - Fluxograma do algoritmo da TH para retas utilizando o espaço de parâmetros.



Fonte: Gonçalves (2011).

Na Figura 41, ao final da varredura da imagem, tem-se uma matriz preenchida com o valor acumulado para cada par de coordenadas ρ, θ. A célula com maior valor indica que seus índices são parâmetros da melhor reta. Caso existam *n* retas a serem identificadas, serão escolhidos os *n* maiores valores na matriz acumuladora (MACEDO, 2005).

2.9.11 Transformada de *Hough* para detecção de círculos

Richard O. Duda e Peter E. Hart (1972) propuseram a utilização da TH para detectar círculos em uma imagem digitalizada. Para tal, é necessário que o acumulador seja uma matriz tridimensional, com informações sobre a posição *(x, y)*, em relação ao centro do círculo, e o raio (DUDA; HART, 1972).

Para a detecção de círculos usando a transformada de *Hough* pode-se usar a fórmula implícita pela Equação (22).

$$(x - x_0)^2 + (y - y_0)^2 = r^2. \tag{22}$$

onde x_0 e y_0 são coordenadas cartesianas do centro do círculo e r o seu raio.

A Equação (22) pode ser representada na forma polar pelas Equações (23) e (24).

$$x_0 = x - \rho cos\theta. \tag{23}$$

$$y_0 = y - \rho sin\theta. \tag{24}$$

O processo para calcular os valores da matriz acumuladora é o mesmo descrito na detecção de linhas. A célula com maior valor acumulado é a melhor candidata a ser o centro do círculo da imagem original.

Em Macedo (2005), é proposto um algoritmo para a detecção de círculos utilizando a TH, o algoritmo é ilustrado na Figura 42.

Figura 42 - Algoritmo para detecção dos parâmetros do círculo.

circulo_hough()

Guardar todos os pixels em um vetor

Limpar acumulador **M(x_0, y_0, ρ)**

Para pixel **I(x,y)**, se este pixel **I(x,y)** estiver aceso na imagem

Para ρ > **minimo** até ρ < **máximo**

Para **θ** = **0** até **θ** < **2π**

$x_0 = x - \rho\cos\theta$

$y_0 = y - \rho\sin\theta$

M(x_0, y_0, ρ) = **M(x_0, y_0, ρ) + 1**

Faça para **θ** = **0** a **2π**

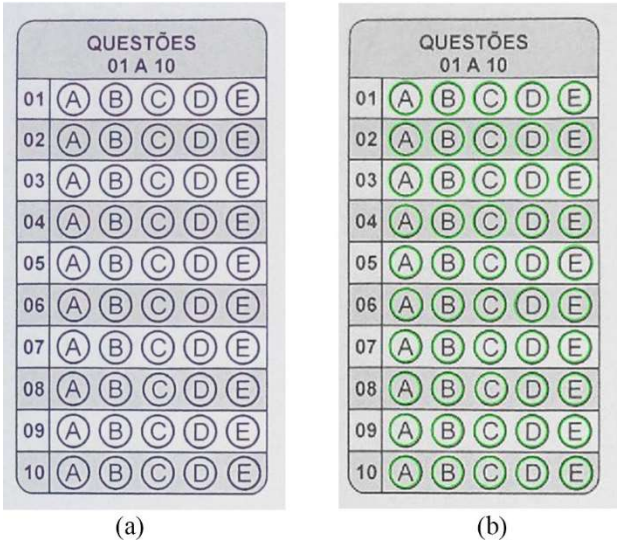
Faça para todos os raios de **mínimo** a **máximo**

Faça para todos os pixels da imagem

Fonte: Adaptado de Macedo (2005).

Na Figura 43 mostra-se o resultado da aplicação da TH para detecção dos círculos em uma imagem.

Figura 43 - (a) Imagem original, (b) Imagem após a aplicação da TH.



Fonte: Elaborado pelo autor.

3 RECONHECIMENTO DE PADRÕES

Nesse capítulo são apresentados os conceitos básicos sobre reconhecimento de padrões em imagens digital e são descritas algumas técnicas utilizadas para a composição da metodologia da pesquisa.

Para Haykin (2001), o reconhecimento de padrões pode ser definido como o processo pelo qual um padrão ou sinal que foi recebido é atribuído a uma classe dentre um número pré-determinado de classes.

Segundo Kulkarni (1994), os sistemas de reconhecimento de padrões utilizam- se de um espaço de características onde um vetor ao qual se procura possa ser mapeado. Através desse vetor pode-se verificar a qual classe ele pertence.

De acordo com Tou e Gonzalez (1974), reconhecimento de padrões é o processo de identificar objetos através das extrações de suas características a partir de dados sobre o objeto. É válido ressaltar que o objeto em questão não precisa ser necessariamente algo concreto, pode ser por exemplo, reconhecer padrões de sinais elétricos, padrões de comportamento, padrões sonoros, padrões numéricos e outros.

3.1 SISTEMAS DE RECONHECIMENTO DE PADRÃO

Segundo Erpen (2004), os sistemas de reconhecimento de padrões compartilham uma estrutura composta geralmente de 5 etapas que são ilustradas na Figura 44.

Figura 44 - Etapas de um sistema de reconhecimento de padrões.



Fonte: Adaptado de Erpen (2004).

3.1.1 Aquisição de imagem

A etapa de aquisição da imagem é o processo tratado no capítulo (2.1), onde ocorrem a atuação dos sensores para captação da imagem.

3.1.2 Pré-processamento

A etapa de Pré-Processamento é considerada opcional, pois, tem como objetivo corrigir imperfeições ou defeitos que foram incorporados durante a aquisição da imagem. As técnicas relacionadas a essa etapa são os filtros de ruídos, aumento de contrastes e operações geométrica para rotação da imagem capturada.

3.1.3 Segmentação

Para Marques e Vieira (1999), a etapa básica da segmentação é dividir a imagem em unidades significativas, onde cada segmento da imagem refere-se a um objeto, ou conjunto de objetos que o segmento representa. As partes que se deseja segmentar, geralmente são distinguíveis pelos seus níveis de cinza ou pelas propriedades de cores dos *pixels*.

3.1.4 Extração de características

A extração de características é um processo que visa a obtenção de informações quantitativas a partir de imagens digitais, que sejam capazes de diferenciá-las, convertendo cada tipo de imagem em dados para reconhecimento. Percebe-se que nessa etapa a entrada é uma imagem e a saída é um conjunto de dados.

3.1.5 Classificação

A etapa de classificação considera a existência de similaridade entre os objetos que tal forma que suas características possibilitem a separação por classe ou agrupamento. De maneira mais simples podemos dizer que essa fase consiste em reconhecer o objeto ou uma forma na imagem. Sendo assim, dado uma imagem de entrada, o sistema determina a que classe o padrão pertence.

Nas literaturas são apresentadas inúmeras técnicas para reconhecimento de padrões que vão desde abordagens estatísticas à implementação de Redes Neurais Artificiais. Na análise do

problema é que se determina qual estratégia deverá ser empregada.

Será apresentada a técnica de Correspondência de Modelo (*Template Matching*) utilizada na metodologia desse trabalho para reconhecimento de uma área de interesse na imagem.

3.2 CORRESPONDÊNCIA DE MODELO

Segundo Mahalakshmi, Muthaiah e Swaminathan (2012), a técnica de correspondência de modelo (*Template Matching*) pode ser usada para classificação de objetos comparando partes de uma imagem com outra. Algumas aplicações utilizam essa técnica para detecção de uma imagem alvo ou como parte do processo para traçar rota para um robô móvel.

Para Kumar (2015), a técnica de correspondência de modelo pode ser subdividida em duas abordagens: baseada em características e baseada em modelo ou área.

A abordagem baseada em características é adequada quando as duas imagens, a de referência e a modelo, têm mais correspondência em relação as características e utiliza pontos de controle definidos, como pontos, curvas, modelos de superfícies que precisam ser coincidentes. O objetivo é localizar conexões de pares entre a referência e o modelo usando relações espaciais ou descritores de características (KUMAR; AGRAWAL, 2015).

Segundo Mahalakshmi, Muthaiah e Swaminathan (2012), a abordagem baseada em modelo ou área é também chamada de método de correlação de semelhança. As correspondências são estimadas com base nos valores de intensidade da imagem e do modelo. Algumas técnicas que fazem uso dessa abordagem são: diferenças quadradas da intensidade, métodos baseados em correlação e métodos de otimização e informação mútua.

Para esse trabalho, optou-se pela implementação baseada em modelo utilizando a abordagem de correlação cruzada, por ser uma técnica mais utilizada.

3.3 CORRELAÇÃO CRUZADA NORMALIZADA

Uma medida de similaridade muito utilizada em processamento de imagens é a correlação cruzada normalizada (CCN). Nesse método, os valores da imagem padrão P são deslocados pela imagem I. Para cada posição do *pixel* em I, multiplica-se pelos valores de dos *pixels* de P. O somatório dessa operação é armazenado em cada posição do deslocamento.

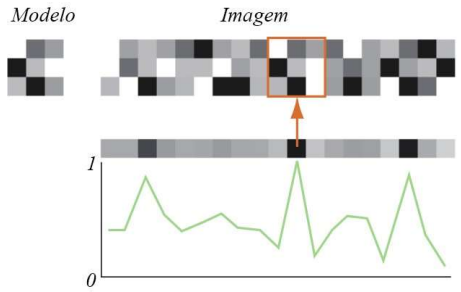
A Equação (25) foi proposta por Russ (1999), e sua aplicação gera uma matriz de

correlação contendo valores no intervalo entre 0 e 1 indicando a proximidade da semelhança entre os padrões.

$$N(m,n) = \frac{\sum k \sum j I(k,j).P(k-m,j-n)}{\sqrt{\sum k \sum j I^2(k,j).\sum k \sum j P^2(k-m,j-n)}}. \quad (25)$$

Na Figura 45 ilustra-se a definição da busca pela correlação entre duas imagens. Os valores mais próximos de 1 indicam maior semelhança entre as imagens.

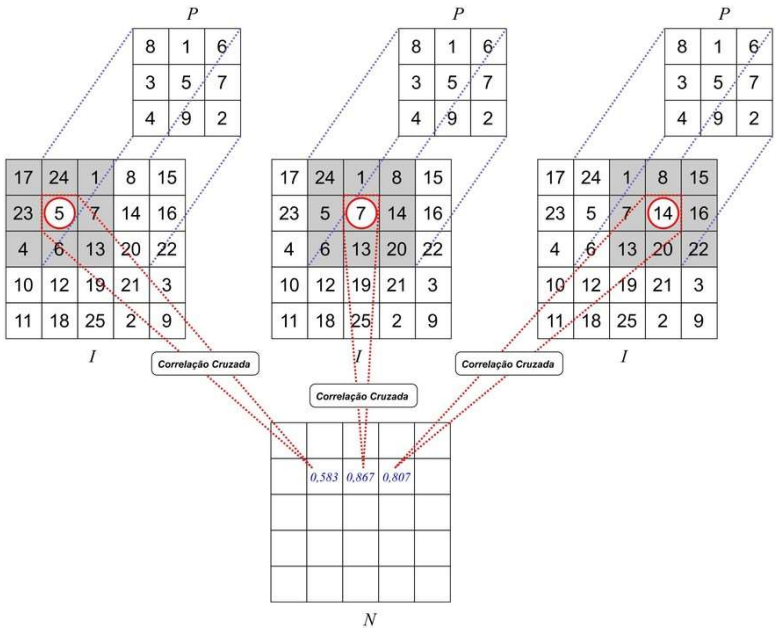
Figura 45 - Resultado da correlação cruzada.



Fonte: Adaptado de Russ (1999).

A Figura 46 é ilustra a aplicação da Equação (25) para a obtenção da matriz de similaridade que foi calculada entre a matriz de *pixels* da imagem *I* e a matriz de *pixels* da imagem *P*.

Figura 46 - Operação de correlação normalizada.



Fonte: Adaptado de Pereira (2006).

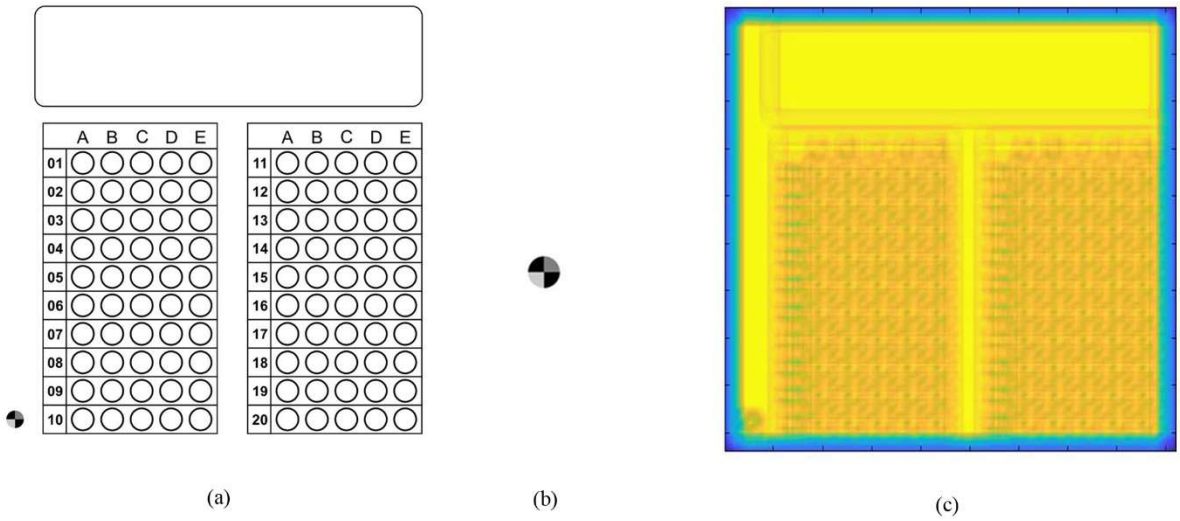
A normalização indicada pela Equação (25) é sensível a variações de média ao longo da imagem, diante disso, Gonzalez e Woods (2010), propõe uma modificação no cálculo para tornar o resultado mais robusto, calculando a média local da imagem *I* e também do padrão *P* antes de se realizar a correlação, conforme a Equação (26) a seguir:

$$N(m,n) = \frac{\sum k'j'\left(P(k',j').I(k+k',j+j')\right)}{\sqrt{\sum k',j'P(k',j')^2.\sum k',j'I(k+k',j+j')^2}}. \quad (26)$$

Sendo *N(m,n)* a matriz resultante, *P(k,j)* a representação da posição dos *pixels* na imagem padrão e *I(k,j)* a posição dos *pixels* na imagem original.

A Figura 47 ilustra a aplicação da CCN, sendo em (a) a imagem principal e em (b) a imagem que busca verificar os *pixels* que mais se assemelham. Utilizando o *software* MATLAB® gerou-se a imagem (c) que mostra o mapa de correlação entre as imagens, sendo as áreas mais escuras os valores mais altos de correlação, indicando possíveis candidatos para a imagem procurada.

Figura 47 - Correlação Cruzada Normalizada - (a) Imagem a ser analisada, (b) Padrão a ser procurado, (c) Mapa de Correlação.



Fonte: Elaborado pelo autor.

Uma deficiência da correlação cruzada normalizada é a sua sensibilidade à escala. Um padrão, se escalonado por um fator de crescimento ou redução em suas dimensões, afeta significativamente o reconhecimento da imagem. Tal problema relacionado à escala pode forçar a utilização de um banco de padrões com escalas diferentes para se evitar reconhecimentos indevidos.

4 METODOLOGIA PROPOSTA

Neste capítulo são apresentados os materiais utilizados e o método proposto para a correção automática de testes de múltipla escolha com base em técnicas de processamento digital de imagem.

O método propõe que baseado em um cartão de respostas onde não foi realizado nenhuma marcação, seja criado um modelo de referência que nesse trabalho é tratado como modelo de referência ou *template*.

O *template* utilizado nessa pesquisa foi desenvolvido utilizando o programa para editoração eletrônica CorelDRAW® Graphics Suite 2017 conforme descrito na Tabela 1.

Tabela 1 - Especificações do *template*.

Especificações	Valor
Largura	200 mm
Altura	120 mm
Quantidade de questões	30
Quantidade de opções para marcação por questão	5

Fonte: Dados da pesquisa do autor.

Foi realizada a impressão do *template* em papel comum em impressora a laser, na sequência o cartão foi digitalizado com um *scanner* de mesa, modelo Fujitsu® SP-1120 com capacidade para digitalizar até 20 páginas por minuto, a bandeja desse equipamento comporta até 50 folhas.

As configurações para digitalização da imagem foram ajustadas para terem as especificações descritas na Tabela 2 .

Tabela 2 - Especificações para a digitalização do *template*.

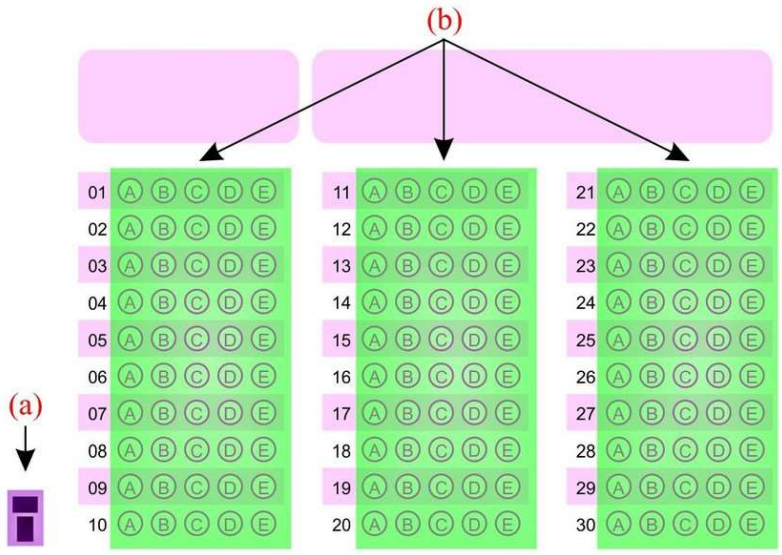
Especificações	Valor
Papel	A4 - 210x297 mm
Resolução	300 dpi
Perfil de cores	RGB
Largura em *pixel*	2480
Altura em *pixels*	3508

Fonte: Dados da pesquisa do autor.

Depois da digitalização do *template* foi realizado a etapa de criação de um arquivo para registrar suas informações. Nessa etapa considerou-se que registraros dados em um arquivo do tipo XML (*eXtensible Markup Language*), Figura 50, seria a forma mais simples para não ter a necessidade de criar um sistema computacional interligado à um banco de dados, pois o arquivo XML seria utilizado como repositório para consultar as parametrizações do *template*.

As áreas de interesse para criação do *template* são ilustradas na Figura 48.

Figura 48 - Áreas de interesse: (a) marcador de borda da página, (b) áreas para marcar questões.



Fonte: Elaborado pelo autor.

Na Figura 49 ilustra-se o diagrama para a criação do *template* em formato XML. A detecção dos círculos utilizando a transformada de *Hough* é realizada nessa etapa para não ter que aplicar em todos os cartões. Ao aplicar a TH nessa etapa, as coordenadas de todos os campos ficam salvos no arquivo XML.

Figura 49 - Diagrama para criação do *template*.



Fonte: Elaborado pelo autor.

A Figura 50 mostra alguns dados do arquivo XML que contém as informações do cartão sem a realização das marcações. Esses dados foram utilizados, por exemplo, para comparar a intensidade de pigmentação dos campos onde são realizadas as marcações.

Figura 50 - Arquivo XML contendo os parâmetros do *template*.

```
<seletorAreaQuestao>
    <areaQuestao type="3" id="0" x="1139.0" y="1155.0" height="1240.0" width="252.0"
        <mCor idArea="0" mX="1162.5" mY="1179.5" mRaio="20" contPixel="570"/>
        <mCor idArea="0" mX="1213.5" mY="1179.5" mRaio="20" contPixel="514"/>
        <mCor idArea="0" mX="1264.5" mY="1179.5" mRaio="20" contPixel="551"/>
        <mCor idArea="0" mX="1314.5" mY="1179.5" mRaio="20" contPixel="560"/>
        <mCor idArea="0" mX="1365.5" mY="1179.5" mRaio="20" contPixel="570"/>
        <mCor idArea="0" mX="1162.5" mY="1229.5" mRaio="20" contPixel="570"/>
        <mCor idArea="0" mX="1213.5" mY="1229.5" mRaio="20" contPixel="514"/>
        <mCor idArea="0" mX="1264.5" mY="1229.5" mRaio="20" contPixel="551"/>
        <mCor idArea="0" mX="1314.5" mY="1229.5" mRaio="20" contPixel="560"/>
        <mCor idArea="0" mX="1365.5" mY="1229.5" mRaio="20" contPixel="570"/>
        <mCor idArea="0" mX="1163.5" mY="1279.5" mRaio="20" contPixel="570"/>
        <mCor idArea="0" mX="1213.5" mY="1280.5" mRaio="20" contPixel="514"/>
        <mCor idArea="0" mX="1264.5" mY="1279.5" mRaio="20" contPixel="551"/>
        <mCor idArea="0" mX="1315.5" mY="1280.5" mRaio="20" contPixel="560"/>
```

Fonte: Dados da pesquisa do autor.

Para validação do método proposto foi desenvolvido um código utilizando a linguagem de programação Python que tem como proposta ser uma linguagem que traz produtividade e legibilidade em sua utilização.

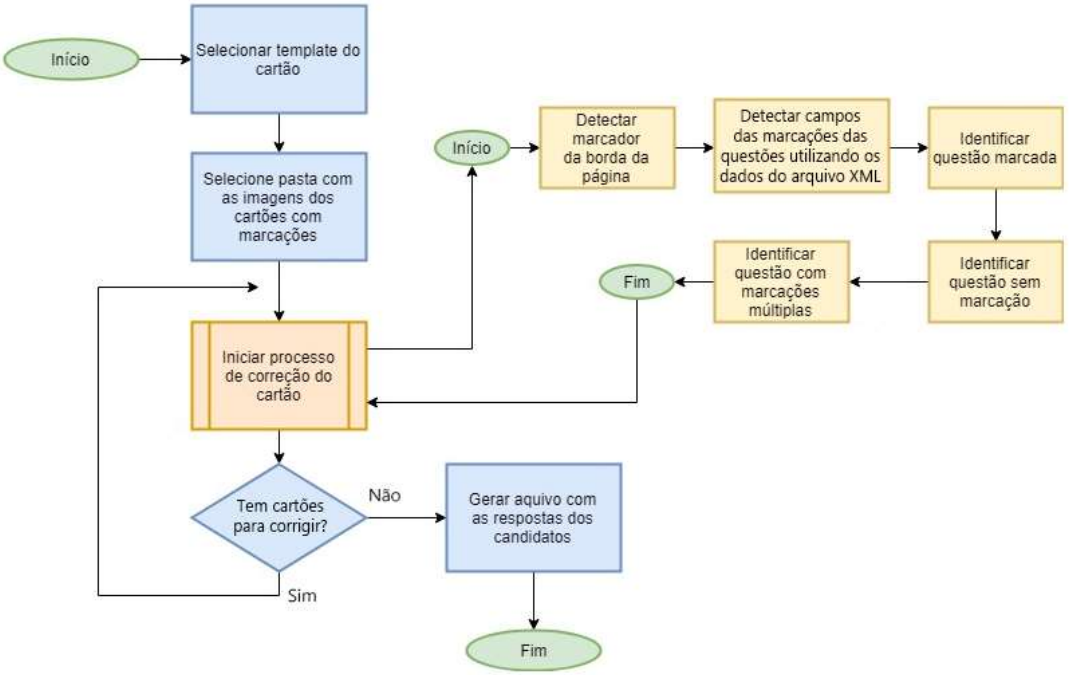
Foi utilizada a biblioteca de processamento de imagens OpenCV que é uma biblioteca multiplataforma e de código fonte aberto e livre para o uso acadêmico ou comercial. Devido a essas particularidades ela é amplamente utilizada em implementações que requerem

processamento de imagens. A vantagem de sua utilização está no fato dela trazer diversos algoritmos para processamento de imagens implementados.

Após a criação do *template*, foram impressos 100 cartões onde foram realizados preenchimentos aleatórios das questões de 1 a 10. Todos os cartões foram preenchidos com caneta na cor azul. A cobertura do preenchimento das opções das questões variou entre 40% e 100% de cobertura de tinta.

Todos os cartões foram digitalizados e colocados em uma pasta no computador. O código desenvolvido para validação dessa proposta seguiu o diagrama especificado na Figura 51 .

Figura 51 - Diagrama com as etapas do processo de correção de um cartão de respostas.



Fonte: Elaborado pelo autor.

4.1 DETECTAR O MARCADOR DE BORDA DA PÁGINA

Para realizar a detecção do marcador de borda utilizou-se a técnica de *Template Matching*. A base desse método é a CCN conforme apresentou a seção 3.2.

Na criação do *template*, quanto foi selecionada a área que definiu o marcador de borda, Figura 48 (a), foram registrados os parâmetros das coordenadas daquela área. No processo de detecção do marcador de borda da página da imagem a ser corrigida, é realizado o recorte na imagem *template* na região definida no arquivo XML.

A imagem recortada do *template* é utilizada como parâmetro na técnica de *Template Matching* juntamente com a imagem do cartão a ser corrigido.

Esse processo resulta na geração de um mapa contendo os resultados das similaridades encontradas entre as imagens, no qual os valores mais altos indicam um grau de similaridade mais próximo do pretendido. Com esses valores é possível identificar as coordenadas no cartão com as marcações que se encontram na imagem do marcador de bordas, fazendo esse posicionamento ser utilizado como referência inicial para posicionamento dos demais elementos do processo.

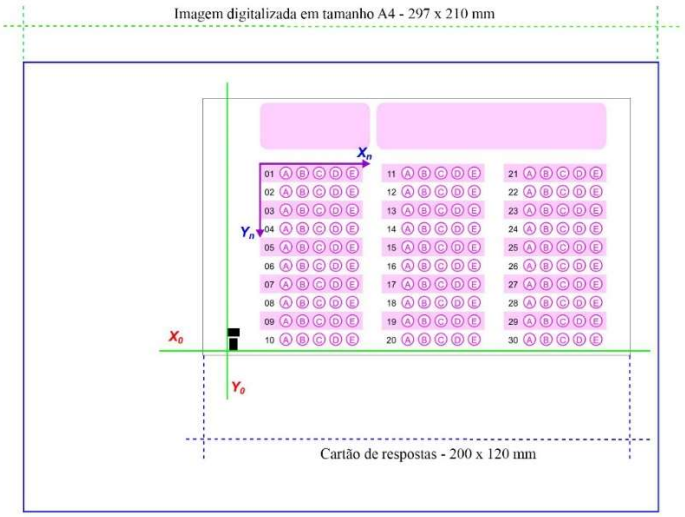
Considerando a definição de uma imagem monocromática do tipo $f(x, y)$, sendo X_0 e Y_0 o posicionamento do marcador de borda e tomando X_n e Y_n como o posicionamento de uma das áreas para marcação de questões, pode-se definir sua distância do marcador de borda utilizando as Equações (27) e (28) a seguir:

$$\Delta Sx_0 = X_n - X_0. \tag{27}$$

$$\Delta Sy_0 = Y_n - Y_0. \tag{28}$$

Na Figura 52 ilustra-se os elementos X_0, Y_0 que representam as coordenadas do marcador de borda e X_n, Y_n que representam as coordenadas de uma área para marcação de questões.

Figura 52 - Identificação do marcador de borda e da área de marcação de questões.



Fonte: Elaborado pelo autor.

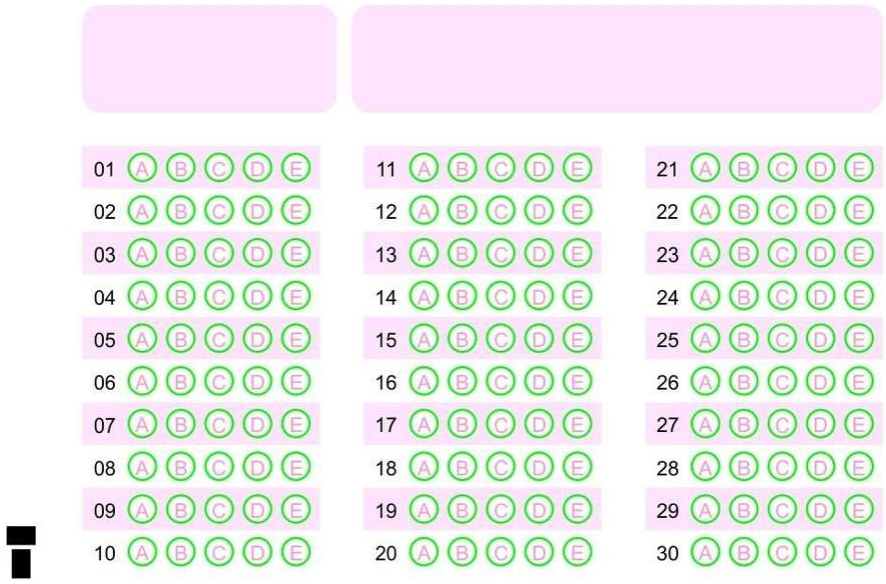
Utilizando os valores de altura e largura da área de marcação da questão que estão registrados no arquivo XML cria-se uma sub imagem para processamento.

4.2 DETECTAR CAMPOS PARA MARCAÇÃO DAS QUESTÕES

Os campos para marcação das questões são identificados utilizando a *Transformada de Hough* na fase de criação do *template*. Nessa fase são registradas as coordenadas de cada campo conforme ilustra a Figura 53.

A TH é aplicada apenas nas áreas que foram selecionadas para serem processadas como áreas de marcação de questão.

Figura 53 - Resultado da aplicação da transformada de *Hough* nas áreas selecionadas para processamento.



Fonte: Elaborado pelo autor.

Conforme a Figura 50, que ilustra os registros contidos no arquivo XML, é possível observar na linha 8, por exemplo, que foram armazenadas informações das coordenadas *(x,y)*, o raio e a contagem de *pixel* da área de marcação da questão.

4.3 DETECÇÃO DAS QUESTÃO MARCADA

O método utilizado como forma de identificar qual campo foi preenchido como opção de resposta da questão foi a contagem de *pixels* para o grupo de opções de uma questão.

A contagem de *pixel* é um método que totaliza a quantidade de *pixels* que atendem a um

limiar pré-definido da tonalidade de cor determinada. Para o método proposto, considerou-se realizar o processamento dos cartões convertendo a imagem digitalizada em tons de cinza representados por valores entre 0 e 255.

Assim, dada uma região de interesse *ri* da imagem de tamanho *MxN*, pode-se realizar a contagem de *pixels* da região de interesse, *CPri,* que atenda a um valor de limiar *Vl* qualquer conforme a Equação (29).

$$CPri = \sum_{M}^{N} \begin{cases} 1\ ri(x,y) < Vl \\ 0\ ri(x,y) \neq Vl \end{cases}. \tag{29}$$

Um algoritmo para realizar a contagem de *pixels* em *ri* é exibido na Figura 54.

Figura 54 - Algoritmo para a contagem de *pixels*.

```
conta_pixel()
        Guardar todos os pixels da região de interesse em uma matriz MxN
        Guardar o valor do limiar, Vl
        Iniciar contador de pixels da região de interesse com zero, CPri = 0
                Para i = 0 até i < M
                        Para j = 0 até j < N
                                Se o valor do pixel na posição MxN < Vl
                                        CPri = CPri + 1
                        Faça para j = 0 até j < N
                Faça para i = 0 até i < M
        Retorna o valor de CPri
```

Fonte: Elaborado pelo autor.

Utilizando o algoritmo da Figura 54 e os dados com as coordenadas de posicionamento de cada campo para a marcação das opções com os dados que constava no arquivo XML criou-se o algoritmo ilustrado na Figura 55 para identificar qual questão foi marcada considerando-se o grupo de opções que podem ser marcadas na questão e o campo que possui o maior valor acumulado.

Figura 55 - Algoritmo para identificar a questão que foi marcada.

opcao_marcada ()

Iniciar o vetor de opções, ***VOp[‘A’,‘B’, ‘C’, ‘D’, ‘E’]***

Pegar as coordenadas das opções no arquivo XML.

Realizar **conta_pixel()** nas coordenadas e guardar em vetor, ***CPOp[]***

Considera o primeiro elemento de ***CPOp[]*** o maior valor, ***Mval*** = ***CPOp[0]***

Para i = 0 até i < Tamanho de ***VOp***

Se ***Mval < CPOp[i + 1]***

Mval = CPOp[i + 1]

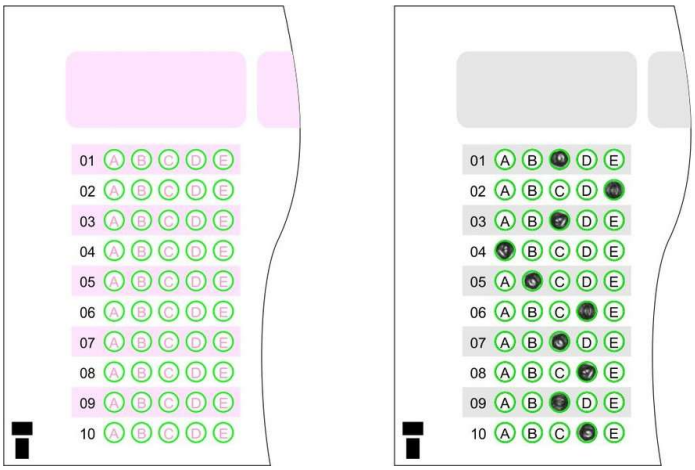
Faça para i = 0 até i < Tamanho de ***VOp***

Retorna o valor no índice de ***VOp[Mval]***

Fonte: Elaborado pelo autor.

Na Figura 56, ilustra-se o processo de utilização das coordenadas do arquivo XML para identificação das áreas onde devem ocorrer a contagem de *pixels*.

Figura 56 - Ilustração da contagem de *pixels*.



Fonte: Elaborado pelo autor.

Na Tabela 3 tem-se os valores acumulados da contagem de *pixels* da imagem anterior que contém as marcações no cartão.

Tabela 3 - Contagem de *pixel* referente ao cartão de resposta.

	A	B	C	D	E
Questão 1	172	183	938	182	194
Questão 2	194	208	195	219	902
Questão 3	209	219	820	224	238
Questão 4	782	254	200	228	212
Questão 5	204	863	190	223	194
Questão 6	209	259	175	687	218
Questão 7	205	226	713	206	190
Questão 8	200	224	158	913	194
Questão 9	223	241	780	210	201
Questão 10	184	198	144	800	194

Fonte: Dados da pesquisa do autor.

É possível identificar na Tabela 3,que os campos que foram marcados no cartão de reposta obtiveram maior contagem de *pixels*.

Observa-se ainda que embora alguns campos não estão marcados o processo de contagem de *pixel* retorna valores para essas áreas.

Para o modelo de cartão de respostas utilizado isso ocorreu pelo fato de queos campos possuem caracteres impressos internamente e cada caractere tem uma área de cobertura diferente do outro.

Para minimizar esse efeito, pode-se utilizar um método para normalizar esses valores. Tal normalização foi realizada considerando-se o valor do acumulador como sendo a diferença entre a contagem de *pixel* após a marcação do campo e a contagem de *pixel* antes da marcação do campo, como demonstra a Equação (30).

$$A_n = A_{pm} - A_{am}. \tag{30}$$

sendo *An* o valor do acumulador normalizado, *Apm* o valor do acumulador após a marcação do campo e *Aam* o valor acumulador antes da marcação docampo.

Dessa forma foi possível minimizar grande parte dos valores apresentados para os campos

sem marcação.

Para melhorar esse resultado, foi definido um limite mínimo *Lm* da contagem de *pixel* entre as diferenças dos acumuladores, que caso não atingido, o acumulador recebe o valor zero, como demonstra a Equação (31).

$$A_{lm} = \begin{cases} A_n \; A_n \geq Lm \\ 0 \; A_n < Lm \end{cases}. \tag{31}$$

Aplicando a Equação (31), e utilizando *Lm* igual a 155, os resultados obtidos para a contagem de *pixel* foram normalizados para uma representação ideal conforme apresentado na Tabela 4.

Tabela 4 - Contagem de *pixel* normalizados pelas Equações (30) e (31).

	A	B	C	D	E
Questão 1	0	0	910	0	0
Questão 2	0	0	0	0	833
Questão 3	0	0	668	0	0
Questão 4	622	0	0	0	0
Questão 5	0	672	0	0	0
Questão 6	0	0	0	524	0
Questão 7	0	0	560	0	0
Questão 8	0	0	0	738	0
Questão 9	0	0	633	0	0
Questão 10	0	0	0	592	0

Fonte: Dados da pesquisa do autor.

4.4 IDENTIFICAR QUESTÃO SEM MARCAÇÃO

No processo de correção do cartão é necessário identificar um grupo de opções de uma questão não teve opções marcadas.

Durante os testes realizados a utilização das Equações (30) e (31) fizeram com que os campos que continham marcação ficassem com os valores zerados.

Dessa forma, foi criada uma função para verificar os valores das contagens de *pixels* para as opções de marcação da questão, se todos os valores forem zeros então a questão é considerada sem marcação.

Na Figura 57 é apresentado o algoritmo para verificar se uma questão foi deixada em branco, ou seja, em marcação.

Figura 57 - Algoritmo para identificar se a questão está em branco.

questão_branco ()

Pegar as coordenadas das opções no arquivo XML

Pegar contagem de pixel do cartão antes das marcação no arquivo XML e guardar em vetor, ***CPam[]***

Pegar valor do limiar, ***Lm***

Realizar **conta_pixel()** nas coordenadas e guardar em vetor, ***CPpm[]***

Considerar total de questões em branco igual a zero, ***TotalQuestBranco*** = *0*

Considerar resultado como falso, ***Resultado*** = *Falso*

Para i = 0 até i < Tamanho de ***CPpm[]***

Alm = ***CPpm[i] - CPam[i]***

Se ***Alm*** < ***Lm***

TotalQuestBranco = ***TotalQuestBranco*** + *1*

Faça para i = 0 até i < Tamanho de ***CPpm[]***

Se **TotalQuest*Branco*** **=** Tamanho de ***CPpm[]***

Resultado = ***Verdadeiro***

Retorna o valor de ***Resultado***

Fonte: Elaborado pelo autor.

4.5 IDENTIFICAR QUESTÃO COM MARCAÇÕES MÚLTIPLAS

Para identificar se uma questão contém múltiplas marcações foi criado uma função que verifica a existência de mais de um valor acima de zero para as opções de marcações da questão.

Isso indica que as equações (30) e (31) não foram capazes de zerar os valores da contagem de *pixels*, indicando que a coluna tem valores acima do limite mínimo parametrizado em *Lm*.

Na Figura 58 é apresentado o algoritmo para verificar a existência de múltiplas marcações em uma questão.

Figura 58 - Algoritmo para identificar se a questão está com marcações múltiplas.

marcou_multiplas ()

Pegar as coordenadas das opções no arquivo XML

Pegar contagem de pixel do cartão antes das marcação no arquivo XML e guardar em vetor, ***CPam[]***

Pegar valor do limiar, ***Lm***

Realizar **conta_pixel()** nas coordenadas e guardar em vetor, ***CPpm[]***

Considerar múltiplas marcações igual a zero, ***MarcouMultiplas*** = *0*

Considerar resultado como falso, ***Resultado*** = *Falso*

Para i = 0 até i < Tamanho de ***CPpm[]***

Alm = ***CPpm[i] - CPam[i]***

Se ***Alm*** ≥ ***Lm***

MarcouMultiplas = ***MarcouMultiplas*** + *1*

Faça para i = 0 até i < Tamanho de ***CPpm[]***

Se ***MarcouMultiplas*** > 1

Resultado = ***Verdadeiro***

Retorna o valor de ***Resultado***

Fonte: Elaborado pelo autor.

5 RESULTADOS

O método proposto nesse trabalho foi testado utilizando a linguagem Python 2.7 e a biblioteca OpenCV na versão 2.4.

O computador usado nos testes dispunha de um processador de 7 núcleos com 2.40 GHz e 8 GB de memória RAM.

Os primeiros testes foram realizados em ambiente simulado com o preenchimento manual dos cartões por um grupo de 3 pessoas que preenchiam aleatoriamente a sequência de cartões que lhes foram apresentados.

Foram analisados 100 cartões contendo 30 perguntas com 5 opções de respostas, mas somente 10 questões foram preenchidas.

Durante os testes para identificar o marcador de borda da página que utilizou a técnica de *Template Matching*, o processamento de cada cartão foi executado em 93 milésimos de segundos.

Já para processar as áreas de marcação das respostas, o tempo de processamento foi de cerca de 11 milésimos de segundos, considerando as verificações dos campos que não foram marcados e as questões com múltiplas marcações.

Considerando esse tempo constante, o tempo total para realizar a correçãode cada cartão de resposta utilizado nesse trabalho foi de 1,26 segundos.

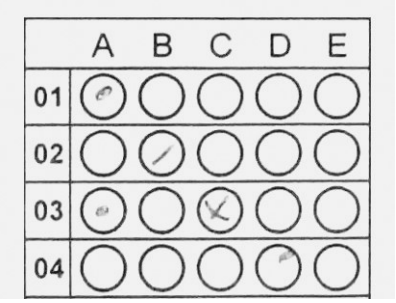
5.1 CAUSAS DE INCONSISTÊNCIAS NA CORREÇÃO

Embora existam instruções para o correto preenchimento dos campos para marcação das questões é possível que alguns candidatos não se atentem para essas informações e realizem o preenchimento dos campos de forma insuficiente.

Para que o método proposto seja eficiente os campos onde serão realizadas as marcações devem ter uma boa cobertura na sua área de preenchimento.

Foram realizados testes com o modelo de gabarito apresentado na Figura 59 para identificar o percentual de cobertura ideal para que não ocorram inconsistências na identificação das questões marcadas.

Figura 59 - Preenchimento insuficiente.



Fonte: Elaborado pelo autor.

Percebe-se que na Figura 59, embora existam marcações em todos os campos essas marcações ocupam um percentual pequeno em relação à área disponível para esse fim.

Os resultados apresentados na Tabela 5 mostram os valores registrados para a contagem de *pixel* realizada nos campos da Figura 59 após a aplicar as Equações (30) e (31) e variando o valor de *Lm* entre 0 e 50.

Tabela 5 - Teste de valores para *Lm* realizados na Figura 59.

Valor	**Questão 1**					**Questão 2**					**Questão 3**					**Questão 4**				
de Lm	**A**	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**
0	89	1	1	2	2	2	60	3	0	1	39	3	143	1	0	3	0	2	30	0
10	89	0	0	0	0	0	60	0	0	0	39	3	143	1	0	0	0	0	30	0
20	89	0	0	0	0	0	60	0	0	0	39	3	143	1	0	0	0	0	30	0
30	89	0	0	0	0	0	60	0	0	0	39	3	143	1	0	0	0	0	0	0
40	89	0	0	0	0	0	60	0	0	0	0	0	143	0	0	0	0	0	0	0
50	89	0	0	0	0	0	60	0	0	0	0	0	143	0	0	0	0	0	0	0

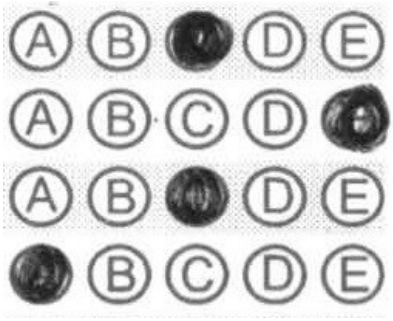
Fonte: Dados da pesquisa do autor.

O valor de *Lm* pode ser ajustado para atender as especificações de cada correção, como exemplo, percebe-se que na questão 3 foram realizadas marcações nas opções A e C. Dependendo do tipo de registro que se deseja obter essa marcação pode ser considerada como "marcação dupla", para os valores de *Lm* entre 0 e 30. Caso se deseje desconsiderar pequenas variações nas marcações, pode-se ajustar o valor de *Lm* para valores entre 40 e 50, assim, como se pode verificar no quadro anterior, a questão 3, opção A e a questão 10, opção D, que estão com valores de contagem de *pixels* próximos, são registrados apenas os valores máximos após a aplicação das Equações (30) e (31) e ajustando o valor de *Lm* de acordo com as necessidade

de correção.

Foi realizado teste para analisar o comportamento do método utilizando outro modelo de gabarito conforme apresentado na Figura 60.

Figura 60 - Modelo com opções ao centro.



Fonte: Elaborado pelo autor.

O modelo de gabarito apresentado na Figura 60 tem a característica de trazer impresso a opção dentro da área de marcação da questão. Isso gera um valor de contagem de *pixels* maior que o modelo de gabarito apresentado na Figura 59, dessa forma os valores de *Lm* utilizados foram maiores. Percebe-se que o preenchimento completo das opções desejadas resulta em valores altos na contagem de *pixel* conforme ilustrado na Tabela 6.

Tabela 6 - Teste de valores para *Lm* realizados na Figura 60.

Valor	**Questão 1**					**Questão 2**					**Questão 3**					**Questão 4**				
de Lm	**A**	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**
60	89	79	725	69	68	65	82	0	86	704	0	0	583	0	0	520	0	0	0	0
70	89	79	725	0	0	0	82	0	86	704	0	0	583	0	0	520	0	0	0	0
80	89	0	725	0	0	0	82	0	86	704	0	0	583	0	0	520	0	0	0	0
90	0	0	725	0	0	0	0	0	0	704	0	0	583	0	0	520	0	0	0	0
100	0	0	725	0	0	0	0	0	0	704	0	0	583	0	0	520	0	0	0	0
110	0	0	725	0	0	0	0	0	0	704	0	0	583	0	0	520	0	0	0	0

Fonte: Dados da pesquisa do autor.

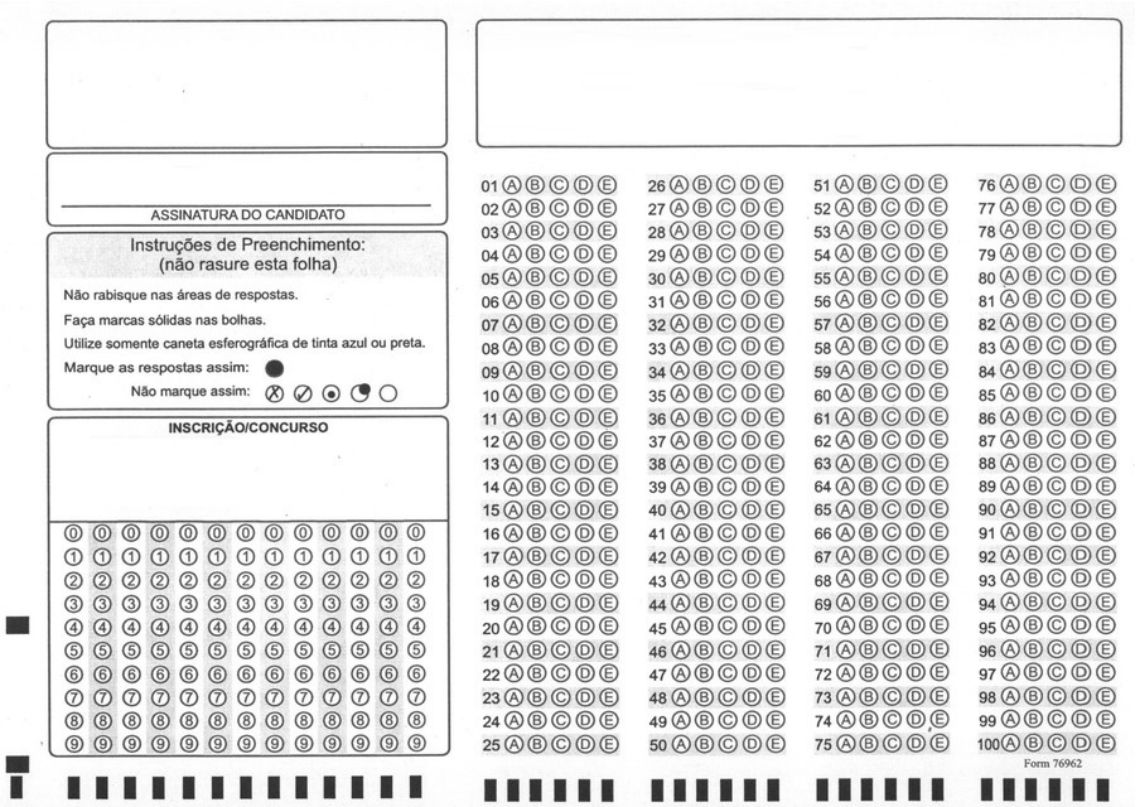
Conforme dados apresentados no quadro anterior é possível identificar que uma maior cobertura da área de marcação gera valores mais altos na contagem de *pixels* do que os apresentados na Tabela 6, como já era esperado, mas tal comportamento gera maior segurança para interpretação das marcações por parte da metodologia proposta. Por esse motivo espera-se

que os cartões tenham cobertura superior a 40%, pois coberturas muito baixas podem produzir inconsistências na identificação de marcação.

5.2 TESTES COM GABARITOS REAIS

Foram realizados testes em gabaritos utilizados no vestibular de uma instituição particular. Foram disponibilizados o total de 1.154 cartões para processamento. O processo utilizado pela instituição para correção dos cartões era realizado por um *software* que criava uma grade sobre a imagem utilizando as coordenadas das áreas de marcação das opções. O deslocamento do cartão no momento da digitalização implicaria na realização do ajuste manual da grade para cada folha de cartão digitalizado, por esse motivo o resultado final do vestibular levava em média 3 dias para serem disponibilizados. O modelo do cartão utilizado no teste real é ilustrado na Figura 61.

Figura 61 - Modelo do gabarito utilizado nos testes reais.



Fonte: Dados da pesquisa do autor.

O computador usado nos testes contava com processador de 7 núcleos com 2.40 GHz e 8 GB de memória RAM e foram apresentadas a quantidade de 40 questões para responder.

O tempo médio para correção de cada cartão foi em torno de 1,39 segundos. Foram realizadas amostragens em 5% dos cartões digitalizados comparando visualmente a valor marcado pelos candidatos e as leituras apresentadas pelo programa.

Ocorreram atrasos pontuais no processo de correção dos cartões de 3 salas pois no momento da digitalização os cartões sofreram inclinações na bandeja do *scanner* durante a alimentação automática do cartão. Devido essa inclinação na folha, o algoritmo de identificação do marcador de bordas não identificou essa região da imagem, pois, conforme mencionado na seção (3.2) o método de *template matching* é altamente sensível a escala. Como o modelo procurado considerava um *template* sem inclinação, o método não foi capaz de encontrar o modelo da imagem digitalizada com a referência inclinada. Os cartões das 3 salas foram digitalizados novamente e o processo foi concluído.

Desconsiderando o tempo de atraso devido a inclinação dos cartões, o tempo total para leitura das marcações de todos os cartões foram de 27 minutos.

O desenvolvimento de um programa utilizando a metodologia proposta possibilitou que o resultado do vestibular fosse disponibilizado em um período de 6 horas, considerando o processo de digitalização, leitura dos cartões, geração do arquivo de saída com a marcação das questões, lançamento das notas de redação e geração da lista de aprovados.

6 CONCLUSÕES

As principais técnicas utilizadas para composição da metodologia foram baseadas na TH, usada para detecção dos círculos onde são realizadas as marcações das opções. A TH foi aplicada em um cartão sem marcação para registrar as coordenadas de cada círculo detectado. Optou-se por esse procedimento para não ter que aplicar a TH em todo cartão a ser processado, reduzindo-se, assim, o tempo de processamento aplicado a cada imagem. Utilizou-se a técnica de *template matching* para identificar uma imagem impressa nos limites do cartão de resposta. Essa imagem na borda foi utilizada para identificar a borda do cartão de resposta, essa marcação serviu como referência de posicionamento para as demais áreas de processamento.

Em testes reais realizado em 1.154 gabaritos o método apresentou um percentual de acerto de 100% no total de 5% das amostras que foram conferidas. O tempo médio para processamento de cada cartão foi de 1,39 segundos para o total de 40 questões apresentadas para marcação.

O método apresenta uma deficiência quando a imagem em processamento apresenta algum grau de inclinação, seja positiva ou negativa. Isso se deve ao fato do método de *template matching* ser sensível a escala para identificação do marcador de borda. Devido à exiguidade de tempo, sugere-se, como trabalho futuro, a implementação de uma melhoria no algoritmo de detecção de imagem da borda para que o mesmo possa realizar a correção automática da inclinação da imagem, quando for o caso.

Outra maneira de melhorar o método proposto é utilizar outras formas de detecção para os campos das questões como, por exemplo, campos para marcação de questão em forma de quadrado para múltiplas opções. A técnica aplicada neste trabalho apresenta apenas a possibilidade de utilizar cartões de respostas onde os campos para marcação sejam circulares.

REFERÊNCIAS

BAYAR, G. *The use of hough transform to develop an intelligent grading system for the multiple choice exam papers.* Turquia: Karaelmas Science and Engineering Journal, 2016. p. 100-104.

BOYLE, W. S.; SMITH, G. E. *Charge coupled semiconductor devices*. New York: *The Bell System Technical Journal*, 1970, v. *49, n. 2*, p. 587-593.

CHAI, D. *Automated marking of printed multiple choice answer sheets*. In: IEEE INTERNATIONAL CONFERENCE ON TEACHING, ASSESSMENT, AND LEARNING FOR ENGINEERING, 2016, Bangkok. Proceedings... Bangkok: IEEE, 2016. p. 145-149.

DENG, H.; WANG, F.; LIANG, B. *A low-cost OMR solution for educational applications*. *IEEE INTERNATIONAL SYMPOSIUM ON PARALLEL AND DISTRIBUTED PROCESSING WITH APPLICATIONS*, 2008, Australia. Proceedings... Australia: IEEE, 2008. p. 967-970.

DUDA, R.; HART, P. *Use of the Hough transform to detect lines and curves in pictures. New York: Graphics and Image processing, 1972. p. 11-15.*

ERPEN, L. R. Reconhecimento de padrões em imagens por descritores de forma. Porto Alegre: Universidade Federal do Rio Grande do Sul, 2004. 109 p.

FARIA, D. Análise e processamento de Imagem. Porto: Universidade do Porto, 2010. 44p.

GONÇALVES, I. S. Desenvolvimento de um sistema reativo para desvio de obstáculo. Feira de Santana: Universidade Estadual de Feira de Santana, 2011. 78 p.

GONZALEZ, R. C.; WOODS, R. E. Processamento digital de imagens. 3. ed. São Paulo: Pearson, 2010. 644 p.

HAYKIN, S. Redes neurais: princípios e prática. 2 ed. Porto Alegre: Bookman, 2001. 908 p.

HOUGH; P.V.C. Method and means for recognizing complex patterns. U.S Patent 3 069 654, Dec, 1962.

ITU-R. Recommendation ITU-R BT.601-7. Genebra: ITU, 2011. <Disponível em http://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.601-7-201103-I%21%21PDF-E.pdf>. Acesso em: 18 de dez. de 2017.

KULKARNI, A. D. *Artificial neural networks for image understanding.* New York: Van Nostrand Reinhold, 1994. 210 p.

KUMAR, A.; AGRAWAL, A. *Template matching in remote sensing & image processing.* In: *INTERNATIONAL CONFERENCE ON SYSTEM MODELING & ADVANCEMENT IN RESEARCH TRENDS*, 4., 2015, Moradabad. Proceedings... Moradabad: IEEE, 2015. p. 109-114.

LIMA, K. G. Estruturas APS resistentes à radiação para aplicações espaciais. Rio de Janeiro: Universidade Federal do Rio de Janeiro, 2006. 105 p.

MACEDO, M. M. Uso da transformada de *hough* na vetorização de moldes e outras aplicações. Niterói: Universidade Federal Fluminense, 2005. 109 p.

MAHALAKSHMI, T.; MUTHAIAH, R.; SWAMINATHAN, P. *An overview of template matching technique in image processing*. Research Journal of Applied Sciences, Engineering and Technology, India, p. 5469-5473, 2012.

MARQUES, F. O.; VIEIRA, N. H. Processamento digital de imagens, Rio de Janeiro: Brasport, 1999. 331 p.

NGUYEN, T. D.; MANH, Q. H.; MINH, P. B.; THANH, L. N.; HOANG, T. M. *Efficient and reliable camera based multiple-choice test grading system*. In: INTERNATIONAL CONFERENCE ON ADVANCED TECHNOLOGIES FOR COMMUNICATIONS, 2011, Vietnã. Proceedings... Vietnã: IEEE, 2011. p. 268-271.

PADMAVATHI, K.; THANGADURAI, K. *Implementation of RGB and grayscale images in plant leaves disease detection - comparative study*. *Indian Journal of Science and Technology*, India, v. 9, n. 6, p. 1-6, 2016.

PEREIRA, M. V. L. Reconhecimento de caracteres estampados em tarugos de aço. Belo Horizonte: Universidade Federal de Minas Gerais, 2006. 95 p.

PREWITT, J. M. S. Object Enhancement and Extraction Picture Processing and Psychopictorics. New York: Academic Press, 1970. p. 75–149

ROBERTS, L. G. Machine Perception of Three-Dimensional Solids. Cambridge: Optical and Electro-Optical Information Processing, 1965. 82 p.

RUSS, J. C. *The image processing handbook.* 3 ed. Boca Raton: CRC Press, 1999. 885 p.

SOBEL, I. E. *Camera Models and Machine Perception*. Palo Alto: Stanford University. 1970. p. 277-284.

SCURI, A. E. Fundamentos da imagem digital. Rio de Janeiro: Tecgraf, 2002. 95 p.

TOU, J. T.; GONZALEZ, R. C. *Pattern recognition principles.* Boston: Addison-Wesley, 1974. 377 p.

YOUNG, T. Teoria do espaço de cores RGB, 1973. Disponível em: <http://www.ufrgs.br/engcart/PDASR/formcor.html#231>. Acesso em 12 de nov. de 2017.