

# ویژگی

## علی نصیری سروی

اطلاعات گزارش	چکیده
تاریخ: 99/03/27	
واژگان کلیدی: نقاط ویژگی گوشه تخمین هندسه تبدیل هندسی	در این تمرین ابتدا سعی در تخمین هندسه تصاویر به کمک نقاط ویژگی میکنیم. سپس سعی میکنیم برخی تصاویر تغییر یافته را به تصویر اورجینال تبدیل کنیم. در آخر نیز الگوریتم گوشه یابی Harris را پیاده سازی میکنیم.

## 1-مقدمه

شناسایی ویژگی یک عملیات سطح پایین بر روی تصویر میباشد. به این صورت که هر پیکسل جداگانه بررسی میشود که آیا میتواند نقطه ویژگی باشد یا خیر. برخی از کاربردهای استفاده از نقاط ویژگی می توان به image matching اشاره کرد. زمانی که دو تصویر با زوایای مختلف از یک صحنه یکسان گرفته شود و بخواهیم آن دو تصویر را متشابه هم در نظر بگیریم از نقاط ویژگی شان استفاده میکنیم.

یکی از کاربرد های دیگر نقاط ویژگی ساخت تصاویر panorama از تصاویر جدا جدا می باشد. یک نمونه از آن در تمرین 1 مشاهده شد.

## 2-شرح تکنیکال

### 7.1 تخمین هندسه

یکی از کاربردهای بسیار مهم نقاط ویژگی تخمین هندسه تصویر می باشد. برای تخمین هندسه یک تصویر تغییر یافته از نقاط ویژگی استفاده میکنیم. برای تخمین زدن هندسه تصویر نیازمند به دست آوردن یک تبدیل هندسی هستیم که تصویر تغییر یافته را به کمک آن به تصویر اصلی مپ کند.

برای اینکار به صورت زیر عمل میکنیم.

1) ابتدا نقاط ویژگی تصویر فرنس را بدست می آوریم.

2) سپس نقاط ویژگی تصویر تغییر یافته را بدست می آوریم.

(3) سپس بین نقاط ویژگی بدست آمده، نقاط

ویژگی متناظر را بدست می آوریم (به کمک

الگوریتم  $k$  nearest neighbor)

(4) اگر فرض کنیم یک نقطه ویژگی در تصویر

اصلی مختصات  $[u, v]$  داشته باشد و مختصات

آن در تصویر تغییر یافته  $[x, y]$

باشد، می خواهیم ماتریس تبدیل  $T$  را به صورتی

پیدا کنیم که  $T[x \ y \ 1]^T = [u, v, 1]$

اگر برای تمام نقاط متناظر بخواهیم این  $T$  را داشته

باشیم، می توان نقاط را در کنار هم در یک ماتریس

قرار داد.

$$X = \begin{bmatrix} x1 & \cdots & xn \\ y1 & \cdots & yn \\ 1 & \cdots & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} u1 & \cdots & un \\ v1 & \cdots & vn \\ 1 & \cdots & 1 \end{bmatrix}$$

در نهایت خواهیم داشت.

$$TX = U$$

چون ماتریس  $X$  لزوما معکوس پذیر (مربعی) نیست با

محاسبه شبه معکوس آن خواهیم داشت:

$$T = UX^+$$

بدین صورت ماتریس تبدیل بدست می آید.

حال برای هر نقطه در ماتریس تغییر یافته، نقطه متناظر

آن را به کمک فرمول  $T[x \ y \ 1]^T = [u, v, 1]$  بدست

می آوریم. در مختصات بدست آمده در تصویر اصلی به

کمک cubic interpolation مقدار عددی پیکسل را

به دست آورده و مقدار هر پیکسل را به این صورت تغییر

میدهیم.

برای بدست آوردن نقاط ویژگی از الگوریتم surf استفاده

میکنیم.

### 7.2.1 گوشه یابی:

برای گوشه یابی از الگوریتم *Harris* استفاده می کنیم. این

الگوریتم با در نظر گرفتن این نکته که در نقاط گوشه با

تغییر جزئی در مکان نقطه تغییرات زیادی در شدت رخ

میدهد کار میکند. الگوریتم *Harris* به صورت زیر

است:

(1) ابتدا به کمک فیلترهایی مانند فیلتر *Roberts*

مشتق های عمودی و افقی  $I_x, I_y$  را محاسبه

میکنیم.

(2) ماتریس  $I$  را به صورت زیر تعریف می کنیم.

$$I = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

(3) ماتریس  $M$  را به صورت زیر تعریف می کنیم.

$$M = \sum_{x,y} w(x,y) I$$

که  $w$  یک فیلتر گاوسین میباشد که بر روی  $I$  آن را

اعمال می کنیم.

(4) بردارهای ویژه ماتریس  $M$  نشان دهنده گوشه بودن

یا نبودن پیکسل می باشند. اگر هر دو بردار ویژه

مقدار بزرگی در دو جهت داشته باشند، نقطه گوشه

می باشد. به همین دلیل معیاری مانند  $R$  تعریف

میکنیم که:

$$R = \lambda_1 \lambda_2 - \alpha (\lambda_1 + \lambda_2)^2$$

در صورتی که مقدار  $R$  از یک *threshold* بیشتر

باشد، آن نقطه گوشه می باشد.

(5) در نهایت به کمک *non max suppression* در

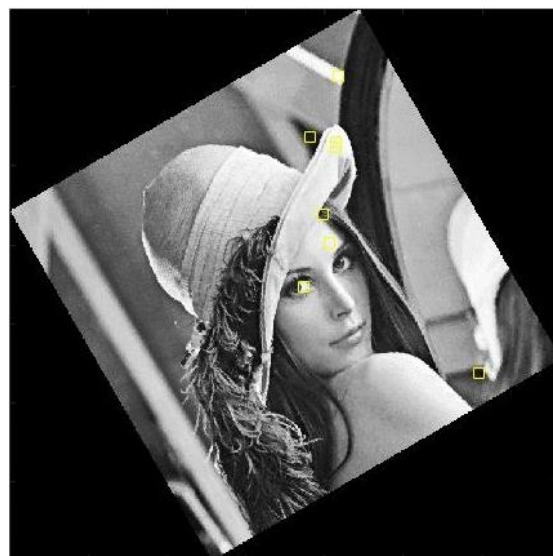
نواحی محلی، گوشه های اصلی را بدست می آوریم.

### 3-شرح نتایج

#### 7.1.1 تخمین هندسی:

ابتدا به سراغ تصویر اول می رویم.

ابتدا تصویر اول از attack1 را با تصویر reference مچ میکنیم:



فیچرهای مچ شده از تصویر اول attack1



فیچرهای مچ شده از تصویر reference

حال ماتریس تبدیل را محاسبه کرده و به تصویر یک از attack2 اعمال میکنیم.



تصویر اول از attack2



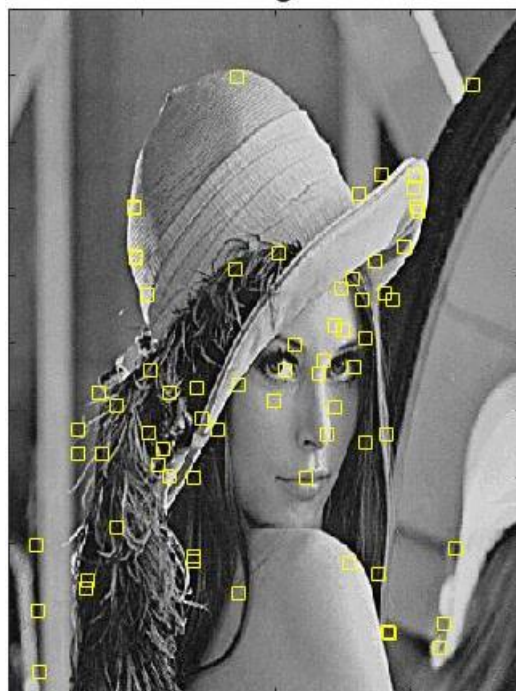
تصویر بدست آمده به کمک تبدیل محاسبه شده

کیفیت تبدیل را میسنجیم:

	SSIM	MSE	MP
Value	0.94	17.26	15

حال سراغ تصویر دوم می رویم.

تصویر دوم از attack1 را با تصویر reference مچ میکنیم:



فیچرهای مچ شده از تصویر دوم attack1

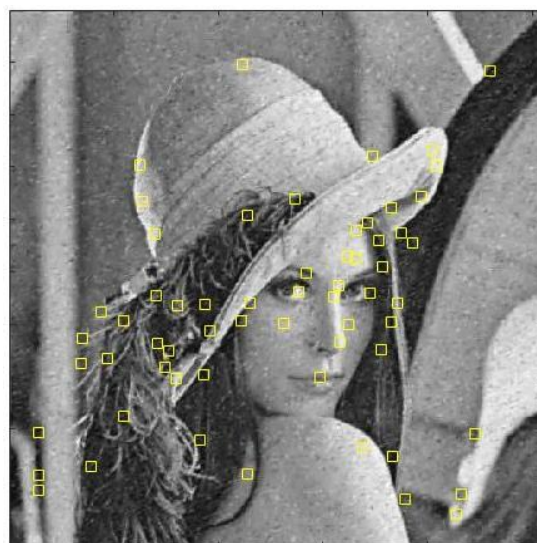
حال ماتریس تبدیل را محاسبه کرده و به تصویر یک از attack2 اعمال میکنیم.



تصویر دوم از attack2



تصویر بدست آمده به کمک تبدیل محاسبه شده



فیچرهای مچ شده از تصویر reference

کیفیت تبدیل را میسنجیم:

	SSIM	MSE	MP
Value	0.58	619.6	68

حال ماتریس تبدیل را محاسبه کرده و به تصویر یک از attack2 اعمال میکنیم.



تصویر سوم از attack2



تصویر بدست آمده به کمک تبدیل محاسبه شده

کیفیت تبدیل را میسنجیم:

	SSIM	MSE	MP
Value	0.61	407.09	13

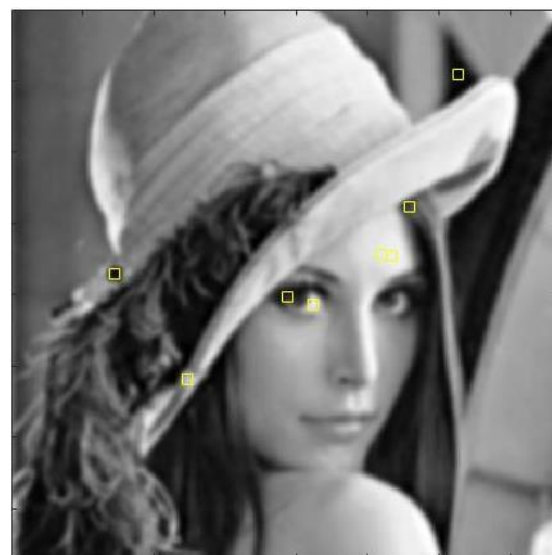
حال سراغ تصویر سوم می رویم.

این تصویر چون crop شده است، مقایسه آن با تصویر

reference اصلی به تنهایی، نتایج اشتباهی به ما

میدهند. پس آنها را با رفرنس crop شده مقایسه میکنیم.

تصویر سوم از attack1 را با تصویر reference-crop مچ میکنیم:



فیچرهای مچ شده از تصویر سوم attack1



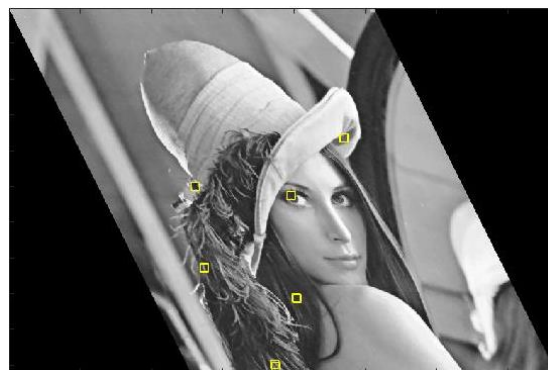
فیچرهای مچ شده از تصویر reference-crop



حال سراغ تصویر چهارم می رویم.

تصویر چهارم از attack1 را با تصویر reference

مچ میکنیم:

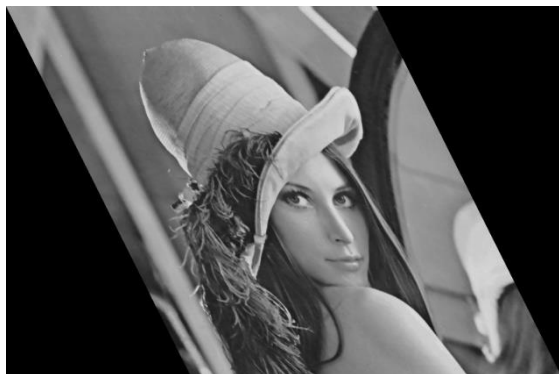


فیچرهای مچ شده از تصویر چهارم attack1



فیچرهای مچ شده از تصویر reference

حال ماتریس تبدیل را محاسبه کرده و به تصویر چهارم از attack2 اعمال میکنیم.



تصویر چهارم از attack2



تصویر بدست آمده به کمک تبدیل محاسبه شده

کیفیت تبدیل را میسنجیم:

	SSIM	MSE	MP
Value	0.92	62.53	19

نتایج کلی

	SSIM	MSE	MP
Mean	0.7625	276.62	28.75

	SSIM	MSE	MP
STD	0.16	248.87	22.76

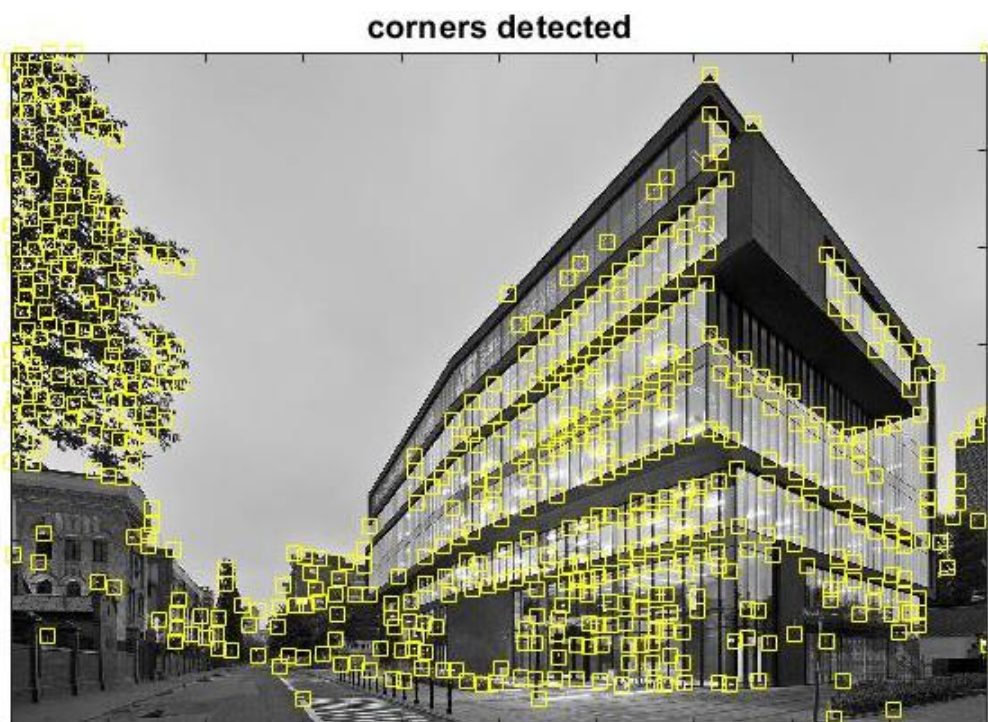
## 7.2.1 شناسایی لبه:

تصویر ساختمان را داریم:



با اعمال کد پیاده سازی شده گوشه یاب و اعمال

threshold و nms خواهیم داشت:



```
x1 =
matchedPoints1.Location(:,
1);
y1 =
matchedPoints1.Location(:,
2);
x2 =
matchedPoints2.Location(:,
1);
y2 =
matchedPoints2.Location(:,
2);
z1 = [x1 y1];
z2 = [x2 y2];

T =
fitgeotrans(z1,z2,'nonrefl
ectivesimilarity');

output =
imwarp(new_img,T,'cubic','
OutputView',
imref2d(size(new_img)));
output =
output(1:512,1:512);

imshow(output);

m =
immse(original,output);
ss =
ssim(original,output);

imwrite(output , 'x.png');

figure, imagesc(img), axis
image, colormap(gray),
hold on
    plot(x1,y1,'ys'),
title('img');

figure, imagesc(ref), axis
image, colormap(gray),
hold on
    plot(x2,y2,'ys'),
title('ref');
```

```
img =
imread('Homeworks\Images\7
\Attack 1\1.bmp');
img = rgb2gray(img);
ref =
imread('Homeworks\Images\7
\Reference.bmp');
ref = rgb2gray(ref);
new_img =
imread('Homeworks\Images\7
\Attack 2\1.bmp');
new_img =
rgb2gray(new_img);
original =
imread('Homeworks\Images\7
\Original.bmp');
original =
rgb2gray(original);

mt = 6500;
feature_img =
detectSURFFeatures(img, 'Me
tricThreshold',mt);
feature_ref =
detectSURFFeatures(ref, 'Me
tricThreshold',mt);

[f1,vpts1] =
extractFeatures(img,featur
e_img);
[f2,vpts2] =
extractFeatures(ref,featur
e_ref);

indexPairs =
matchFeatures(f1,f2);
matchedPoints1 =
vpts1(indexPairs(:,1));
matchedPoints2 =
vpts2(indexPairs(:,2));
```



```

    cim =
    (cim==mx) & (cim>thresh);
    % Find maxima.

```

```

    [r,c] = find(cim);
    % Find row,col coords.

```

```

end
function output =
norm(img)
    output =
mat2gray(img);
    Max =
max(max(output));
    Min =
min(min(output));
    output = (255/(Max-
Min))*output;

```

```

end

```

```

function
output=filtering(img,mask)
    [R,C] = size(img);
    [x,y] = size(mask);
    img =
padarray(img,[1,1]);
    output =
zeros(R,C,'double');
    for i=1:R
        for j=1:C
            part =
img(i:i+x-1,j:j+y-1);
            mult =
part.*mask;
            out =
sum(mult,'all');
            output(i,j) =
out;
        end
    end
end
end

```

```

img =
imread('Homeworks\Images\7
\Building.jpg');
img =
double(rgb2gray(img));

```

```

dx = [-1 0 1;-1 0 1;-1 0
1]/6;
dy = [-1 -1 -1;0 0 0 ; 1 1
1]/6;
sigma = 0.5;
g =
fspecial('gaussian',max(1,
fix(6*sigma)), sigma);

```

```

Ix = filtering(img,dx);
Iy = filtering(img,dy);
Ix2 = conv2(Ix.^2, g);
Iy2 = conv2(Iy.^2, g);
Ixy = conv2(Ix.*Iy, g);

```

```

k = 0.04;
R11 = (Ix2.*Iy2 - Ixy.^2)
- k*(Ix2 + Iy2).^2;
thresh = 10;
cond = R11>thresh;
R = R11.*cond;
R = norm(R);
imshow(R);
[r,c] = nms(R,20,thresh);
figure, imagesc(img), axis
image, colormap(gray),
hold on
    plot(c,r,'ys'),
title('corners detected');

```

```

function [r,c] =
nms(cim,radius,thresh)
    size = 2*radius+1;
    % Size of mask.
    mx =
ordfilt2(cim,size^2,ones(sz
e)); % Grey-scale dilate.

```