SIMPLES: UM DESCRITOR DE CARACTERÍSTICAS LOCAIS RÁPIDO E SIMPLES

MARCOS CESAR VOLTOLINI, HAE YONG KIM

Dept. Eng. Sistemas Eletrônicos, Escola Politécnica, USP
Av. Prof. Luciano Gualberto, trav. 3, 158, CEP 05508-010, São Paulo, SP, Brasil.
E-mails: marcos.voltolini@usp.br, hae@lps.usp.br

Abstract— Lowe, in the well-known paper that introduced SIFT, raised the possibility of using the local image intensities around the keypoints as the local image features. The distance between two sets of image intensities can be measured using normalized correlation. However, he discarded this idea and ended up using histogram of gradients as the features. We decided to test experimentally the abandoned idea and obtained unexpectedly good results. We named "SIMPLES" the descriptor of features obtained using this idea. Our results suggest that the accuracy of SIMPLES is lower than SIFT but similar to SURF. On the other hand, SIMPLES is much faster to compute than both descriptors. In our experiments, it was 90 times faster than SIFT and 6 times faster than SURF.

Keywords— Computer Vision, SIFT, SURF, Local Features

Resumo— Lowe, no conhecido artigo que introduziu o SIFT, levantou a possibilidade de usar as intensidades na região em volta dos pontos-chave como descritores de características locais. A distância entre dois conjuntos de intensidades de imagens pode ser medida usando correlação normalizada. Entretanto, ele descartou a ideia e acabou utilizando histograma de gradientes como descritores. Decidimos testar experimentalmente a ideia abandonada e acabamos obtendo resultados inesperadamente bons. Nomeamos de "SIMPLES" o descritor de características obtido com esta ideia. Nossos resultados sugerem que a acuracidade do SIMPLES é menor que a do SIFT mas comparável a do SURF. Entretanto, SIMPLES pode ser calculado muito mais rapidamente que ambos descritores. Nos nossos experimentos, foi 90 vezes mais rápido que SIFT e 6 vezes mais rápido que SURF.

Palavras-chave— Visão Computacional, SIFT, SURF, Características Locais

Introdução

Descritores de características locais computados no entorno de pontos-chave têm sido aplicados com sucesso em diferentes áreas, como casamento de imagens apresentado por Bay et al. (2006), detecção e reconhecimento de objetos por Fergus et al. (2003) e Csurka et al. (2004) e estabilização de vídeo por Pinto & Anurenjan (2011). Neste artigo, analisamos experimentalmente uma ideia que Lowe levantou em seu famoso artigo, mas que não deu continuidade. Lowe (2004) escreveu: "*One obvious approach would be to sample the local image intensities around the keypoint at the appropriate scale, and to match these using a normalized correlation measure. However, simple correlation of image patches is highly sensitive to changes that cause misregistration of samples, such as affine of 3D viewpoint change or non-rigid deformations.*" Lowe acabou usando histograma de gradientes como descritor de características locais.

Implementamos esta ideia abandonada e a testamos experimentalmente. Surpreendentemente, percebemos que esta ideia não é ruim: é rápida computacionalmente e apresenta acuracidade comparável a outros descritores de características locais em determinados casos. Apesar de não mostrarmos neste artigo uma nova ideia, acreditamos que a comunidade de visão computacional possa estar interessada em saber que um descritor de características locais baseado na correlação normalizada de regiões pode apresentar uma boa acuracidade de casamento e ao mesmo tempo ser computacionalmente rápido.

Trabalhos relacionados

Os descritores SIFT criado por Lowe (2004) e SURF criado Bay et al. (2006) são quase padrões de referência para descritores de características locais. SIFT apresenta alta acuracidade mas tempo de processamento um tanto longo. SURF apresenta acuracidade ligeiramente inferior mas é rápido. Estes trabalhos são até hoje utilizados como comparativo para novos detectores de pontos-chave e descritores de características locais. SIFT utiliza como base para descrição de pontos-chave o histograma de gradientes orientados na região em torno do ponto. SURF procura obter um descritor similar mas de forma computacionalmente mais rápida. Para acelerar os cálculos, SURF utiliza imagem integral para calcular respostas a wavelet de Haar em sentidos horizontal e vertical. Estas respostas são amostradas numa grade regular e as suas somatórias (com e sem sinal) servem como descritores. Este processo leva a resultados mais rápidos porém menos acurados.

Descritor de características locais "SIMPLES"

Vamos supor que algum detector de pontos-chave ache a localização, escala e orientação de cada ponto-chave de uma imagem. O descritor de características

locais deve descrever a região em torno de cada ponto-chave de uma forma invariante a distorções como mudança de iluminação e pequenas variações de ponto de vista.

O descritor de características locais que denominamos de "SIMPLES" é realmente simples. Ele consiste em amostrar pontos pré-determinados em torno do ponto-chave. Por exemplo, a Figura 1 mostra 127 pontos de amostragem que geram um vector com 127 características. As posições dos pontos precisam ser escaladas e rotacionadas de acordo com a escala e a orientação de cada ponto-chave.

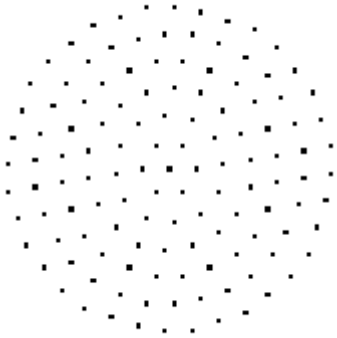

Figura 1: Pontos de amostragem

Duas imagens do mesmo objeto podem apresentar uma grande diferença nos níveis de cinza se forem fotografadas sob diferente iluminações. Com isso, a distância euclidiana entre os vetores de características obtidos por amostragem de níveis de cinza pode não medir devidamente a diferença visual entre duas regiões. A correlação normalizada poderia ser utilizada para computar a similaridade entre dois vetores de características, pois é invariante por mudança de brilho e contraste como mostrado por Lewis (1995). Entretanto, o cálculo de uma correlação normalizada para cada comparação é computacionalmente proibitivo.

Assim, resolvemos normalizar o vetor de características para torná-lo invariante a mudanças de brilho e contraste. Para isso, dado um vetor de características $u = (u_1,...,u_n)$, subtraímos o valor da média μ de cada elemento u_i e dividimos pelo desvio padrão σ, obtendo o vetor de características normalizado $\hat{u} = (\hat{u}_1,...,\hat{u}_n)$:

$$\hat{u}_i = \frac{u_i - \mu}{\sigma},\ 1 \le i \le n \qquad (1)$$

Evidentemente, não podemos normalizar o vetor de características de uma região com níveis de cinza constantes, porque isto causaria uma divisão por zero. Entretanto, nenhum detector de pontos-chave escolherá uma região com níveis de cinza constante como um ponto-chave, pois não há como escolher um pixel especial numa região constante.

Dois vetores normalizados podem ser comparados diretamente utilizando distância euclidiana, sendo esta distância invariante a mudanças de brilho e contraste e consequentemente robusta a variações de iluminação. Estes 127 números devem descrever a vizinhança de um ponto-chave.

Para tornar o SIMPLES robusto a pequenas imprecisões na localização, rotação e escala dos pontos-chave e a pequenas mudanças do ponto de vista 3D, em vez de pegar amostras em pixels definidos (como na Figura 1) podemos tirar uma média aritmética local em torno dos pontos amostrados. Para isso, temos três possíveis estratégias:

Estratégia 1: Filtrar a região em torno de cada ponto-chave utilizando um filtro passa-baixas gaussiano com desvio-padrão $\sigma = ks$, onde s é o tamanho do ponto-chave e k é uma constante. Em seguida, os 127 pontos desejados são amostrados. Esta é a estratégia testada por nós utilizando $k = 0.1$

Estratégia 2: Filtrar a imagem utilizando filtros passa-baixas gaussianos. O tamanho do núcleo deve variar de acordo com a escala do ponto-chave. Para acelerar o processamento, podemos filtrar previamente a imagem utilizando um conjunto de diferentes núcleos gaussianos, obtendo um volume espacial 3D de espaço de escalas, onde cada camada corresponde a imagem original filtrada utilizando um certo núcleo gaussiano. Então, um simples acesso ao determinado voxel no volume corresponderá a região filtrada com o núcleo gaussiano seguido pela amostragem. Esta operação pode ser ainda mais eficiente porque alguns detectores de pontos-chave (como o SIFT) necessitam computar o espaço de escala para uso próprio. Então o SIMPLES não necessitaria construir este volume 3D: poderia usar o volume computado pelo detector de pontos-chave.

Estratégia 3: Ao invés de fazer a amostragem dos pixels, é possível computar explicitamente a média dos níveis de cinza dentro de círculos em torno dos pontos de amostragem, como mostra a Figura 2. Esta solução é computacionalmente mais custosa (e menos acurada) que as anteriores, porém apresenta implementação mais direta e simples.

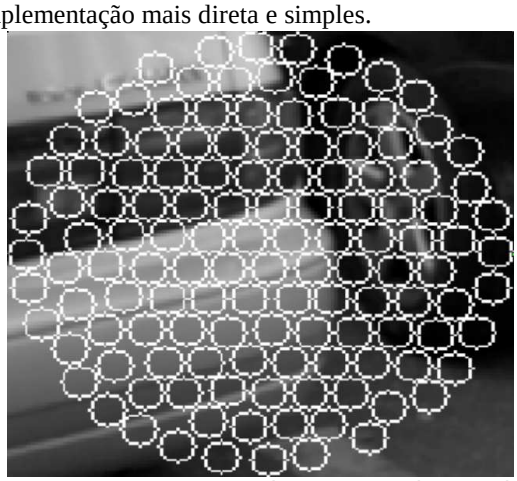

Figura 2: Representação dos círculos onde a média local é calculada

Implementação

Para podermos comparar SIMPLES com SIFT e SURF, implementamos o descritor proposto utilizando OpenCV 2.2. Esta é uma biblioteca de rotinas de visão computacional bastante conhecido na comunidade acadêmica e os algoritmos SIFT e SURF estão implementados nela, tanto a detecção quanto a descrição de pontos-chave.

Resultados Experimentais

Para avaliar o desempenho do novo descritor, utilizamos o método e o conjunto de imagens descrito por Mikolajczyk & Schmid (2005), que aborda justamente o problema de avaliar o desempenho de descritores de características locais. Este método consiste em pegar duas imagens de uma cena fotografadas sob diferentes condições. "Diferentes condições" pode significar, por exemplo, mudança de iluminação, rotação, mudança de escala, borrão de foco, compressão com perdas, etc. Para melhor simular condições reais, estas transformações não são geradas computacionalmente, mas obtidas fotografando a mesma cena diversas vezes sob diferentes condições. Em seguida, os pontos-chave são detectados nas duas imagens (usando detector SIFT ou SURF). Por fim, efetua-se o casamento entre os pontos-chave da primeira e da segunda imagem. Como a transformação é conhecida, é possível classificar cada casamento como "verdadeiro" ou "falso" usando a matriz homográfica dada. Um exemplo de casamento de pontos-chave pode ser visto na Figura 4

Para achar os casamentos entre duas imagens *A* e *B*, o algoritmo calcula, para cada descritor da imagem base *A*, a distância euclidiana com todos os descritores da imagem transformada *B*. Assim, para cada ponto-chave da imagem base *A*, são encontrados os descritores de *B* mais "parecidos". Se todos os descritores de *A* formassem casamento com o descritor de *B* mais semelhante, haveria um grande número de falsos casamentos. Assim, para diminuir o número de falsos casamentos, Lowe (2004) sugeriu utilizar a seguinte a regra: para que um descritor de *A* forme um casamento com o descritor de menor distância em *B*, essa menor distância deve ser no mínimo 0,8 vezes menor (20%) que a segunda menor distância. Variando o parâmetro 0,8 é possível alterar o número de casamentos possíveis, sendo assim permitido aumentar a taxa de verdadeiros positivos com o preço de também se aumentar a taxa de falsos positivos. Para gerar as curvas de desempenho, este parâmetro é variado de 0 até 1 em passos incrementais de 0,01. Note que os casamentos de *A* para *B* são diferentes dos casamentos de *B* para *A*.

O desempenho é avaliado através da curva "total de casamentos possíveis" versus "taxa de casamentos positivos verdadeiros". Este gráfico é o mesmo utilizado por Mikolajczyk & Schmid (2005).

O banco de dados de imagens apresentado por Mikolajczyk & Schmid (2005) consiste em 8 classes de imagens distorcidas:

1. Bikes – borrão de foco.
2. Trees – borrão de foco.
3. Graf – mudança do ponto de vista.
4. Wall – mudança do ponto de vista.
5. Bark – zoom e rotação.
6. Boat – zoom e rotação.
7. Leuven – mudança na iluminação.
8. UBC – compressão JPEG.

Para não tornar o artigo demasiado longo, mostraremos os resultados experimentais de apenas quatro classes de imagens: Bikes, Graf, Boat e Leuven (Figura 3). No caso do borrão de foco, testamos apenas a classe Bikes, pois as imagens de Trees foram tiradas com as folhas balançando ao vento e assim não apresenta pontos-chave confiáveis (quaisquer sejam o detector e o descritor usados). Os resultados de outros casos não mostrados neste artigo são similares aos mostrados.

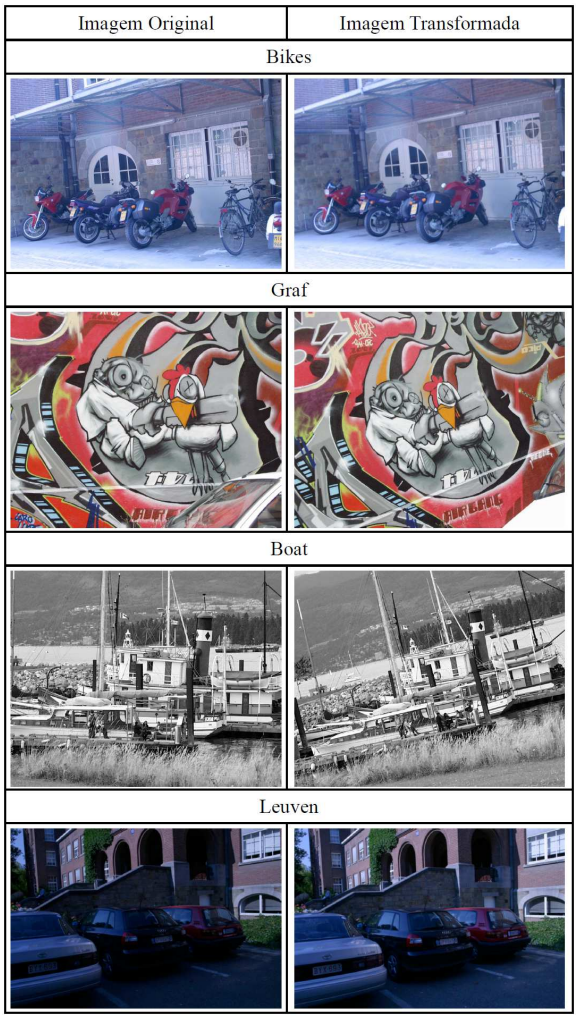

Figura 3: Classes de imagens utilizadas para os testes

Para avaliar o desempenho do descritor SIMPLES, traçamos sua curva de desempenho juntamente com as curvas do SIFT e do SURF-128

(versão do SURF que possuí um descritor de 128 dimensões). Utilizamos SURF-128 em vez de SURF-64 habitual para que todos os descritores tenham dimensões aproximadamente iguais (128 no SIFT e SURF e 127 no SIMPLES). Os pontos-chave foram detectados utilizando SIFT e SURF. Os pontos-chave que se encontravam no limite da imagem (perto da borda) foram removidos. Utilizando o detector de pontos-chaves de SIFT, o descritor SIMPLES apresenta resultados comparáveis ao descritor SURF mais inferiores ao descritor SIFT (Figura 5). Utilizando o detector SURF, o descritor SIMPLES continuou a apresentar resultados similares aos testes anteriores (Figura 6).

Uma das características do descritor proposto é a sua simplicidade que leva a um algoritmo computacionalmente eficiente. O SIMPLES não necessita rotacionar nem mudar a escala da região em torno do ponto-chave, como é feito na extração dos descritores SURF e no SIFT. Basta copiar pixel a pixel a região, aplicar o filtro gaussiano e amostrar os pontos. Com isso, consegue-se um tempo de extração consideravelmente menor que os atuais descritores (Tabela 1): 90 vezes menor que SIFT e 6 vezes menor que SURF. O tempo de SIMPLES da Tabela 1 utiliza a implementação "estratégia 1". Porém, conforme já discutimos, a "estratégia 2" deve diminuir ainda mais (e de forma substancial) o tempo de processamento do SIMPLES. Neste caso, não haveria mais a necessidade de copiar a região em torno do ponto-chave nem aplicar o filtro gaussiano. Bastaria acessar os elementos do volume 3D pré-calculado. Os testes foram feitos utilizando um processador Intel Core i7 – 2.6 GHz. As implementações do SIFT e SURF são as do OpenCV 2.2, usando parâmetros padrões.

Tabela 1. Tempo de extração de 20000 descritores, utilizando a estratégia 1. O tempo de processamento de SIMPLES deve diminuir substancialmente utilizando a estratégia 2.

SIFT (s)	SURF (s)	SIMPLES (s)
29.04	2.08	0.31

Conclusões

A acuracidade do novo descritor é comparável a do descritor SURF e em alguns casos até mesmo superior a este. O descritor proposto tem como grande vantagem sua simplicidade, deixando o processo computacionalmente mais rápido. Isto pode ser percebido pelo menor tempo de processamento. Atualmente o mercado de aplicações embarcadas ao consumidor final está bastante aquecido. Este fato pode ser percebida de forma clara no mercado de celulares inteligentes. Todos estes celulares já apresentam como padrão uma câmera de boa qualidade, logo, aplicações envolvendo visão computacional serão cada vez mais comuns. Porém celulares são alimentandos por baterias e consequentemente exigem aplicações eficientes computacionalmente. Neste cenário acreditamos que o SIMPLES seria bastante útil, pois sua implementação simples juntamente com sua eficiência o tornam amigável a bateria do celular.

Referências Bibliográficas

Bay, H., Tuytelaars, T. & Gool, L. Van, 2006. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, 3951, pp.404–417.

Csurka, G. et al., 2004. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision ECCV*. Citeseer, p. 22.

Fergus, R., Perona, P. & Zisserman, A., 2003. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Published by the IEEE Computer Society, pp. II–264–II–271 vol.2.

Lewis, J., 1995. Fast normalized cross-correlation. *Vision interface*, 10(1), pp.120–123.

Lowe, D.G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), pp.91–110.

Mikolajczyk, K. & Schmid, C., 2005. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10), pp.1615–1630.

Pinto, B. & Anurenjan, P., 2011. Video stabilization using Speeded Up Robust Features. In *Communications and Signal Processing ICCSP 2011 International Conference on*. IEEE, pp. 527–531.

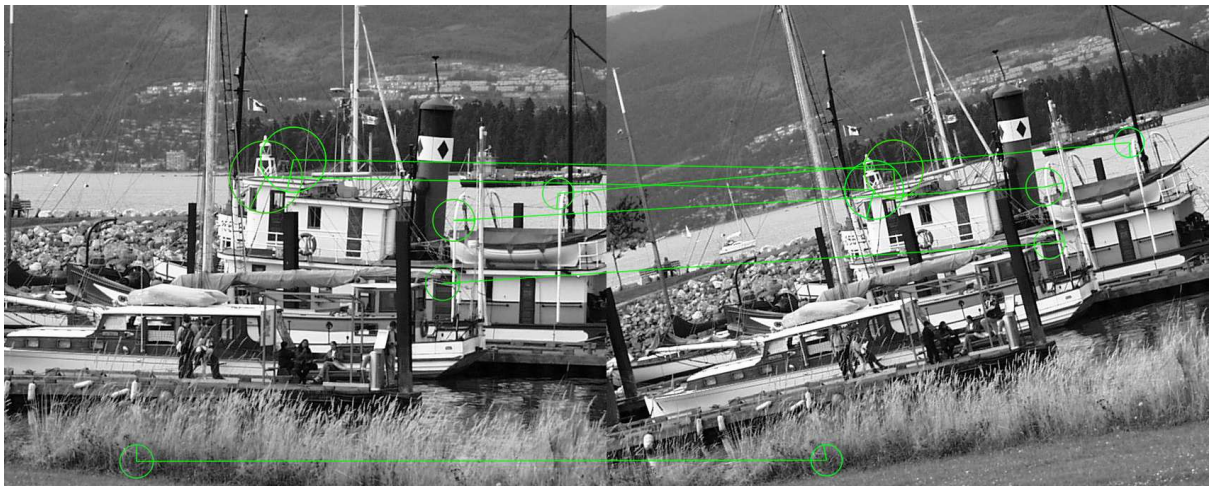

Figura 4: Exemplo de casamentos de pontos-chave

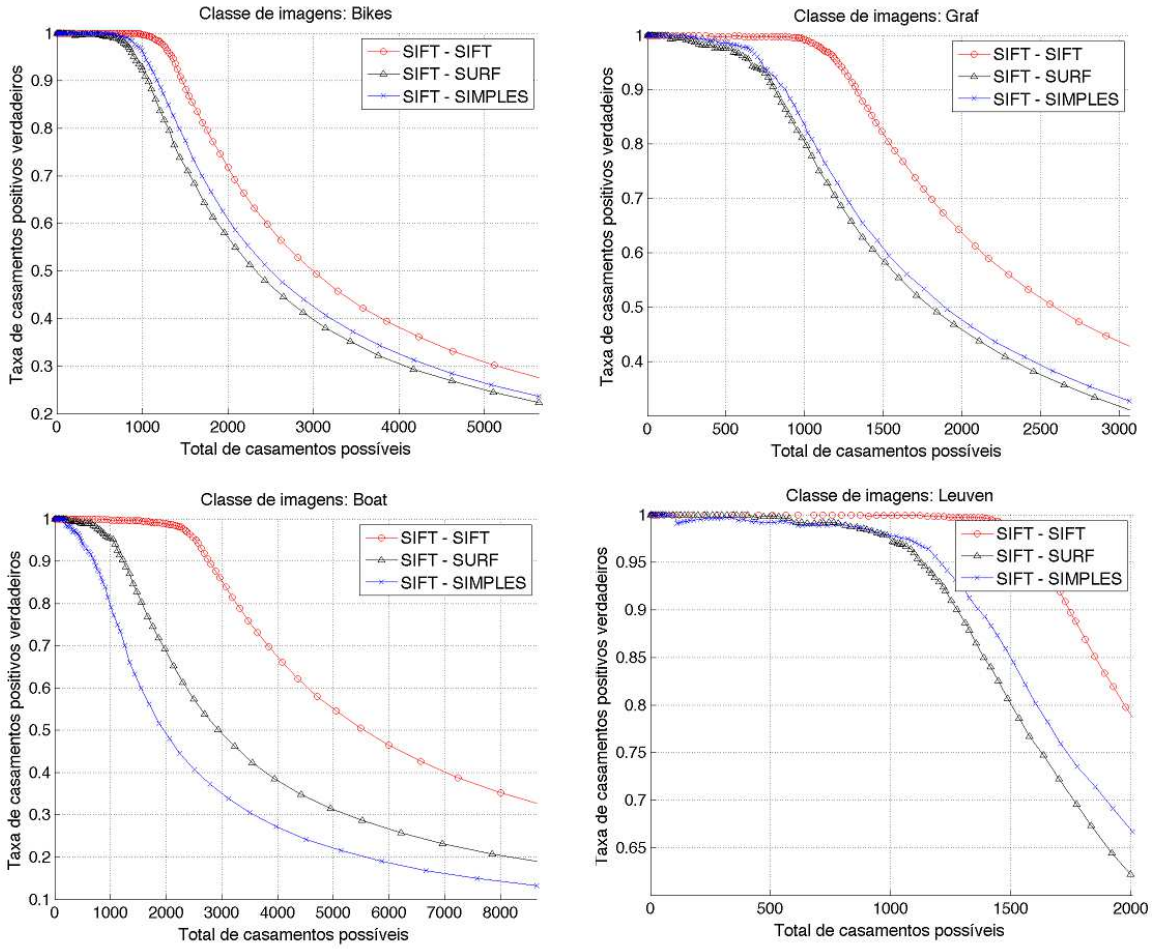

Figura 5: Curvas de desempenho para o os descritores SIFT, SURF-128 e SIMPLES sendo os pontos-chave detectados com o SIFT

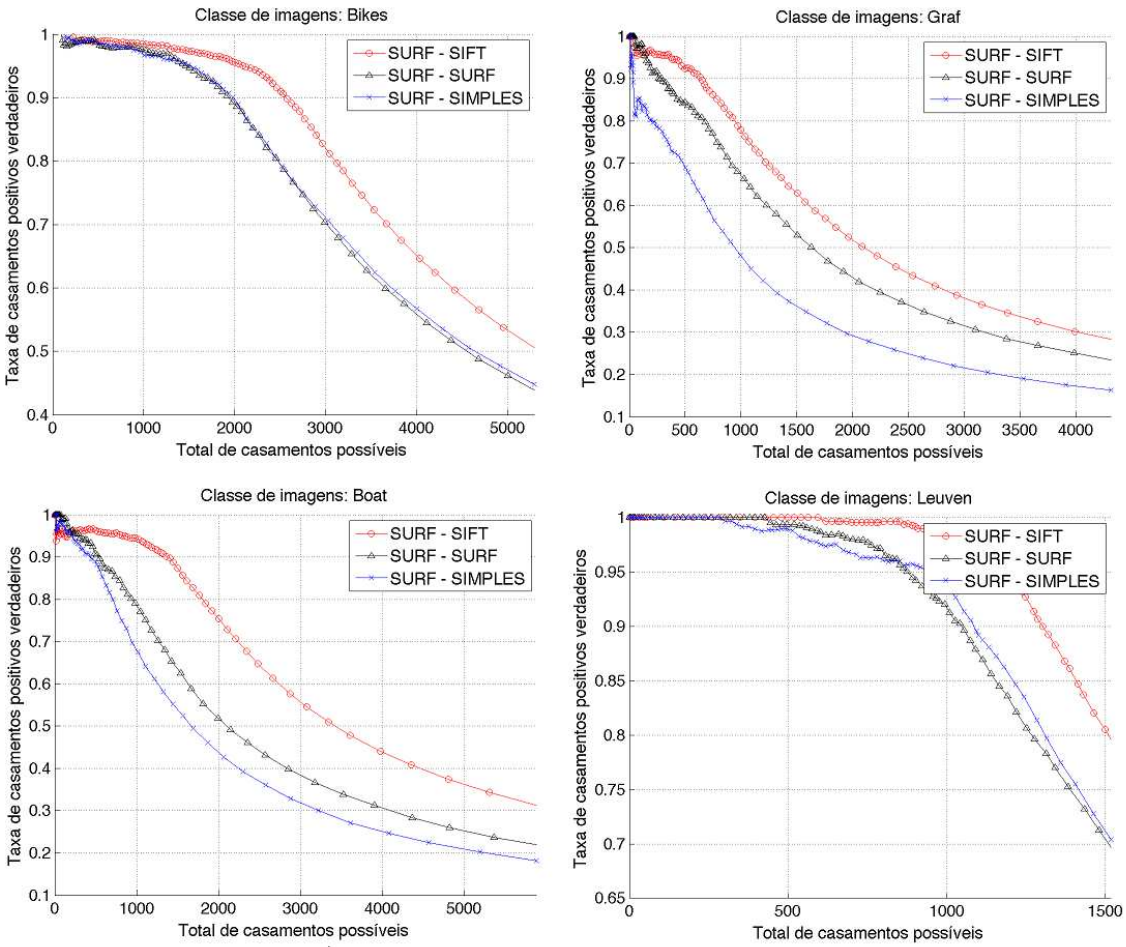

Figura 6: Curvas de desempenho para o os descritores SIFT, SURF-128 e SIMPLES sendo os pontos-chave detectados com o SURF