

# Reconhecimento de Padrões Invariante Afim Baseado nos Coeficientes de Fourier de Projeções Radiais

Hae Yong Kim, Guillermo Ángel Pérez López  
Escola Politécnica, Universidade de São Paulo, Brasil  
hae@lps.usp.br, guillermo.angel@poli.usp.br

## Resumo

O reconhecimento de padrões consiste em achar ocorrências de uma imagem padrão ou modelo  $Q$  numa outra imagem a ser analisada  $A$ . Este problema torna-se bem mais difícil de se resolver se as imagens  $Q$  e  $A$  podem ser vistas de diferentes pontos de vista, gerando deformações em perspectiva. Neste trabalho, descrevemos os resultados preliminares de um sistema de reconhecimento invariante a transformações afins que consegue reconhecer as imagens independentes do ponto de vista. A proposta está baseada em duas técnicas: (1) o casamento de padrões baseado nos Coeficientes de Fourier de Projeções Radiais (Forapro), para obter a invariância à rotação, escala, brilho e contraste; (2) a simulação de pontos de vista da imagem  $Q$  para obter a invariância afim e consequentemente invariância aos pontos de vista.

## 1. Introdução

O reconhecimento de padrões em imagens digitais (*template matching*) é uma tarefa fundamental em visão computacional. Um reconhecedor de padrões procura uma imagem modelo ou padrão  $Q$  (*query*) dentro da imagem de procura a analisar  $A$ .

As principais dificuldades do reconhecimento do padrão  $Q$  em  $A$  são: (1)  $Q$  pode aparecer em diferentes ângulos e escalas; (2) as instâncias de  $Q$  em  $A$  podem sofrer efeitos fotométricos que incluem mudança de brilho e contraste; (3) o conjunto ótico da câmera borra as imagens (principalmente com o foco mal-ajustado); (4)  $Q$  pode sofrer deformação em perspectiva devido à mudança na posição do observador.

Recentemente, foi proposto o algoritmo de reconhecimento de padrões Forapro (*Fourier Coefficients of Radial Projection*) [1]. Este algoritmo é invariante à rotação, escala, translação, brilho e contraste. Ele é bastante robusto a borrão e ruídos. Porém, este algoritmo é pouco robusto a deformação em perspectiva.

SIFT (*Scale-Invariant Feature Transform*) [2] é uma das técnicas mais populares e eficientes para calcular as correspondências entre os pontos-chaves de duas imagens, invariante a escala e rotação. SIFT detecta os pontos-chaves através de DoG (*Difference-of-Gaussian*) e um descritor baseado na distribuição de orientação do gradi-

ente local. SIFT original não faz reconhecimento de padrões, mas apenas calcula os pontos-chaves correspondentes. Para que SIFT possa reconhecer padrões, as correspondências dos pontos-chaves devem ser agrupadas pela transformada de Hough generalizada [3]. Lowe sugere que se utilizem grupos de pelo menos três correspondências de pontos-chaves para identificar um padrão.

Morel e You propõem uma extensão do SIFT que o torna invariante a transformações afim denominada de ASIFT [4]. A idéia deste algoritmo é procurar as correspondências dos pontos-chaves em imagens que simulam as distorções afins.

Este trabalho apresenta os resultados preliminares do esforço para mostrar a potencialidade dos coeficientes de Fourier de projeções radiais na construção de um algoritmo para o reconhecimento de objetos invariante a transformações afins.

## 2. Transformações afins

Uma transformação afim de uma imagem  $u$  é uma transformação linear com seis graus de liberdade e pode ser descrita como:

$$u(x, y) \rightarrow u(ax+by+e, cx+dy+f).$$

As transformações afins incluem rotação, mudança de escala uniforme, não-uniforme e cisalhamento. Uma transformação afim no espaço euclidiano tem as seguintes propriedades:

1. Preserva a relação de colinearidade entre pontos, isto é, três pontos colineares continuam a ser colineares após a transformação.
2. Preserva a razão das distâncias ao longo de uma linha, isto é, para pontos colineares distintos  $p_1, p_2$  e  $p_3$ , a razão  $|p_2 - p_1|/|p_3 - p_2|$  é preservada.

Como consequência da primeira propriedade, a transformação afim preserva o paralelismo das retas. A projeção em perspectiva não mantém o paralelismo das retas. Porém, qualquer transformação em perspectiva pode ser aproximada localmente por uma transformação afim. Observe o trapezóide preto no chão da figura 1. Este trapezóide era originalmente um retângulo, mas tornou-se trapezoidal devido à deformação em perspectiva. Porém, repare que cada ladrilho é aproximadamente paralelogramo. Assim, pode-se procurar instâncias de  $Q$  que apareçam com deformação em perspectiva em  $A$  procurando sub-imagens  $T_i$  de  $Q$  que apareçam com deformação afim

em  $A$ . Os resultados das buscas podem ser agrupados através da transformada de Hough para localizar a instância de  $Q$  com distorção em perspectiva.



Figura 1: Uma transformação em perspectiva pode ser aproximada localmente por uma transformação afim.

### 3. Simulação de distorções

SIFT (*Scale-Invariant Feature Transform*) [2] é uma técnica bem-conhecida, popular e altamente eficiente para calcular as correspondências dos pontos-chaves entre duas imagens. SIFT é invariante à rotação e escala, mas não a distorção em perspectiva. Morel e You propõem uma extensão do SIFT que o torna invariante a transformações afim e que eles denominam de ASIFT [4]. Eles mostram que qualquer transformação afim (desconsiderando a translação):

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

pode ser decomposta de forma única como produto de quatro transformações:

$$A = H_\lambda R_1(\psi) T_t R_2(\phi) = \lambda \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix}$$

onde  $H_\lambda$  com  $\lambda > 0$  é a mudança de escala,  $R_1$  e  $R_2$  são duas rotações, e  $T_t$  com  $t = 1/\cos \theta$  é a mudança de inclinação. Figura 2 apresenta os parâmetros utilizados na equação acima.

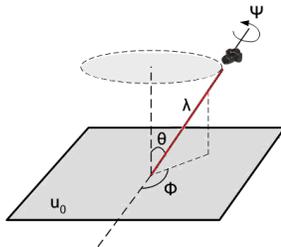


Figura 2: Decomposição de uma transformada afim.

SIFT é invariante por rotação e escala, ou seja, é invariante às mudanças de  $\psi$  e  $\lambda$ . Assim, se gerar as imagens distorcidas pelas mudanças de  $\theta$  e  $\phi$  e achar as correspondências dos pontos-chaves entre todas elas, é possível obter a invariância afim.

Podemos resumir ASIFT em: 1) dadas duas imagens  $Q$  e  $A$ ; 2) ASIFT realiza as transformações afins sobre as

duas imagens variando os parâmetros  $\theta$  e  $\phi$ ; 3) ASIFT compara todas as imagens simuladas utilizando SIFT para calcular as melhores correspondências dos pontos-chaves.

### 4. Forapro

Forapro (*Fourier coefficients of Radial Projections*) é um algoritmo de reconhecimento de padrões intrinsecamente invariante por rotação, brilho e contraste [1]. Ele é invariante à mudança de escala dentro de um intervalo especificado. Definiremos que duas imagens  $x$  e  $y$  são equivalentes sob variação de brilho/contraste se existem fator de correção de contraste  $\beta > 0$  e fator de correção de brilho  $\gamma$  tais que  $y = \beta x + \gamma I$ , onde  $I$  é a matriz de 1's.

Forapro calcula várias características invariantes por rotação, brilho e contraste, e uma característica que discrimina a rotação local (a “orientação canônica”) para cada pixel de  $A$ . Isto é feito a partir dos coeficientes de Fourier de projeções radiais. Forapro também utiliza os coeficientes de Fourier de projeções circulares para obter outras características invariantes por rotação.

Quando este processo termina, diferentes padrões circulares  $T_i$  podem ser buscados rapidamente em  $A$  (de forma invariante por rotação) comparando somente o vetor de características.

Para buscar uma imagem  $Q$  em  $A$ , Forapro extrai várias sub-imagens circulares  $T_i \subset Q$ , calculando as suas características invariantes por rotação e procurando em  $A$  os pixels com características semelhantes. Depois, essas correspondências são agrupadas através de transformada de Hough, resultando na detecção final.

No Forapro, achar  $N$  imagens circulares  $T_i$  numa única imagem  $A$  é muito mais rápido do que achar essas imagens  $T_i$  em  $N$  imagens diferentes. Esta propriedade facilita transformá-lo num reconhecedor de padrões invariante por escala e por transformação afim. Por exemplo, para obter a invariância à escala, Forapro redimensiona  $Q$  para várias escalas e procura-os todos em  $A$ . Este processo é relativamente rápido, pois as características de  $A$  só precisam ser calculadas uma única vez.

#### 4.1 Coeficientes Radiais de Fourier

Nesta e nas próximas subseções, descreveremos as características invariantes à rotação utilizadas no Forapro. Mais detalhes podem ser obtidos em [1].

Dada uma imagem  $A$  em níveis de cinza, definimos a projeção radial  $R_\alpha^\lambda(A(x, y))$  como o nível de cinza médio dos pixels de  $A$  localizados sobre a linha radial com um dos vértices em  $(x, y)$ , comprimento  $\lambda$  e inclinação  $\alpha$ :

$$R_\alpha^\lambda(A(x, y)) = \frac{1}{\lambda} \int_0^\lambda A(x + t \cos(\alpha), y + t \sin(\alpha)) dt.$$

O vetor de  $M$  projeções radiais discretas no pixel  $A(x, y)$  com raio  $\lambda$  é obtido variando o ângulo  $\alpha$ :

$$R_{A(x,y)}^\lambda[m] = R_{2\pi m/M}^\lambda(A(x,y)), \quad 0 \leq m < M.$$

Figura 3a mostra as  $M=36$  projeções radiais do pixel central.



Figura 3: Projeções radiais e circulares.

O vetor de projeções radiais  $R_{A(x,y)}^\lambda[m]$  caracteriza a vizinhança de raio  $\lambda$  de  $A(x,y)$ . Se  $A$  rotacionar, este vetor desloca circularmente. A  $k$ -ésima coeficiente de Fourier do vetor de projeções radiais  $R$  é (omitimos os índices  $A(x,y)$  e  $\lambda$ ):

$$r[k] = \sum_{m=0}^{M-1} R[m] \exp(-j2\pi km/M), \quad 0 \leq k < M.$$

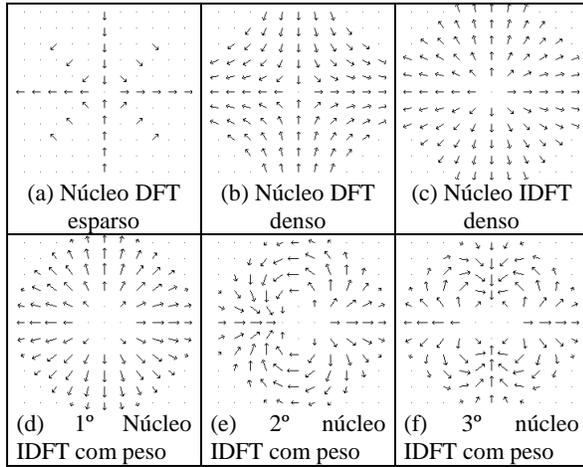


Figura 4: Núcleos radiais.

Os coeficientes de Fourier de um vetor de projeções podem ser calculados diretamente fazendo convolução de  $A$  com núcleos apropriados  $K$ , sem a necessidade de se calcular explicitamente as projeções radiais. Figura 4a mostra o núcleo DFT (Discrete Fourier Transform) esparsos  $K$  (com  $M=8$  ângulos) tal que a convolução  $A * \tilde{K}$  resulta no primeiro coeficiente de Fourier das 8 projeções radiais (onde  $\tilde{K}(x,y) = K(-x,-y)$ ):

$$(A * \tilde{K})(x,y) = \sum_p \sum_q A(p,q) K(p-x, q-y).$$

É amplamente conhecido que a convolução  $A * \tilde{K}$  pode ser calculada através de multiplicações no domínio da frequência:

$$A * \tilde{K} \leftrightarrow A\tilde{K},$$

onde  $A$  e  $\tilde{K}$  são respectivamente as transformadas Fourier discretas de  $A$  e  $\tilde{K}$ .

O núcleo esparsos da Figura 4a não leva em conta a maioria dos pixels externos e assim não gera características robustas. Para superar este problema, “núcleo DFT denso” da Figura 5b pode ser usado. Ele preenche todos os pixels do núcleo, exceto o elemento central. O filtro linear usando este núcleo não tem um significado intuitivo. Usando a transformada discreta de Fourier inversa (IDFT), o resultado da convolução adquire um significado: torna-se análogo ao gradiente. Figura 4c mostra o núcleo obtido usando IDFT e  $k=1$ . Para tornar o núcleo mais “estável”, isto é, para fazer o resultado da convolução menos sensível às pequenas perturbações como translação ou rotação por sub-pixel, atribuímos menos peso aos pixels nas regiões externa e central, resultando em núcleo com peso da Figura 4d. O núcleo radial resultante é:

$$\mathfrak{R}_k[x,y] = \sqrt{r(\lambda-r)} \exp(jk \angle(x+iy))$$

onde  $r = \sqrt{x^2 + y^2}$  e  $\lambda$  é o raio do núcleo. Os núcleos usados para obter coeficientes IDFT para  $k=2$  e  $k=3$  são mostrados nas figuras 4e e 4f. Chamaremos a convolução de  $A(x,y)$  com a reflexão dupla do  $k$ -ésimo núcleo radial de “ $k$ -ésimo coeficiente radial” e a denotaremos por  $r_k(A(x,y))$  ou simplesmente  $r_k$ . Chamaremos  $\angle r_k$  e  $|r_k|$  respectivamente “ $k$ -ésimo ângulo radial” e “ $k$ -ésima magnitude radial”.

## 4.2 Orientação Canônica

A característica que discrimina a rotação local ou a “orientação canônica” é o primeiro ângulo radial  $\angle r_1$ . A orientação canônica  $\angle r_1(A(x,y))$  indica a direção local de  $A(x,y)$  na vizinhança de raio  $\lambda$ . Se  $A$  rotacionar  $\theta$  radianos, o vetor de projeções radiais  $R_{A(x,y)}^\lambda[m]$  desloca circularmente  $\theta$  radianos e consequentemente o primeiro ângulo radial  $\angle r_1(A(x,y))$  rotaciona-se pelo mesmo ângulo. Uma mudança de brilho/contraste não altera a orientação canônica.

## 4.3 Vetor de Magnitudes Radiais

Magnitudes radiais são invariantes por rotação pois se  $A$  rotaciona, então o vetor de projeções radiais  $R_{A(x,y)}^\lambda[m]$  desloca circularmente, e um deslocamento circular não muda as magnitudes dos coeficientes IDFT (somente os seus ângulos são afetados). As magnitudes radiais  $|r_k|$ ,  $k \geq 1$ , também são invariantes a brilho pois a mudança de brilho somente afeta o coeficiente  $r_0$  (o componente CC). Finalmente, as razões entre magnitudes radiais são invariantes a contraste, pois uma alteração no contraste multiplica pelo mesmo fator todos os coeficientes radiais. Portanto utiliza o seguinte vetor de magnitudes radiais  $v_m$

que leva em conta magnitudes de todos os coeficientes radiais até o grau  $K$ :

$$v_{rm} = \upsilon [ |r_1|, |r_2|, \dots, |r_K| ]$$

Onde  $\upsilon$  significa  $L^1$ -versor que consiste em dividir cada elemento do vetor pelo seu comprimento  $L^1$ . Este vetor é invariante por rotação, brilho e contraste. A função distância  $\Lambda$  entre dois  $v_{rm}$ 's é definida como:

$$\Lambda(v_{rm}(A(x, y)), v_{rm}(T(x_o, y_o))) = \frac{1}{2} \|v_{rm}(A(x, y)) - v_{rm}(T(x_o, y_o))\|_1$$

#### 4.4 Vetor de Ângulos Radiais

Se  $A$  rotacionar  $\theta$  radianos, então o vetor de projeções radiais  $R_{A(x,y)}^\lambda[m]$  desloca circularmente  $\theta$  radianos e o  $k$ -ésimo coeficiente radial  $r_k(A(x, y))$  é multiplicado por  $\exp(jk\theta)$ . Além disso, uma mudança de brilho/contraste não afeta  $\angle r_k$ . Assim, a diferença entre  $\angle r_k$  e  $k \angle r_1$  é invariante por rotação, brilho e contraste. Denominamos este valor de diferença entre ângulos radiais  $k$  e 1:

$$\text{dra}_k = \text{mod}(\angle r_k - k \angle r_1, 2\pi), \quad k \geq 2.$$

Esta característica deve ser calculada em módulo  $2\pi$ , pois é um ângulo. Empacotamos todas as diferenças de ângulos radiais até ordem  $K$  numa estrutura denominada vetor de ângulos radiais  $v_{ra}$ :

$$v_{ra} = [\text{dra}_2, \text{dra}_3, \dots, \text{dra}_K].$$

Este vetor é invariante por rotação, brilho e contraste. A distância  $\Lambda$  entre dois  $v_{ra}$ 's é definida como:

$$\begin{aligned} \Lambda[v_{ra}(A(x, y)), v_{ra}(T(x_o, y_o))] = & (w_2 / (\pi w_t)) \Phi[\text{dra}_2(A(x, y)), \text{dra}_2(T(x_o, y_o))] + \\ & (w_3 / (\pi w_t)) \Phi[\text{dra}_3(A(x, y)), \text{dra}_3(T(x_o, y_o))] + \\ & \dots + \\ & (w_K / (\pi w_t)) \Phi[\text{dra}_K(A(x, y)), \text{dra}_K(T(x_o, y_o))] \end{aligned}$$

onde  $w_k = 1/k$  ( $2 \leq k \leq K$ ),  $w_t = w_2 + w_3 + \dots + w_K$  e  $\Phi$  é a diferença entre dois ângulos:

$$\Phi(a, b) = \min(\text{mod}(a - b, 2\pi), 2\pi - \text{mod}(a - b, 2\pi)).$$

#### 4.5 Vetor de Características Circulares

Foraprop utiliza as características derivadas das projeções circulares (figura 3b) junto com aquelas derivadas de projeções radiais. O núcleo circular utilizado é:

$$\mathfrak{J}_l[x, y] = \begin{cases} \frac{1}{2\pi r} \exp\left(\frac{jlr}{\lambda}\right), & \text{se } r > 0 \\ 0,73, & \text{se } r = 0 \end{cases}$$

onde  $r = \sqrt{x^2 + y^2}$  e  $\lambda$  é o raio do núcleo. O peso  $1/2\pi r$  é o inverso do perímetro do círculo onde o pixel está situado. O peso para  $r = 0$  foi definido como 0,73 para distri-

buir uniformemente os ângulos da imagem complexa resultante da convolução.

A convolução de  $A(x, y)$  com a dupla reflexão de primeiro núcleo circular é denotado  $c_l(A(x, y))$  ou simplesmente  $c_l$ . O “vetor de características circulares” abaixo leva em conta os componentes reais e imaginários de todos os coeficientes circulares até o grau  $L$ :

$$v_{cf} = \upsilon [\text{re}(c_1), \text{im}(c_1), \text{re}(c_2), \text{im}(c_2), \dots, \text{re}(c_L), \text{im}(c_L)],$$

onde  $\upsilon$  significa  $L^1$ -versor, e “re” e “im” são respectivamente as partes real e imaginário do número complexo. Este vetor é invariante por rotação, brilho e contraste. A distância  $\Lambda$  entre dois  $v_{cf}$ 's é:

$$\Lambda(v_{cf}(A(x, y)), v_{cf}(T(x_o, y_o))) = \frac{1}{2} \|v_{cf}(A(x, y)) - v_{cf}(T(x_o, y_o))\|_1$$

#### 4.6 Combinando as Características

Nas subseções anteriores, obtivemos três classes de características invariantes por rotação, brilho e contraste e as empacotamos em três vetores:  $v_{rm}$ ,  $v_{ra}$ , e  $v_{cf}$ . Agruparemos esses três vetores numa outra estrutura denominada de “vetor de características”:

$$v_f = (v_{rm}, v_{ra}, v_{cf}).$$

Definimos a função distância  $\Lambda$  entre dois vetores de características como uma média ponderada da distância dos três vetores constituintes:

$$\begin{aligned} \Lambda(v_f(A(x, y)), v_f(T(x_o, y_o))) = & (w_m / w_t) \Lambda(v_{rm}(A(x, y)), v_{rm}(T(x_o, y_o))) \\ & + (w_a / w_t) \Lambda(v_{ra}(A(x, y)), v_{ra}(T(x_o, y_o))) \\ & + (w_c / w_t) \Lambda(v_{cf}(A(x, y)), v_{cf}(T(x_o, y_o))) \end{aligned}$$

onde  $w_m = w_a = K-1$ ,  $w_c = 2L-1$  e  $w_t = w_m + w_a + w_c$ .

#### 4.7 Transformada de Hough

Para detectar  $Q$  em  $A$  com robustez, extraímos  $N$  sub-imagens circulares  $T_1, \dots, T_N \subset Q$ , as detectamos em  $A$  e agrupamos os resultados usando a transformada de Hough. É possível que a detecção de uma única sub-imagem gere erro. Porém, se muitas sub-imagens concordam em apontar um pixel como o ponto de casamento, a probabilidade de erro é minimizada.

O algoritmo usando a transformada de Hough é: Dada uma imagem  $Q$ , extrair  $N$  sub-imagens circulares “apropriadas”  $T_1, \dots, T_N$ . Para cada  $T_i$ , gere a imagem de distâncias  $D_i$ , onde cada pixel  $D_i(x, y)$  é a distância entre as características de  $A(x, y)$  e  $T_i$ :

$$D_i(x, y) = \Lambda(v_f(A(x, y)), v_f(T_i(x_o, y_o))).$$

Localize o conjunto  $C_i$  de pixels candidatos ao casamento selecionando  $n_c$  pixels com as menores distâncias. A partir dos pixels candidatos e das suas orientações canônicas,

calcule o pixel central de  $Q$  em  $A$ . Incremente a matriz  $H$  de acumulação da transformada de Hough por  $1 - D_i(x, y)$ . Os casamentos são aqueles pixels de  $H$  com os maiores valores. Quanto maior este valor, melhor é a qualidade do casamento. Para acelerar o processamento, a transformada de Hough tradicional pode ser substituída pela “transformada de Hough matricial” descrita em [1]. Também é possível calcular  $C_i$  sem calcular explicitamente  $D_i$ .

Para obter a invariância à escala, a imagem  $Q$  deve ser redimensionada para diferentes escalas e as imagens resultantes devem ser procuradas em  $A$ . Observa-se experimentalmente que 4 escalas por oitava é suficiente para obter a invariância à escala.

## 5. Aforapro

Vamos denominar Forapro invariante às transformações afins de Aforapro. Para tornar Forapro invariante às distorções em perspectiva, distorcemos a imagem  $Q$  com diferentes parâmetros  $\theta$  e  $\phi$ , obtendo as imagens distorcidas  $Q_j$ . Supomos que a imagem  $Q$  foi fotografada de frente mas que pode aparecer na imagem  $A$  com distorção. Isto é, diferentemente do algoritmo ASIFT, não consideramos as “inclinações de transição” onde ambas  $Q$  e  $A$  podem estar inclinadas em direções diferentes. Seguindo a sugestão de Morel e You, distorcemos a imagem  $Q$  tomando:

- $t = \left\{ \sqrt{2}^0, \sqrt{2}^1, \sqrt{2}^2, \sqrt{2}^3, \sqrt{2}^4 \right\}$ , onde  $t = 1/\cos \theta$ .
- $\phi = \{0, b/t, \dots, kb/t\}$ , onde  $b = 72^\circ$  e  $kb/t < 180^\circ$

A Figura 5 mostra algumas distorções simuladas.

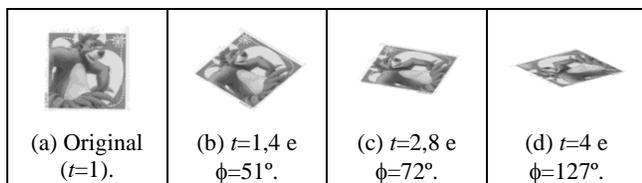


Figura 5: Imagens que simulam distorções afins.

A nossa implementação preliminar está executando em dois passos. No primeiro passo, escolhemos a distorção que melhor corresponde à imagem  $A$ . Para isso, executamos Forapro e escolhemos a imagem distorcida  $Q_j$  que gerou o maior valor na matriz de acumulação  $H$  da transformada de Hough. No segundo passo, executamos novamente Forapro para gerar a imagem que mostra as correspondências entre  $Q_j$  e  $A$ .

## 6. Experimentos Preliminares

Apresentamos abaixo alguns resultados preliminares do algoritmo proposto. Ainda não realizamos testes sistemáticos para verificar a acuracidade do algoritmo.

### 6.1 Testes Iniciais

Figura 6 mostra um dos testes iniciais do Aforapro. Obtivemos as imagens  $Q$  e  $A$  do banco de imagens do ASIFT<sup>2</sup>. Pela informação fornecida por You e Morel, entre os algoritmos existentes só o ASIFT consegue um reconhecimento destas imagens. Figura 6c mostra a imagem distorcida  $Q_j$  que melhor se casa com a imagem  $A$ , escolhida automaticamente pelo Aforapro. Figura 6d mostra as correspondências detectadas pelo Aforapro para reconhecer o padrão. Diferentemente de ASIFT, o número de correspondências é um parâmetro escolhido pelo usuário. Pequeno número de correspondências não indica uma detecção pobre. Figura 7 mostra outro teste do Aforapro. Obtivemos as imagens  $Q$  e  $A$  do banco de imagens de Mikolajczyk<sup>3</sup>.

### 6.2 Aforapro e ASIFT

Nesta subseção apresentamos alguns casos em que ASIFT apresenta dificuldade para reconhecer um padrão, mas que Aforapro consegue reconhecê-lo com sucesso. Existem vários outros casos em que ocorre o inverso (Aforapro falha, mas ASIFT sucede) mas que não foram relatados neste artigo. É necessário um estudo mais detalhado para caracterizar as situações em que cada algoritmo se comporta melhor.

Figura 8 mostra a dificuldade do ASIFT para fazer reconhecimento quando há excesso de textura. Aforapro consegue reconhecer o padrão com sucesso. Figura 9 mostra a dificuldade do ASIFT para fazer reconhecimento quando há diferença excessiva de brilho e contraste entre  $Q$  e  $A$ . Aforapro consegue reconhecer o padrão com sucesso.

## 7. Conclusões e Trabalhos Futuros

Forapro é um algoritmo recente de reconhecimento de padrões invariante à rotação, brilho e contraste, e robusto a variações de escala. Este trabalho apresentou os resultados preliminares do esforço para torná-lo invariante às mudanças de pontos de vista que geram distorções em perspectiva. Ainda são necessárias várias melhorias na implementação e mais testes experimentais para completar a validação do algoritmo proposto. Entre eles, o uso de alguma estrutura de dados para acelerar a busca do pixel com as características similares (por exemplo, usando árvore de decisão, “boosting” ou árvore aleatória) e o uso de estrutura de pirâmide para tornar Aforapro realmente invariante à escala.

<sup>2</sup> <http://www.cmap.polytechnique.fr/~yu/research/ASIFT/demo.html>

<sup>3</sup> <http://lear.inrialpes.fr/people/mikolajczyk/Database>

## 8. Referências

- [1] H. Y. Kim, "Rotation-Discriminating Template Matching Based on Fourier Coefficients of Radial Projections with Robustness to Scaling and Partial Occlusion", *Pattern Recognition*, vol. 43, no. 3, pp. 859-872, 2010.
- [2] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- [3] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981.
- [4] J. Morel and G. Yu, "A Fully Affine Invariant Image Comparison Method", in *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp 1597-1600, 2009.

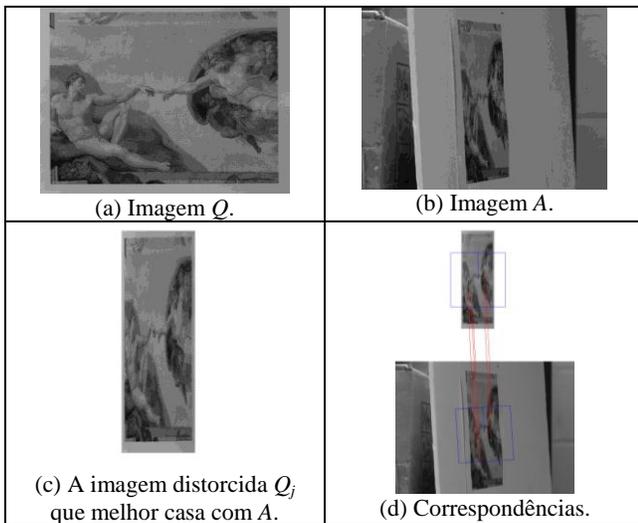


Figura 6: Exemplo de uso do Aforapro.

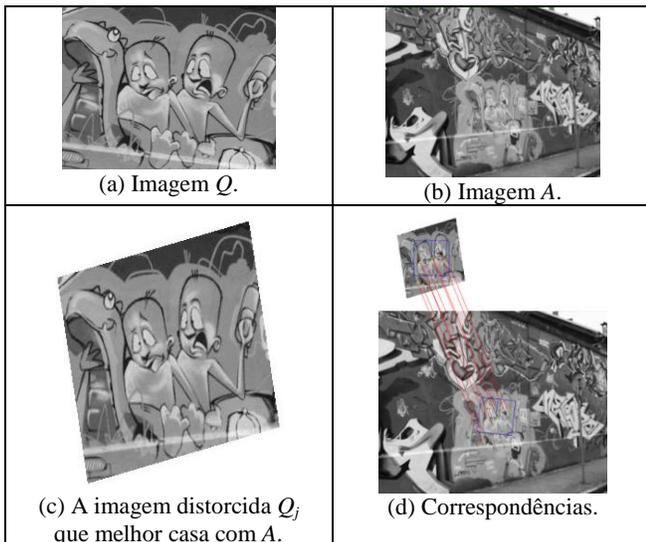


Figura 7: Exemplo de uso do Aforapro.

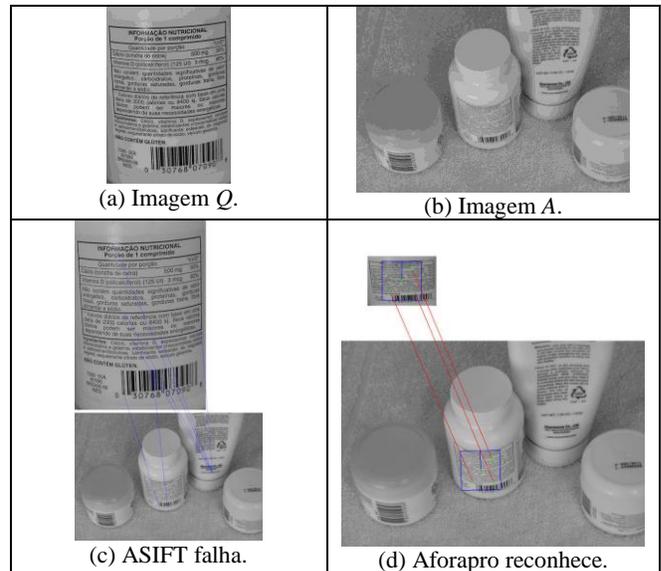


Figura 8: Aforapro pode funcionar melhor que ASIFT quando as imagens apresentam excesso de texturas.

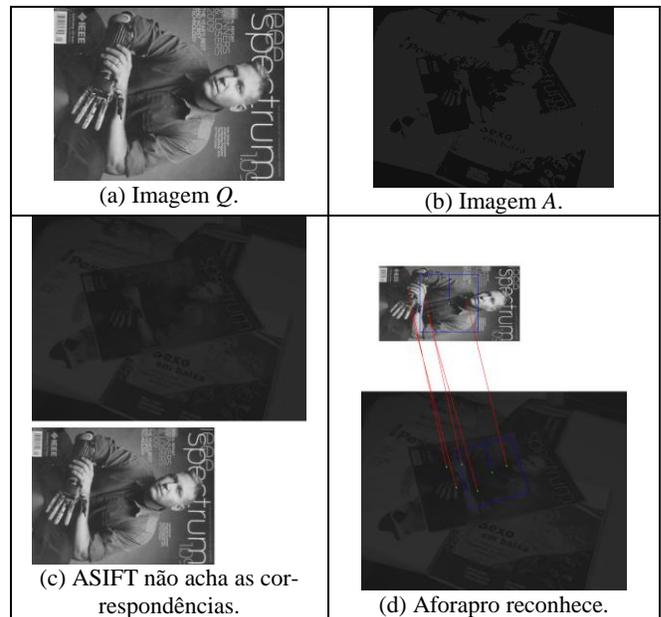


Figura 9: ASIFT pode falhar ao reconhecer uma instância com baixo contraste, enquanto que Aforapro consegue.