colloquium exactarum

DETECÇÃO E RECONHECIMENTO DE OBJETOS EM IMAGENS UTILIZANDO ALGORITMOS DE EXTRAÇÃO DE PONTOS CHAVE

OBJETCT DETECTION AND RECOGNITION IN IMAGES USING KEYPOINT EXTRACTION ALGORITHMS

Leonardo Almeida Rodrigues[1]; Francisco Assis da Silva[1]; Danillo Roberto Pereira[1]; Almir Olivette Artero[2]; Marco Antonio Piteri[2]

[1]Faculdade de Informática – FIPP, Universidade do Oeste Paulista – UNOESTE
e-mail: leorodrigues@unoeste.edu.br, {chico, danilopereira}@unoeste.br
[2]Faculdade de Ciências e Tecnologia – FCT, Universidade Estadual Paulista – Unesp
e-mail: {almir, piteri}@fct.unesp.br

RESUMO - Este trabalho apresenta um estudo comparativo de algoritmos descritores de pontos chave fazendo uma combinação de detectores e descritores. São combinados os detectores dos algoritmos SIFT, SURF e HARRIS com os descritores dos algoritmos SIFT e SURF. Foram realizados três experimentos com imagens com características bem diferentes, em dois desses experimentos foram usadas imagens de objetos reais, em um terceiro experimento, foram realizadas algumas degradações em uma imagem, como redução de escala, rotação, borramento, escurecimento e adição de ruído, para analisar o comportamento dos algoritmos. Também foram realizados experimentos com todas as degradações juntas. Para analisar as combinações propostas, foram extraídos o tempo de processamento e o número de *inliers*. Esse trabalho contribui para aplicações que busquem utilizar algoritmos descritores de pontos chave para serem usados na detecção e reconhecimento de objetos em imagens.

Palavras-chave: Reconhecimento de Objetos; Características Locais; Detector de Pontos Chave; Descritor de Pontos Chave.

ABSTRACT - This paper presents a comparative study of descriptors keypoint algorithms doing a combination of detectors and descriptors. We combined the SIFT, SURF and Harris detector algorithms with the SIFT and SURF descriptor algorithms. Three experiments with images with different characteristics were performed, in two of these experiments were used images of real objects, in a third experiment, some degradation in an image such as scale reduction, rotation, blurring, darkening and addition of noise have been performed, to analyze the behavior of the algorithms. Experiments were also performed with all degradations together. To analyze the proposed combination, we extracted processing time and the number of inliers. This work contributes to applications that require using keypoint descriptors algorithms in detection and recognition of objects in images.

Keywords: Object Recognition; Local Features; Keypoint Detector; Keypoint Descriptors.

Recebido em: 03/06/2014
Revisado em: 30/08/2014
Aprovado em: 20/09/2014

1 INTRODUÇÃO

A Visão Computacional vem a ser o ramo da Ciência da Computação que reúne todas as teorias e tecnologias desenvolvidas com a finalidade de possibilitar que imagens sejam interpretadas por sistemas artificiais implementados em computadores (MAIA, 2010). O reconhecimento de objetos em imagens é uma das principais áreas de aplicação de Processamento de Imagens e Visão Computacional (TREIBER, 2010). A tarefa de detecção e reconhecimento de objetos em imagens é um dos vários ramos que trata a Visão Computacional.

Em geral, a descrição de objetos é realizada em duas etapas, a primeira etapa consiste na utilização de um detector de pontos chave, e em seguida, com a utilização de um descritor, ser capaz de gerar valores ou atributos que sejam suficientes para descrever os pontos chave (SILVA et al., 2013).

É observado em Silva (2012) que o reconhecimento visual tem uma variedade de aplicações potenciais incluindo muitas áreas de inteligência artificial e de recuperação de informações. O autor afirma que os descritores representam as partes características de uma imagem.

Mudanças na escala, na orientação, pontos de vista, ou distorções como borramentos, alterações de iluminação ou oclusão torna a tarefa de reconhecimento de objetos ainda mais difícil (SILVA et al., 2013).

A inclusão dos algoritmos mais relevantes na biblioteca de visão computacional OpenCV tem sido de grande ajuda para a implementação de sistemas que se utilizam destes recursos. No entanto, a escolha de uma desses algoritmos nem sempre é fácil, e simples de ser realizada por falta de informações a respeito de suas vantagens e desvantagens. Assim, neste artigo, é apresentada uma análise da qualidade de diferentes combinações entre detectores e descritores de dois dos algoritmos mais conhecidos, com a implementação do algoritmo HARRIS-SIFT proposta deste trabalho.

O presente trabalho encontra-se organizado da seguinte maneira: a Seção 2 apresenta os trabalhos relacionados, que serviram de motivação e contribuíram para a realização deste trabalho; na Seção 3 são apresentados os conceitos fundamentais que descrevem o funcionamento e as características dos algoritmos utilizados; a Seção 4 descreve a metodologia utilizada, os passos que foram realizados no desenvolvimento deste trabalho. Na Seção 5 são apresentados os experimentos que foram realizados, os testes realizados com a combinação dos detectores e descritores dos algoritmos de extração de pontos chave, e os resultados obtidos; e por fim na Seção 6 são

apresentadas as considerações finais e trabalhos futuros.

2 TRABALHOS RELACIONADOS

O trabalho apresentado por Maia (2010) investiga a utilização de métodos supervisionados, em particular aqueles baseados em classificadores não-lineares para conduzir ou auxiliar a etapa de detecção de pontos chave. Mais especificamente, propõe uma metodologia para a incorporação de mecanismos supervisionados para aprendizagem de máquina na etapa de detecção de pontos chave. O autor desenvolve um estudo de caso por meio da combinação de um método existente com a metodologia proposta, realiza uma análise crítica dos resultados a partir de experimentos que caracterizem a utilização da abordagem proposta no contexto de aplicações do mundo real.

O trabalho proposto por (SILVA et al., 2013) apresenta um estudo comparativo usando combinações diferentes de detectores de pontos chave e descritores, aplicado em pares de imagens digitais (Objeto / Cena), em que as imagens das cenas foram degradadas por: borramento, escala, iluminação, rotação, ruído e todas essas degradações ao mesmo tempo. Todas as combinações foram analisadas usando os detectores SIFT, SURF, FAST, STAR, MSER, GFTT (com Harris), GFTT e ORB, e os descritores: SIFT, SURF, BRIEF e ORB. Os parâmetros observados nesse trabalho são tempo de processando (TP), número de *inliers* (NIn) e geração de matriz homográfica capaz de realizar satisfatoriamente a correspondência entre objetos e imagens das cenas.

3 CONCEITOS FUNDAMENTAIS

A identificação de pontos com características em comum entre duas imagens não é um trabalho simples. A primeira dificuldade está em encontrar pontos chave (*keypoints*) em uma das imagens (imagem objeto) e, em seguida, localizá-los na outra imagem (imagem cena). Neste trabalho, está proposta a utilização do algoritmo SIFT (LOWE, 2004), SURF (BAY et al., 2006) e HARRIS-SIFT (HARRIS; STEPHENS, 1988) para encontrar e descrever os pontos chave das imagens de entrada (imagem cena) e imagem dos *templates* (imagens dos objetos buscados). Depois de encontrados os pontos chave correspondentes entre as imagens, é utilizado o algoritmo RANSAC para eliminar falsas correspondências entre esses pontos chave. Ambos os algoritmos são descritos nas seções seguintes.

3.1 SIFT

SIFT (*Scale Invariant Feature Transform*) é um algoritmo de Visão Computacional proposto e publicado por David G. Lowe em 1999 (LOWE, 1999). Um algoritmo que permite a detecção e extração de descritores que tem muitas propriedades que são desejáveis para correspondência (*matching*) de diferentes imagens de um objeto ou cena. Esses descritores são razoavelmente invariantes à mudanças de rotação e escala, e parcialmente invariante a iluminação e a projeções 3D.

É um algoritmo basicamente dividido em duas partes, o detector e o descritor. O detector é baseado no cálculo de diferenças gaussianas e o descritor utiliza histogramas de gradientes orientados para descrever a vizinhança dos pontos de interesse.

Os descritores que o algoritmo SIFT fornece são bem localizados, reduzindo assim a possibilidade de não haver correspondência.

A obtenção de descritores de uma imagem pelo algoritmo SIFT é realizada por meio de quatro estágios principais. Os dois primeiros estágios descrevem a parte do detector e as duas seguintes a formação do descritor. Esses estágios são (LOWE, 2004):

Detecção de extremos no espaço escala: Nessa primeira etapa os pontos chave são detectados por busca de características estáveis que identificam candidatos que são invariantes a escala e orientação. Isto é feito utilizando-se uma função chamada de espaço escala (WITKIN, 1983), no algoritmo SIFT é usada a função Gaussiana como sendo o núcleo (*kernel*) da função espaço escala. De maneira sucinta, são detectados extremos (máximos ou mínimos) em uma pirâmide da imagem convoluída com a função de diferença de filtros Gaussianos. O espaço escala de uma imagem é definido como uma função $L(x, y, \sigma)$, que é dada a partir da convolução de uma Gaussiana $G(x, y, \sigma)$, com uma imagem de entrada $I(x, y)$, conforme a Equação 1:

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

onde * é a operação de convolução em *x* e *y*, e função Gaussiana é dada por:

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

Esse primeiro estágio de computação faz a busca em todas as escalas e locais de imagem. Isto é feito utilizando-se a diferença de filtros Gaussianos de modo a identificar pontos de interesse invariáveis à escala e orientação. Este é o estágio mais custoso computacionalmente do algoritmo (LOWE, 2004).

No filtro Gaussiano, o valor de σ representa o parâmetro que define o fator de suavização de imagem.

Foi proposta por Lowe (1999) uma forma eficiente para detectar pontos chave estáveis usando os extremos do espaço escala com o uso de uma função de Diferença de Gaussianas (DoG – *Difference of Gaussian*).

A diferença de Gaussianas, *D(x, y,* σ), é computada pela diferença de duas imagens filtradas em escalas próximas, separadas por uma constante multiplicadora *k.* A função DoG é definida pela Equação:

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \quad (3)$$

Esta convolução é a diferença entre imagens suavizadas por um filtro Gaussiano em escalas *σ* e *kσ.* A função DoG consegue detectar variações de intensidade na imagem, elimina detalhes indesejados e realça características fortes.

Localização de pontos chave: Para cada local candidato, em que foi detectado um extremo (máximo ou mínimo), um modelo detalhado é ajustado para se determinar a localização e escala. Pontos chave, ou pontos de interesse, são então selecionados baseando-se em suas medidas de estabilidade; são rejeitados os pontos de baixo contraste e localizados ao longo de bordas.

Segundo em Lowe (2004), todos os pontos detectados como extremos são possíveis candidatos a pontos chave, e o próximo passo é executar um ajuste detalhado aos pixels próximos em relação a escala, localização e relação de curvaturas principais.

Determinação de orientação: Uma ou mais orientações são associadas para cada imagem. Todas as operações seguintes são realizadas nos dados da imagem relativamente transformados em relação à orientação, escala e localização de cada ponto chave, assim provendo a invariância a estas transformações (MAIA, 2010).

A escala do ponto chave é utilizada para selecionar a imagem suavizada *L* com o filtro Gaussiano. Dessa maneira, todas as computações passam a ser executadas com invariâncias à escala. Para cada imagem $L(x,y,\sigma)$, na mesma escala σ, a magnitude do gradiente *m*(*x*, *y*) e a orientação θ(*x*, *y*) são computadas usando diferenças de pixels, conforme as Equações 4 e 5.

$$m(x,y) = \sqrt{\Delta x^2 + \Delta y^2} \quad (4)$$

$$\theta(x,y) = \tan^{-1}(\Delta y / \Delta x) \quad (5)$$

onde $\Delta x = L(x+1,y) - L(x-1,y)$ e $\Delta y = L(x,y+1) - L(x,y-1)$.

Descritor dos pontos chave: Neste estágio da computação, os gradientes locais da imagem são medidos na escala selecionada, na região ao redor de cada ponto chave. Estas medidas são então transformadas para uma representação que permite observar níveis significantes de distorção de forma e mudança na iluminação. São então criados histogramas de orientação para compor o descritor (Figura 1).

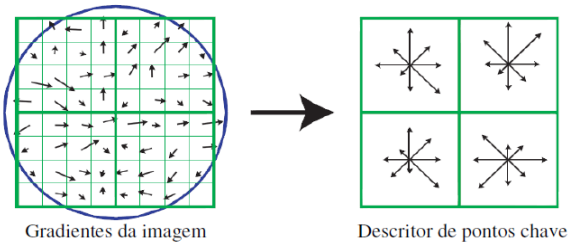

Figura 1. Gradiente da imagem e descritor de pontos chave.
Fonte: (LOWE, 2004).

O peso referente à magnitude de cada pixel é atenuado pela função Gaussiana. A função Gaussiana não é aplicada de modo idêntico ao do estágio anterior. É utilizado um peso α para interpolar a direção relativa no histograma.

Para cada imagem, são construídos diversos descritores, cada um referente a um ponto chave. Quando se aplica o algoritmo SIFT em uma imagem, tem-se como resultado um conjunto de descritores, que são usados para se fazer a correspondência entre duas imagens.

3.2 SURF

Segundo Bay et al. (2006), o algoritmo SURF (*Speeded Up Robust Features Algorithm*) pode ser divido em três passos: primeiramente, pontos de interesse são selecionados em localizações distintas da imagem, como cantos, borrões, e Junções *T*. A propriedade de maior valor de um detector de pontos de interesse é a sua repetibilidade, isto é, se é confiável, se ele encontra os mesmos pontos de interesse em diferentes pontos de vista. Em seguida, a vizinhança de todos os pontos de interesse é representada por um vetor de características. Este descritor tem que ser distintivo e ao mesmo tempo, robusto ao ruído, à detecção de erros e deformações geométricas e fotométricas. Finalmente, os vetores descritores são combinados entre imagens diferentes. A correspondência é frequentemente baseada em uma distância entre os vetores, por exemplo, a distância Mahalanobis ou a distância Euclidiana.

Segundo Bay et al. (2006) detectores de pontos de interesse (*Interest Point Detector*), provavelmente, o mais usado é o detector de cantos de Harris (HARRIS; STEPHENS, 1988), proposto em 1988, baseado em autovetores (*eigenvalues*) da matriz de segundo momento. No entanto, os cantos de Harris não são invariantes a escala. Lindeberg introduziu o conceito de seleção automática de escala (LINDEBERG, 1998). Isto

permite detectar pontos de interesse em uma imagem, cada um com sua própria escala e característica.

O descritor SURF baseia-se nas propriedades similares, com uma complexidade ainda mais simplificada. A primeira etapa consiste em fixar uma orientação reprodutível com base em informações a partir de uma região circular em torno do ponto de interesse, como mostra a Figura 2. É construída uma região quadrada alinhada com a orientação selecionada, e extrai o descritor SURF a partir dessa região.

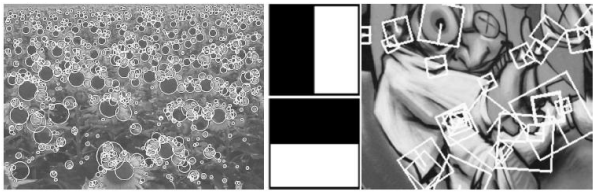

Figura 2. Esquerda: pontos de interesse detectados para um campo de girassol. Esse tipo de cena mostra claramente a natureza dos descritores baseados em detectores Hessian. Centro: Haar wavelet utilizada no SURF. Direita: Detalhe de uma cena grafite mostrando o tamanho da janela do descritor em diferentes escalas.
Fonte: (BAY et al., 2006).

Segundo Rachid e Pereira (2009) o algoritmo SIFT e SURF têm maneiras ligeiramente diferentes de detectar características. SIFT identifica as posições chave no espaço-escala através da busca por posições de máximo ou mínimo em uma pirâmide da imagem utilizando uma função diferença de Gaussianas (DoG), a qual é uma aproximação do Laplaciano da Gaussiana (LoG). Por outro lado, o SURF se baseia no uso de integral de imagens Fast-Hessian, por meio de uma aproximação do *kernel* gaussiano de segunda ordem, para determinar características. Para a determinação do descritor é utilizada a soma das respostas Haar wavelet 2D em diferentes orientações. A versão padrão do SURF cria um descritor de 64 posições, sendo muito eficiente, porém menos preciso.

Segundo Bay et al. (2006) o algoritmo SURF é um detector e descritor de pontos chave invariante a rotação e a escala, que é computacionalmente muito rápido. O detector de descritores SURF é baseado na matriz Hessiana. O determinante da matriz Hessiana é usado para determinar a localização e escala do descritor. Dado um ponto $p = (x, y)$ na imagem I, a matriz Hessiana $H(x, \sigma)$ em x na escala σ é definida como segue:

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \quad (6)$$

onde $L_{xx}(x,\sigma)$ é a convolução da derivada de segunda ordem da Gaussiana $\frac{\partial^2}{\partial x^2} g(\sigma)$ com a imagem I no ponto x, e similarmente a $L_{xy}(x,\sigma)$ e $L_{yy}(x,\sigma)$. A matriz de determinantes Hessianos é escrita como:

$$\det(H_{\text{approx}}) = D_{xx}D_{yy} - (0{,}9D_{xy})^2 \quad (7)$$

Para localizar pontos de interesse sobre escalas, é aplicada uma supressão não máxima em uma vizinhança 3 x 3 x 3. O descritor SURF é extraído em duas etapas: a primeira etapa é a atribuição de uma orientação com base nas informações de uma região circular em torno dos pontos de interesse detectados. A orientação é computada usando respostas Haar-wavelet, nas direções *x* e *y*, que são ponderadas com uma Gaussiana (σ = 3.3s) centrada no ponto de interesse, a fim de aumentar a robustez às deformações geométricas, e respostas Wavelet em direções *dx* horizontal e vertical *dy* que são adicionadas em cada sub-região. Os valores absolutos |*dx*| e |*dy*| são somados a fim de obter informação sobre a polaridade das alterações da intensidade da imagem. Portanto, cada sub-região tem um vetor *v* de descritor de quatro dimensões.

$$V = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \quad (8)$$

Isso resulta em um vetor de descritores para todas as sub-regiões 4 x 4 de tamanho 64.

3.3 HARRIS-SIFT

Mikolajczyk e Schmid (2005) definem que o bom desempenho do algoritmo SIFT em comparação com os outros descritores é notável. Usando pontos fortes e orientação dos gradientes reduz-se o efeito de alterações fotométricas.

O algoritmo HARRIS-SIFT faz a combinação do detector de cantos Harris (HARRIS; STEPHENS, 1988) com o algoritmo SIFT (LOWE, 2004), que determina descritores com alta repetibilidade e boas propriedades de correspondência com um tempo de processamento menor em relação ao da execução do algoritmo SIFT original (AZAD; ASFOUR; DILLMANN, 2009).

O algoritmo detector de cantos HARRIS segundo (MAIA, 2010) representa uma abordagem clássica proposta por Harris e Stephens (HARRIS; STEPHENS, 1988), sendo um aprimoramento da técnica proposta por Moravec. Os autores propuseram usar os autovalores β1 e β2 da matriz Hessiana, para a classificação de pontos da imagem.

$$H_I = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix} \quad (9)$$

Essa matriz representa um tensor contendo as derivadas de segunda ordem em cada pixel, de forma que os autovalores de H_I correspondem às curvaturas principais locais naquele ponto da imagem. A curvatura local contém informação suficiente para que cantos sejam detectados de forma confiável (MAIA, 2010).

- Quando β1 e β2 são grandes, e β1 ~β2, o ponto da imagem representa uma junção;
- Quando β1 >> β2 ou β2 >> β1, o ponto da imagem pertence a uma aresta;
- Finalmente, quando β1 e β2 tendem a zero, a região é considerada plana. Logo, não há pontos de interesse.

Como a computação dos autovalores é computacionalmente custosa, os autores propuseram a adoção de uma métrica alternativa para que a detecção de cantos, na qual uma constante *k* é determinada empiricamente dependendo da aplicação.

$$Mc = \beta_1.\beta_2 - k(\beta_1 + \beta_2)^2 \\ = \det(H_I) - kTr^2(H_I) \quad (10)$$

Esse operador é invariante a rotação e a deformações lineares na *luminância* na imagem. Apesar disso, essa métrica não é invariante a escala, o que dificulta sua pronta utilização em diversos contextos (MAIA, 2010).

Neste trabalho foi implementado o algoritmo HARRIS-SIFT substituindo a fase de detecção realizada originalmente pelo algoritmo SIFT, pelo algoritmo de detecção de cantos HARRIS. No algoritmo SIFT são criadas pirâmides gaussianas e DoG (*Diference of Gaussians*) como alguns dos recursos para detectar pontos chave, neste trabalho tem-se a proposta da substituição destes passos pelo algoritmo HARRIS.

3.4 RANSAC

O algoritmo RANSAC (*RANdom SAmple Consensus*) (FISCHLER; BOLLES, 1981) é um método de estimação robusto projetado para extração dos *inliers*[1] e *outliers*[2] do conjunto de pontos chave. Tem sido muito usado para o reconhecimento de objetos (COLLET et al., 2009) (OKABE; SATO, 2003), pois permite encontrar correspondências geometricamente consistentes para resolver o problema de junção de pares de imagens, mesmo em condições extremas, ou com algum tipo de *outlier*.

Ao contrário das técnicas convencionais que usam grande quantidade de dados para obter uma solução inicial, e em seguida eliminar os *outliers*, o RANSAC usa um conjunto com um número mínimo de pontos para uma primeira estimativa e continua o processo, aumentando o conjunto de pontos de dados consistentes (FISCHLER; BOLLES, 1981).

[1] *inliers*: pontos de dados que se ajustam com um determinado modelo desejado dentro de uma certa tolerância de erro.

[2] *outliers*: pontos de dados que não se ajustam ao modelo correspondente ao objeto desejado, estão fora de uma certa tolerância de erro.

4 METODOLOGIA PARA DETECÇÃO DE OBJETOS

Nesta seção é apresentada a metodologia aplicada neste trabalho para se realizar a detecção dos objetos em imagens. São definidas as imagens que foram utilizadas nos experimentos dos algoritmos: imagens objeto (*templates*), que são as imagens usadas como alvo da detecção dos algoritmos; e imagens da cena (imagens de entrada), que contem o objeto alvo da detecção.

4.1 Algoritmos Utilizados

Neste trabalho, os algoritmos utilizados foram o SIFT (LOWE, 2004) que é detector e descritor razoavelmente invariante a mudança de iluminação, ruído, rotação, escala e pequenas mudanças no ponto de vista. O algoritmo SURF (BAY et al., 2006) que é um detector e descritor de pontos chave invariante a escala e parcialmente a rotação, que é computacionalmente muito rápido. A seção 3 descreve com detalhes os estágios do algoritmo SIFT e SURF. E por fim, o algoritmo HARRIS-SIFT, que é uma combinação do detector de cantos HARRIS (HARRIS; STEPHENS, 1988) com o descritor do algoritmo SIFT (LOWE, 2004).

O algoritmo HARRIS-SIFT foi implementado, neste trabalho, utilizando detector de cantos HARRIS adicionando o descritor de pontos chave do algoritmo SIFT. O algoritmo está descrito na seção 4.2.

4.2 Implementação do algoritmo HARRIS-SIFT

Para a implementação do detector do algoritmo HARRIS-SIFT, foi utilizado o detector de canto HARRIS (HARRIS; STEPHENS, 1988) combinado com a segunda fase do detector SIFT.

O algoritmo detector de cantos HARRIS, em sua execução natural não retorna, junto com os valores das coordenadas dos pontos, o valor do ângulo de orientação do gradiente do ponto chave, que é um valor importante para a fase da descrição do algoritmo SIFT. Sendo assim, foi utilizado o código original do algoritmo SIFT que realiza o cálculo do ângulo de orientação modificado para se adaptar ao algoritmo HARRIS.

5 EXPERIMENTOS E RESULTADOS

Os experimentos apresentados nesta seção fazem o uso de imagens de dois objetos reais, e uma imagem com degradações (redução de escala, rotação, borramento, escurecimento e adição de ruído). É comparado o desempenho das diferentes combinações dos algoritmos citados na Seção 3. As combinações utilizadas foram realizadas alternando os detectores SIFT, SURF e HARRIS e descritores SIFT e

SURF. O computador usado no experimento possui um processador Intel Core i3-3220 3,30 GHz com 4 GB de memória RAM, e com um sistema operacional Windows 7 de 64 bits.

Foi utilizada a Biblioteca de Visão Computacional OpenCV-2.4.8 na linguagem de programação C++ para a implementação deste trabalho.

Os parâmetros observados no experimento, para o processamento das imagens do *template* (objeto) e imagem de entrada (cena), são:

- TP – Tempo de processamento – tempo necessário para processar as duas imagens (Detecção, Descrição, Correspondência e RANSAC – em milisegundos). Foram realizados, para cada exemplo, cinco testes e desses foram descartados o maior e menor tempo. Com os três tempos restantes, foi calculada uma média aritmética para corrigir possíveis variações a cada ocorrência de execução;
- NIn – Número de *inliers* – consistem nos pontos chaves com correspondência validadas pelo algoritmo RANSAC.

Foram utilizados três conjuntos de imagens, sendo dois deles de imagens com objetos reais, e o terceiro com uma imagem adicionada degradações.

O primeiro conjunto (Conjunto 1) é apresentado na Figura 3.

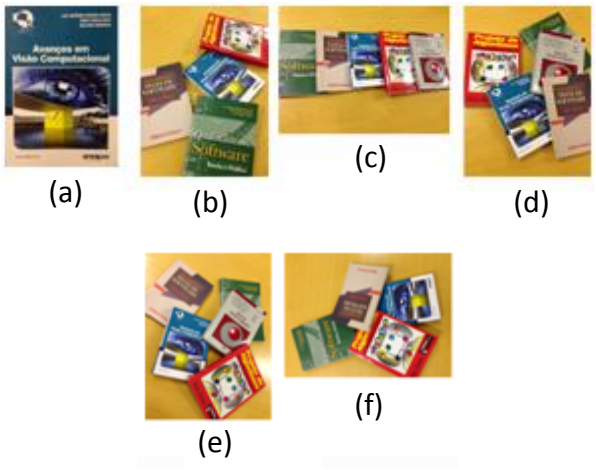

Figura 3. Imagens do Conjunto 1 utilizadas no experimento. Em (a) tem-se a imagem objeto real, e nas demais as imagens representando as cenas de busca.

O Conjunto 2 é composto pela imagem da Lena (muito utilizada em aplicações de processamento de imagens) apresentada na Figura 4. Na Figura 4 (a) tem-se a imagem da Lena original (imagem objeto). Em (b) tem-se a imagem com redução de escala, em (c) a imagem com rotação, em (d) a imagem com borramento, em (e) tem-se a imagem com escurecimento, em (f) a imagem com adição de ruído e em (g) a imagem com todas as degradações juntas.

Figura 4. Imagens do Conjunto 2.

O terceiro conjunto (Conjunto 3), é apresentado na Figura 5.

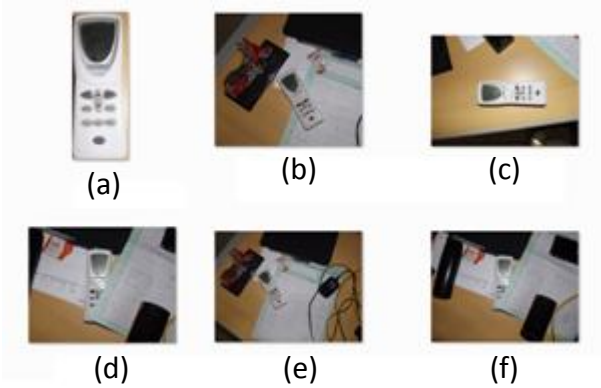

Figura 5. Imagens do Conjunto 3. Em (a) tem-se a imagem objeto, e nas demais as imagens representando as cenas de busca.

Na Figura 6 é apresentado o resultado de um experimento com a combinação SIFT-SIFT. Na Tabela 1, essa foi a única combinação que retornou resultados expressivos, as demais combinações não apresentaram resultados esperados e satisfatórios para o experimento.

Figura 6. Resultado de um dos testes com o detector-descritor SIFT-SIFT para o Conjunto 1 de imagens.

Na Figura 7 pode-se observar uma resposta negativa em que a combinação SURF-SURF retornou no teste realizado com uma imagem cena da Figura 3 (f).

Na Tabela 1 são apresentados os tempos de processamento (em milisegundos) de cada imagem de teste do Conjunto 1 para cada combinação proposta entre detector-descritor. Os valores que representam resultados que não obtiveram respostas satisfatórias estão representados por um * (asterisco).

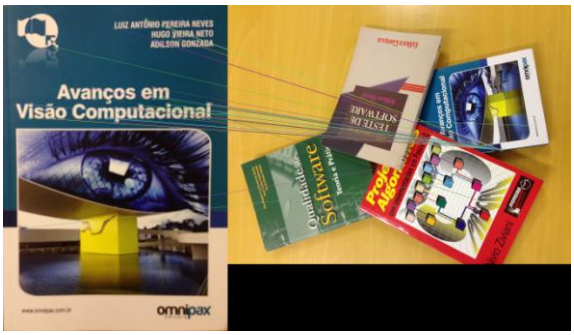

Figura 7. Resultados não satisfatórios retornados pelo algoritmo SURF-SURF no teste para a imagem (f) da Figura 3 (Conjunto 1).

Tabela 1. Tempos de processamento (em milisegundos) do Conjunto 1.

	Cena 1	Cena 2	Cena 3	Cena 4	Cena 5
SIFT-SIFT	6537	6334	6428	6552	6412
SIFT-SURF	3744*	3900*	3728*	3869*	3791*
SURF-SIFT	37752*	34554*	35708*	39952*	39312*
SURF-SURF	7067*	6677*	6817*	7161*	7145*
HARRIS-SIFT	305417*	306525*	304372*	306025*	305021*
HARRIS-SURF	303670*	308054*	307364*	307064*	309376*

Na Tabela 2 são apresentados os *inliers* retornados nos testes realizados com as imagens de entrada do Conjunto 1.

Tabela 2. Quantidade de *inliers* obtida com a experimentação do Conjunto 1.

	Cena 1	Cena 2	Cena 3	Cena 4	Cena 5
SIFT-SIFT	288	148	286	256	238
SIFT-SURF	29*	26*	20*	33*	30*
SURF-SIFT	38*	45*	41*	60*	101*
SURF-SURF	19*	15*	27*	21*	43*
HARRIS-SIFT	30*	43*	23*	35*	35*
HARRIS-SURF	29*	37*	25*	32*	27*

Para o Conjunto 2, a combinação SIFT-SURF não obteve resultados satisfatórios para as imagens degradadas com rotação de 90º negativo, ruídos gaussianos 10%, e na imagem que combina todas as degradações.

Na Tabela 3 são mostrados os resultados dos tempos de processamento (em milisegundos) dos testes realizados com as imagens do Conjunto 2.

Tabela 3. Tempos de processamento (em milisegundos) do Conjunto 2.

	Escala (Reduzida 50%)	Rotação (-90°)	Borramento Gaussiano 2	Iluminação	Ruído Gaussiano 10%	Todos
SIFT-SIFT	593	842	1107	796	905	718
SIFT-SURF	373	780*	593	499	655*	484*
SURF-SIFT	3650	5222*	3978	4212	9469	3760*
SURF-SURF	718	1061*	796	749	9484	749*
HARRIS-SIFT	21481*	33727	32557	33010	38657	23572*
HARRIS-SURF	21247*	33587*	31964*	32856	38594*	23275*

Na Figura 8 observa-se a aplicação do algoritmo HARRIS-SIFT, que retornou o maior número de *inliers*, com um NIn = 412 (Tabela 4), sendo este o maior número entre todos os algoritmos. Esse resultado foi obtido no teste realizado com a imagem da Lena original Figura 4 (a) como imagem objeto, sendo detectada na imagem Figura 4 (c), rotacionada em 90° negativo. Outro algoritmo que também apresentou resultados satisfatórios para esse mesmo conjunto de imagens teste foi o algoritmo SIFT-SIFT, que apresentou um número de *inliers* menor do que o algoritmo HARRIS-SIFT, contudo apresentando um tempo de processamento menor, TP = 842 ms. Já

algoritmo HARRIS-SIFT apresentou um TP = 33.727 ms, que é bem menor se comparado ao do SIFT-SIFT. A Tabela 4 mostra o tempo de processamento (em milisegundos) dos algoritmos para o conjunto 2.

Tabela 4. Quantidade de *inliers* do Conjunto 2.

	Escala (Reduzida 50%)	Rotação(-90°)	Borramento Gaussiano 2	Iluminação	Ruído Gaussiano 10%	Todos
SIFT-SIFT	81	221	92	83	98	44
SIFT-SURF	72	17*	62	75	14*	27*
SURF-SIFT	115	19*	174	189	263	25*
SURF-SURF	113	19*	171	191	263	32*
HARRIS-SIFT	21*	412	41	251	160	11*
HARRIS-SURF	16*	9*	17*	165	9*	11*

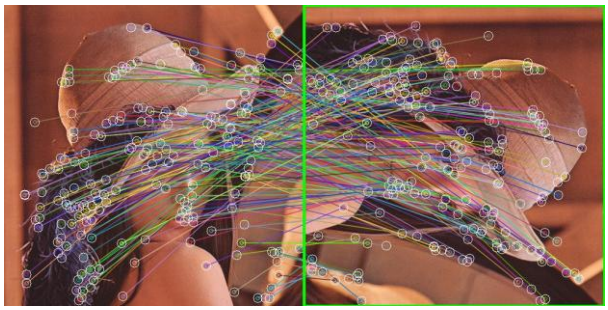

Figura 8. Resultado do algoritmo HARRIS-SIFT que obteve o maior número de *inliers,* com um NIn = 412.

O único algoritmo que retornou resultado satisfatório no processamento da Imagem (g) da Figura 4, com todas as degradações foi o algoritmo SIFT-SIFT, que apresentou um tempo de processamento de TP = 718 ms e número de *inliers* NIn = 44. Essa é a maior quantidade de *inliers* se comparado com os outros algoritmos para a mesma imagem de teste. A Figura 9 apresenta o resultado do SIFT-SIFT para imagem (g) da Figura 4.

Figura 9. Resultado do algoritmo SIFT-SIFT que obteve o maior número de *inliers,* com um NIn = 44, o melhor e único resultado satisfatório entre todas as combinações dos algoritmos para a imagem (g) da Figura 4.

A Figura 10 mostra o resultado com a combinação que retornou a melhor razão entre tempo de processamento e *inliers*, TP/NIn. Essa combinação foi a do SIFT-SIFT com a imagem (c) da Figura 4. Pode-se observar na Tabela 4 que o algoritmo SIFT-SIFT obteve 221 *inliers* com um TP = 842 ms, tendo uma razão de TP/NIn = 3,809 (milissegundo por *inlier*).

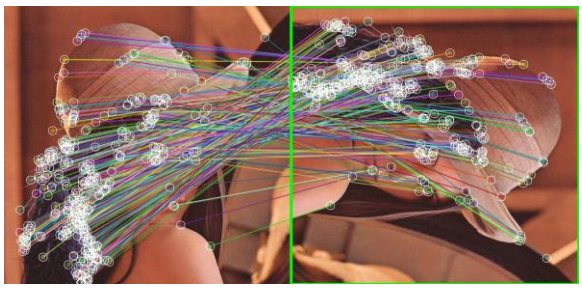

Figura 10. Resultado do algoritmo SIFT-SIFT que obteve a melhor razão entre TP e NIn para a imagem da Lena.

Para o Conjunto 3, o algoritmo que obteve os melhores resultados foi o algoritmo SIFT-SIFT, que das cinco cenas, obteve sucesso na detecção do objeto em três delas. O algoritmo HARRIS-SIFT foi satisfatório na detecção do objeto no processamento da cena 2, onde pode-se observar na Figura 11 e na Tabela 5 de tempos de processamento que o mesmo possuiu um tempo de processamento TP = 18.767 ms e *inliers* NIn = 40 (Tabela 6), o melhor resultado entre todas as combinações dos algoritmos.

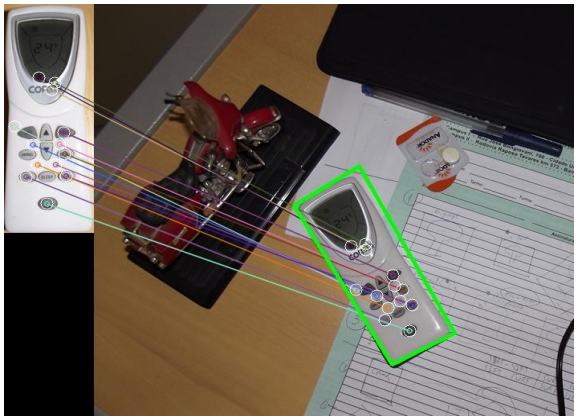

Figura 11. Resultado do algoritmo HARRIS-SIFT, que obteve NIn = 40 e TP = 18.767 ms.

Tabela 5. Tempos de processamento do Conjunto 3.

	Cena 1	Cena 2	Cena 3	Cena 4	Cena 5
SIFT-SIFT	827	593	577*	920	733*
SIFT-SURF	484*	530*	296*	640*	406*
SURF-SIFT	3541*	2137*	2246*	4649*	2902*
SURF-SURF	718*	655*	468*	905*	593
HARRIS-SIFT	22526*	18767	13057*	27263*	20732*
HARRIS-SURF	22355*	18626*	12954*	26957*	20759*

O algoritmo SIFT-SIFT apresentou melhores resultados em comparação aos outros algoritmos para as imagens de entrada do Conjunto 3. Na Figura 12 é apresentado o resultado de um teste do algoritmo SIFT-SIFT.

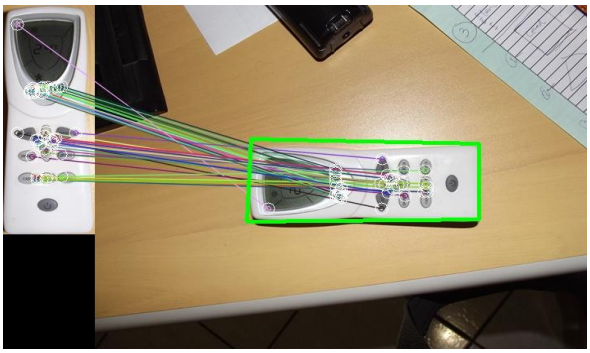

Figura 12. Resultado do processamento do algoritmo SIFT-SIFT.

Na Tabela 6 são apresentados as quantidades de *inliers* para os testes realizados com o Conjunto 3.

Tabela 6. Quantidade de *inliers* do Conjunto 3.

	Cena 1	Cena 2	Cena 3	Cena 4	Cena 5
SIFT-SIFT	16	12	10*	16	10*
SIFT-SURF	11*	12*	10*	9*	10*
SURF-SIFT	16*	15*	13*	8*	12*
SURF-SURF	12*	14*	9*	15*	9*
HARRIS-SIFT	13*	40	21*	13*	21*
HARRIS-SURF	7*	8*	10*	9*	8*

Para imagens com muitos detalhes como no caso do teste do Conjunto 1, com imagens que apresentam muitas informações, o algoritmo SIFT-SIFT obteve melhores resultados que as outras combinações. O algoritmo HARRIS não apresentou resultados satisfatórios, mesmo com a presença de cantos e retas, características abrangidas pelo algoritmo HARRIS, nas imagens dos testes do Conjunto 1. Nos testes realizados com as imagens do Conjunto 3, o algoritmo SIFT-SIFT obteve melhores resultados, com um teste positivo para o algoritmo HARRIS-SIFT, que apresentou vários *inliers* tanto no símbolo da marca do controle remoto quanto nos botões (Figuras 11 e 12).

Os testes realizados com as imagens do Conjunto 2, é um teste diferente dos outros dois por não usar um objeto real imageado. Nesse teste foi introduzido na imagem original, transformações a fim de alterar as propriedades visuais da imagem, tornando-a uma imagem diferente, mas com as mesmas características.

Nos testes realizados com o Conjunto 2, os algoritmos SIFT e SURF obtiveram resultados satisfatórios. O algoritmo SIFT-SIFT obteve os melhores resultados em todos os testes realizados com todas as imagens transformadas. O algoritmo HARRIS, em sua combinação com o descritor SIFT, obteve resultados satisfatórios retornando grandes quantidades de *inliers*, demonstrando ser um excelente detector de pontos. Em conjunto com o descritor SIFT, o detector HARRIS mostrou ser um descritor capaz de ser utilizado para detecção de objetos. Por detectar vários pontos, esse algoritmo seria bem útil em imagens que apresentem uma grande quantidade de detalhes.

6 CONSIDERAÇÕES FINAIS

A utilização de métodos baseados em descritores locais consiste em três etapas: detecção de pontos chave, cálculo de descritores e obtenção de correspondências. A primeira etapa é uma tarefa essencial por resultar em conjuntos de partes notáveis da imagem (os pontos chave), e que são a base da representação dos objetos utilizando nessa abordagem. Os algoritmos existentes ainda possuem dificuldades com certos padrões de imagens e a não detecção de pontos corretos foi apresentado em grande parte dos testes dos algoritmos.

Diversos algoritmos para detecção de pontos chave têm sido trabalhados e estudados nas áreas tanto de Visão Computacional quanto da Matemática. Diversos descritores de características têm sido criados para os processos de correspondências de imagens, recuperação de imagens e detecção de objetos, entre

outros. Embora esses descritores sejam bastante eficientes, a quantidade de características detectadas e o tamanho dos vetores de características que eles descrevem elevam o seu custo computacional.

Por fim, sugere-se o desenvolvimento dos seguintes temas como trabalhos futuros, os quais são considerados interessantes e produtivos:

- Implementação de outros algoritmos para uma maior abrangência nas qualidades e especialidades dos algoritmos;
- Testes com outros algoritmos de detecção de pontos chave para uma ampliação do estudo e conhecimento das funcionalidades e especialidades dos algoritmos existentes.

REFERÊNCIAS

AZAD, P.; ASFOUR, T.; DILLMANN, R. **Combining Harris interest points and the sift descriptor for fast scale-invariant object recognition**. Germany: Institute for Anthropomatics, University of Karlsruhe, 2009.

BAY, H. et al. Speeded up robust features. **Lecture Notes in Computer Science, Chapter Computer Vision – ECCV 2006**, p. 404-417, 2006.

COLLET, A. et al. Object recognition and full pose registration from a single image for robotic manipulation. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, ICRA'09, **Proceedings…** 2009. p. 48–55.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, v. 24, p. 381–395, 1981.

HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: ALVEY VISION CONFERENCE, 4. **Proceedings...** Manchester, UK, 1988. p. 147-151.

LINDEBERG, T. Feature detection with automatic scale selection. **IJCV**, v.30, n.2, p. 79-116, 1998.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, v. 60, n. 2, p. 91–110, 2004.

LOWE, D. G. Object recognition from local scale-invariant features. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION. **Proceedings...** Corfu, Greece, 1999. p. 1150-1157.

MAIA, J. G. R. **Detecção e reconhecimento de objetos usando descritores locais.** [s.l.]: Departamento de Computação, Centro de Ciências, Universidade Federal do Ceará, 2010.

MIKOLAJCZYK, K.; SCHMID, C. A Performance Evaluation of Local Descriptors. **IEEE Trans. on Pattern Analysis and Machine Intelligence**, v. 27, n. 10, p. 1615-1630, 2005.

OKABE, T.; SATO, Y. Object recognition based on photometric alignment using RANSAC. In: COMPUTER SOCIETY CONF. ON COMPUTER VISION AND PATTERN RECOGNITION, **Proceedings...** 2003. p.221-228.

RACHID, C. L.; PEREIRA, A. A. S. **Algoritmos de busca SIFT e SURF no uso de dispositivos móveis**. Minas Gerais: Ciência da

Computação, Universidade Presidente Antônio Carlos, 2009.

SILVA, F. A. **Georreferenciamento automático de placas de sinalização com imagens obtidas com um sistema móvel de mapeamento**. 2012. Tese (Doutorado) - Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos.

SILVA, F. A. et al. Evaluation of keypoint detectors and descriptors. In: WORKSHOP DE VISÃO COMPUTACIONAL - WVC 2013, 9., Rio de Janeiro. **Anais…** 2013.

TREIBER, M. **An Introduction to object recognition:** selected algorithms for a wide variety of application. London: Springer-Verlag, 2010. 201p.

WITKIN, A. P. Scale-space filtering. In: THE INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE OF BIOLOGICAL SYSTEMS. **Proceedings…** v.1, n.1, p.1019-1022, 1983.