

Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

**VISÃO COMPUTACIONAL PARA DEFICIENTES
VISUAIS: UMA APLICAÇÃO IOS**

DOUGLAS DE SOUZA FERREIRA

Americana, SP

2014

Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

VISÃO COMPUTACIONAL PARA DEFICIENTES VISUAIS: UMA APLICAÇÃO IOS

DOUGLAS DE SOUZA FERREIRA
douglas.ferreira10@fatec.sp.gov.br

Trabalho Monográfico, desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Fatec-Americana

Área: Visão Computacional

Ferreira, Douglas de Souza

F441v

Visão computacional para deficientes visuais: uma aplicação IOS. / Douglas de Souza Ferreira. – Americana: 2014.

42f.

Monografia (Graduação de Tecnologia em Análise e Desenvolvimento de Sistemas). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza.

Orientador: Prof. Raul Paiva de Oliveira

1. Informática – medicina I. Oliveira, Raul Paiva de II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.

CDU:681.3:615

Douglas de Souza Ferreira

VISÃO COMPUTACIONAL PARA DEFICIENTES VISUAIS: UMA APLICAÇÃO IOS

Trabalho Monográfico, desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Fatec-Americana

Área: Visão Computacional

Americana, 24 de Junho de 2014.

Banca Examinadora:

Raul Paiva de Oliveira
Graduado
Faculdade de Tecnologia de Americana

Luciene Maria Garbuio Castello Branco
Mestre
Faculdade de Tecnologia de Americana

Wladimir da Costa
Mestre
Faculdade de Tecnologia de Americana

AGRADECIMENTOS

Agradeço a todos que me apoiaram desde sempre com o desenvolvimento e pesquisas referente a área de visão computacional para deficientes visuais.

Agradeço aos colegas de classe e companheiros de trabalho que me ajudaram na resolução de dúvidas e nos experimentos necessários.

E aos professores e orientador que me guiaram para a conclusão deste projeto.

“Uma imagem vale mais que mil palavras”
CONFÚCIO, filósofo chinês.

RESUMO

Tendo em vista o grande esforço da comunidade científica do poder público e privado, voltado ao desenvolvimento de tecnologias assistivas, a pesquisa tem como intuito de levantar informações sobre a utilização de ferramentas da visão computacional e o desenvolvimento de um aplicativo que tem o propósito de auxiliar deficientes visuais no momento da compra detectando notas do Real com dispositivos móveis. São fornecidas técnicas de uso da biblioteca OpenCV, as diferenças de algoritmos de processamento de imagens e reconhecimento de objetos. Os resultados oferecidos demonstram a viabilidade do projeto e a necessidade de trabalhos para aperfeiçoamento da técnica.

Palavras Chave: Acessibilidade; visão computacional; notas monetárias; iOS.

ABSTRACT

Considering the great effort of the scientific community of public and private power, focused on the development of assistive technologies, the project has the aim of providing information on the use of the computer vision tools and the development of an application that aims to help disabled visual people while buying detecting Brazilian currency with mobile devices. Using technics with OpenCV library, the differences of algorithms for image processing and object recognition. The results demonstrate the viability of the project and the need to work on perfecting the technique.

Keywords: Accessibility; computer vision; currency; iOS.

LISTA DE FIGURAS

Figura 1 - Faixada do Bloco A da FATEC Americana	16
Figura 2 - Matriz numérica “vista” por um computador analisando o retrovisor esquerdo do veículo.	18
Figura 3 - Imagem original (esquerda) e imagem após a aplicação do ruído do tipo “sal e pimenta” (direita).....	19
Figura 4 - Imagem com ruído após aplicação do filtro de média com matriz 3x3 (A), 5x5 (B) e 7x7 (C).....	20
Figura 5 - Imagem com ruído após aplicação do filtro Gaussiano com matriz 3x3 (A), 5x5 (B) e 7x7 (C).....	20
Figura 6 - Imagem com ruído após aplicação do filtro de mediana com matriz 3x3 (A), 5x5 (B) e 7x7 (C).....	21
Figura 7 - Dilatação da imagem F pelo elemento estruturante H (matriz 3x3), que resulta na imagem G, com a dilatação evidenciada pela tonalidade de cinza claro .	22
Figura 8 - Dilatação da imagem utilizando elemento estruturante (matriz 3x3), para restauração de caracteres digitalizados.....	23
Figura 9 - Exemplo de erosão. A configuração do elemento estruturante (H) altera o resultado final do processamento	23
Figura 10 - Operação de Abertura (erosão seguida de uma dilatação).....	24
Figura 11 - Operação de Fechamento (dilatação seguida de uma erosão).....	24
Figura 12 - Exemplo de segmentação usando o operador de Canny (à esquerda a imagem original e à direita a imagem segmentada)	26
Figura 13 - Modelo da arquitetura básica da biblioteca OpenCV	28
Figura 14 - Simulação no reconhecimento de objetos	30
Figura 15 - Exemplo da segmentação dos pontos envolta do ponto central p	32
Figura 16 - Comparativos velocidade e número de pontos-chave detectados dos algoritmos SIFT, SURF e ORB.....	34
Figura 17 - Ícone da aplicação na página inicial do dispositivo	35
Figura 18 - Diagrama do fluxo da aplicação de detecção de notas	37
Figura 19 - Demonstração da aplicação de detecção testada em um iPhone 4S detectando uma nota de 2 Reais.....	39

LISTA DE TABELAS

Tabela 1 - Resultado da avaliação das notas pela aplicação.....	37
---	----

LISTA DE SIGLAS E ABREVIATURAS

BRIEF - *Binary Robust Independent Elementary Features*

CV - *Computer Vision*

DLL - *Dynamic Linked Library*

FAST - *Features from Accelerated Segment Test*

HMM - *Hidden Markov Model*

IPL - *Image Processing Library*

MLL - *Machine Learning Library*

OpenCV - *Open Source Computer Vision*

ORB - *Oriented FAST and Rotated BRIEF*

RNA - *Redes Neurais Artificiais*

SUMÁRIO

1. INTRODUÇÃO	13
2. VISÃO COMPUTACIONAL	16
2.1. Processamento de Imagens.....	17
2.1.1. Filtros	19
2.1.2. Filtro de Média.....	19
2.1.3. Filtro Gaussiano	20
2.1.4. Filtro de Mediana.....	21
2.2. Operadores Morfológicos	21
2.2.1. Dilatação	22
2.2.2. Erosão	23
2.2.3. Abertura.....	24
2.2.4. Fechamento.....	24
2.3. Segmentação.....	25
2.3.1. Segmentação por Descontinuidade.....	25
2.3.2. Segmentação por Similaridade	27
2.4. OpenCV	27
2.4.1. Arquitetura da biblioteca OpenCV	28
3. RECONHECIMENTO DE OBJETOS.....	30
3.1. Algoritmos de reconhecimento de objetos.....	31
4. CASO DE TESTE	35
4.1. Metodologia.....	36
4.2. Experimento.....	37
4.3. Resultados	38
5. CONSIDERAÇÕES FINAIS.....	40
REFERÊNCIAS.....	41
APÊNDICE A – DIAGRAMA DE SEQUÊNCIA.....	43

1. INTRODUÇÃO

De acordo com Bertanha *et al.* (2013), com passar do tempo, a mobilização das entidades públicas e privadas no desenvolvimento e embarcação da tecnologia computacional em cenários que tendem a possibilitar o acesso universal à tecnologia, vem crescendo e adaptando-se cada vez mais à população que o utiliza.

Segundo Stephanidis (2001), a abordagem de um Design Universal (*Universal Design*) ou Interface para todos (*Interface for All*), visa minimizar a necessidade de retrabalho posteriores ao lançamento dos produtos, buscando designs que possam adaptar-se ao usuário desde o primeiro momento. Em casos onde essa abordagem não se aplica, é causada uma defasagem entre os usuários que se já entendem o produtos e os outros que precisam esperar novos produto com soluções específicas. De acordo com dados do Instituto Brasileiro de Geografia e Estatística (2010), no Brasil 45 milhões de pessoa, cerca de 23,9%, possuem algum tipo de deficiência - visual, auditiva, motora e mental ou intelectual.

Nas abordagens atuais, as soluções para adaptação das interfaces que se distanciam do emprego de hardwares específicos (como luvas, capacetes de visualização, controles espaciais e dispositivos dedicados) se aproximam do conceito de interação baseada em comandos ou mediada pelo reconhecimento natural das ações do usuário apresentam-se como as alternativas mais promissoras.

Neste trabalho, analisa-se a implementação de algoritmos de pontos de interesse (*feature points*) no desenvolvimento de sistemas de realidade aumentada que possam auxiliar pessoas com deficiência em seu dia a dia.

A visão computacional, em nossa proposta, é empregada, também, como alternativa à necessidade de retrabalho posterior de dispositivos já aplicados.

Para tanto o estudo se **justificou** devido a cerca de 29 milhões de pessoas no país se declararem com alguma deficiência visual e destes, cerca de 6 milhões possuem grandes dificuldades. No contexto atual de sistemas que auxiliam essa população se encontra escasso e restrito à uma pequena parcela, portanto este

estudo se justifica devido a demanda crescente de acessibilidade por parte dos deficientes em suas tarefas cotidianas, como: estudo, entretenimento, trabalho, entre outros.

Já o **Problema** foi: Devido à complexidade de um deficiente se relacionar na sociedade e a falta de conhecimento por parte dos desenvolvedores de sistemas que utilizam plataformas com base em visão computacional que auxiliam a acessibilidade de pessoas com deficiência visuais na sociedade.

Como **Pergunta** que se buscou responder: é possível desenvolver uma aplicação que utilize as técnicas de visão computacional com a finalidade de auxiliar os portadores de deficiência visual?

As **Hipóteses** foram: a) Por ser uma tecnologia de algoritmos de nível de complexidade alta, torna-se difícil o desenvolvimento de aplicações que auxiliam um portador de deficiência. b) Apesar de ser um assunto abrangente que afeta 1/3 da população, poucos possuem interesse em desenvolver um sistema do gênero. c) Os sistemas com o nível de tecnologia atual não apresenta usabilidade suficiente para auxiliar um deficiente.

O **objetivo geral** consistiu em: estudar e demonstrar meios onde sistemas de visão computacional focados em realidade aumentada pode beneficiar portadores de deficiência visual em seu cotidiano, objetivando fazer um levantamento dos meios disponíveis atualmente e das características necessárias para desenvolver um caso de teste.

Os **objetivos específicos** foram: a) Analisar métodos onde a visão computacional em forma de realidade aumentada pode ser aplicada para deficientes, visando b) Analisar possíveis locais onde os sistemas possam ser implementados, buscando c) Desenvolver um caso de teste de uma aplicação do sistema de realidade aumentada, objetivando d) Analisar se a proposta pode ser aplicável e se os resultados obtidos são satisfatórios.

Como **metodologia** para o desenvolvimento deste trabalho, foi utilizada quanto ao ponto de vista da natureza, a pesquisa básica, que de acordo com Gil (2005, p.20), a pesquisa possuem um caráter pragmático, na qual objetiva gerar conhecimentos novos úteis para o avanço da ciência sem aplicação prática prevista, visando o conceito e utilização de aplicações de visão computacional com foco em realidade aumentada.

Para que os objetivos da pesquisa sejam alcançados, serão utilizados dois modos de pesquisa: a exploratória e descritiva.

A pesquisa exploratória, para Silva e Menezes (2005, p.21), visa proporcionar maior familiaridade com o problema com vistas a torna-lo explícito, visto que será a explicação do sistema. Já a pesquisa descritiva é citada pelos autores da seguinte forma:

[...] visa descrever as características de determinada população ou fenômeno ou o estabelecimento de relações entre variáveis. Envolve o uso de técnicas padronizadas de coleta de dados: questionário e observação sistemática. [...].

O trabalho foi estruturado em 4 capítulos, sendo que o primeiro será a conceituação onde apresentará a origem e definições de visão computacional; o segundo trará de modo específico a explicação dos algoritmos de descrição e extração de pontos de interesse; o terceiro apresentará um caso de teste com base nas informações conseguidas a partir dos estudos realizados nos capítulos anteriores, o quarto capítulo se reserva às Considerações Finais.

2. VISÃO COMPUTACIONAL

A visão é um dos mais fascinantes sentidos do ser humano, que possibilita a percepção de informações visuais do ambiente que o cerca. Através da visão podemos identificar texturas, cores, brilho, profundidade entre outros elementos. O cérebro humano tem a capacidade de identificar imagens e associa-las facilmente com objetos previamente vistos, como por exemplo, ao ver uma foto de amigos é possível reconhecer cada pessoa na foto, se são conhecidos ou não, a idade aparente de cada um e até mesmo o estado emocional com base na expressão facial do momento. Tais técnicas do sistema visual humano não são facilmente aplicadas quando se refere a um computador, uma simples variação de cor torna uma imagem diferente de outra para um computador, conforme demonstrado na Figura 1.

Figura 1 - Faixada do Bloco A da FATEC Americana



Fonte: Adaptado de <http://www.pontojovemnet.com.br/noticia/estilo-de-vida/1060/fatec-abre-inscriaees-para-o-vestibular-2014>

Ambas imagens da Figura 1 são referentes a mesma faixa da do bloco A da FATEC Americana, tiradas em momentos diferentes. É simples um ser humano notar que se trata do mesmo local, porém, para um computador, estes dois locais são totalmente diferentes. Enquanto conseguimos fazer o reconhecimento de objetos, um computador precisa processar essas imagens e converte-las para dados numéricos para que seja possível assim serem interpretadas.

Segundo Solem (2012), visão computacional é extração automatizada de informações de imagens. Destas informações podem derivar muitos dados, como: modelos 3D, posição da câmera e reconhecimento de objetos. Muitas vezes a visão computacional tenta imitar a visão humana, seja usando dados ou estatísticas

aproximadas, mas algumas vezes geometria é a chave para resolver problemas, portanto definindo na prática visão computacional como uma mistura de programação, modelagem e matemática.

Em outra definição, Basu e Li (1993, p. 14) afirmam que “O objetivo geral da visão computacional é derivar informações de uma cena pela análise das imagens daquela cena por um computador”.

Basu e Li (1993) ainda afirmam que a visão computacional é dividida em várias áreas de aplicação, que incluem: o processamento de documentos (reconhecimento de objetos, por exemplo), sensores remotos, análise médica de imagens, reconhecimento de alvos e o orientação de robôs. Estas aplicações têm provido maiores incentivos para a continuidade das pesquisas do campo de visão computacional.

Contudo, o avanço do campo de visão computacional almeja ter um computador que seja capaz de analisar e interpretar imagens ao mesmo nível de uma criança de dois anos, capaz de identificar e contar os animais de uma foto, por exemplo. Neste caso, a visão computacional se torna limitada, pois muitas vezes se torna necessário fazer a análise de objetos em uma cena com dados insuficientes, e assim tendo que recorrer a técnicas com base em estatística e física para complemento das informações para que seja possível fazer a correlação de objetos entre possíveis soluções. (SZELISK, 2010)

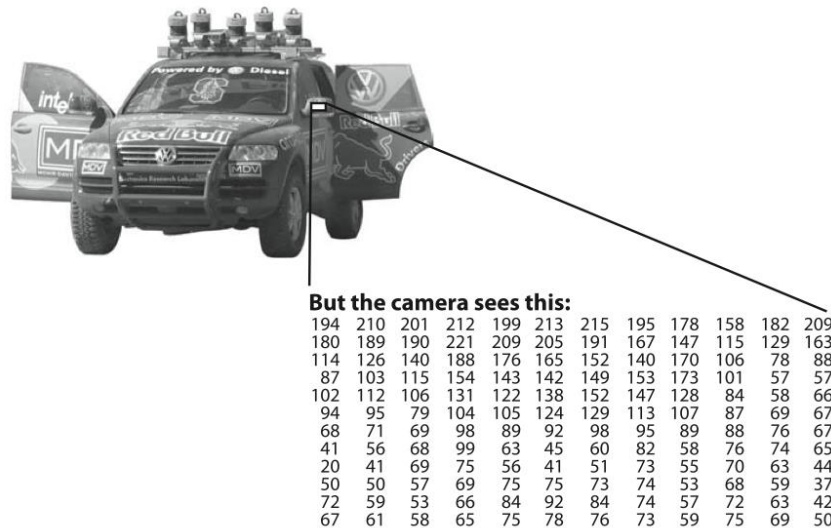
2.1. Processamento de Imagens

Segundo Gonzalez e Woods (2002), o interesse em processamento de imagens surge da aplicação em duas áreas principais: melhoria da informação para interpretação humana e também o processamento de imagens para reconhecimento automático por máquinas. Portanto os objetivos do processamento digital de imagens são melhorar os aspectos visuais de uma imagem para análise humana e fornecer subsídios para sua interpretação, extração de dados de uma imagem e conseqüentemente seu processamento.

Para que um computador possa entender uma imagem, é necessário digitalizar a imagem em forma de matriz de numérica, onde cada combinação representa um pixel da imagem. A partir dessas matrizes numéricas o computador consegue processar e extrair as informações desejadas da imagem. A Figura 2 representa a

“visão” do computador no retrovisor esquerdo do automóvel, ou seja, para um computador, o retrovisor é apenas uma matriz de números.

Figura 2 - Matriz numérica “vista” por um computador analisando o retrovisor esquerdo do veículo.



Fonte: Adaptado de BRADISK; KAEHLER, 2008, p. 3.

O processamento de imagens pode ser dividido em três níveis: baixo, médio e alto. O baixo-nível são aplicadas operações primitivas, como a redução de ruído ou efeitos de contrastes em uma imagem. Os processos de nível-médio são do tipo de segmentação ou reconhecimento; e os processos de alto-nível estão relacionados com as funções de aprendizado, associadas com o sentido da visão humana, como por exemplo, criar um cenário que “faça sentido” dado um conjunto de objetos reconhecidos. (GONZALEZ; WOODS, 2002)

É comum encontrar diversos tipos de ruídos em imagens que podem derivar de várias fontes, como: luminosidade do ambiente, condições climáticas, resolução da câmera fotográfica ou degradação da imagem, estas são interferências que podem causar impacto na no reconhecimento e interpretação de imagens. A seguir serão apresentadas algumas operações que tratam diferentes casos de ruído em imagens.

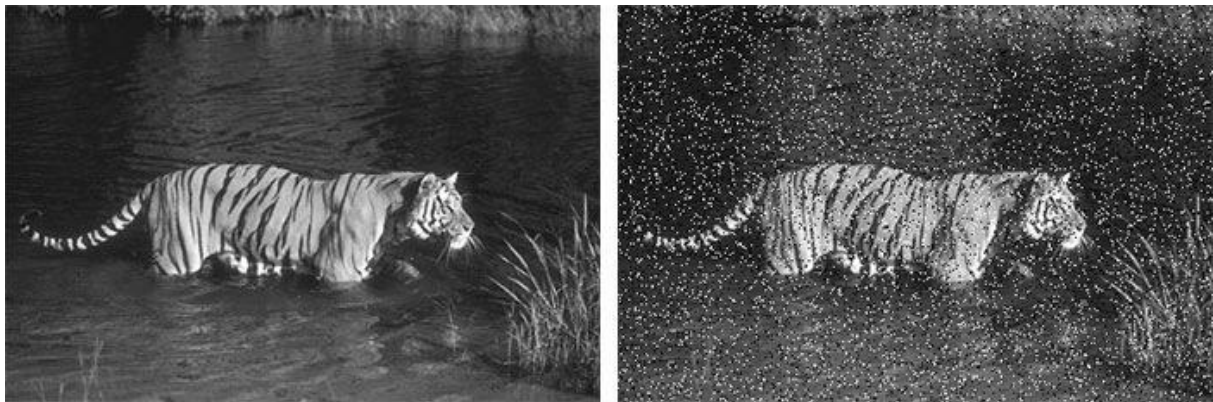
2.1.1. Filtros

As ferramentas básicas para remover ruídos de imagens são os filtros, que podem atuar diretamente na imagem (domínio de espacial) ou transformando-a para frequência por meio da transformada de Fourier¹ (domínio de frequência). A filtragem trata-se de transformações da imagem que são aplicadas pixel a pixel e dependem da escala de cinza de um determinado pixel e do valor da escala de cinza dos pixels ao redor.

São vários os algoritmos de filtros de imagem que tem como objetivo remover ruídos de imagens. A seguir serão demonstrado alguns dos filtros mais utilizados para tratamento de ruídos em imagens.

A imagem da direita na Figura 3 apresenta ruídos do tipo “sal e pimenta”, e esta será utilizada para demonstração dos efeitos de cada filtro.

Figura 3 - Imagem original (esquerda) e imagem após a aplicação do ruído do tipo “sal e pimenta” (direita).



Fonte: Autoria própria

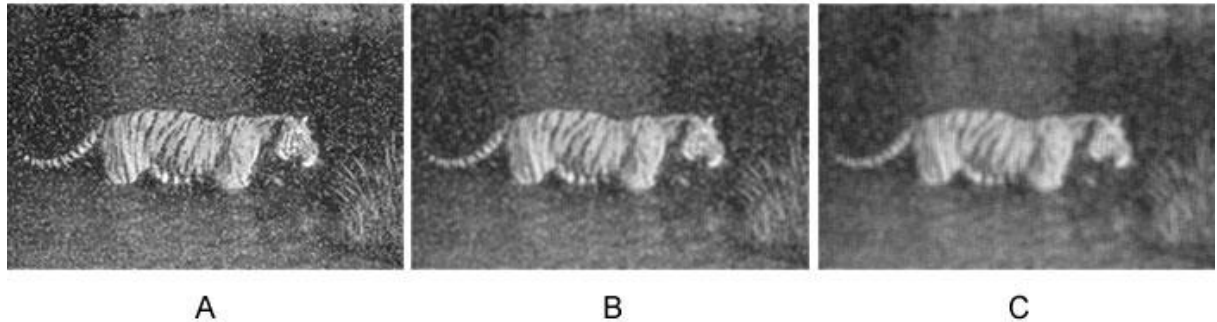
2.1.2. Filtro de Média

A aplicação do filtro de média é comumente utilizada por ser uma técnica simples e relativamente rápida, pois não necessita de processamentos complexos,

¹ Técnica que permite ter uma visão da imagem a ser analisada no domínio da frequência (frequência de determinados pixels na imagem), facilitando sobremaneira esta análise e o seu processamento.

podendo ser resolvida com somas e divisões. A quantidade de pixels que fazem parte da média são determinados pelo tamanho da matriz aplicada.

Figura 4 - Imagem com ruído após aplicação do filtro de média com matriz 3x3 (A), 5x5 (B) e 7x7 (C).

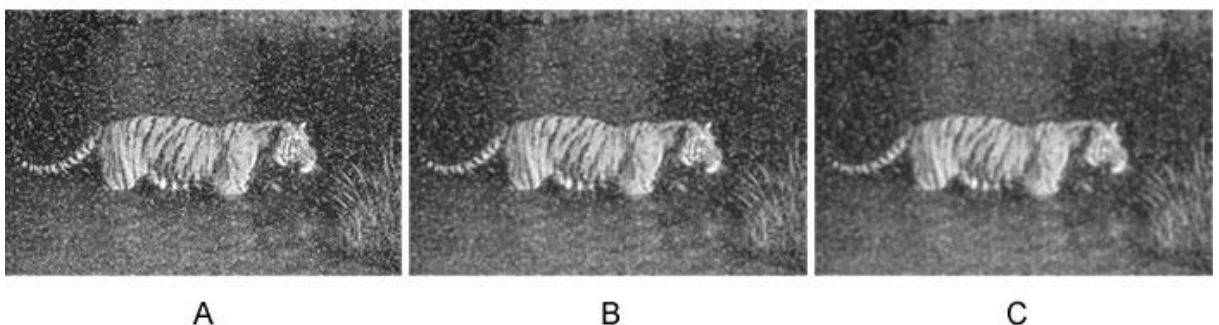


Fonte: Autoria própria

Como pode-se observar na Figura 4, o resultado após a aplicação do filtro de média é a suavização da imagem original, de modo que sejam diminuída a interferência de ruídos. Porém, como um aspecto negativo da aplicação deste filtro é a diminuição da nitidez da imagem, pela diminuição da percepção das bordas que as compõem.

2.1.3. Filtro Gaussiano

Figura 5 - Imagem com ruído após aplicação do filtro Gaussiano com matriz 3x3 (A), 5x5 (B) e 7x7 (C).



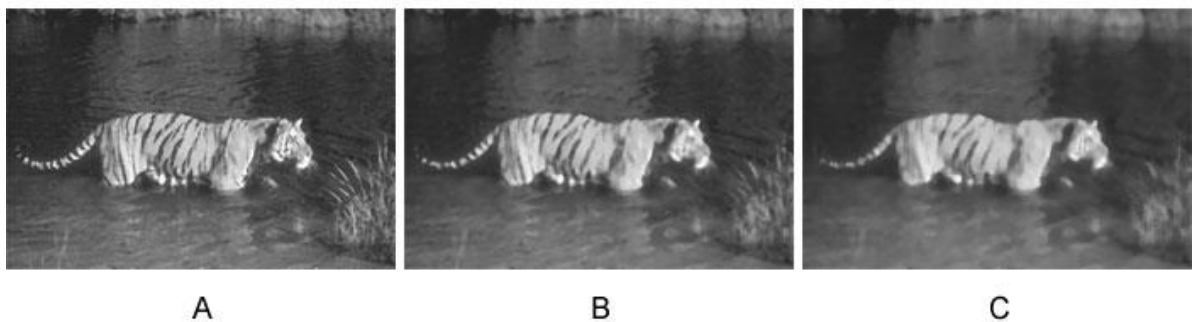
Fonte: Autoria própria

O filtro Gaussiano (Figura 5) em comparação com o filtro de média, mostra-se menos ofensivo na degradação da nitidez da imagem com ruídos, como pode-se observar nas imagens C da Figura 4 e 5, ainda que usando um filtro com uma matriz grande (7x7) em ambas as imagens, a que sofreu filtragem por Gaussiano se apresenta mais conservada.

2.1.4. Filtro de Mediana

O filtro de mediana é um dos filtros mais utilizados para suavização de imagens, em modo geral o nível de cinza de cada pixel é substituído pelo cálculo da mediana dos níveis de cinza dos pixels da vizinhança, proporcionando assim um polimento da imagem.

Figura 6 - Imagem com ruído após aplicação do filtro de mediana com matriz 3x3 (A), 5x5 (B) e 7x7 (C).



Fonte: Autoria própria

Dentre os filtros apresentados para remoção de ruídos do tipo “sal e pimenta”, o filtro de Mediana demonstra-se ser mais efetivo para a o polimento da imagem sem causar grande degradação na nitidez da imagem.

2.2. Operadores Morfológicos

A morfologia é o estudo das imagens sob o ponto de vista das formas que as compõe. Cada imagem é composta por elementos, denominados pixels, e para destacar aspectos específicos das formas de uma imagem, certas operações matemáticas de composto de pixels podem ser utilizadas, possibilitando que sejam contadas ou reconhecidas, dependendo da aplicação desejada (ORTH, 2000).

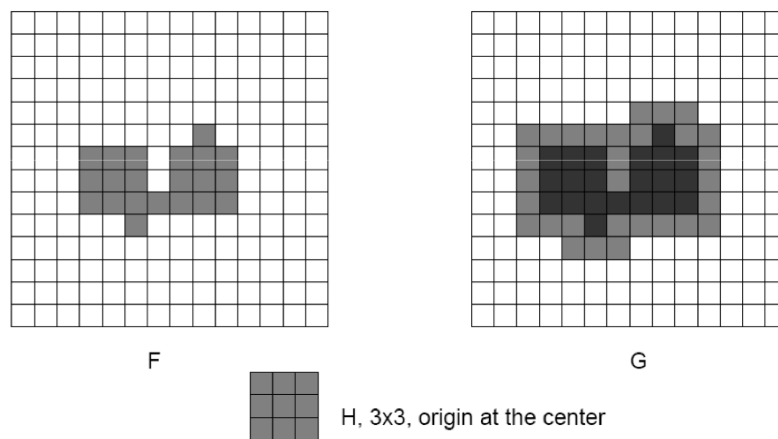
As operações morfológicas, sejam elas em imagens em cores ou tons de cinza, não implicam na retirada ou inclusão de um pixel (substituição de seu valor por preto ou branco), mas sim na modificação parcial de seu valor.

Todos operadores utilizados processamento morfológico de uma imagem são construídos pela junção de duas operações básicas: a Dilatação e a Erosão. E a partir destas podem ser criadas facilmente outras duas operações: Abertura e Fechamento.

2.2.1. Dilatação

A Operação de Dilatação consiste em aumentar a área de um elemento em relação à imagem original. Exemplificado na Figura 7, dada uma imagem (F) e um elemento estruturante de matriz 3x3 (H), a dilatação resulta na imagem (G) cuja área demonstra-se maior.

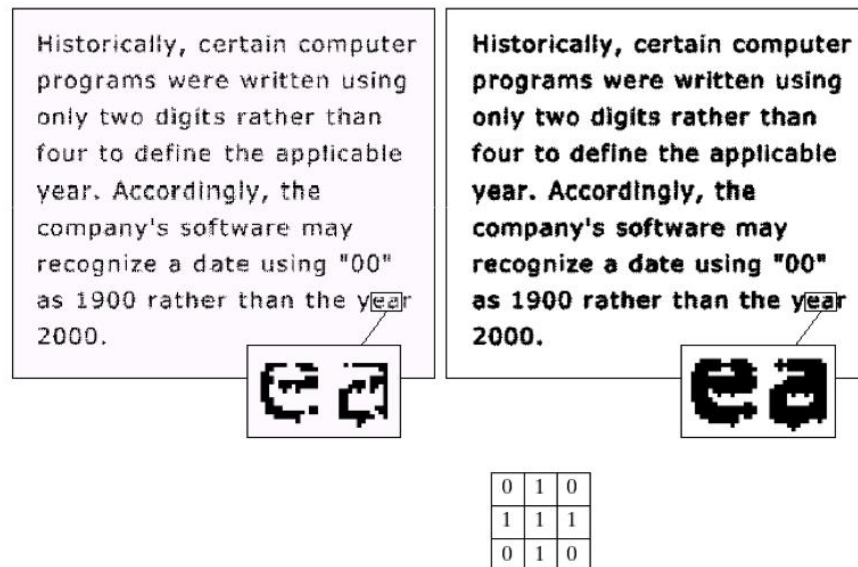
Figura 7 - Dilatação da imagem F pelo elemento estruturante H (matriz 3x3), que resulta na imagem G, com a dilatação evidenciada pela tonalidade de cinza claro



Fonte: Adaptado de REN, 2010, p. 11.

A técnica de dilatação é utilizada para reconexão de elementos ou preenchimento de espaço, como é o caso da Figura 8, onde o processo de digitalização de imagens acaba "corroendo" pontos do texto.

Figura 8 - Dilatação da imagem utilizando elemento estruturante (matriz 3x3), para restauração de caracteres digitalizados

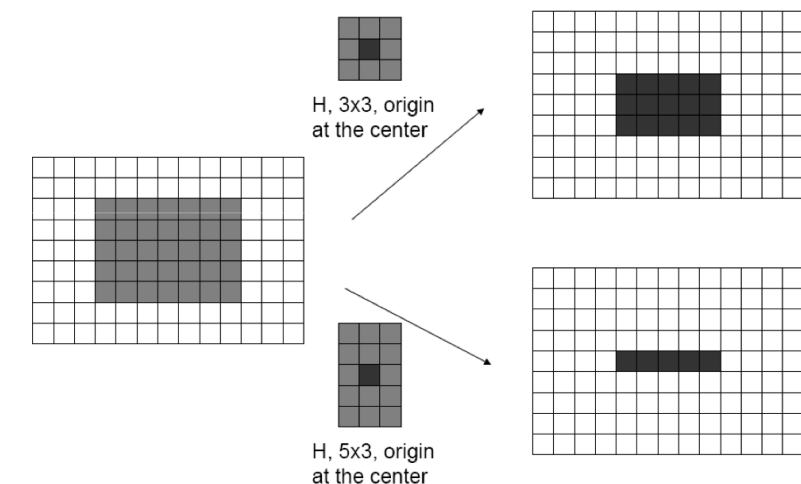


Fonte: Adaptado de REN, 2010, p. 12.

2.2.2. Erosão

A Operação de Erosão consiste em diminuir a área de um elemento em relação à imagem original. Exemplificado na Figura 9, dada uma imagem e um elemento estruturante de matriz 3x3 e 5x3 (H), a erosão resulta na imagem cuja área demonstrase menor.

Figura 9 - Exemplo de erosão. A configuração do elemento estruturante (H) altera o resultado final do processamento

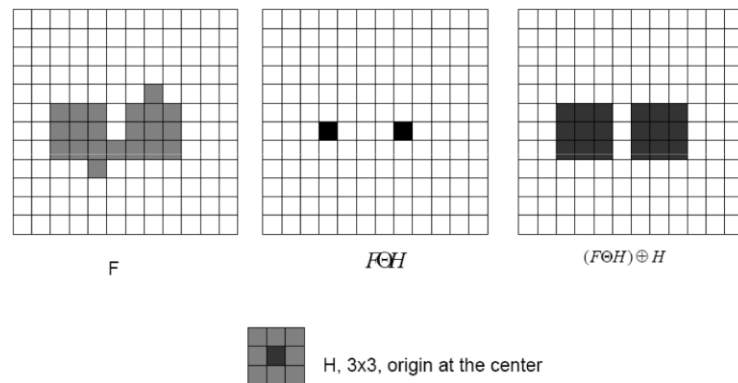


Fonte: Adaptado de REN, 2010, p. 15.

2.2.3. Abertura

Abertura é composta por uma erosão seguida por uma dilatação. Apesar de serem operações opostas, uma não invalida a outra. Tendo com o efeito uma suavização das bordas, eliminação de ruído ou pequenos defeitos, ou o aumento (abertura) de espaços. (BAUERMANN,2008).

Figura 10 - Operação de Abertura (erosão seguida de uma dilatação)

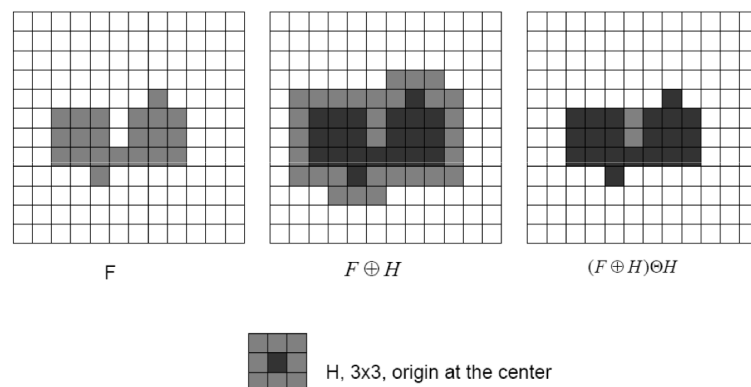


Fonte: Adaptado de REN, 2010, p. 22.

2.2.4. Fechamento

Fechamento utiliza dos mesmo elementos que a abertura, porém, com uma alteração entre as operações: uma dilatação seguida por uma erosão. Causando um efeito de preenchimento de texturas ou pode unir pixel que estavam próximos.

Figura 11 - Operação de Fechamento (dilatação seguida de uma erosão)



Fonte: Adaptado de REN, 2010, p. 24.

2.3. Segmentação

Na área de processamento de imagens o termo segmentação consiste no processo de particionar uma imagem em regiões ou vários objetos, guiado por características, como: cor, proximidade, pontos, linhas, bordas, dentre outros. O nível de detalhes no processo de segmentação varia de acordo com a tarefa a ser executada e da resolução da imagem (MARENGONI; STRINGHINI, 2009), assim esperando que o resultado final de um processo de segmentação seja um número finito de regiões homogêneas, que individualizam as diferentes regiões contidas em uma imagem. (ACHARYA; RAY, 2005).

São diversas as técnicas de segmentação que podem ser aplicadas em uma imagem, portanto os principais algoritmos de segmentação usam como base a descontinuidade ou similaridade dos níveis de cinza da imagem.

2.3.1. Segmentação por Descontinuidade

Nos algoritmos com base em descontinuidade são buscados pontos, linhas ou limites (bordas) que possam definir um objeto. Essas formas sobressaem-se em uma imagem, seja por uma diferença nos tons de cinza da região inseridas (caso dos pontos e linhas) ou por demonstrarem alterações bruscas nos tons de cinza entre regiões (caso de bordas e linha). Portanto, esta abordagem de segmentação envolve a utilização de filtros para analisar a imagem e identificar suas estruturas. Tal identificação pode ocorrer tanto no domínio espacial quanto no domínio de frequência, conforme explicado no subcapítulo de Filtros. (GONZALEZ; WOODS, 2002).

Detecção de Bordas

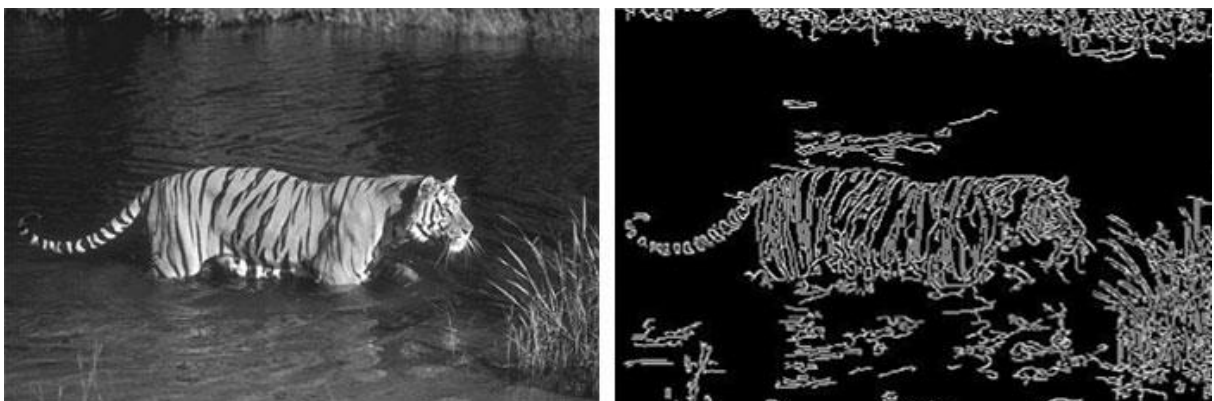
Dentre as técnicas de segmentação por descontinuidade, a detecção de bordas é a abordagem mais utilizada (GONZALEZ; WOODS, 2002). Em uma imagem a borda é definida como o limite entre duas regiões ou objetos com diferentes propriedades, normalmente o nível de intensidade dos *pixels*, portanto, os detectores de bordas são definidos para encontrar essa variação de *pixels*.

Na abordagem de detecção de bordas são vários os algoritmos seguidos ou precedidos por uma limiarização (*threshold*), processo de definição de um nível de

cinza dentro da imagem que se tonará a fronteira quais o *pixels* serão “acesos” ou “apagados”. O resultado é uma imagem binária, definida pelo histograma da imagem. O histograma é o indicador percentual da quantidade de pixels com determinado tom de cinza ou intensidade de cor na imagem. De acordo com o agrupamento dos pixels na imagem e com o objetivo do tratamento da imagem (separar objetos do fundo, identificar bordas) o ponto de corte é escolhido (CONCI; AZEVEDO; LETA, 2008).

Segundo Marengoni e Stringhini (2009), a técnica de detecção de borda desenvolvida por John F. Canny em 1986, que leva seu sobrenome (Canny), embora seja a mais complexa, é também a de melhor desempenho, pois utiliza duas limiarizações diferentes, uma para contornos forte e uma para contornos fracos. O detector de bordas Canny possui três objetivos básicos: baixa taxa de erros (todas as bordas devem ser encontradas), os pontos da borda devem ser bem localizados (as bordas localizadas devem ser próximas das bordas reais) e uma resposta única para os pontos de uma borda (o operador deve retornar apenas um ponto para cada ponto sobre a borda). A Figura 12 demonstra um exemplo de segmentação de imagem utilizando o operador de Canny.

Figura 12 - Exemplo de segmentação usando o operador de Canny (à esquerda a imagem original e à direita a imagem segmentada)



Fonte: Autoria própria.

2.3.2. Segmentação por Similaridade

Algoritmos de segmentação por similaridade baseiam-se em duas abordagens principais: limiarização e crescimento de regiões. O processo de limiarização aplicados a segmentação por descontinuidade descrita anteriormente é o tipo mais simples de limiarização. Este tipo de limiarização utilizando o histograma é também chamado de limiarização global e só apresenta resultados ótimos, em geral, quando é possível controlar a iluminação da cena.

No caso da segunda abordagem por crescimento de regiões, o princípio da técnica é agrupar os pixels ou sub-regiões em regiões maiores por meio da análise de *pixels* vizinhos em busca de similaridades como intensidade, cor, textura, etc. Portanto o crescimento ocorre até que todos os *pixels* da imagem sejam processados e, assim, formadas diferentes regiões. (GONZALEZ; WOODS, 2002)

2.4. OpenCV

A Biblioteca OpenCV foi desenvolvida inicialmente pela Intel em 2000, com o objetivo de tornar mais simples e eficaz o desenvolvimento de ferramentas que utilizam visão computacional para programadores das áreas de robótica e interação humano-computador. Com uma comunidade de cerca de 47 mil desenvolvedores e um número estimado de 7 milhões de downloads, a biblioteca possui mais de 2500 algoritmos para diversos tipos de aplicações como: reconhecimento de rostos, detectar objetos, classificar ações humanas em vídeos, rastrear movimentos de câmeras, extrair modelos 3D de objetos, remover olhos vermelhos de uma foto tirada com flash, reconhecer cenários e estabelecer marcações com realidade aumentada, etc.

Escrita nativamente em C++, a OpenCV possui interfaces em C++, C, Python, Java e MATLAB com suporte em Windows, Linux, Android, Mac OS e iOS, facilitando ainda mais a portabilidade de aplicações em diversas plataformas. (OPENCV, 2014)

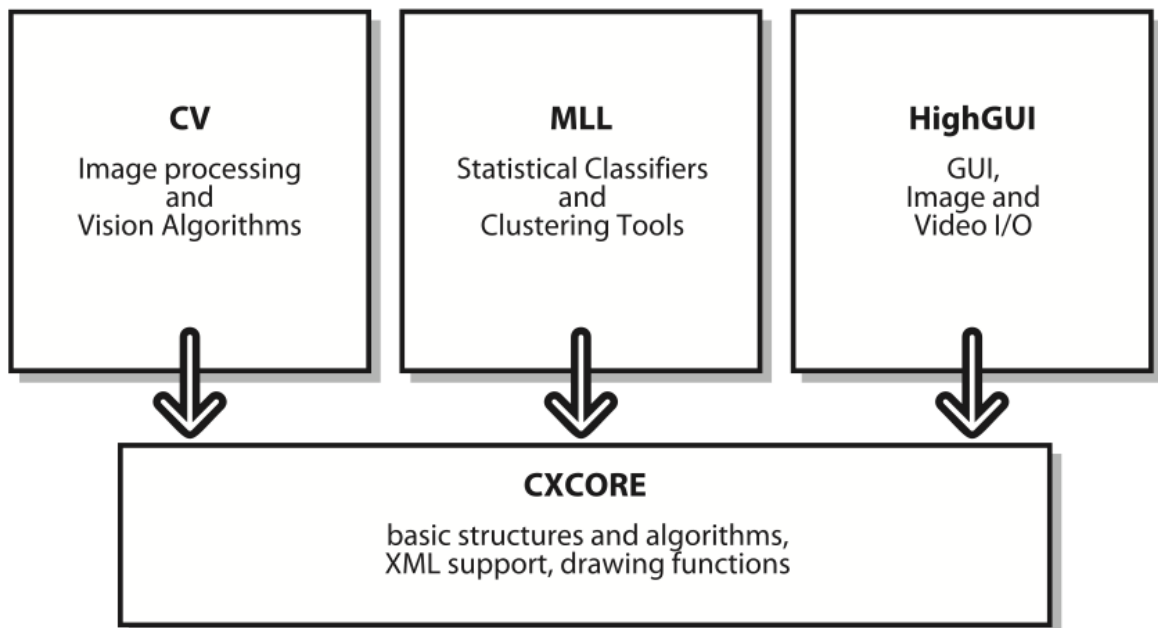
A biblioteca está disponível com o código fonte e os executáveis (binários) otimizados para os processadores Intel. Um programa OpenCV, ao ser executado, invoca automaticamente uma DLL que detecta o tipo de processador e carrega, por sua vez, a DLL otimizada para este. Juntamente com o pacote OpenCV é oferecida a

biblioteca IPL, da qual a OpenCV depende parcialmente, além de documentação e um conjunto de códigos exemplos. (MARENGONI; STRINGHINI, 2009)

2.4.1. Arquitetura da biblioteca OpenCV

A biblioteca OpenCV é dividida em cinco componentes principais, quatro destes podem ser observados na Figura 13:

Figura 13 - Modelo da arquitetura básica da biblioteca OpenCV



Fonte: Adaptado de BRADISK; KAEHLER, 2008. p. 13

CV

Contém o básico do processamento de imagem, algoritmos de visão de computacional de alto nível, detecção de movimento, padrões de reconhecimento e calibragem da câmera.

MLL

É a camada de aprendizado de máquina que inclui vários classificadores estatísticos;

HighGUI

Contém as rotinas de entrada e saída e funções para armazenamento e carregamento de imagens e vídeos;

CXCORE

Possuí estruturas de dados, funções, persistência de objetos, gerenciamento de memória e erros, e também funções para desenho, texto e matemática básica.

A Figura 13 não contém a camada **CvAux**, que é trata algoritmos de reconhecimento de faces, HMM e algoritmos experimentais de segmentação em *background* e *foreground*, pois a CvAux ainda não foi bem documentada e está em fase de testes.

3. RECONHECIMENTO DE OBJETOS

O ato de reconhecer algo, implica no significado de conhecer de novo, processo onde existe um conhecimento prévio e algum tipo de armazenamento do conhecimento já obtido sobre o objeto a ser reconhecido. Neste ponto os sistemas de visão computacional possuem uma intersecção com a área de inteligência artificial. Para que seja possível reconhecer um objeto, a aplicação de visão computacional precisa possuir uma base de conhecimento dos objetos a serem reconhecidos, esta base pode ser implementada diretamente no código ou essa base pode ser “aprendida” a partir de um conjunto de amostra dos objetos a serem reconhecidos utilizando técnicas de aprendizado de máquina (MARENGONI; STRINGHINI, 2009), conforme exemplificado na Figura 14:

Figura 14 - Simulação no reconhecimento de objetos



Fonte: Adaptado do Filme O Exterminador do Futuro 2: O Julgamento Final, Dierção: James Cameron, 1991.

A utilização da análise de contornos e borda pode auxiliar em operações onde haja variações nas formas de um objeto, mesmo que pequenas. Porém, frequentemente objetos do mundo real possuem diversas cores, formas e tamanhos que dificultam com a comparação em uma pequeno conjunto de valores. Devido a isso, o reconhecimento de objetos tem sido uma das tarefas mais desafiantes da visão computacional (SZELISKI, 2010).

Durante o processo de reconhecimento de objetos, podem ser utilizados algoritmos que extraiam características das imagens baseando-se na ideia de ao

invés de olhar na imagem como um todo. Ela é segmentada e pontos especiais (pontos de interesse) são utilizados para classificar de melhor forma cada objeto com menor custo computacional.

Detectar pontos chave em uma imagem é atualmente o núcleo de muitas tecnologias com foco em Visão Computacional, como reconhecimento de objetos, reconstrução 3D, recuperação de imagens e localização de câmeras. Da forma que a aplicação dessas tecnologias devem lidar com mais dados ou para rodar em dispositivos móveis com recursos computacionais limitados, há uma necessidade crescente em utilizar descritores locais menores para agilizar os cálculos, fazer rápidas combinações e usar memória a suficiente (CALONDER et al, 2010).

3.1. Algoritmos de reconhecimento de objetos

São muitos os algoritmos que realizam a extração de informações de uma imagem, conhecidas como descritores, para que seja possível fazer uma análise profunda e consistente de cada setor da imagem. Segundo Lowe e Bay, em geral, esse processo é composto por três etapas:

- Extração e seleção dos pontos de interesse que caracterizam uma imagem. Nesta etapa também são obtidas a orientação do objeto, posição e também a escala;
- Relação entre pontos-chave da vizinhança de modo a montar um vetor de características;
- Analisar e definir a correspondência entre esses pontos, de modo a determinar a igualdade de duas imagens (LOWE, 1999; BAY *et al.*, 2008).

Para obtenção de bons resultados durante a etapa de extração e correspondência entre os pontos de interesse de uma imagem a característica que é considerada como a mais importante para um algoritmo de reconhecimento de objetos é a sua consistência na repetição, ou seja, um descritor de pontos de sempre selecionar o mesmo conjunto de pontos de um objeto ou cena, mesmo que sob diferentes condições de visualização (BAY *et al.*, 2008).

Segundo Lowe (1999), um descritor de pontos de interesse ideal deve possuir invariância a rotação e escala, que são associadas a um ponto no cálculo do vetor de

características, e também ser robusto em relação a erros como distorções devido a diferentes pontos de vista da cena e distorções causada pela iluminação.

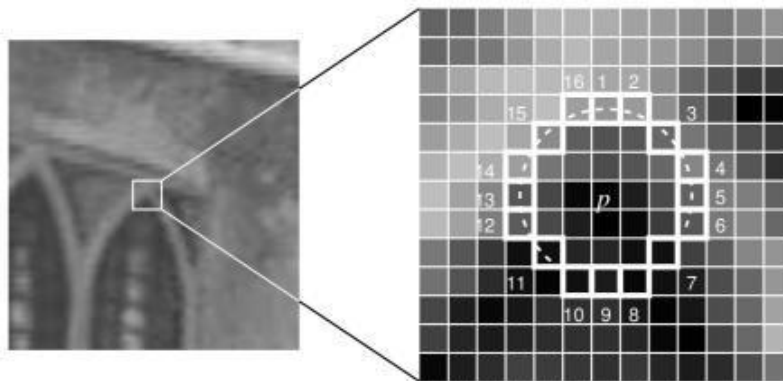
A seguir serão apresentadas uma breve introdução dos principais algoritmos utilizados no reconhecimento de objetos:

- Detector: FAST;
- Descritor: BRIEF;
- Detectores e Descritores: SIFT, SURF e ORB;
- Combinadores: BruteForce.

FAST

O FAST é um detector de pontos de interesse que utilizado algoritmo de Bresenham² para analisar um círculo de 16 *pixels* de diâmetro em volta do pixel candidato à ponto-chave. Se pelo menos nove destes *pixels* dos forem mais brilhantes, ou a totalidade de pixels mais escuros que o pixel central somado, ou subtraído de uma certa limiarização, então o centro daquele círculo é considerado um ponto-chave (ROSTEN; DRUMMOND, 2006).

Figura 15 - Exemplo da segmentação dos pontos envolta do ponto central p .



Fonte: Adaptado de ROSTEN; DRUMMOND, 2006, p. 434.

² Também conhecido como algoritmo do ponto médio, foi criado por Jack Bresenham em 1962, que determina como será desenhada uma reta entre dois pontos em um plano.

BRIEF

O BRIEF é um robusto descritor binário que utiliza como base os testes de diferença de intensidade dos pontos vizinhos de um ponto-chave e faz comparações de forma rápida e com alta taxa de reconhecimento. Como o BRIEF não calcula uma orientação do ponto-chave encontrado, ele se torna invariante à rotações da imagem analisada (CALONDER; 2010).

SIFT

O algoritmo SIFT proposto por Lowe (1999), arquiva pontos-chave de uma pirâmide de escala de modo à ordenar e gerar características que sejam invariantes à escala. Além disso, os descritores extraídos do SIFT são invariantes à rotação e possuem alta tolerância à ruídos ou distorções causadas por mudanças da iluminação.

SURF

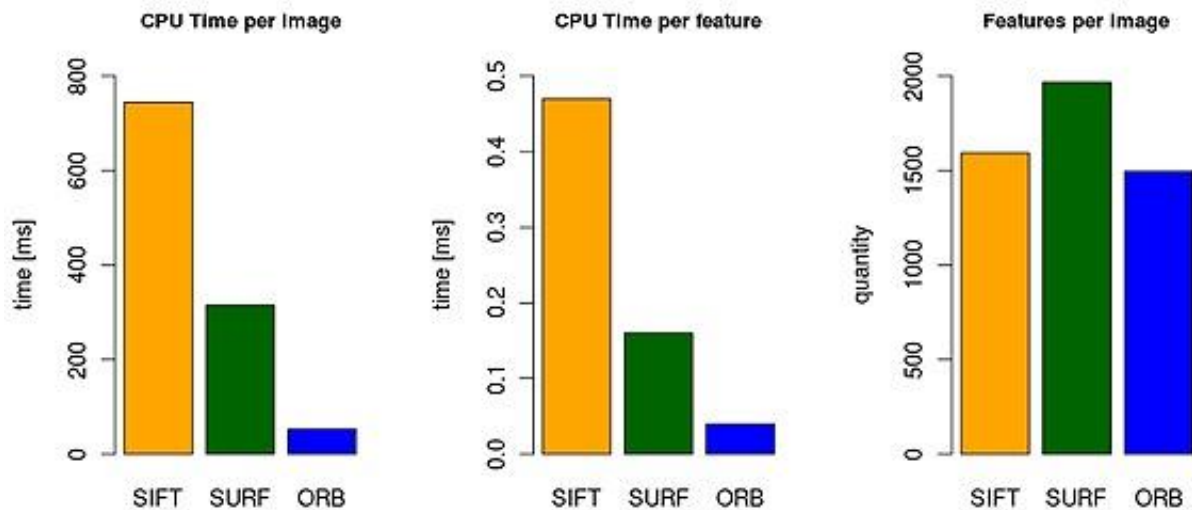
Inspirado parcialmente no SIFT, o algoritmo de detecção de objetos proposto por Bay *et al.* (2006) que utiliza como base em seu detector a matriz Hessiana para performance em computação e precisão. O descritor do SURF é extraído levando em consideração a orientação computadas usando retornos Haar-wavelet para aumentar a resistência à distorções geométricas, tornando assim o algoritmo invariante a rotações e escala.

ORB

O mais recente dos algoritmos apresentados, proposto por Rublee *et al.* (2011), é ao contrário dos algoritmos SIFT e SURF, distribuído gratuitamente a partir da versão 2.3.1 da biblioteca OpenCV. O ORB faz uso da combinação do detector invariante à orientação, conhecido como oFAST (uma versão derivada do FAST), e o descritor dos pontos-chave BRIEF melhorado com alterações que o torna invariante a rotação e ruídos. Absorvendo o melhor de ambos os algoritmos, o ORB apresenta-se um descritor binário com alto desempenho com baixo consumo computacional,

tornando-se a alternativa mais eficiente para dispositivos de baixo nível de processamento, como *Smartphones* (BERTANHA *et al.* 2013).

Figura 16 - Comparativos velocidade e número de pontos-chave detectados dos algoritmos SIFT, SURF e ORB.



Fonte: Adaptado de ADORF, 2011, p. 78.

O algoritmo ORB em comparação com SIFT e SURF, mostra-se ser até 14 vezes mais ágil que o SIFT no tempo de do processado e até 5 vezes mais rápido que o algoritmo SURF.

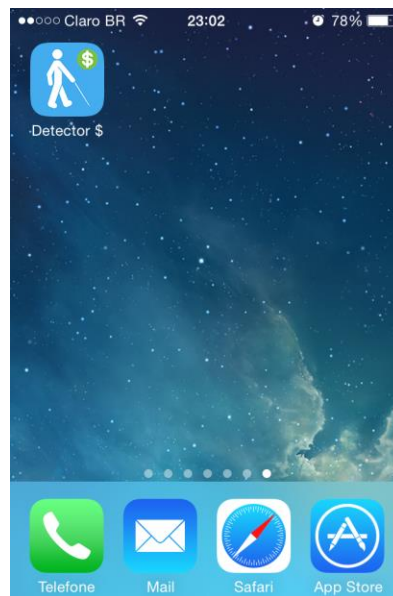
BRUTEFORCE

O algoritmo *BruteForce*, também conhecido Força Bruta, recebe esse nome devido à busca exaustiva na qual ele combina e compara cada ponto de interesse entre dois vetores de descritores. Portanto, o algoritmo de força bruta tem como objetivo obter, de modo eficiente e como boa tolerância à erros, a correspondência entre pontos-chave encontrados em descritores (BERTANHA *et al.* 2013).

4. CASO DE TESTE

Este capítulo tem como objetivo apresentar uma técnica para reconhecimento de notas utilizando aparelhos que executem o sistema operacional iOS (iPod, iPhone e iPad), combinado com a biblioteca OpenCV e algoritmos gratuitos, gerando assim um baixo custo de processamento computacional, assim podendo ser utilizado por deficientes visuais para conferência do dinheiro recebido no momento de uma compra, por exemplo. Como os dispositivos móveis atualmente possuem um forte apelo visual, torna-se desafiante a elaboração de uma interface voltada para deficientes visuais, assim a audição e o toque foram os meios escolhidos para transmitir as ações da aplicação criada.

Figura 17 - Ícone da aplicação na página inicial do dispositivo



Fonte: Autoria própria.

A plataforma iOS foi escolhida por possuir suporte à biblioteca OpenCV e também ao suporte à linguagem C++ que é utilizada nativamente no OpenCV.

A expectativa é que a aplicação seja capaz de identificar de forma precisa todas as notas da moeda brasileira, o Real, tanto sua frente como o verso da cédula. Cabe destacar que o trabalho proposto não tem a intenção de validar a veracidade da nota apresentada, mas sim interpretar o valor da cédula.

4.1. Metodologia

Descrito por Ampudia (2011), o deficiente visual é aquele que tem o comprometimento total ou parcial de 40 a 60% da visão. Não são consideradas deficientes visuais pessoas que possuem doenças como astigmatismo, miopia ou hipermetropia, que podem ser corrigidas com o uso de lentes ou em cirurgias.

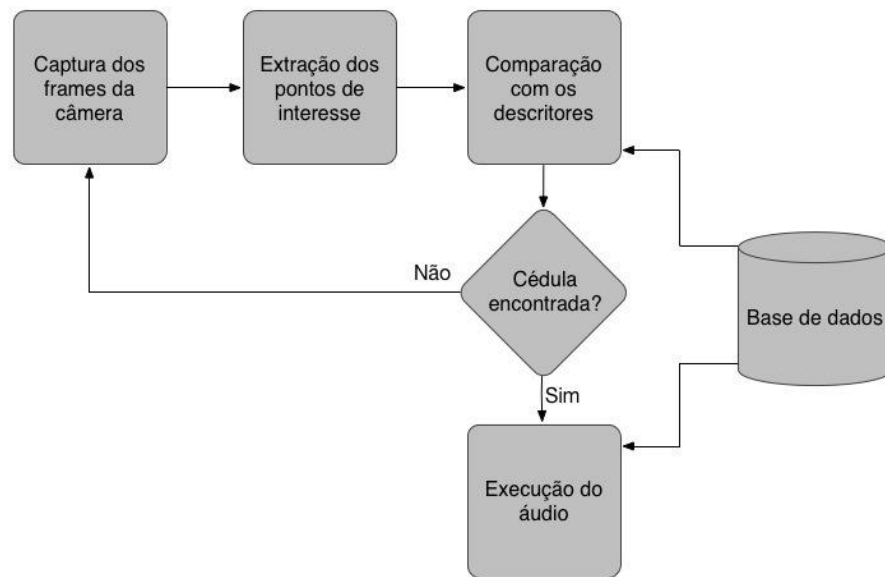
Segundo critérios estabelecidos pela Organização Mundial da Saúde os níveis de acuidade visual podem ser classificados em:

- **Baixa visão:** compensada com o uso de lentes de aumento, lupas, telescópios, com o auxílio de bengalas e de treinamentos de orientação.
- **Próximo à cegueira:** quando a pessoa ainda é capaz de distinguir luz e sombra, mas já emprega o sistema braile para ler e escrever, utiliza recursos de voz para acessar programas de computador, locomove-se com a bengala e precisa de treinamentos de orientação e de mobilidade.
- **Cegueira:** quando não existe qualquer percepção de luz. O sistema braile, a bengala e os treinamentos de orientação e de mobilidade, nesse caso, são fundamentais (AMPUDIA, 2011).

Com isso, desenvolvimento do algoritmo de reconhecimento de cédulas para deficientes visuais foi baseado na análise de pontos de interesse com a utilização do detector e descritor do algoritmo ORB, por ser uma alternativa gratuita e com ótimo desempenho no processamento de imagens em dispositivos móveis, e com o algoritmo força bruta para a comparação e escolha dos melhores pontos.

A Figura 18 ilustra o fluxo do sistema proposto. Quando ativado pelo toque do usuário, a aplicação inicia o processo de captura de *frames* (quadros) pela câmera do dispositivo. Esses *frames* são analisado pelo algoritmo ORB e assim os pontos de interesse são extraídos e comparados com os descritores de cada cédula encontrados na base de dados da aplicação, caso o algoritmo de força bruta faça uma comparação satisfatória entre os pontos de interesse, então um áudio é executado, caso contrário o fluxo será reiniciado e processado novamente.

Figura 18 - Diagrama do fluxo da aplicação de detecção de notas



Fonte: Autoria própria.

4.2. Experimento

Para colocar em prática a proposta do sistema, foram analisadas 10 notas de cada valor de real: 2, 5, 10, 20, 50 e 100 em ambos modelos, tanto o antigo quanto o novo, a Figura 19 exemplifica as notas reconhecidas e utilizadas na aplicação. Os teste foram executados utilizando um *smartphone* iPhone 4S com processador de 800 MHz e câmera de 8 *megapixels*. Para cada nota detectada, a aplicação emitia um som de confirmação referente à aquela nota e mostrava um alerta na tela.

Durante o experimento foram convidados voluntários na FATEC de Americana (professores e alunos graduandos do curso de análise de sistemas de ambos os sexos) e no Instituto de Pesquisas Eldorado (funcionários), aleatoriamente, para exibirem para o sistema alguma nota que portavam em suas carteiras. O experimento durou até que o número mínimo de 10 notas de cada valor fosse avaliado. A cada exibição, o sistema emitia um aviso sonoro sobre o valor da nota apresentada. Os teste foram realizados em diferentes períodos e com variações de ambiente e iluminação.

4.3. Resultados

Os testes com o algoritmo executando no iPhone mostram que o verso das notas foram mais facilmente detectadas pela aplicação. Na parte frontal da nota, o tempo para detecção era maior em relação ao verso devido ao fato em que o verso da nota possui mais desenhos e uma variação mais destacada das bordas, tornando assim a identificação mais fácil para o algoritmo.

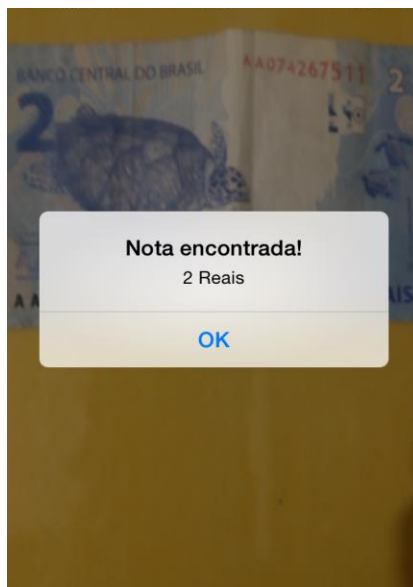
Tabela 1 - Resultado da avaliação das notas pela aplicação

Notas	Testes	Acertos	Erros	% acerto
2	10	10	0	100,0%
5	10	10	0	100,0%
10	10	10	0	100,0%
20	10	9	1	90,0%
50	10	9	1	90,0%
100	10	10	0	100,0%
Média de acerto:				96,7%

É possível notar na Tabela 1 que a média de acerto chega perto a perfeição dos 100% desejáveis. No caso do erro da nota de 50, acredita-se que devido a faixa holográfica no canto esquerdo da parte frontal, que varia de acordo com a posição e iluminação na nota, torna mais difícil a detecção da nota. Já no erro da nota de 20, o algoritmo de detecção confundiu a nota com uma de 2 reais, possivelmente por causa do número 2 e as poucas diferença de bordas na parte frontal da nota.

A Figura 20 mostra a interface simples da aplicação que ao detectar uma nota reproduz um som correspondente a aquela nota e exibe um alerta informando a nota encontrada.

Figura 19 - Demonstração da aplicação de detecção testada em um iPhone 4S detectando uma ³⁹
nota de 2 Reais



Fonte: Autoria própria.

5. CONSIDERAÇÕES FINAIS

Com base no caso de teste, percebeu-se que o desenvolvimento de um sistema que utiliza visão computacional com foco no uso para deficientes visuais possui pode não se tornar complexo, sem a necessidade de aplicações robustas que implementam algoritmos de treinamento de RNA deixando assim o usuário final livre da necessidade de retreinamento do algoritmo e ajustes característicos de sistemas baseados em RNA.

Aplicável à plataformas de fácil acesso como *smartphones* e *tablets*, assim uma interface simples de fácil acesso, com interação com o deficiente através de sons, torna mais eficaz a imersão das pessoas com deficiência visuais no mundo da visão computacional.

Durante o desenvolvimento e experimentos da aplicação de reconhecimento de notas, surgiram novas ideias e necessidades que os voluntário identificaram como possível melhorias, do tipo: validação de cores, moedas e notas de outra espécie (dólar). Vale ressaltar que o campo de visão computacional é vasto e muitas outras metodologias e aplicações podem ser elaboradas a modo de estudar novas áreas de desenvolvimento para auxiliar o cotidiano das pessoas.

A partir desta experiência e dos resultados alcançados, despertam-se oportunidades de trabalhos futuros decorrente da proposta de sistemas de visão computacional utilizando algoritmos de extração de pontos de interesse e tratar com mais ênfase a exequibilidade do sistema juntamente com o usuário final de maior benefício (deficientes visuais), podendo ser esta, tarefa decisiva para a continuidade do projeto.

REFERÊNCIAS

ACHARYA, Tinku; RAY, Ajoy K. **Image processing: principles and applications**. John Wiley & Sons. Hoboken, 2005.

ADORF, Julius. **Object detection and segmentation in cluttered scenes through perception and manipulation**. 2011. 78p. Tese (Bacharel em Ciência da Computação) – Universidade Técnica de Munique. Munique.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Citação**: NBR-10520/ago - 2002. Rio de Janeiro: ABNT, 2002.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Referências**: NBR-10520/ago - 2002. Rio de Janeiro: ABNT, 2002.

BASU, Anup; LI, Xiaobo. **Computer vision: Systems, Theory and Applications**. College Park: World Scientific, 1993.

BAUERMANN, Gabriela. **Abertura e fechamento**. 2008. Disponível em: <<http://www.imagesurvey.com.br/2008/12/abertura-e-fechamento/>>. Acesso em: 17 mai. 2014. 12h.

BAY, Herbert *et al.* **SURF**: Speeded up robust features. In: Computer Vision–ECCV 2006. Springer Berlin Heidelberg, 2006. p. 404-417.

BERNANHA, Luiz L. *et al.* **Recognition of Brazilian Currency Notes via Feature Point Extraction**. Americana, 2013.

BRADISK, Gary; KAEHLER, Adrian. **Learning OpenCV – Computer Vision With OpenCV Library**. 1ªed. O’Reilly Media, 2008.

CALONDER, Michael *et al.* **BRIEF**: Binary robust independent elementary features. In: Computer Vision–ECCV 2010. Springer Berlin Heidelberg, 2010. p. 778-792.

CONCI, Aura; AZEVEDO, Eduardo; LETA, Fabiana. **Computação gráfica volume 2: Processamento de Imagens Teoria e Prática**. Rio de Janeiro: Campus/Elsevier, 2008.

SILVA, Edna Lúcia; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação**. UFSC. Florianópolis, 2001.

GONZALEZ, Rafael C.; WOODS, Richard E. **Digital image processing**. 2ªed. Upper Saddle River, New Jersey: Prentice-hall, 2002.

IBGE, Instituto Brasileiro de Geografia e Estatística. **Censo Demográfico 2010**. Disponível em: <http://www.ibge.gov.br/home/estatistica/populacao/censo2010/caracteristicas_religiao_deficiencia/default_caracteristicas_religiao_deficiencia.shtm>. Acesso em: 16 set. 2013. 20h.

LOWE, David G. **Object recognition from local scale-invariant features**. In: Computer vision, 1999. The proceedings of the seventh IEEE international conference on. IEEE, 1999. p. 1150-1157.

MARENGONI, Maurício; STRINGHINI, Denise. **Tutorial: Introdução à Visão Computacional usando OpenCV**. Universidade Presbiteriana Mackenzie: Revista de Informática Aplicada - RITA, Volume XVI, Número 1, 2009.

OpenCV. **About**. Disponível em: <<http://www.opencv.org/about.html>>. Acesso em: 07 mai. 2014. 10h.

ORTH, Alexandre. **Morfologia Matemática**. Disponível em: <<http://www.inf.ufsc.br/~visao/2000/morfologia/morfologia.html>>. Acesso em: 17 mai. 2014. 09h.

REN, Tsang I. **Processamento de Imagem Morfológica**. UFPE, 2010. Disponível em: <<http://www.cin.ufpe.br/~tir/ComputacaoCientifica/7.Morfologia%20Matematica.pdf>> Acesso em: 17 mai. 2014. 10h.

AMPUDIA, Ricardo. **O que é deficiência visual?**. Disponível em: <<http://revistaescola.abril.com.br/formacao/deficiencia-visual-inclusao-636416.shtml>>. Acesso em: 05 jul. 2014. 23h

ROSTEN, Edward; DRUMMOND, Tom. **Machine learning for high-speed corner detection**. In: Computer Vision—ECCV 2006. Springer Berlin Heidelberg, 2006. p. 430-443.

RUBLEE, Ethan *et al.* **ORB: an efficient alternative to SIFT or SURF**. In: Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011. p. 2564-2571.

SZELISK, Richard. **Computer Vision: Algorithms and Applications**. New York: Springer - Verlag, 2010.

SOLEM, Jam. **Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images**. 1ªed. O'Reilly Media, 2012. p. 9.

STEPHANIDIS, C. **Adaptative Techniques for Universal Access in: User Modeling and User-Adapted Interaction**. Kluwer Academic Publishers. 2001, p.159-179.

APÊNDICE A – DIAGRAMA DE SEQUÊNCIA

