Análise e Reconhecimento de Imagens Através do Algoritmo SIFT

Pedro Augusto Silveira de Almeida Veloso[1], Sandro Roberto Fernandes[2]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais – Câmpus Juiz de Fora
Rua Bernardo Mascarenhas, 1283 – Fábrica – 36.080-001 – Juiz de Fora – MG – Brasil

[2]Núcleo de Informática - Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais – Câmpus Juiz de Fora

pedrosilveira.apple@gmail.com, sandro.fernandes@ifsudestemg.edu.br

Abstract. *The main objective of this work was to develop a computational tool for the use of the SIFT and SURF algorithm, determining its possible applications. These two algorithms allow you to locate an object in one image within another image. The developed application used the Java programming language and the OpenCV library.*

Resumo. *O objetivo principal deste trabalho foi desenvolver uma ferramenta computacional para o uso do algoritmo SIFT e SURF, determinando suas possíveis aplicações. Esses dois algoritmos permitem localizar um objeto dentro de outra imagem. O aplicativo desenvolvido utilizou a linguagem de programação Java e a biblioteca OpenCV.*

1. Introdução

Pode-se observar, nas últimas décadas, grandes avanços na área de computação gráfica. Diversos trabalhos estão sendo realizados visando usar uma imagem como base de dados e extrair da mesma, características que sejam suficientes para identificá-la em outra imagem.

Para se obter esse resultado leva em conta perspectivas 3D, para que se crie um objeto de modelo comparativo. Para reconhecimento em 2D, são analisadas distâncias entre os pontos no histograma para se conseguir mapear a imagem de base dados.

Uma das formas de reconhecimento e correspondência de imagens em 3D permite representar a imagem por um conjunto de características, vetores, denominadas SIFT (Scale In-variant Feature Transform). Esses vetores correspondem a diferentes posições, rotação e levam em conta a deformação da imagem, bem como sua iluminação. O SIFT reconhece a máxima e a mínima da imagem modelo, utilizando funções gaussianas na escala de espaço. Esse conjunto de características formam um gradiente com as regiões vizinhas dos pontos demarcados. O algoritmo SURF semelhante ao SIFT também é capaz de reconhecer imagens independente da rotação. Pesquisadores afirmam que se trata de uma solução mais rápida e robusta. O SURF reconhece os pontos chaves de uma Imagem Base para localizá-los na Imagem objeto. Neste trabalho o foco principal foi o uso e estudo do SIFT.

2. Metodologia

2.1. SIFT e SURF

O algoritmo SIFT tem como objetivo extrair das imagens as principais características e assim poder classificá-las, pois esse processo é essencial na computação gráfica. Esse algoritmo é capaz de reconhecer pontos chaves da imagem, os chamados keypoints, que não variam com a rotação, escala ou iluminação. O primeiro processo do SIFT é fazer a detecção desses keypoints. Com esses pontos determinados, se forma um vetor de 128 posições para cada ponto encontrado. As informações contidas nestes vetores são os gradientes dos pontos vizinhos com seus valores normalizados, isso garante a invariância da rotação e iluminação. O processo se divide entre a detecção dos keypoints e a formação dos vetores com as características das imagens.

Na detecção dos Keypoints, é utilizada uma função Gaussiana onde uma imagem I (x,y) é definida por L (x,y,σ), no espaço de escala. Produzimos então uma função através da convolução desta função gaussiana G (x,y,σ) com a imagem I (x,y), temos (GONZÁLES):

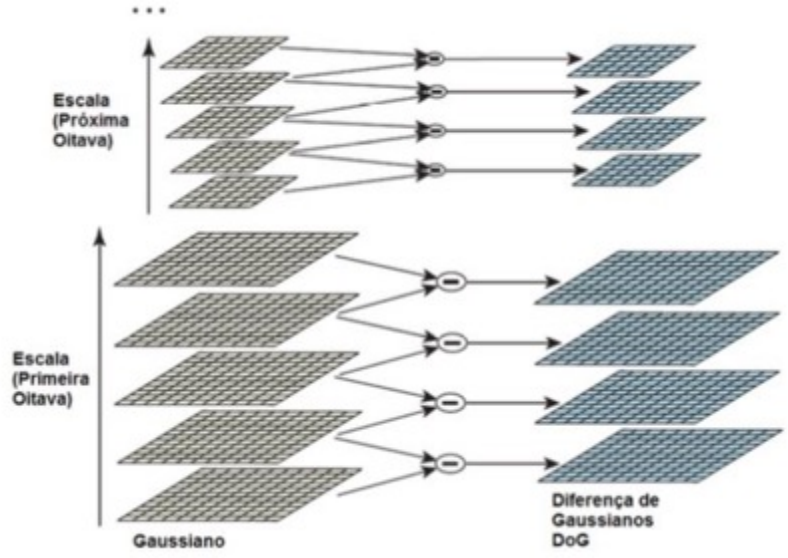
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \qquad (1)$$

onde

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

O objetivo de se usar funções Gaussiana é conseguir amostras das imagens com menos detalhes indesejados, ruídos e características muito fortemente ressaltadas. Na imagem a seguir podemos ver o processo de obtenção das diferenças Gaussianas:

Figura 1. Processo de obtenção das diferenças Gaussianas



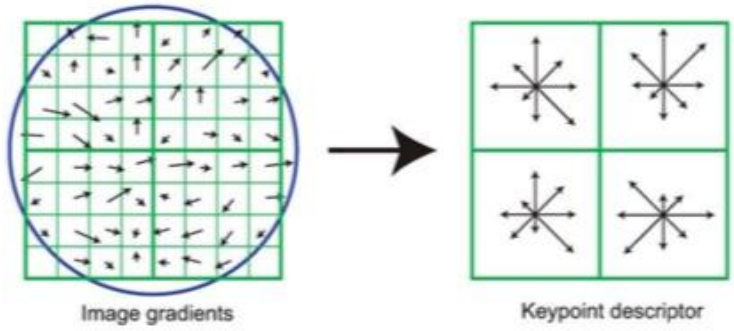
Fonte: GONZÁLES, Giancarlo Luís Gómez

Passando desse processo, desejamos encontrar e localizar os pontos chaves da imagem, que é objeto do nosso estudo. Para isso se usa uma expansão de Taylor da Diferença de Gaussiano aplicada à imagem, D (x,y,σ), que pode ser observada abaixo:

$$D(\bar{x}) = D + \frac{\partial D^T}{\partial \bar{x}}\bar{x} + \frac{1}{2}\bar{x}^T\frac{\partial^2 D}{\partial x^2}\bar{x} \ldots \qquad (3)$$

Para cada ponto chave localizado é atribuída uma orientação que será usada na construção dos descritores invariantes à rotação, obtemos essa invariância pelas características locais da imagem. Para a construção do descritor a imagem é dividida em regiões 4 x 4, em cada uma dessas regiões se extrai um histograma com 8 bins, que são as faixas de orientação (SILVA). A figura que segue ilustra o processo de forma simplificada:

Figura 2. Exemplo simplificado do construtor de imagens



Fonte: GONZÁLES, Giancarlo Luís Gómez

Depois de todos esses procedimentos é feito o match entre as imagens, esse processo consiste na extração dos pontos chaves de cada imagem e na correspondência entre eles como ilustrado na imagem abaixo:

Figura 3. Match entre a imagem de estudo e a imagem da cena

Fonte: VELOSO, Pedro Augusto Silveira de Almeida

O algoritmo SURF foi proposto por Bay et al. (2006) e se trata de uma solução em computação gráfica para extração e descrição de características de interesse em uma imagem computacional rápida e robusta. Ele foi parcialmente inspirado no algoritmo SIFT, porém, pesquisadores afirmam que este apresenta pontos mais representativos e rápidos em relação ao SIFT. Ambos compartilham da invariância rotacional na busca da imagem. Assim como o SIFT as etapas de identificação consistem na extração e descrição dos pontos de principais características. O SURF possui um poderoso descritor baseado na Transformada de Haar, assim é possível usar a imagem de forma integral sem grande custo computacional (BORTH et al.).

2.2. OpenCV

A Biblioteca OpenCV (Open Source Computer Vision Library) é uma das bibliotecas mais utilizadas mundialmente no meio acadêmico e comercial, pode ser obtida através do endereço listado na referência. Essa biblioteca é multiplataforma que pode ser usada em diversos sistemas operacionais como Linux, Windows e, como neste estudo, no macOS. Essa foi desenvolvida nas linguagens de programação C e C++. Com essa biblioteca é possível desenvolver softwares em diversas linguagens de programação, como Pyhon, Visual Basic, Ruby, dentre outras e Java, que foi o foco deste estudo (SILVA, CHAVES, AQUINO).

A biblioteca OpenCV foi desenvolvida pela Intel e possui cerca de 500 funções. Foi idealizada para tornar acessível a visão computacional tanto para usuários, quanto para programadores em áreas como interação em tempo real de homens e computadores e robótica. Tal biblioteca está disponível e otimizada para processadores Intel, por meio de códigos fonte e arquivos executáveis (binários). Quando um programa que utiliza a OpenCV é carregado, carrega também a DLL (Dynamic Linked Library) que identifica o processador e otimiza a OpenCV para suas configurações. Adicionalmente há também a biblioteca IPL (Image Processing Library) que além da documentação disponibiliza também um conjunto de exemplos de códigos (MARENGONI, STRINGHINI).

Entre as funções principais da biblioteca podemos destacar: Processamento de imagens, Análise de estrutural, Análise de movimento, rastreamento de objetos, Reconhecimento de padrões e Calibração de câmera e reconstrução 3D. Podemos observar na imagem abaixo um trecho de código utilizando a biblioteca OpenCV.

Figura 4. Exemplo de código em Java utilizando a Biblioteca OpenCV com o SIFT

```
MatOfKeyPoint objectDescriptors = new MatOfKeyPoint();
//Nessa linha mudar de SURF para SIFT caso queira usar outro Descritor
DescriptorExtractor descriptorExtractor = DescriptorExtractor.create(DescriptorExtractor.SIFT);
System.out.println("Computing descriptors...");
descriptorExtractor.compute(objectImage, objectKeyPoints, objectDescriptors);

//Criar a matrix da imagem de saída
Mat outputImage = new Mat(objectImage.rows(), objectImage.cols(), Highgui.CV_LOAD_IMAGE_COLOR);
Scalar newKeyPointColor = new Scalar(255, 0, 0);

System.out.println("Drawing key points on object image...");
Features2d.drawKeypoints(objectImage, objectKeyPoints, outputImage, newKeyPointColor, 0);
```

2.3. Software

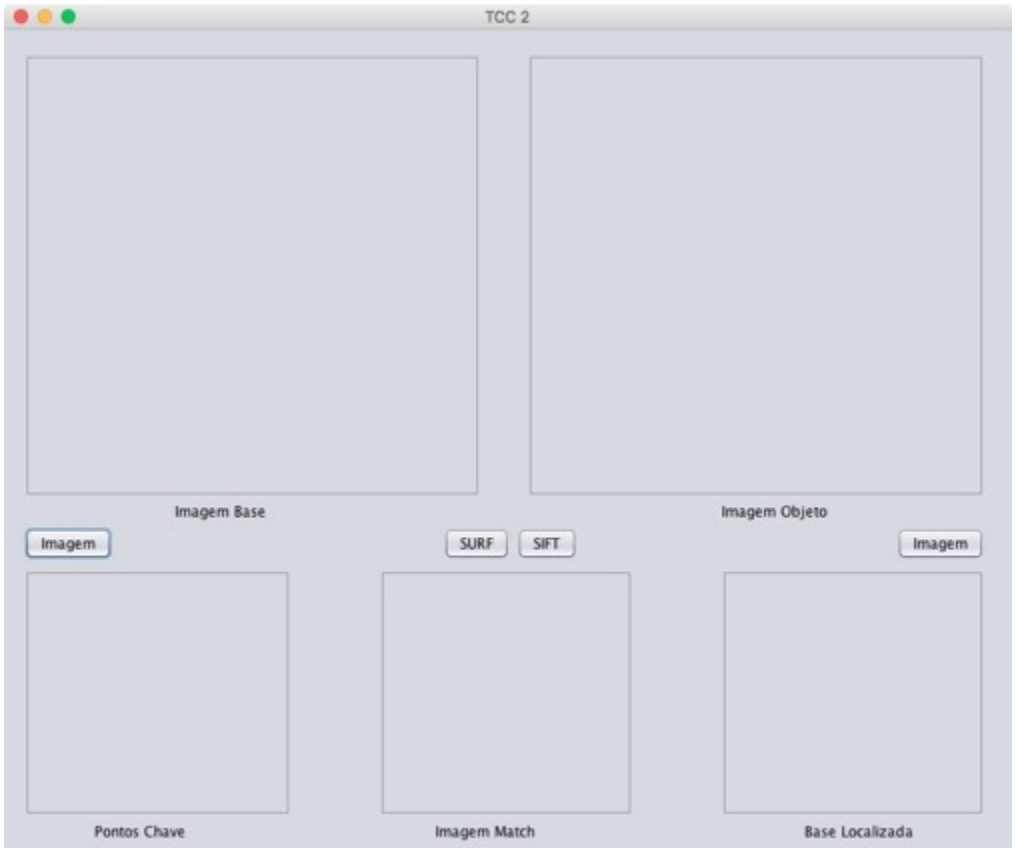
Para o desenvolvimento do aplicativo utilizamos a linguagem de programação Java, a biblioteca OpenCV na versão 2.4.13, no ambiente de desenvolvimento NetBeans e o sistema operacional macOS. A criação desta aplicação teve como base diversos exemplos que abordam o uso da biblioteca OpenCV.

Aplicação consiste em um painel onde se carrega a "Imagem Base" que é a imagem que possui o elemento que desejamos localizar, também chamada de imagem de estudo, e a "Imagem Objeto" que se trata da cena onde o objeto de estudo se encontra. Temos a opção de utilizar o descritor SIFT ou o SURF, para cada um deles foi criado um botão.

Depois de carregar as duas imagens, os procedimentos feitos pela aplicação envolvem em aplicar o descritor desejado de acordo com o botão pressionado pelo usuário, a criação da matriz de saída, detecção e impressão dos pontos chaves na Imagem base, o match entre as duas imagens, a localização da Imagem base dentro da imagem Objeto e por fim a impressão das imagens com os resultados obtidos.

Da esquerda para direita no painel da aplicação na parte inferior temos a impressão dos pontos chave na imagem de estudo, o match entre as duas imagens e o objeto base encontrado na imagem objeto. Na figura 4 temos um screenshot da tela da aplicação.

Figura 5. Aplicação em Java que utiliza a biblioteca OpenCV para o reconhecimento de imagens



2.4. Uso da Ferramenta

Podemos integrar em um cluster de treinamento, várias imagens que darão origem a um único objeto modelo. As imagens de treinamento que possuem um ponto de vista similar serão agrupadas nos mesmos objetos. O modelo SIFT é formado a partir da extração da localização, orientação, escala, dentre outras características da imagem de treinamento. O modelo SIFT aceita até 20 pontos em qualquer direção.

O processo de combinação dos modelos nas imagens segue a abordagem da transformação de Hough seguida pelos mínimos quadrados de verificação geométrica. Cada vizinho forma uma combinação com o mais próximo de acordo com a distância euclidiana da imagem banco de dados. O modelo deve ter pelo menos 20% da imagem a ser analisada. Isso previne erros de combinação, bem como permite diferenças de orientação. Porém dificulta trabalhos com reconhecimento de face por exemplo.

Foi utilizado essa versão da biblioteca tendo em vista que as mais recentes descontinuaram o suporte ao SIFT e o SURF. Para fazer a instalação da biblioteca no macOS foi utilizado o aplicativo Brew.

A opção de utilizar a linguagem Java, foi feita com o objetivo de aproveitar seus recursos de multiplataforma. Ao fazer o desenvolvimento no sistema Operacional MacOS, reforçamos essa questão, pois será aproveitada em outra aplicação que está sendo desenvolvida em outro sistema operacional. A versão mais nova da biblioteca OpenCV, por descontinuar as ferramentas desejadas, foi descartada. Existem opções para a instalação de add-ons para o uso destas ferramentas, mas não foi satisfatório para o desenvolvimento da ferramenta.

Com o intuito de testar a aplicação, foram simuladas situações diversas e observados os resultados obtidos, dentre esses foram selecionados os mais relevantes para serem discutidos nesse trabalho. No primeiro utilizou-se como objeto de estudo um livro posicionado em uma mesa com outros objetos. Para essa situação teste, a aplicação conseguiu ser bem-sucedida na localização da Imagem base, como se observa abaixo.

Figura 6. Livro como imagem de estudo (esquerda) ou Imagem base e a cena onde está localizado o objeto de estudo

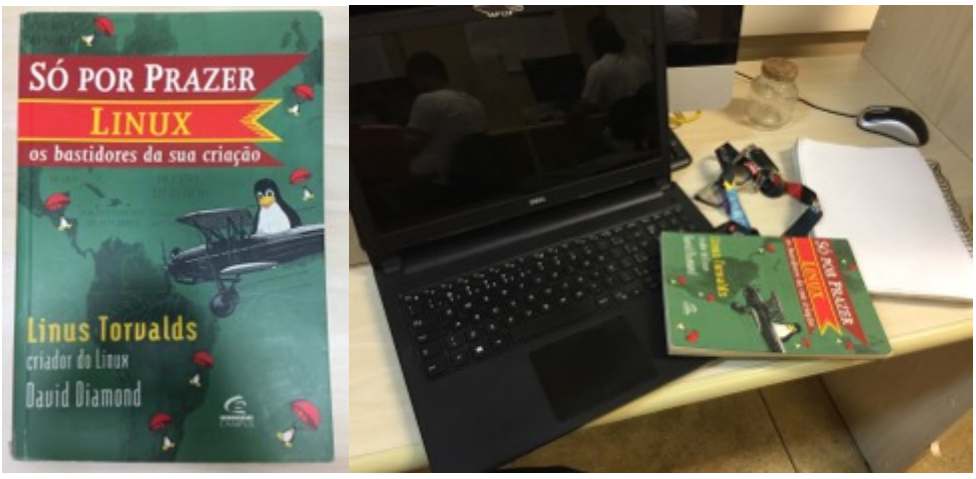


Figura 7. Keypoints localizados na Imagem base usando o SIFT (esquerda) e Keypoints localizados utilizando o SURF

Figura 8. Match entre as Imagens base e objeto usando o SIFT

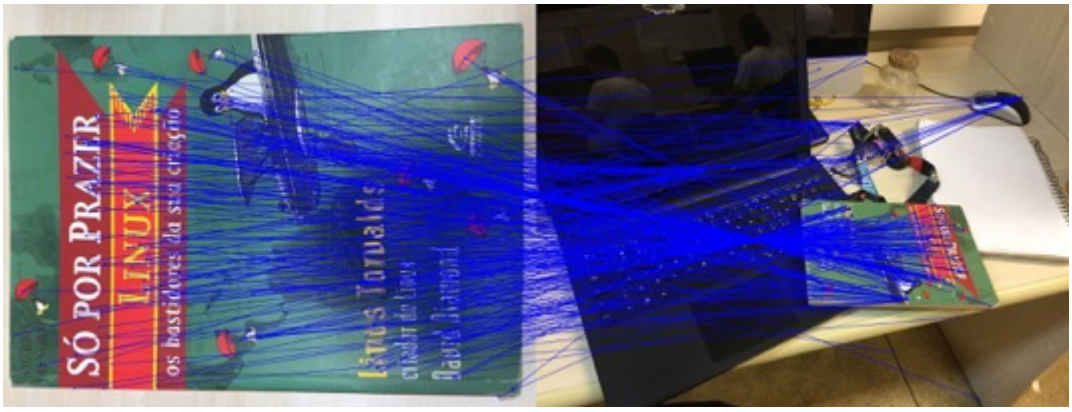


Figura 9. Match entre as imagens usando o SURF

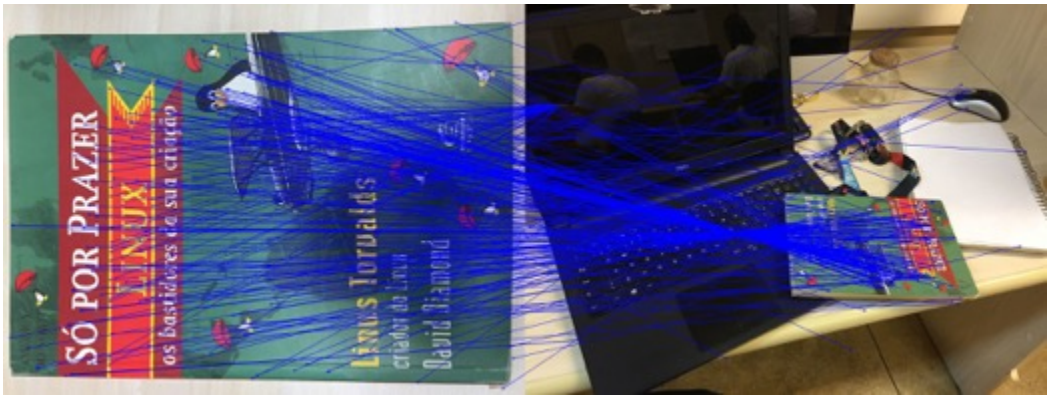
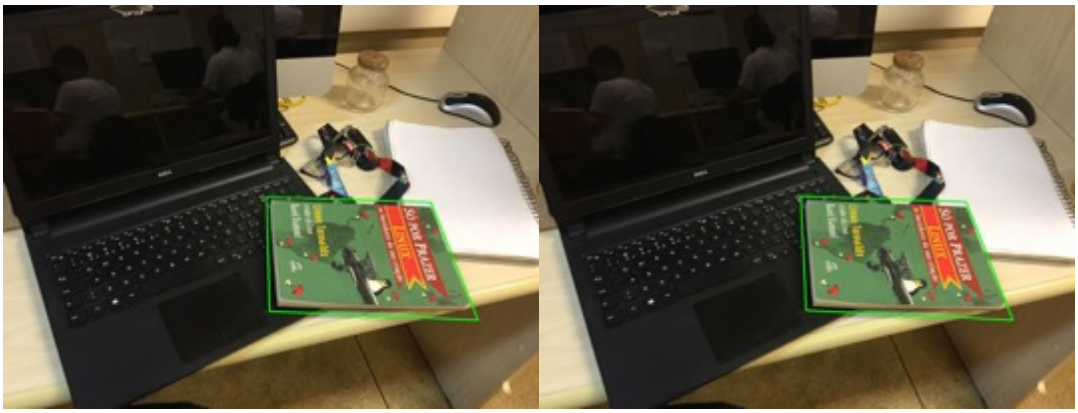


Figura 10. Objeto base encontrado na Imagem Objeto usando o SIFT (esquerda) e objeto base encontrado na Imagem Objeto usando o SURF



No segundo teste realizado, utilizamos novamente um livro como objeto de estudo. O objeto que serviu para Imagem de base foi colocado em uma cena distinta para formar uma nova Imagem objeto. Foi verificado que a aplicação teve êxito em localizar o objeto de estudo como pode ser observado nas imagens abaixo.

Figura 11. Objeto de estudo utilizado para formar a Imagem base (esquerda) e cena utilizada para formar a imagem Objeto

Figura 12. Keypoints localizados na Imagem Base usando o SIFT (esquerda) e Keypoints localizados usando o SURF

Figura 13. Match entre as duas Imagens usando o SIFT

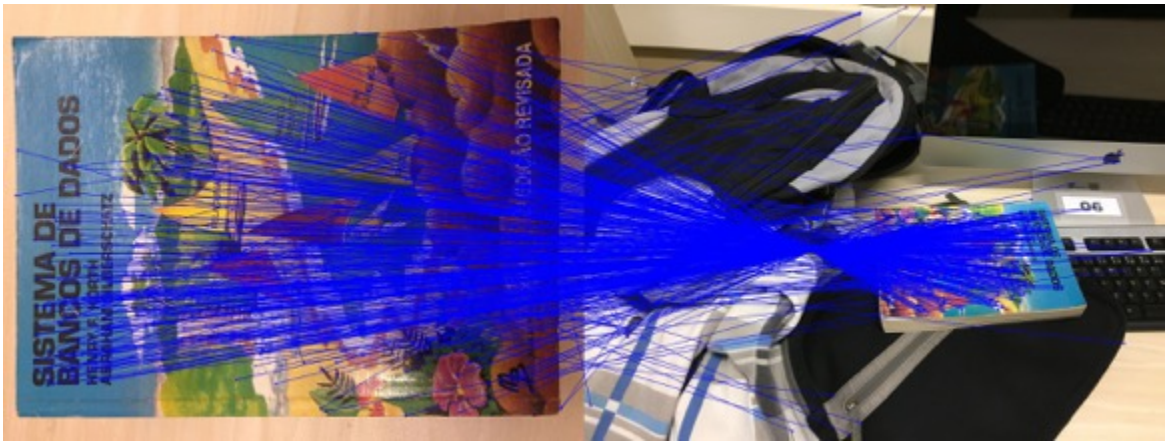


Figura 14. Match entre as duas imagens usando o SURF

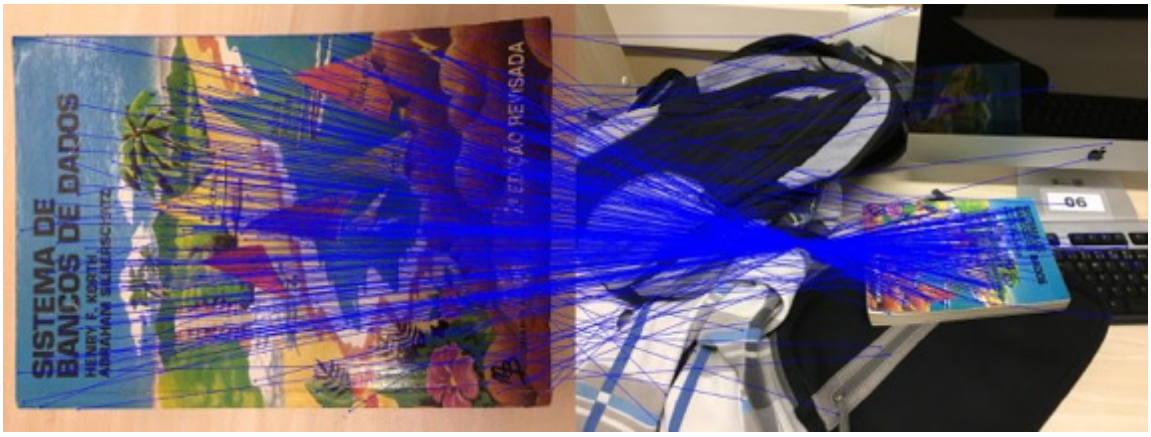
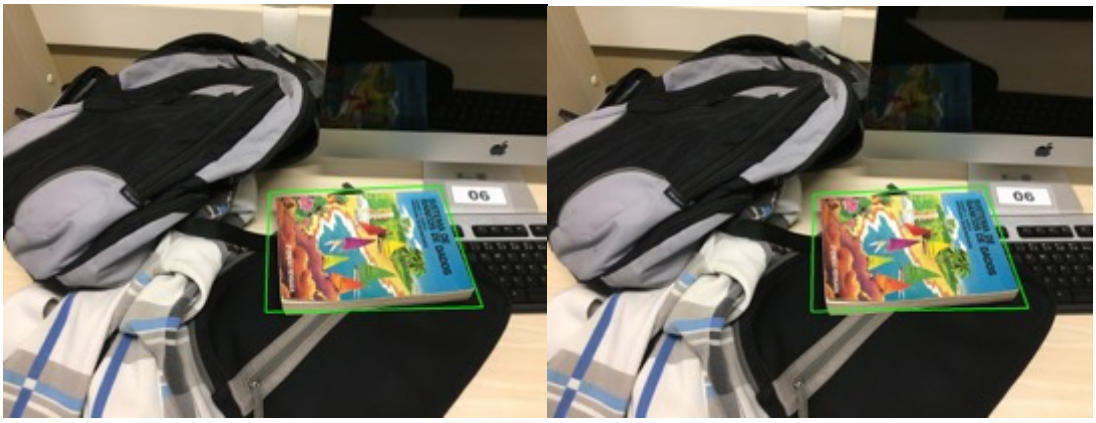


Figura 15. Objeto de estudo localizado na Imagem objeto usando o SIFT (esquerda) e Objeto de estudo localizado usando o SURF



Foi obtido sucesso com a realização de um teste utilizando um CD como objeto de estudo sobre uma mesa com vários outros objetos em volta, como pode ser observado nas imagens abaixo.

Figura 16. CD utilizado como objeto de estudo para criação da Imagem base (esquerda) e cena utilizada para formação da Imagem Objeto

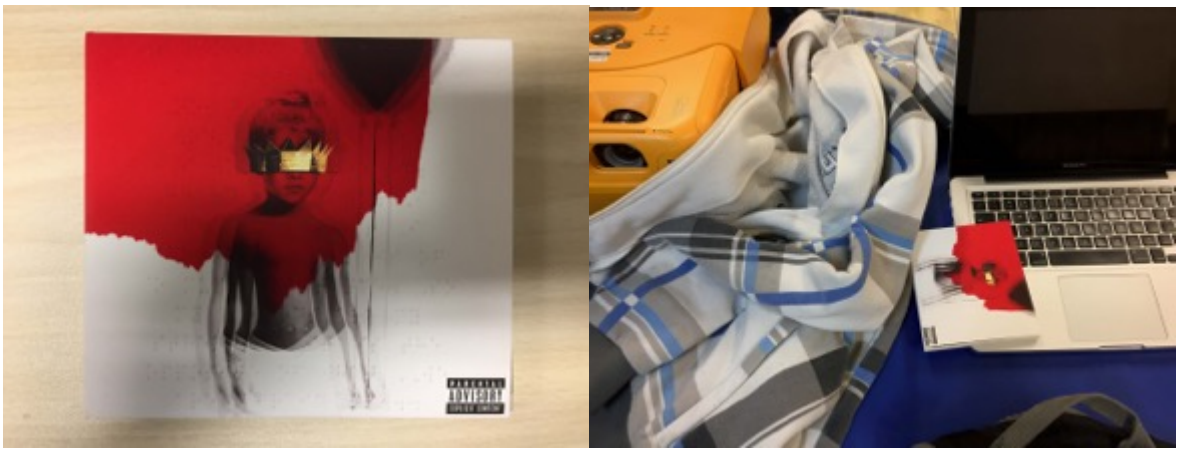


Figura 17. Localização os Keypoints na Imagem Base usando o SIFT (esquerda) e localização dos Keypoints usando o SURF

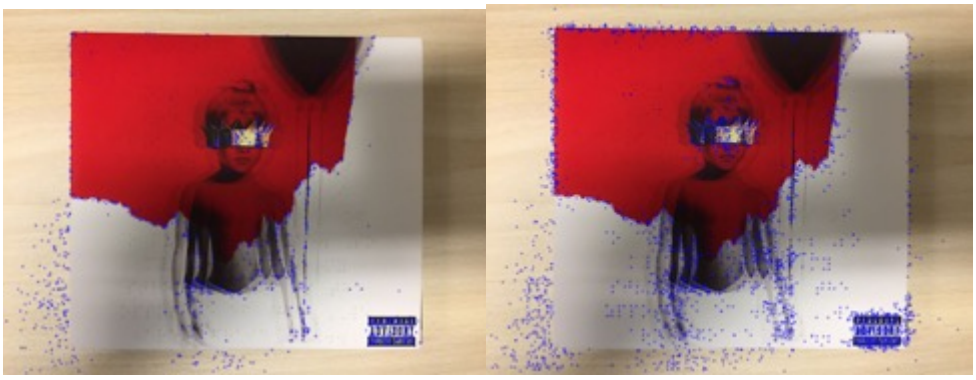


Figura 18. Match entre as duas Imagens usando o SIFT

Figura 19. Match entre as duas imagens usando o SURF

Figura 20. Objeto de estudo localizado usando o SIFT (esquerda) e objeto de estudo localizado usando o SURF

Foi realizado um segundo teste com outro CD porém em uma mesa com uma estrutura mais rugosa para criação da imagem Base. Para a montagem da Imagem objeto, foram utilizados outros CDs em uma mesa com rugosidades, além da estampa das cadeiras em volta. Nessa situação, a aplicação também localizou pontos chaves na rugosidade da mesa. Obtivemos sucesso nessa situação, porém na localização da Imagem base a aplicação identificou uma área maior.

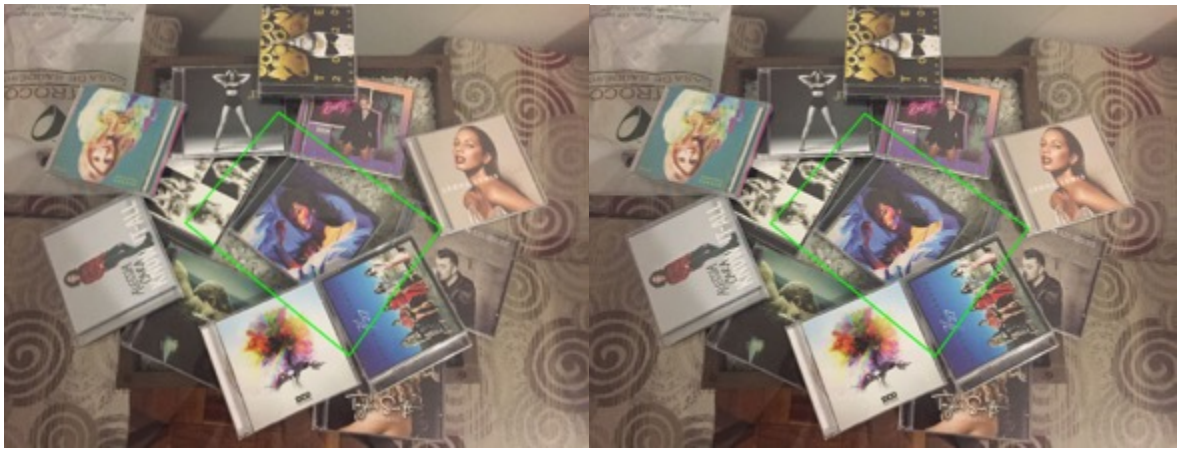
Figura 21. CD utilizado para criação da Imagem base em uma mesa com rugosidade (esquerda) e cena utilizada para criação da Imagem objeto

Figura 22. Keypoints localizados na Imagem base usando o SIFT (esquerda) e Keypoints localizados usando o SURF

Figura 23. Match entre as duas imagens usando o SIFT

Figura 24. Match entre as duas imagens usando o SURF

Figura 25. Objeto de estudo localizado usando o SIFT (esquerda) e objeto de estudo localizado usando o SURF



Foi realizado um teste com uma frase escrita em uma página de livro, para criação da Imagem base. Como Imagem objeto foi fotografada a página do livro em que a frase estava contida. Na correspondência da imagem match alguns pontos chaves foram localizados em outras partes do texto. Para essa situação a aplicação também conseguiu localizar a Imagem base como esperado.

Figura 26. Frase utilizada para criação da Imagem base (esquerda) e página do livro utilizada como Imagem objeto

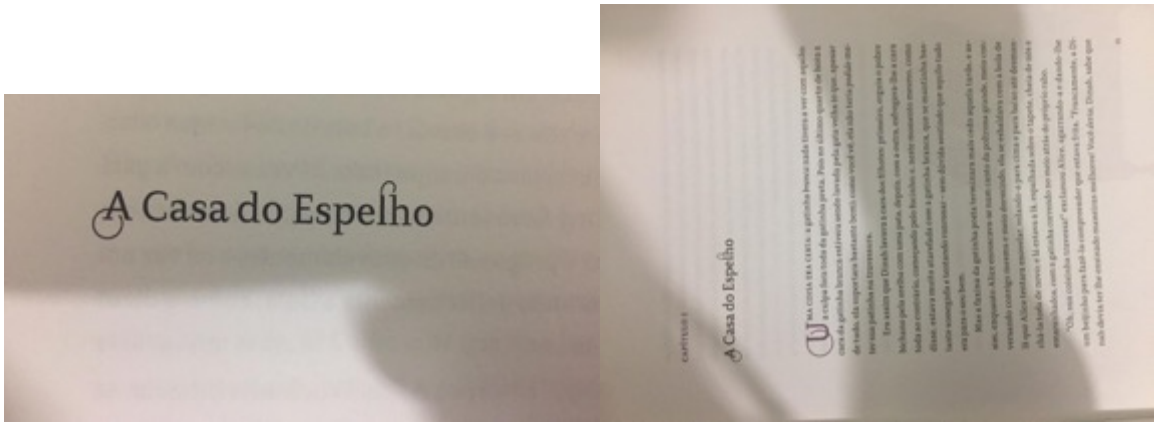


Figura 27. Keypoints localizados usando o SIFT (esquerda) e Keypoints localizados usando o SURF

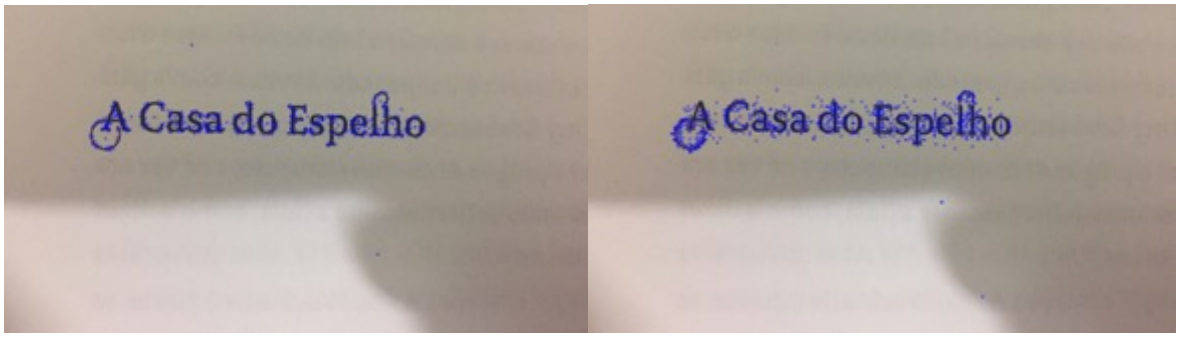


Figura 28. Match entre as duas imagens usando o SIFT

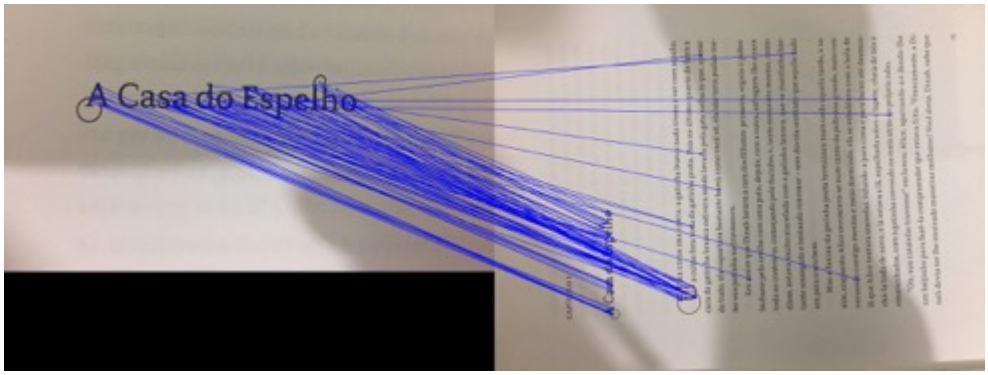


Figura 29. Match entre as duas imagens usando o SURF

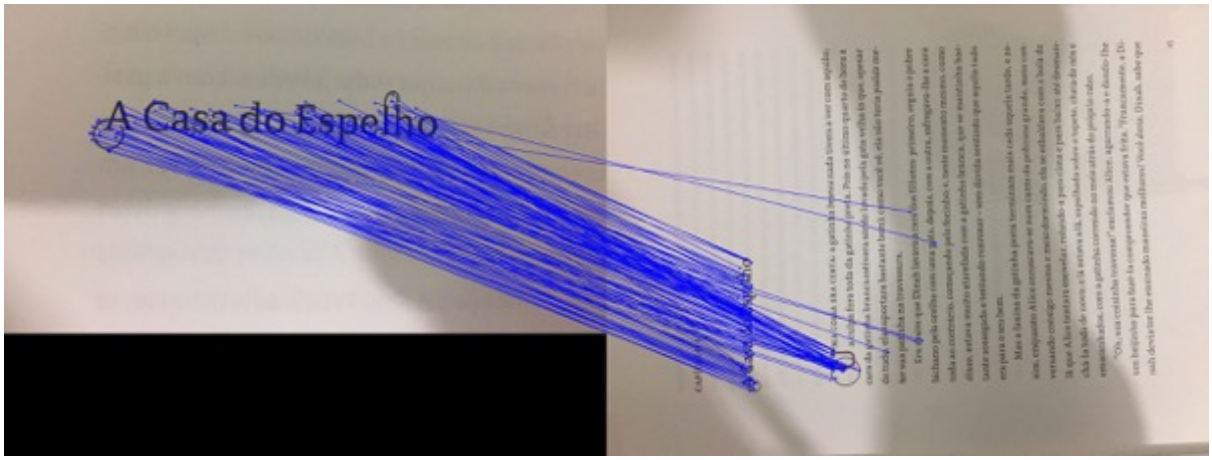
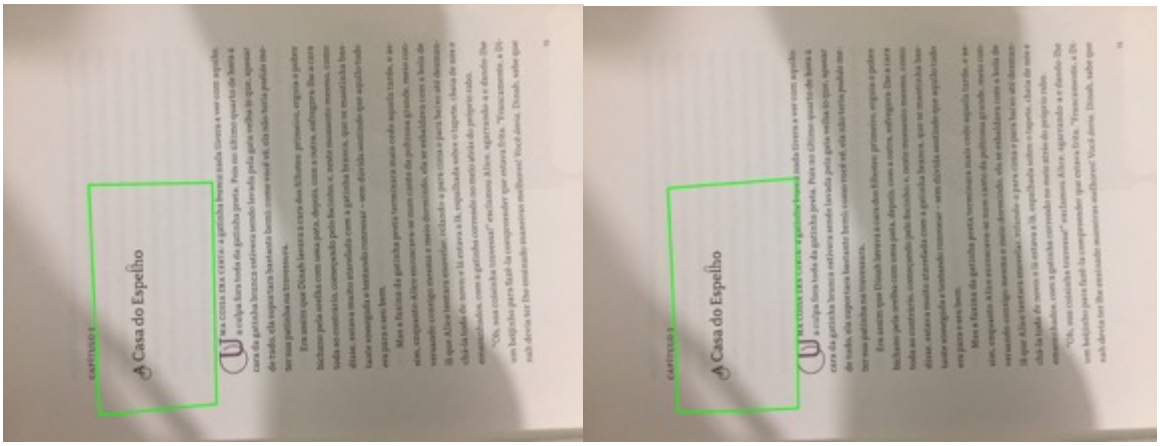


Figura 30. Frase localizada usando o SIFT (esquerda) e objeto de estudo localizado usando o SURF



Foi realizado um teste com uma embalagem de sabão líquido para Imagem base em uma área de serviço como Imagem objeto. Para essa situação, o SIFT conseguiu localizar o objeto de estudo, porém cortou uma parte dele cuja cor era muito similar ao fundo. O algoritmo SURF não localizou o objeto de estudo como esperado.

Figura 31. Objeto de estudo utilizado para criar a Imagem base (esquerda) e cena utilizada para criação da Imagem objeto

Figura 32. Keypoints obtidos com o SIFT (esquerda) e Keypoints obtidos com o SURF

Figura 33. Match das imagens obtidos com o SIFT

Figura 34. Match das imagens obtidas com o SURF

Figura 35. Objeto de estudo não localizado utilizando o SURF (esquerda) e objeto de estudo localizado com o SIFT

Como último teste a ser ressaltado nesse trabalho, visando também demonstrar a dificuldade tanto do SIFT quanto do SURF em localizar faces, se utilizou-se um brinquedo com feições humanas para criação da Imagem base sobre uma mesa com poucas rugosidades e poucos elementos em volta para criação da Imagem objeto. Nem o SIFT nem o SURF foram bem-sucedidos em encontrar a Imagem de estudo como esperado.

Figura 34. Brinquedo com feições humanas (esquerda) e cena utilizada para criação da imagem objeto

Figura 35. Keypoints localizados com o SIFT (esquerda) e Keypoints localizados com o SURF

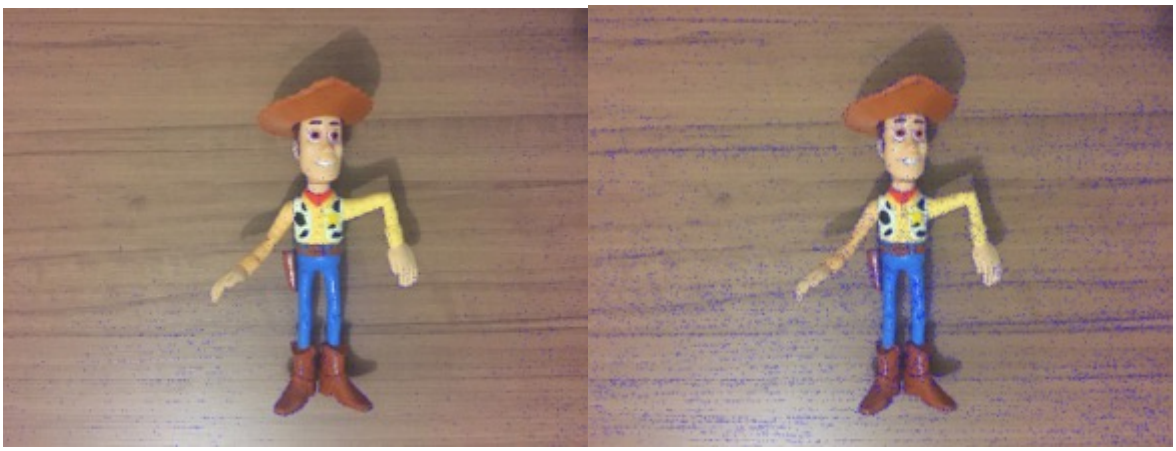


Figura 36. Match das imagens com o SIFT

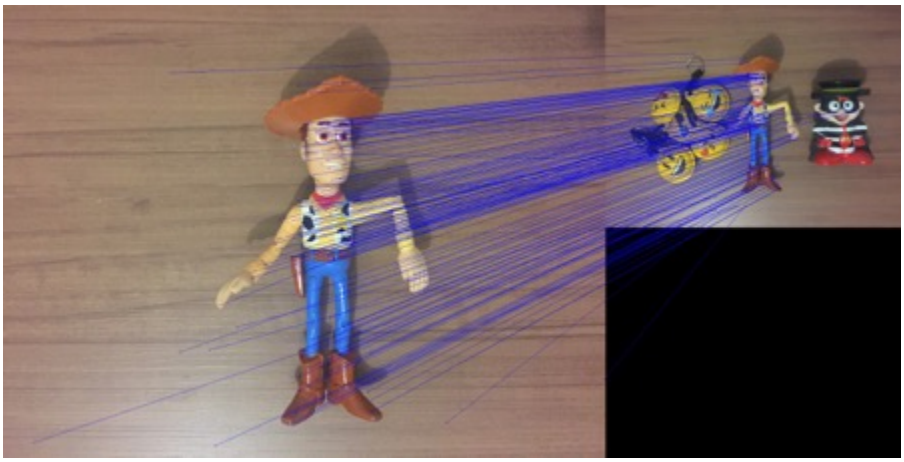


Figura 37. Match das imagens usando o SURF

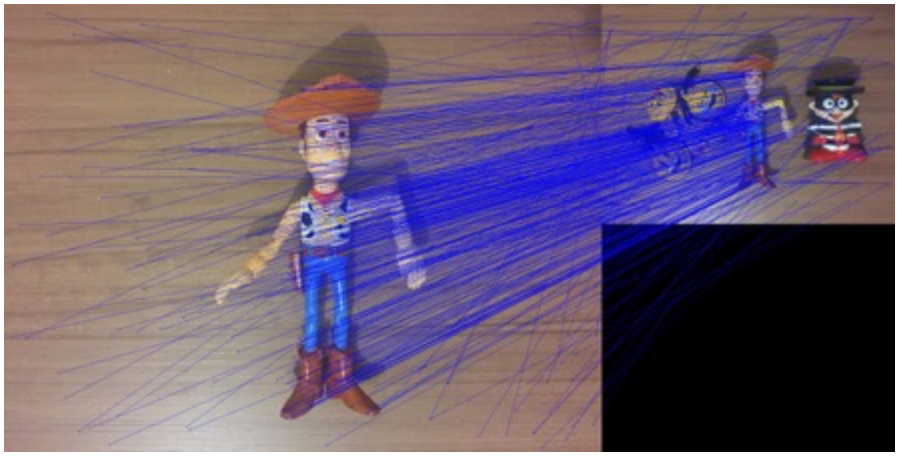


Figura 38. Objeto de estudo não localizado como esperado usando o SIFT (esquerda) e objeto de estudo não localizado como esperado utilizando o SURF

3. Considerações gerais e Trabalhos Futuros

O algoritmo SIFT, bem como o SURF podem ser usados na correspondência entre imagens aéreas e terrestres, utilização no agronegócio, estimação de altitudes, reconhecimento de sinalizações de trânsito, segurança (reconhecimento facial de pessoas desaparecidas e identificação de terroristas em áreas turísticas).

Este trabalho obteve resultados satisfatórios na localização dos objetos de estudo mesmo em cenas que apresentavam rugosidades ou onde o plano de fundo possuía cores semelhantes à da Imagem base. Também o algoritmo obteve êxito na localização de texto.

A biblioteca OpenCV possui uma grande variedade de funções e métodos prontos que facilitam a criação dos códigos. Todos os cálculos matemáticos que foram estudados para chegar à estrutura dos algoritmos foram simplificados nestas funções.

Para criação da aplicação deste estudo, utilizamos várias destas funções. Com o auxílio dessa ferramenta foi possível transformar a imagem em uma matriz possível de ser trabalhada pelo SIFT ou SURF, também localizar e imprimir na imagem os pontos chaves, fazer o match entre as duas imagens, além de imprimir na imagem o espaço onde o objeto de estudo se encontrava.

Por ter sido desenvolvido em Java, a aplicação criada por esse estudo pode ser executada em qualquer sistema operacional que tenha uma máquina virtual Java instalada. Este fator, aliado a escolha da biblioteca OpenCV, que identifica o processador em que está sendo a executada a aplicação, permite um menor custo computacional.

Dentre os resultados que foram obtidos se percebeu uma dificuldade em reconhecer objetos de estudo com feições humanas, mesmo em situações em que existiam um número considerável de pontos chaves na imagem base. Percebeu-se também uma dificuldade em reconhecer objetos de estudo com pouco contraste, em especial na cor preta, ainda que tivessem um número considerável de Keypoints localizados. Observamos que para esses casos se faria necessário alterar a cena para realizar a localização do objeto de estudo, ou fazer alterações de contraste.

Verificou-se ainda que as imagens devem ter a mesma largura para que a matriz seja montada corretamente. Foi padronizado no trabalho o uso de imagens com 1600 pixels de

largura. As que tinham orientação vertical precisaram sofrer ajuste de rotação de 90º para serem compatíveis com o estudo. Imagens com resoluções muito acima do supracitado exigiam muito poder de processamento computacional e inviabilizavam a pesquisa com o equipamento usado.

Para o primeiro par de imagens testados, verificou-se que o número de keypoints encontrados pelo SURF foi largamente superior aos encontrados pelo SIF, possivelmente devido a esse fator, o tempo de execução foi dobro do segundo descritor. Em ambos os casos o objeto de estudo foi localizado como esperado. Para o segundo par, a diferença entre o número de keypoints e o tempo de execução foram muito próximos sem haver maiores discrepâncias em relação aos descritores analisados, para esse teste o objeto de estudo também foi localizado.

No terceiro par, considerando que o objeto de estudo era mais simples, o número de keypoints localizados foi o segundo mais tímido se considerarmos os pares que obtiveram sucesso. Para esse teste houve uma discrepância média do número de keypoints localizados e tempo de execução próximo. Para o quarto par analisado, se obteve sucesso em encontrar o objeto de estudo analisado em ambas as soluções e um volume de keypoints mais densos no descritor SURF, não houve discrepâncias consideráveis no tempo de execução.

O quinto par de estudo com foco na busca de textos, foi o mais tímido em número de keypoints localizados. Porém o tempo de execução no descritor SIFT se mostrou muito superior para um teste tão simples, equiparando a testes com imagens complexas. Para ambas soluções o objeto de estudo foi localizado. O sexto objeto de estudo, que apresentava cor semelhante ao fundo, foi localizado apenas no SIFT e com uma pequena falha. Nesse teste o número de keypoints localizados pelo SURF foi superior e tempo de execução inferior de comparado ao SIFT. Para o sétimo e ultimo par testado, não foi possível localizar em nenhuma das soluções o objeto de estudo desejado. Novamente o SURF teve maior densidade no número de keypoints e tempo de excução menor. Todos os testes podem ser observados na tabela 1.

Exemplo	**Key points SURF**	**Key Points SIFT**	**Tempo de Execução SURF**	**Tempo de execução SIFT**	**Objeto Encontrado** pelo SURF	**Objeto Encontrado** pelo SIFT
1º par	10.585	4.635	12 s	6 s	Sim	Sim
2º par	12.260	8.684	6 s	7 s	Sim	Sim
3º par	3.080	1.142	4 s	5 s	Sim	Sim
4º par	9.744	4.672	6 s	7 s	Sim	Sim
5º par	946	615	3 s	6 s	Sim	Sim
6º par	4.582	3.251	4 s	7 s	Não	Sim – com pequena falha
7º par	3.047	1.079	3 s	6 s	Não	Não

Tabela 1. Comparativo entre o SIFT e o SURF

Conclui-se que na grande maioria dos testes o SURF mostrou um poder maior de localização de pontos chaves e menor tempo de execução. O SIFT por sua vez mostrou um maior poder de localização de objetos de estudo que apresentavam cor semelhante ao fundo da imagem objeto. Ambas soluções falharam na localização de objetos de estudo com feições humanas.

Como sugestão de trabalhos futuros, podemos citar o aperfeiçoamento do algoritmo para reconhecer faces, o que atualmente é uma limitação. Com esse aperfeiçoamento seria possível localizar pessoas desaparecidas usando por exemplo uma foto 3x4. Esse algoritmo também pode ser aperfeiçoado para localização de placas de carros ou para fazer o reconhecimento de sinalizadores de trânsito e assim integrar os sistemas de segurança de centrais de trânsito.

4. Conclusão

A computação gráfica motivou diversos avanços na informática, robótica, controle de qualidade, automação, dentre outros. Quando uma rede social reconhece a face de um dos nossos amigos existem diversos algoritmos trabalhando por trás desses resultados e avanços significativos ainda podem ser feitos.

O uso do SIFT e do SURF pode contribuir em muito para esses avanços se adaptados corretamente a cada necessidade. A criação de sistemas computacionais que utilizam esses algoritmos em uma realidade próxima dos programadores facilita a criação de sistemas mais robustos, rápidos, eficientes e inteligentes. Esses sistemas por sua vez, facilitam a interação dos usuários que pouco ou nada sabem de programação com as realidades da computação gráfica.

Esse trabalho teve foco no estudo o algoritmo SIFT e seu semelhante SURF aliados a biblioteca OpenCV, colocando em prática os conhecimentos em programação na linguagem Java adquiridos durante o curso. O sistema resultante dessa pesquisa possibilita gerar e comparar resultados usando os dois algoritmos.

Os resultados obtidos mostram que ambos algoritmos possuem uma grande eficiência em gerar os pontos chaves corretos e localizá-los nas Imagens objeto. Percebeu-se que, mesmo em um sistema em sua fase inicial, já é possível gerar resultados assertivos na maioria dos casos de estudo.

Percebeu-se uma necessidade de aprimorar a aplicação, que está em um estado mais genérico, para se obter resultados mais assertivos para determinadas situações como reconhecimento de faces. Um avanço maior na pesquisa seria possibilitado, por exemplo, pelo uso da biblioteca OpenCV para reconhecimento de Objetos de estudo em vídeos.

Referência

ALVES, Wonder Alexandre Luz; ARAÚJO, Sidnei Alves de; LIBRANTZ, André Felipe H. *Reconhecimento de Padrões de Texturas em Imagens Digitais Usando uma Rede Neural Artificial Híbrida*. 8f. Artigo. Departamento de Ciências Exatas – Centro Universitário Nove de Julho (UNINOVE) São Paulo – SP – Brasil. São Paulo – SP

BOMBARDELLI, Felipe Gustavo. *Estudo sobre reconhecimento facial*. 5f. Artigo. Universidade Federal do Paraná. Curitiba- PR

BORTH, Marcelo Rafael; PISTORI, Hemerson; GONÇALVES, Ariadne Barbosa; FREITAS, Uéliton. *Análise da Extração de Atributos do Algoritmo SURF em Espécies de Peixe*. 2013. 10f. Artigo. IFMS – Instituto Federal de Mato Grosso do Sul, UCDB – Universidade Católica Dom Bosco, UCDB – Universidade Católica Dom Bosco, UFMS – Universidade Federal de Mato Grosso do Sul. Ponta Porã - MS. 2013

CASTRO, Armando Antonio Monteiro de; PRADO, Pedro Paulo Leite do. *Algoritmos para reconhecimento de padrões*. 2008. 18f. Artigo. Departamento de Engenharia Elétrica. Universidade de Taubaté. Taubaté - SP. 2008.

Dummy's Codes: Using SIFT/SURF for Object Recognition in OpenCV Java, Disponível em <http://dummyscodes.blogspot.com/2015/12/using-siftsurf-for-object-recognition.html>. Acesso em 1 junho de 2018

GONZÁLES, Giancarlo Luis Gómez. *Aplicação da Técnica SIFT para Determinação de Campos de Deformações de Materiais*. 2010. 109f. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro - RJ. 2010.

MARENGONI, Maurício; STRINGHINI, Denise. *Tutorial: Introdução à Visão Computacional usando OpenCV*. 2009. 36f. Artigo. Universidade Presbiteriana Mackenzie, Faculdade de Computação e Informática e Pós Graduação em Engenharia Elétrica, Universidade Presbiteriana Mackenzie, Faculdade de Computação e Informática. 2009.

OpenCV library, Disponível em <https://opencv.org>. Acesso em 1 de junho de 2018

RIBEIRO, Matheus V. L.; SALOMÃO, João M. *Desenvolvimento de um sistema para reconhecimento automático de placas de Trânsito*. 2014. 8f. Artigo. Coordenadoria de Engenharia Elétrica, Campus Vitória, Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo - IFES. Vitória - ES. 2014

SILVA, Marco Antônio de Albuquerque. *Extração e Comparação de Características Locais para o Reconhecimento Facial por Meio de Retratos Falados*. 2014. 81f. Dissertação de Mestrado. Universidade Federal de Ouro Preto. Ouro Preto - MG. 2014.

SILVA, Marcos Aurélio Medeiros; CHAVES, Anderson Pinheiro de Araújo; AQUINO, Francisco José Alves de. *Projeto E Desenvolvimento De Uma Ferramenta Educativa Para Ensino De Processamento De Imagens Baseado Na Biblioteca Opencv*. 9f. Artigo. Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE). Belém - PA. 2012.