

# Research on Remote Sensing Image Embedding by Digital Tag as Dark Watermark

Jianbo Ouyang Yabin Lin

Yunnan Remote Sensing Center, Kunming, Yunnan, 650032, China

## Abstract

Remote sensing image dark watermark embedding technology is a means of embedding copyright tags in remote sensing images to achieve copyright protection for image distribution department. The paper introduces how to embed and extract dark watermarks in the spatial and transformation domains, and verifies the performance of various embedding methods in watermark attacks through a series of experiments.

## Keywords

remote sensing image; dark watermark; transform domains; spatial domains

## 遥感影像数字标签暗水印嵌入技术研究

欧阳剑波 林亚斌

云南省遥感中心, 中国·云南昆明 650034

## 摘要

遥感影像暗水印嵌入技术是在遥感影像中嵌入版权标识来实现影像分发单位版权保护的一种手段, 论文介绍了如何在空间和变换域进行暗水印嵌入和提取的方法, 并通过一序列试验验证了各种不同嵌入方法在水印攻击中的表现。

## 关键词

遥感影像; 暗水印; 变换域; 空间域

## 1 引言

图像暗水印技术是指用技术手段在数字图像数据中嵌入对于大部分感知系统来说都是透明不可见的特定密文信息, 用于实现对图像标识来源、版权以及完整性认证的一种技术。这种技术可以让图像发行方进行标识版权, 用于作为图像文件被盗版后的事后维权取证; 也可以成为对保密性保真度要求较高的学术期刊、新闻、医学、电子发票、法律文书等图片的应用场景中用于完整性校验。

图像暗水印按照其应用分为易失性和鲁棒性两种技术路线刚好相反的方向, 前者要求图像中所嵌入暗水印具有易失性, 即载体图像只要轻微改变该水印即无法被检测到, 一般用于原版性验证。后者要求嵌入图像的水印具有鲁棒性, 一般图像修改操作很难将其抹去, 一般用于版权标识, 论文所介绍的正是后一种。

## 2 遥感影像与暗水印

遥感影像作为国家基础设施建设与地球信息科学研究

【作者简介】欧阳剑波(1975-), 男, 中国云南丽江人, 硕士, 正高工程师, 从事遥感影像处理研究。

的重要基础地理信息数据, 是国民经济中重要的战略资源, 在各行各业中都有着广泛应用, 随着地理信息数据介入民众日常生活程度的程度越来越高, 数据的获取、复制与传播也越来越方便, 数据的保密安全面临威胁外遭受盗版、窃取和私自扩大使用范围的情况等也越来越多, 因此面向数据生产和数据统筹单位的侵权行为也越来越多, 而因为数字产品易复制、易修改的特点, 如果缺乏有效的技术手段进行防范, 会给版权主张带来很大挑战<sup>[1]</sup>。

2017年4月新修订的《中华人民共和国测绘法》明确提出: “测绘成果保管单位应当采取措施保障测绘成果的完整和安全。”随着地理数据应用和共享的广泛展开, 如何研发并使用具有实用性、可靠性和安全性的地理数据数字版权信息嵌入产品需求会变得越来越迫切。

## 3 数字暗水印技术

数字暗水印技术的基本思路是通过一定的算法, 将特定的信息嵌入宿主对象(图像)中, 与宿主对象的数据融合在一起, 要求嵌入信息尽量不影像到宿主图像的内容, 达到不为人类视觉系统或机器所感知的程度, 并要求具有一定的鲁棒性, 不容易被破坏。按照信息嵌入宿主图像的维度不同,

一般有空域嵌入和变换域嵌入两种方式。

### 3.1 空间域嵌入

空间域对于数字图像信息来说就是图像亮度 - 色度空间，是研究坐标位置与像素 RGB 三原色亮度值函数关系的信息空间，空域层面图像是一个具有多层结构的平面数字矩阵，在图像平面上的每一个点代表该位置处图像一个颜色通道的亮度值，在空间域上嵌入信息，就是通过直接改变一序列像素值来实现将需要隐藏的水印信息“调制”到图像上的方法，这种方法很显然是会引起图像特定像素的亮度或颜色值发生改变的，但我们可以通过 LSB (Least Significant Bits) 即“最低有效位”嵌入的方式，也就是每个像素值只改变它最低位上的一个 bit 位来让这种亮度和颜色的改变无法通过人眼察觉出来。因为对于宿主影像来说，任何一个通道的像素亮度改变只是在该值的基础上变化了  $\pm 1$ ，人眼根本无法识别出这种改变，能达到很好的隐藏效果。

图 1 是按 LSB 方法嵌入的示意图。

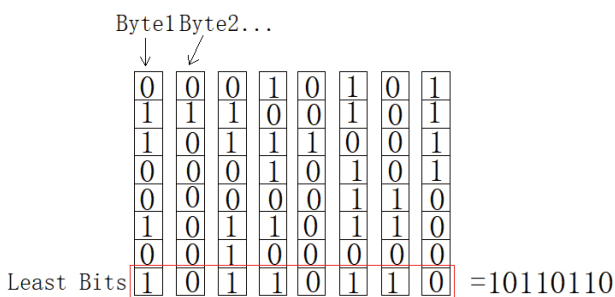


图 1 按 LSB 方法嵌入的示意图

### 3.2 变换域嵌入

变换域是经过一系列正交变换函数将空间域信号变换到可以用“频率”“波数”等表征量上的数据信息空间，对应于数字图像，我们一般通过傅里叶变换、余弦变换、小波变换等可逆的正交变换，将空间域信号变换到用信号幅度与相位角表征的频域（波数）空间<sup>[2]</sup>。论文介绍的主要是使用数字傅里叶变换 (DFT) 和数字余弦变换 (DCT) 来进行的，二者的区别是在变换后的结果序列上，前者变换结果是复数值，后者则是实数值，二者的变换结果高低频分布位置也不一样相对来说，数字余弦变换低频能量更集中，其变换序数更适合用来作为调制信号的载体。

这是离散信号傅里叶变换的公式表达：

$$X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) e^{-j2\pi n_1 k_1 / N_1} e^{-j2\pi n_2 k_2 / N_2}$$

其中  $k_1 = 0, 1, \Lambda, N_1 - 1$ ;  $k_2 = 0, 1, \Lambda, N_2 - 1$ 。

$$x(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} X(k_1, k_2) e^{-j2\pi n_1 k_1 / N_1} e^{-j2\pi n_2 k_2 / N_2}$$

其中  $n_1 = 0, 1, \Lambda, N_1 - 1$ ;  $n_2 = 0, 1, \Lambda, N_2 - 1$ 。

这是离散信号余弦变换的公式表达：

$$X_c(m, n) = \frac{2}{\sqrt{NM}} c(m, n) \left[ \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} x(i, j) \cos \frac{(2i+1)m\pi}{2N} \cos \frac{(2j+1)n\pi}{2M} \right]$$

$$x(i, j) = \frac{2}{\sqrt{NM}} \left[ \sum_{m=0}^{N-1} \sum_{n=0}^{M-1} c(m, n) X_c(m, n) \cos \frac{(2i+1)m\pi}{2N} \cos \frac{(2j+1)n\pi}{2M} \right]$$

式中  $m, i = 0, 1, \Lambda, N - 1$ ;  $n, j = 0, 1, \Lambda, M - 1$ ;

$$c(m, n) = \begin{cases} 1/\sqrt{2} & m = 0, n = 0 \\ 1 & \text{其他} \end{cases}$$

二者的区别在于前者是从实信号到复数频域的变换，所以有幅度谱与相位谱，后者则是实信号到实数频域的变换，只有幅度谱。图 2、图 3 是对同一区域图像分别进行傅里叶变换和余弦变换后的三维频谱图像：

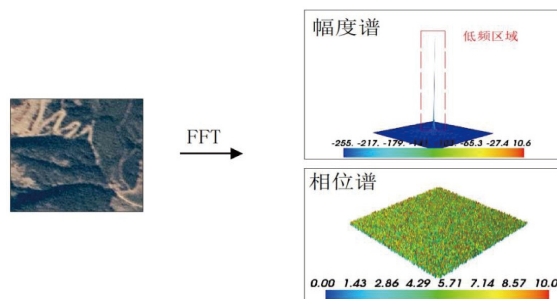


图 2 图像进行 DFT 变换后的频谱

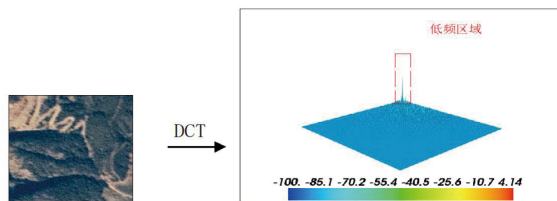


图 3 图像进行 DCT 变换后的频谱

可以看出，遥感影像的频域能量谱主要集中在低频区域，对去中心化后的信号进行变换时，离散傅里叶变换的低频在变换图像的中间，高频在四周，而离散余弦变换的高频在变换图像左上角，低频在右下角，离散余弦变换后的能量更为集中。

在变换域中低频部分对应着源图像整体的色调和颜色变化，轻微修改低频区域都会带来源图像的巨大外观变化，而高频区域对应着图像的细节部分，对其进行修改只会带来清晰度方面的细微变化，人眼对这种变化并不敏感，因此数字暗水印的嵌入其实就是针对图像高频区域的变换系数的一种调制。而因为离散余弦变换的低频部分能量更集中，所以有更大的区域来提供数字水印调制，所以离散余弦变换更适合用来进行数字暗水印的嵌入设计。

## 4 嵌入和提取算法

### 4.1 用 LSB 方法进行嵌入和提取

LSB 方法进行嵌入时是将调制信号按照 1bit 对应基底图像一个 byte 的对应关系通过图像 I/O 读取出一个大小合适的区域，可以是单通道也可以是多通道，如果是用 3 通道同时进行调制则读写区域也可以相应缩小 3 倍，但调制过程

的难度也会相应复杂一点<sup>[3]</sup>。对需要嵌入的通道像素依次考察最低位（其实如果为了尽量减小图像受影响区域可以同时考察多位，笔者就用了一个像素的低4位来调制信息，图像没有受到明显可见的影响），如果最低位与希望调制的信息比特位一致则直接放过，如果不一致则对该位取反，循环完所有像素后这些像素亮度值的最末位就被调制上了希望嵌入的信息。提取过程则更简单：每次取出被调制区域的8个像素数据，将其最末位取出来分别左移与其顺序位一致的位置，比如第1个像素末位提取出来的左移第1位，第2个像素提取出来的左移到第2位……依此类推，8个像素就解调出了一个完整的字节，直到把嵌入信息全部解调出来，就实现了提取，关键代码如下：

代码中 Sign\_mat 是要调制进去的信号数据，Band 是基底图像数据（单通道），N 是被调制的方形区域的边长。

void LSB\_ByteModu(BYTE \*Sign\_mat,BYTE \*Band,int N)// 调制

```
{
    int i,j;
    BYTE mt;
    unsigned char va;
    if(!Sign_mat||!Band) return;
    for(i=0;i<N*N;i++)
    {
        for(j=0;j<8;j++)
        {
            mt=Sign_mat[i];
            mt=mt>>(7-j);
            mt=mt&0x01;
            if(mt==0) Band[i*8+j]=Band[i*8+j]&0xFE;
            else Band[i*8+j]=Band[i*8+j]|0x01;
        }
    }
}
```

int LSB\_ByteDemodu(BYTE \*Band,int N,BYTE \*Sign\_mat)// 解调

```
{
    int i,j;
    BYTE va,mt;
    if(!Band||!Sign_mat) return 1;
    for(i=0;i<N*N;i++)
    {
        va=0;
        for(j=0;j<8;j++)
        {
            mt=Band[i*8+j];
            mt=mt&0x01;
```

```
mt=mt<<(7-j);
            va=va+mt;
        }
        Sign_mat[i]=va;
    }
    return 0;
}
```

## 4.2 用 DFT 系数进行嵌入和提取

从上面的变换公式可以看出，对于二维的图像数据来说，进行离散傅里叶变换或离散余弦变换的计算复杂度达到了 $(O^4)$ 如果直接采用上面变换公式进行<sup>[4]</sup>，那效率将低的可怕，对于动辄上亿像素的遥感影像进行密集的水印嵌入的话几乎没法达到实用化的程度，因此我们可以采用麻省理工开发的开源模块 FFTW3 来进行快速变换，通过不完全试验，对 $200 \times 200$  像素的一块三通道彩色图像区域进行离散余弦变换改用 FFTW3，速度从 4172ms 缩短到了 94ms 效率提升了 44 倍多。用 FFTW3 进行嵌入和提取的关键代码如下：

fftw\_complex \*FFT2D(BYTE \*Sour\_data,int Hei,int Wid)  
// 快速傅里叶变换

```
{
    int i,j;
    if(!Sour_data) return NULL;
    fftw_complex
        *data_in=(fftw_complex *)fftw_malloc(sizeof(fftw_
        complex)*Wid*Hei);
    fftw_complex *fft=(fftw_complex *)fftw_
        malloc(sizeof(fftw_complex)*Wid*Hei);
    fftw_plan plan_f=fftw_plan_dft_2d(Hei,Wid,data_
        in,fft,FFTW_FORWARD,FFTW_ESTIMATE);
```

```
for(i=0;i<Hei;i++)
    {
        for(j=0;j<Wid;j++)
        {
            data_in[i*Wid+j][0]=pow(-1,i+j)*(double)Sour_
            data[i*Wid+j]; data_in[i*Wid+j][1]=0.0;
        }
    }
    fftw_execute(plan_f);
    fftw_destroy_plan(plan_f);
    fftw_free(data_in);
    return fft;
}

fftw_complex *IFFT2D(fftw_complex *fft_data,int Hei,int
Wid)// 快速傅里叶逆变换
{
    {
```

```

int i,j;
fftw_complex
*ifft=(fftw_complex *)fftw_malloc(sizeof(fftw_
complex)*Wid*Hei);
fftw_plan plan_b=fftw_plan_dft_2d(Hei,Wid,fft_
data,ifft,FFTW_BACKWARD,FFTW_ESTIMATE);
fftw_execute(plan_b);
fftw_destroy_plan(plan_b);
for(i=0;i<Hei;i++)
{
for(j=0;j<Wid;j++)
{
ifft[i*Wid+j][0]=pow(-1,i+j)*ifft[i*Wid+j][0]/(double
(Hei*Wid);
ifft[i*Wid+j][1]=pow(-1,i+j)*ifft[i*Wid+j][1]/(double
(Hei*Wid);;
}
}
return ifft;
}
/* 信号调制
fft 为基底图像的傅里叶变换系数矩阵, Wprint 为水印
信息
*/
int SignFFTModulation(fftw_complex *fft,int fft_hei,int
fft_wid,BYTE *Wprint,int Hei,int Wid,int X,int Y)
{
int i,j,i0,j0;;
fftw_complex *fft_wprint;
fft_wprint=FFT2D(Wprint,Hei,Wid);
for(i=0;i<Hei;i++)
{
for(j=0;j<Wid;j++)
{
i0=Y+i;
j0=X+j;
fft[i0*fft_wid+j0][0]=fft_wprint[i*Wid+j][0];
fft[i0*fft_wid+j0][1]=fft_wprint[i*Wid+j][1];
}
}
fftw_free(fft_wprint);
return 0;
}
/*
从傅里叶变换系数中解调, Band 为调制后的数据,
data 为解调出来的信息
*/

```

```

int SignFFTFetch(BYTE *Band,int bmp_hei,int bmp_
wid,BYTE *data,int Hei,int Wid,int X,int Y)
{
int i,j,i0,j0;
unsigned char dd;
fftw_complex *fft,*fft_print,*ifft_print;
fft=FFT2D(Band,bmp_hei,bmp_wid);
fft_print=(fftw_complex*) fftw_malloc(sizeof(fftw_
complex)*Hei*Wid);
for(i=0;i<Hei;i++)
{
for(j=0;j<Wid;j++)
{
i0=Y+i;
j0=X+j;
fft_print[i*Wid+j][0]=fft[i0*bmp_wid+j0][0];
fft_print[i*Wid+j][1]=fft[i0*bmp_wid+j0][1];
}
}
fftw_free(fft);
HighPassFilter(fft_print,Hei,Wid,0.1);
ifft_print=IFFT2D(fft_print,Hei,Wid);
for(i=0;i<Hei;i++)
{
for(j=0;j<Wid;j++)
{
if((ifft_print[i*Wid+j][0]-(int)ifft_print[i*Wid+j
][0]>0.499) data[i*Wid+j]=255-((unsigned char)ifft_
print[i*Wid+j][0]+1);
else data[i*Wid+j]=255-(unsigned char)ifft_
print[i*Wid+j][0];
}
}
fftw_free(fft_print);
fftw_free(ifft_print);
return 0;
}

```

### 4.3 用 DCT 系数进行嵌入和提取

```

void FFTW3_Dct(BYTE* mat,double *mat_dct,int N) //
快速 dct 变换
{
double *In=new double[N*N];
fftw_plan plan;
int NN=N*N;
for(int i=0;i<NN;i++) In[i]=(double)mat[i];
plan = fftw_plan_r2r_2d(N,N,In,mat_dct,FFTW_
REDFT10,FFTW_REDFT10,FFTW_ESTIMATE);

```

```

fftw_execute(plan);
delete []In;
fftw_destroy_plan(plan);
}
void FFTW_IDct(double *dct,double *idct,int N) // 快速
dct 逆变换
{
fftw_plan plan;
int NN;
NN=N*N;
double NN4;
NN4=NN*4.0;
plan=fftw_plan_r2r_2d(N,N,dct,idct,FFTW_
REDFT01,FFTW_REDFT01,FFTW_ESTIMATE);
fftw_execute(plan);
fftw_destroy_plan(plan);
for(int i=0;i<NN;i++) idct[i]=idct[i]/NN4;
}
int DCTModu(BYTE *Mat,int M,BYTE *Sign_mat,int N)
//dct 系数调制
{
double *Dct,*IDct,*Dct_Sign;
int x,y,flag;
Dct=FFTW3_Dct3Band(Mat,M);
Dct_Sign=FFTW3_Dct3Band(Sign_mat,N);
flag=0;
x=M-1;
y=M-1;
for(int i=0;i<N;i++)
{
for(int j=0;j<N;j++)
{
flag=Zig(&x,&y,M,flag); // 对像素位置进行蛇型遍历
Dct[y*M*3+x*3]=Dct_Sign[i*N*3+j*3]/10.0; // 高频
序数嵌入前幅度压缩到原来的 1/10
Dct[y*M*3+x*3+1]=Dct_Sign[i*N*3+j*3+1]/10.0;
Dct[y*M*3+x*3+2]=Dct_Sign[i*N*3+j*3+2]/10.0;
}
}
delete []Dct_Sign;
IDct=FFTW_IDct3Band(Dct,M);
delete []Dct;
IDctToBands(IDct,Mat,M,M);
delete []IDct;
return 0;
}

```

```

int DCTDemodu(BYTE *Mat,int M,BYTE *Sign_mat,int
N) //dct 系数解调
{
double *Dct,*Dct_sign,*IDct_sign;
int x,y,flag;
Dct=FFTW3_Dct3Band(Mat,M);
Dct_sign=new double[N*N*3];//74*74*3
flag=0;
x=M-1;
y=M-1;
for(int i=0;i<N;i++)
{
for(int j=0;j<N;j++)
{
flag=Zig(&x,&y,M,flag); //200*200 像素中蛇型遍历
Dct_sign[i*N*3+j*3]=Dct[y*M*3+x*3]*100.0; // 恢
复前把序数幅度增大 10 倍
Dct_sign[i*N*3+j*3+1]=Dct[y*M*3+x*3+1]*100.0;
Dct_sign[i*N*3+j*3+2]=Dct[y*M*3+x*3+2]*100.0;
}
}
delete []Dct;
IDct_sign=FFTW_IDct3Band(Dct_sign,N);
delete []Dct_sign;
IDct2Bands_View(IDct_sign,Sign_mat,N,N);
delete []IDct_sign;
return 0;
}

```

## 5 鲁棒性测试

因为二维码能够方便表达大量信息，又具有很强的容错能力，所以可以选用二维码作为暗水印调制信息，通过以上三种嵌入方法在遥感影像中嵌入二维码信息，下面对这些嵌入信息后的影像区域进行水印攻击的一些试验<sup>[5]</sup>。

图4是LSB嵌入的暗水印分别经历亮度拉伸、压缩后的表现。



图4 LSB嵌入的暗水印分别经历亮度拉伸、压缩后的表现

图5是DFT系数中嵌入的暗水印分别是原图经历亮度拉伸、压缩、色平衡调整、去色、锐化、钝化(模糊)、色调调整、加入椒盐噪声的表现:

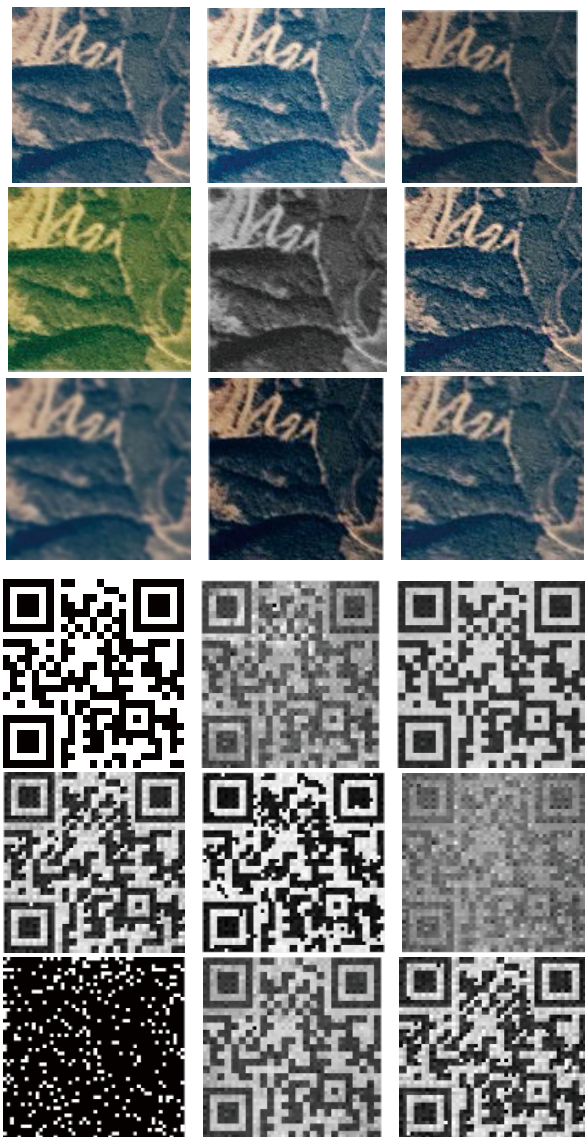


图5 DFT系数中嵌入的暗水印

DCT系数中嵌入的暗水印与在DFT中嵌入的表现类似,但效果更理想,鲁棒性更高。

## 6 结论

在进行一序列暗水印嵌入和提取的试验后可以得出以下的一些结论:

①在空间域中嵌入影像的暗水印能接受某些空间域变换的一些水印攻击,比如裁剪等,但对频域的攻击很敏感,轻微调整亮度色度都可以让嵌入信息失效,因此空间域嵌入的暗水印可以只作为易失型的脆弱水印使用。

②变换域的水印在对抗频域的攻击,比如调色、调亮度、去色等等具有相当强的鲁棒性。

③锐化、钝化及椒盐噪声对各种方式调制的暗水印都有较大影响,但离散余弦变换序数调制相对具有最高鲁棒性。

④暗水印调制信号相当于引入被调制图像的噪声,通过对调制基底图像的变换域系数进行适当的位置编码,可以让调制信号尽可能不泄露到中、低频段中,能最大限度保证暗水印的透明度。

⑤在实际应用中可以将空间域的暗水印嵌入与变换域的暗水印嵌入结合起来,可以最大限度保证效率、提升鲁棒性。

以上只是一些个人经验,希望能与广大读者相互学习,共同进步。

## 参考文献

- [1] 彭金虎.DCT域内的数字图像加密算法[J].煤炭技术,2013(8).
- [2] 朱长青.地理数据数字水印和加密控制技术进展[J].测绘学报,2017(10).
- [3] 董佳宾.多图像加密技术研究[J].上海师范大学学报,2018(5).
- [4] 文小爽.基于频域的彩色图像加密算法研究[D].郑州:河南大学硕士学位论文,2018.
- [5] 张威.空间域和频率域相结合的图像水印研究[J].辽宁石油化工大学学报,2017(6).