第46卷第9期 中国科学技术大学学报 Vol. 46, No. 9
2016年9月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Sep. 2016

文章编号:0253-2778(2016)9-0743-06

图像矩在图像区域复制粘贴检测中的快速计算算法

赖玥聪[1],黄添强[2],林 晶[1]

(1. 福建师范大学数学与计算机科学学院, 福建福州 350007;
2. 福建师范大学大数据分析与应用福建省高校工程研究中心,福建福州 350007)

摘要:在图像区域复制粘贴篡改检测中,特征提取是非常重要的一个步骤,而图像矩(如Zernike矩,圆谐-傅里叶矩和指数矩)是一种比较常见的特征向量.针对目前大多数检测算法在提取某些图像矩运行时间过长的问题,提出一种图像矩的快速算法.利用计算机仿真计算图像矩时,要把积分形式的表达式离散化,将积分转化为求和.在最终的计算图像矩的离散表达式中,可以将表达式分成固定值和像素值的乘积.将两部分分开计算,只计算固定值一次,像素值 N_{um} 次,然后得到它们的乘积.由此大大减少了计算的次数,从而大幅度降低了运行时间.只要图像的分辨率不变和计算的图像矩个数足够多,相比于经典的计算方法,图像矩的快速算法能极大的提高算法的运行效率.

关键词:图像矩;快速算法;区域复制粘贴;图像取证

中图分类号:TP18 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2016.09.005

引用格式:赖玥聪,黄添强,林晶. 图像矩在图像区域复制粘贴检测中的快速计算算法[J]. 中国科学技术大学学报,2016,46(9):743-748.

LAI Yuecong, HUANG Tianqiang, LIN Jing. A fast algorithm of image moments in copy-move forgery detection[J]. Journal of University of Science and Technology of China, 2016,46(9):743-748.

A fast algorithm of image moments in copy-move forgery detection

LAI Yuecong[1], HUANG Tianqiang[2], LIN Jing[1]

(1. *School of Mathematics and Computer Science, Fujian Normal University, Fuzhou* 350007, *China*;
2. *Fujian Provincial University Engineering Research Center of Big Data Analysis and Application, Fujian Normal University, Fuzhou* 350007, *China*)

Abstract: In copy-move forgery detection, feature extraction is an important step. The image moments (e.g., Zernike moments, radial harmonic Fourier moments, Exponential Fourier moments) are common feature vectors. In view of the excessive length of running time of most existing feature extraction algorithms, a fast algorithm of image moments in copy-move forgery detection was proposed. Integral expression should be discretized and turn integration turned into sum. In the final discretization expression, we can divide it into two parts. One is the fixed section, the other is the grey level of the image. In the classical computing algorithm of image moments, the two parts should be calanlated N_{um} times to get their product if N_{um} image moments are to be obtoined. A fast computing method was put

收稿日期:2016-03-01;**修回日期**:2016-09-17

基金项目:国家自然科学基金(61070062),福建省高校产学合作科技重大项目(2015H6007),福州市科技计划(2014-G-7 6),福建省高等学校新世纪优秀人才支持计划(JAI1038),福建省科学厅K类基金(JK2011007),福建省教育厅A类基金(JA10064),福建师范大学研究生教育改革研究项目(MY201414)资助.

作者简介:赖玥聪,男,1991年生,硕士生. 研究方向:多媒体篡改检测、数据挖掘. E-mail:jx8791608@163.com.

通讯作者:黄添强,博士/教授. E-mail:fjhtq@fjnu.edu.cn.

forward to calculate the two parts independently. The fixed parts are calculated just once and the grey level N_{um} times before the product was obtained. This reduced the running time because of the decline of the computing times. If the resolution ratios of the images are invariant and the numbers of the image moments sufficiently large, the fast algorithm of image moments can greatly reduce the running time compared with the classical method.

Key words: image moments; fast computation; copy-move forgery; image forensics

0 引言

随着图像编辑软件的日益流行和网络传播速度的加快,使得篡改一张图像变得越发容易.由此带来的一个负面影响是人们难以判定一幅图像的真伪.图像取证技术就是在这样的背景下发展起来的,它与图像篡改技术的关系类似于矛与盾的关系.虽然图像篡改技术层出不穷,越来越高明,但是图像取证技术也在迅速发展,能更好地检测越来越多的篡改类型.图像区域复制粘贴篡改是图像篡改中一种简单而且难以察觉的篡改方式,即将图像中的一个区域复制并粘贴到同一幅图像的另一个不重叠的区域[1].

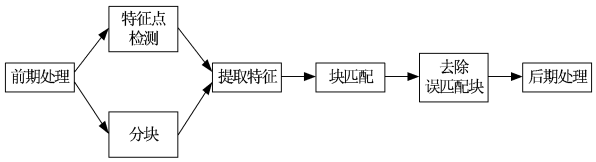

图1 图像区域复制-粘贴篡改检测的经典算法流程

Fig. 1 The radial basis functions of the different image moments

目前,已经提出许多算法用于检测图像区域复制粘贴篡改.如图1所示,大部分检测算法可以分为基于特征点的方法和基于块的方法[1].其中,在基于块的方法中,图像矩是特征提取步骤中比较常用的特征向量,如文献[2-3]的Zernike矩,文献[4]的圆谐-傅里叶矩,文献[5]的指数矩,文献[6]的polar cosine transform (PCT),文献[7]的polar sine transform (PST)等.上述文献在提取图像矩时,往往采用经典的计算方法,导致所用的时间较长,从而降低了整个区域复制粘贴检测算法的效率.也就是说,在分块之后,要提取每个子块的图像矩作为特征向量,上述文献将图像矩的计算方法看成一个"黑盒子",即输入一个图像子块,输出一个图像矩构成的特征向量.这种计算方式往往比较耗时,本文通过分析"黑盒子"的结构,利用图像矩经典计算方法中固定值部分的值不变的性质,得到图像矩快速计算方法,大幅缩减了计算时间.

本文提出了一种计算这些图像矩的快速计算算法.只要图像的分辨率和图像矩的参数不变,最终计算图像矩的式子中的固定部分的值就不变.图像矩的快速算法利用这个性质,大大减少了计算次数,从而提高了提取图像矩特征算法的时间效率.其次,与图像矩的经典计算方法相比,快速算法并不会改变图像矩的计算结果,所以并不会给复制粘贴篡改检测的后续步骤带来未知的影响.

1 图像矩的快速算法

图像矩就是把图像函数投影到基函数上得到的一组系数,具有多畸变不变性,即当图像进行平移、缩放、旋转变换后,图像的矩不变量是保持不变的[8].不同的基函数可以得到不同的图像矩,如Zernike矩,圆谐-傅里叶矩和指数矩等.下面以指数矩为例,类似的可以得到Zernike矩[9],圆谐-傅里叶矩(RHFMs)[10],指数矩(EFMs)[11],PCT[12]及PST的定义.

指数矩的定义如下:

在极坐标系(r,θ)中,定义函数系$P_{nm}(r,\theta)$为:

$$P_{nm}(r,\theta)=Q_n(r)\exp(jm\theta) \tag{1}$$

式中,$Q_n(r)$为径向基函数,$\exp(jm\theta)$为角向函数,j为虚数单位,n和m的取值范围是所有整数.

$$Q_n(r)=\sqrt{\frac{1}{r}}\exp(j2n\pi r) \tag{2}$$

函数系$P_{nm}(r,\theta)$在区间$0\leqslant r\leqslant 1, 0\leqslant\theta\leqslant 2\pi$内是正交的,即

$$\int_0^{2\pi}\int_0^1 P_{nm}(r,\theta)P_{kl}(r,\theta)r\mathrm{d}r\mathrm{d}\theta=2\pi\delta_{nmkl} \tag{3}$$

式中,δ_{nmkl}是Kronecker符号,$r=1$为在特定情形下遇到物体的最大尺寸.按照函数正交理论,在极坐标系中图像$f(r,\theta)$可以分解为函数系$P_{nm}(r,\theta)$的无限加权和.

$$f(r,\theta)=\sum_{n=-\infty}^{+\infty}\sum_{m=-\infty}^{+\infty}\mathrm{EFMs}(n,m)Q_n(r)\exp(jm\theta) \tag{4}$$

式中，

$$\mathrm{EFMs}(n,m)=\frac{1}{2\pi}\int_0^{2\pi}\int_0^1 f(r,\theta)Q_n(r)\exp(-jm\theta)r\mathrm{d}r\mathrm{d}\theta \quad (5)$$

EFMs(n, m)就是指数矩. 对 n 和 m 取不同的整数，可以得到(n, m)阶指数矩. 例如，当 n 等于 0 且 m 等于 0 时，EFMs(0,0)表示(0,0)阶指数矩. 同时，本文定义 Zernike(n, m)表示(n, m)阶 Zernike 矩；RHFMs (n, m)表示(n, m)阶圆谐-傅里叶矩，PCT (n, m)表示(n, m)阶 PCT 矩；PST (n, m)表示(n,m)阶 PST 矩.

如表 1 所示，采用不同的径向基函数，可以得到不同的图像矩.

表 1　不同图像矩的径向基函数

Tab. 1　The radial basis functions of the different image moments

图像矩	径向基函数($Q_n(r)$)
Zernike 矩	$Q_{nl}(r)=\sum_{s=0}^{(n-\lvert l\rvert)/2}(-1)^s r^{n-2s}\times\frac{(n-s)!}{s!((n+\lvert l\rvert)/2-s)!((n-\lvert l\rvert)/2-s)!}$
EFMs	$Q_n(r)=\sqrt{\frac{1}{r}}\exp(j2n\pi r)$
PCT	$Q_n(r)=\cos(\pi n r^2)$
PST	$Q_n(r)=\sin(\pi n r^2)$
RHFMs	$Q_n(r)=\begin{cases}\frac{1}{\sqrt{r}}, n=0\\ \frac{2}{\sqrt{r}}\cos(\pi n r), n\text{ 为偶数}\\ \frac{2}{\sqrt{r}}\sin(\pi(n+1)r), n\text{ 为奇数}\end{cases}$

1.1　经典计算算法

根据文献[8,11]，下面给出指数矩在直角坐标系下的经典计算方法.

式(5)是在极坐标下的表达式，为了得到直角坐标下的表达式，首先要进行极坐标和直角坐标之间的转换. 二维坐标平面上任一点的极坐标(r,θ)和直角坐标(x,y)转换公式如下：

$$x=r\cos\theta, y=r\sin\theta \quad (6)$$

$$r_{x,y}=\sqrt{x^2+y^2}, \theta_{x,y}=\arctan\frac{y}{x} \quad (7)$$

由此可以得到指数矩在直角坐标系下的表达式：

$$\mathrm{EFMs}(n,m)=\frac{1}{2\pi}\iint_{x^2+y^2\leqslant 1} f(x,y)Q_n(r)\exp(-jm\theta)\mathrm{d}x\mathrm{d}y \quad (8)$$

假设图像 $f(i, j)$的大小为 N×N，则令直角坐标系的原点位于图像中心，可以得到

$$r_{i,j}=\sqrt{(c_1 j-c_2)^2+(c_2-c_1 i)^2} \quad (9)$$

$$\theta_{i,j}=\arctan\frac{c_2-c_1 i}{c_1 j-c_2} \quad (10)$$

式中，如果使用外接圆归一法，c_1和 c_2可由下式计算得到.

$$c_1=\frac{\sqrt{2}}{N} \quad (11)$$

$$c_2=\frac{N+1}{\sqrt{2}N} \quad (12)$$

最后可得出直角坐标下指数矩的离散形式的表达式：

$$\mathrm{EFMs}(n,m)=\frac{1}{2\pi N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} f(i,j)Q_n(r_{i,j})\exp(-jm\theta_{i,j}) \quad (13)$$

式(13)中，$f(i, j)$表示图像在(i, j)处的像素值. 同理可以得到其他 4 种图像矩的经典计算方法.

1.2　快速计算算法

图像矩的快速计算方法前面的步骤与图像的经典计算方法是一样的，不同的是最后对离散形式表达式的处理方式. 快速计算方法在得到式(13)的指数矩的离散形式的表达式之后，将表达式分成两部分：图像像素值部分和固定值部分.

假设矩阵 $\boldsymbol{A}$ 和矩阵 $\boldsymbol{B}$ 的行数和列数都相等，则定义 $\boldsymbol{A}*\boldsymbol{B}$ 为 $\boldsymbol{A}$ 点乘 $\boldsymbol{B}$，即矩阵 $\boldsymbol{A}$ 和矩阵 $\boldsymbol{B}$ 中相同位置的元素相乘. 为了便于观察，将式(13)写作：

$$\mathrm{EFMs}(n,m)=\frac{1}{2\pi N^2}\sum\left(\begin{bmatrix}f(1,1)\cdots f(1,N)\\ \cdots\\ f(N,1)\cdots f(N,N)\end{bmatrix}_{N\times N}*\begin{bmatrix}Q_n(r_{1,1})\exp(-jm\theta_{1,1})\cdots Q_n(r_{1,N})\exp(-jm\theta_{1,N})\\ \cdots\\ Q_n(r_{N,1})\exp(-jm\theta_{N,1})\cdots Q_n(r_{N,N})\exp(-jm\theta_{N,N})\end{bmatrix}_{N\times N}\right) \quad (14)$$

通过式(14)可以得到图像像素值部分和固定值部分. 图像像素值部分即式(14)中括号的第一个矩阵，剩余的为固定值部分. 如果图像的分辨率不变，(即式(14)中的 N 不变)，加上指数矩的阶数确定

(即 EFMs(n, m)的 n 和 m),固定值部分的值是不变的. 图像矩的快速算法就是通过保留固定值部分的值来减少计算量,从而提高算法的时间效率. 也就是说只计算固定值部分一次.

此外,文献[13]给出了在极坐标下计算 Zernike 矩的离散表达式. 该离散表达式依然能分解成固定值部分和图像像素值部分,所以同理能得到图像矩的快速算法. 为了计算的方便,在本文实验中,经典计算方法与快速计算方法都是在直角坐标系下实现的.

如果要计算 t 个大小为 8×8 的子块的(0,0)阶指数矩 EFMs(0,0),并假设图像像素值部分和固定值部分分别用矩阵 $\boldsymbol{A}$ 和矩阵 $\boldsymbol{B}$ 表示. 那么经典计算方法的步骤如下:对于每个子块,计算一次矩阵 $\boldsymbol{A}$ 和计算一次矩阵 $\boldsymbol{B}$,然后将它们的点乘的结果求和,得到这个子块的 EFMs(0,0),如此重复 t 次,得到 t 个子块的 EFMs(0,0).

快速计算方法的步骤如下:先计算矩阵 $\boldsymbol{B}$ 一次,然后将结果保存在矩阵 $\boldsymbol{C}$ 中. 对于每个子块,计算一次矩阵 $\boldsymbol{A}$,然后将矩阵 $\boldsymbol{A}$ 和矩阵 $\boldsymbol{C}$ 点乘的结果求和,得到这个子块的 EFMs(0,0),如此重复 t 次,得到 t 个子块的 EFMs(0,0). 与图像矩的经典计算方法不同,快速计算方法这 t 次都是不用计算矩阵 $\boldsymbol{B}$ 的,因为矩阵 $\boldsymbol{B}$ 的结果已经保存在矩阵 $\boldsymbol{C}$ 中. 在这个例子中,经典计算方法计算固定值部分(即矩阵 $\boldsymbol{B}$) t 次,而快速计算方法只计算固定值部分 1 次.

通过上述例子可以得到两个结论:①如果 t 足够大,那么快速计算方法就能减少大量重复无用的计算,从而大幅度减少图像矩计算时间. ②因为固定值部分的值是不变的,所以图像矩经典计算方法和快速计算方法得到的图像矩的值是一样的. 也就是说,假设用图像矩经典计算方法计算某个图像的 EFMs(0,0)为 20,那么用图像矩快速计算方法得到的该图像的 EFMs(0,0)也为 20.

如图 1 所示,在进行图像区域复制粘贴篡改检测时,如果是基于块的方法,首先要对图像进行分块操作. 假设待检测图像分辨率为 $L\times S$,且子块大小为 $b\times b$,则子块由图像左上角进行扫描,每次移动一个像素点,一共能得到$(L-b+1)\times(S-b+1)$个子块. 下一步就要提取每个子块的特征向量. 如果是用图像矩作为特征向量的话,图像矩的经典计算方法要计算固定值部分$(L-b+1)\times(S-b+1)\times N_{\text{dim}}$次,而图像矩的快速计算方法只要计算固定值部分 N_{dim}次. 其中,N_{dim}是特征向量的维数. 图像的分辨率越大,图像矩的经典计算方法所计算的固定值部分越多,花费的运行时间越长. 而只要图像矩的参数(即 n 和 m)确定和分辨率(即 b)确定,固定值部分的值是不变的,没有必要进行多次的重复的计算.

2 实验结果与分析

本文实验使用的图像由两部分组成. 一部分来自 UCID 图像数据库[14],一共有 100 张图像,分辨率为 512×384. 另一部分是来自 GRIP 图像数据库[15], 一共有 80 张图像,分辨率为 768×1024. 实验的运行的硬件条件为:主频为 3.2GHz 的 Intel core i5 的处理器, 4GB 内存; 软件环境为 Matlab R2012a.

为了更好地对比采用经典计算算法和采用快速计算算法的运行时间,本文使用了多种不同的图像矩构成的特征向量. 特征向量有 Zernike 矩[3],圆谐-傅里叶矩[4],指数矩[5],PCT[6],PST[7]. 具体可以参考表 2.

表 2 不同图像矩的特征向量维数

Tab. 2 The dimension of the different image moments' feature vectors

图像矩	特征向量维数(N_{dim})
Zernike 矩	12
RHFMs	7
EFMs	4
PCT	8
PST	9

2.1 UCID 数据库

UCID 图像数据库的分辨率为 512×384,如果对图像进行分块操作,并且假设图像子块大小为 8×8,则可以分成(512−8+1)×(384−8+1)=190385 个子块. 以提取文献[3]的 Zernike 矩为例,图像矩的经典计算算法需要计算固定值部分 190385×12=2284620 次,而快速计算方法只需要计算固定值部分 12 次. 因此图像矩快速算法的运行时间大幅度降低.

表 3 为 UCID 数据库采用不同方法计算每张图像的图像矩所用的平均运行时间,一共采用了 5 种特征向量,3 种子块大小,每种特征向量都分别采用图像矩的经典计算方法和快速计算方法. 特征向量有 Zernike 矩,圆谐-傅里叶矩,指数矩,PCT,PST.

子块的大小分别为 8×8,16×16 和 24×24. 表中所用的时间单位为秒.

表 3　UCID 数据库采用不同方法计算图像矩的时间

Tab. 3　The runtime of two different methods in UCID database

方法		运行时间/s		
		8×8	16×16	24×24
Zernike 矩	经典算法	357.83	429.45	545.64
	快速算法	4.08	4.84	6.31
RHFMs	经典算法	180.53	538.04	1111.80
	快速算法	4.34	4.62	5.22
EFMs	经典算法	61.17	183.33	367.58
	快速算法	2.65	2.80	3.13
PCT	经典算法	125.51	380.88	770.13
	快速算法	4.91	5.13	5.98
PST	经典算法	145.98	446.29	892.76
	快速算法	5.93	6.07	6.57

为了更好地表述表格中数据的含义,下面以 357.83 s 为例,阐述它表达的含义. 为了检测分辨率为 512×384 的 UCID 数据库图像是否经过复制粘贴篡改,需要对图像分成重叠的 8×8 的子块,然后提取每个子块的 12 维特征向量 Zernike 矩. 平均一张 UCID 图像提取所有子块特征向量所用的时间为 357.83 s.

表 3 中,横向来看,随着子块大小的增大,不论是图像矩经典计算算法还是快速计算算法,运行时间都有所增加,但是通过计算可以发现,图像矩经典计算方法的增幅要大于快速计算方法.

从表 3 的纵向来看,图像矩经典计算算法和快速计算算法的运行时间都正相关于特征向量的维数. 特征向量的维数越多,运行时间越长,但是不论采用什么特征向量,图像矩经典计算方法所用的时间要远大于快速算法所用的时间. 这是因为图像矩经典算法在反复的计算图像矩固定值部分,浪费了大量的时间.

2.2　GRIP 数据库

为了更好地对比分别采用图像矩经典计算方法和快速计算方法所用的运行时间,本文增加 GRIP 图像数据库的 80 张图像的实验结果.

检测 GRIP 图像数据库的分辨率为 768×1024,如果对图像进行图 1 所示分块操作,并且假设图像子块大小为 8×8,则可以分成(768−8+1)×(1024−8+1)=773937 个子块. 以提取圆谐-傅里叶矩为例,图像矩的经典计算算法需要计算固定值部分 773937×7=5417559 次,而快速计算方法只需要计算固定值部分 7 次. 固定值部分的计算量的大幅度降低提高了算法的运行效率.

表 4 为 GRIP 数据库的图像采用不同方法计算图像矩所用的平均运行时间. 实验方法与 UCID 数据库的类似. 表中的时间单位为秒.

因为子块大小为 16×16 和 24×24 时,图像矩的经典计算方法的运行时间太长,所以只选了 GRIP 的 40 张图像进行实验,并得到平均每张图像的运行时间. 其他剩余的实验都是用 GRIP 的 80 张图像得到的实验结果.

表 4　GRIP 数据库采用不同方法计算图像矩的时间

Tab. 4　The runtime of two different methods in GRIP database

方法		运行时间/s		
		8×8	16×16	24×24
Zernike 矩	经典算法	1463.05	1792.73	2281.40
	快速算法	16.59	19.28	28.20
RHFMs	经典算法	730.88	2278.30	4692.44
	快速算法	16.59	19.28	22.25
EFMs	经典算法	245.01	757.72	1552.51
	快速算法	10.91	11.71	13.50
PCT	经典算法	508.37	1578.42	3288.41
	快速算法	20.64	21.76	24.77
PST	经典算法	591.87	1808.44	3806.52
	快速算法	22.58	24.84	27.48

从表 4 的横向来看,随着子块大小的增大,不论是图像矩经典计算算法还是快速计算算法,运行时间都有所增加,但图像矩经典计算方法的增幅要大于快速计算方法. 例如,特征向量为 7 维的圆谐-傅里叶矩(RHFMs),当子块大小从 16×16 变成 24×24 时,经典计算方法的运行时间从 2278.30 s 变成了 4692.44 s,提高了 105%,而快速计算方法从 19.28 s变成了 22.25 s,只提高了 15%.

在表 4 中,纵向来看,图像矩经典计算方法所用的时间要远大于快速算法所用的时间. 例如,当特征向量为 8 维的 PCT,同时子块大小为 24×24 时,经典计算方法的运行时间是 3288.41 s,而快速计算方法只要 24.77 s,运行时间提高了 131 倍左右.

综合 UCID 数据库和 GRIP 数据库的图像矩经

典计算方法和快速计算方法的实验结果来看,图像矩的快速算法能大幅度减少图像区域复制粘贴篡改检测在提取子块图像矩的运行时间.如果图像的分辨率越大,提取的图像矩维数越多,相比于图像矩的经典计算方法,快速计算方法提高的时间就越多.原因有两个:第一,每个子块的分辨率是固定的.第二,子块数目非常多,也就是要提取的图像矩非常多.如图1所示,图像在经过分块步骤之后,得到的子块数目都是上万甚至是上百万的,分辨率越大,子块数目越多.在这种情况下,图像矩的快速算法就有非常大的时间优势,因为它减少了上万次甚至是上千万次的固定值部分的计算.而在特征提取步骤时间的大幅度减少,能大幅度提高整个图像区域复制粘贴篡改检测算法的时间效率.

3 结论

针对传统的计算图像矩的方法时间效率不高的问题,本文提出一种快速计算指数矩的方法.图像矩的快速计算方法通过减少固定值部分的计算次数来提高算法的时间效率.只要图像的分辨率不变,所要提取图像矩的图像越多,图像矩的快速计算算法减少的运行时间越明显.实验结果表明,相比于图像矩的经典计算方法,快速计算方法在图像复制粘贴篡改检测中大幅度减少了运行时间.此外,还能将图像矩的快速计算方法运用到其他方面,如提取整个人脸图像数据库的图像矩特征.

参考文献(References)

[1] CHRISTLEIN V, RIESS C, JORDAN J, et al. An evaluation of popular copy-move forgery detection approaches[J]. IEEE Transactions on Information Forensics and Security, 2012, 7(6): 1841-1854.

[2] RYU S J, LEE M J, LEE H K. Detection of copy-rotate-move forgery using Zernike moments[C]// Proceedings of the 12th International Conference on Information Hiding. Springer, 2010: 51-65.

[3] RYU S J, KIRCHNER M, LEE M J, et al. Rotation invariant localization of duplicated image regions based on Zernike moments [J]. IEEE Transactions on Information Forensics and Security, 2013, 8(8): 1355-1370.

[4] 秦娟, 李峰, 向凌云, 等. 采用圆谐-傅里叶矩的图像区域复制粘贴篡改检测[J]. 中国图象图形学报, 2013, 18(8): 919-923.
QIN Juan, LIFeng, XISNG Lingyun, et al. Detection of image region copy-move forgery using radial harmonic Fourier moments[J]. Journal of Image and Graphics, 2013, 8(8): 919-923.

[5] 赖玥聪, 黄添强, 蒋仁祥. 采用指数矩的图像区域复制粘贴篡改检测[J]. 中国图象图形学报, 2015, 20(9): 1212-1221.
LAI Y C, HUANG T Q, JIANG R X. Image region copy-move forgery detection based on Exponential-Fourier moments[J]. Journal of Image and Graphics, 2015,20(9): 1212-1221.

[6] LI Y N. Image copy-move forgery detection based on polar cosine transform and approximate nearest neighbor searching [J]. Forensic science international, 2013, 224(1): 59-67.

[7] LI L D, LI S S, ZHU H C, et al. Detecting copy-move forgery under affine transforms for image forensics [J]. Computers & Electrical Engineering, 2014, 40(6): 1951-1962.

[8] 姜永静. 指数矩及其在模式识别中的应用[D]. 北京: 北京邮电大学, 2011.

[9] TEAGUE M R. Image analysis via the general theory of moments [J]. Journal of the Optical Society of America, 1980, 70(8): 920-930.

[10] REN H P, PING Z L, BO W, et al. Multidistortion-invariant image recognition with radial harmonic Fourier moments[J]. Journal of the Optical Society of America, 2003, 20(4): 631-637.

[11] 孟敏, 平子良. 基于指数矩的图像分解和重建 [J]. 内蒙古师范大学学报, 2011, 40(3): 258-260.

[12] YAP P T, JIANG X D, KOT A C. Two-dimensional polar harmonic transforms for invariant image representation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(7): 1259-1270.

[13] MUKUNDAN R, RAMAKRISHNAN K R. Fast computation of Legendre and Zernike moments [J]. Pattern Recognition, 1995, 28(9): 1433-1442.

[14] UCID. Uncompressed Colour Image Database [DB/OL]. http://homepages. lboro. ac. uk/~ cogs/datasets/ucid/ucid. html.

[15] GRIP. Image processing research group of Università degli Studi di Napoli, Federico II[DB/OL]. http://www. grip. unina. it/.