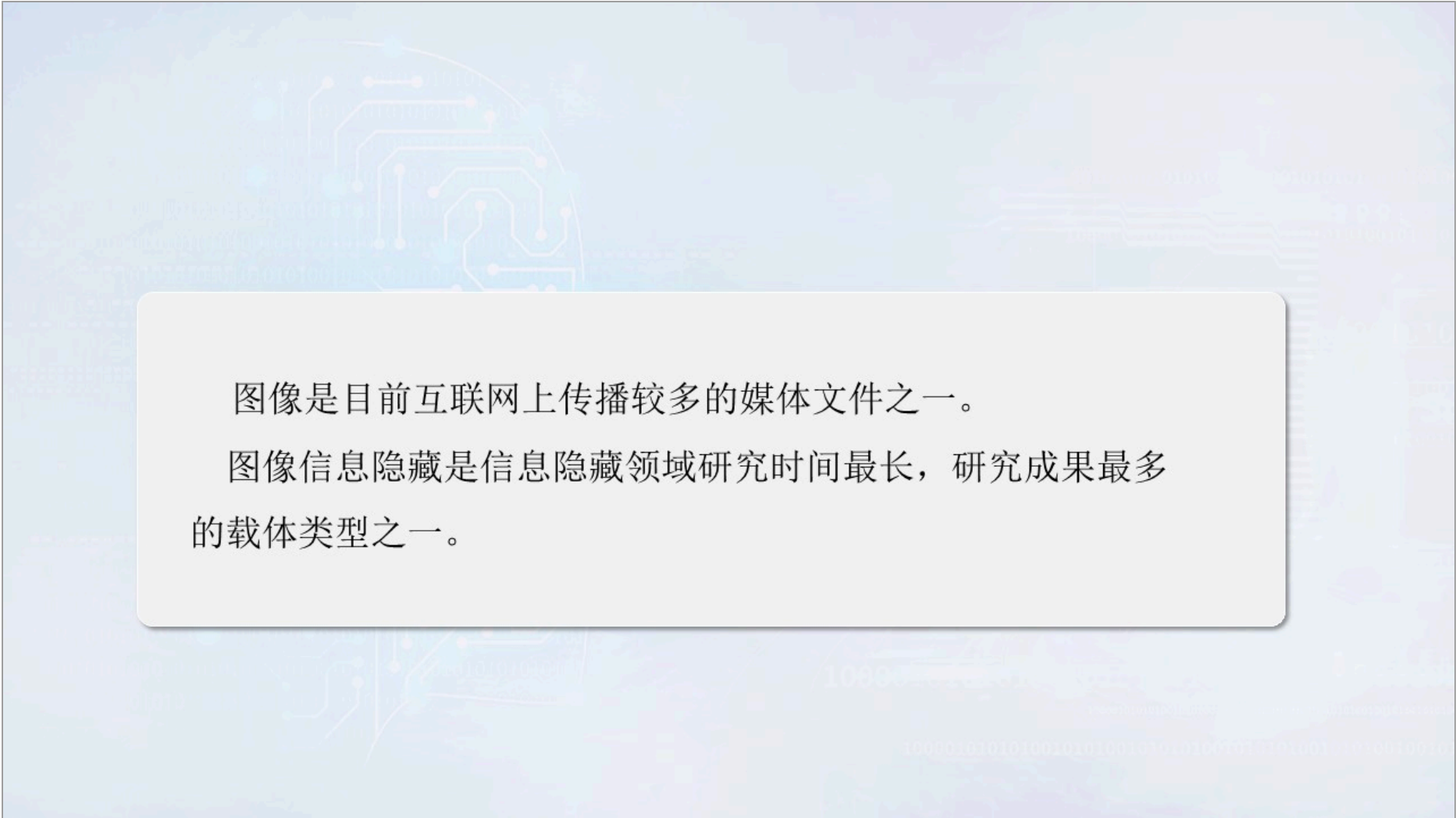




## 图像的位平面概念



图像是目前互联网上传播较多的媒体文件之一。

图像信息隐藏是信息隐藏领域研究时间最长，研究成果最多的载体类型之一。



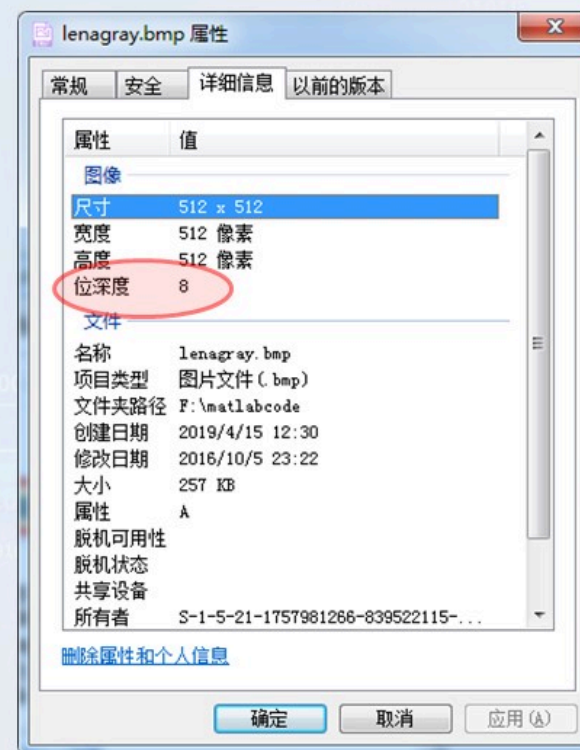
## 数字图像中哪些地方可以用来隐藏信息?

主要考察像素的各个比特对图像的贡献:

- ✓ 贡献大的保留;
- ✓ 贡献微弱的则可用于隐藏秘密信息的比特。



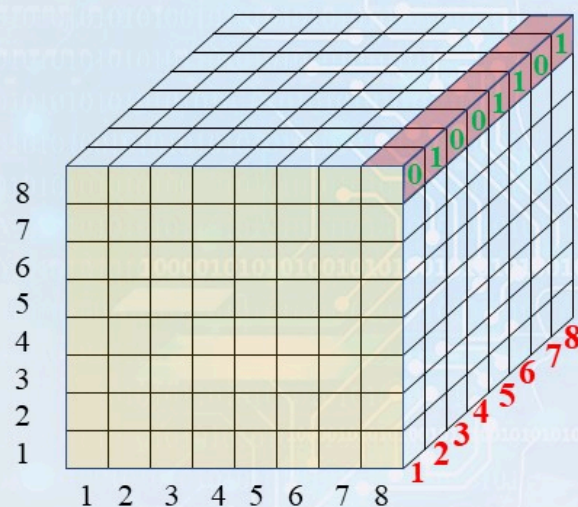
## 256级灰度图像







## 图像的位平面概念



图像像素的灰度表示

图像的数据表示:

- bit /像素点: 256色(0 ~ 255)
- 各像素位置相同的位形成一个平面, 称为“位平面”。
- 不同位平面的重要程度不同

高位: 图像信息

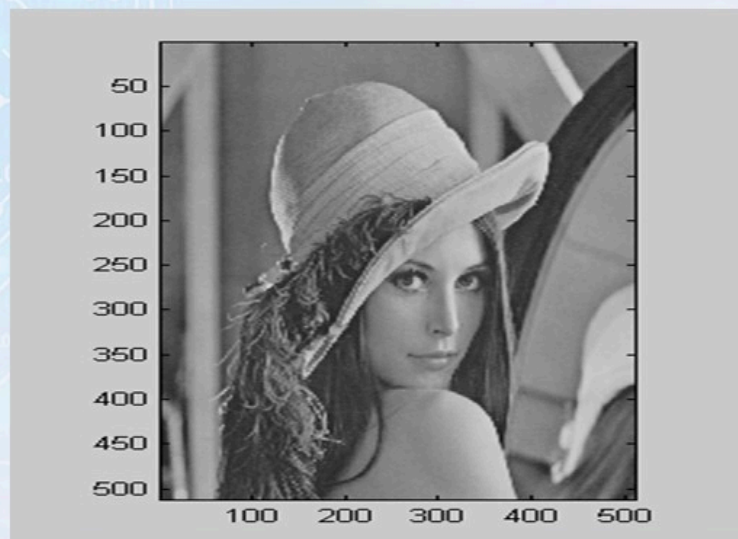
低位: 噪声

人眼对随机噪声不敏感。



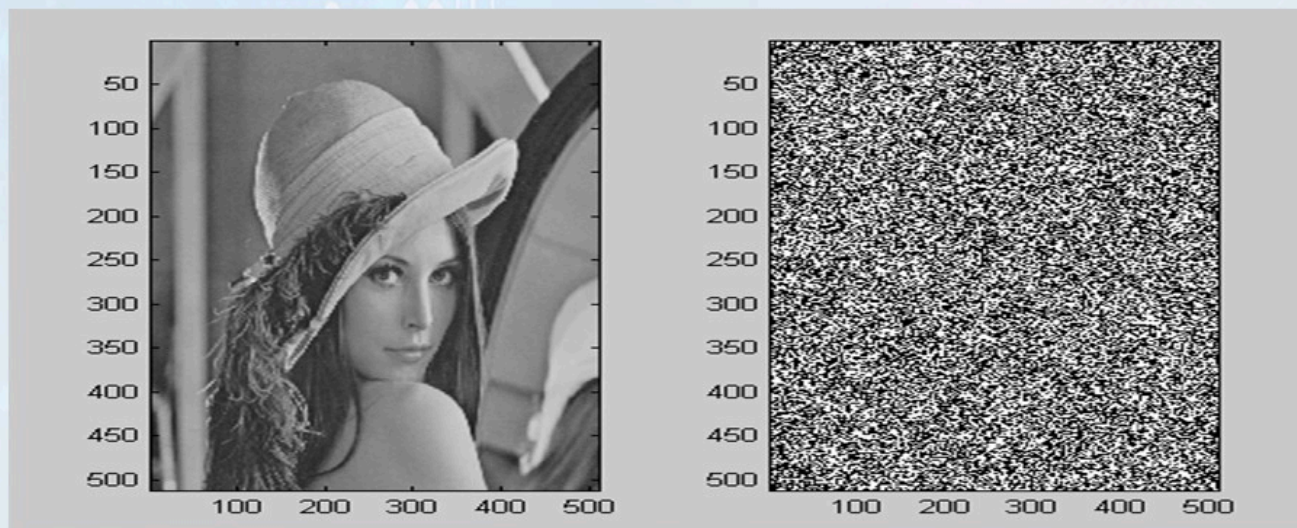
## 图像的位平面概念

原始图像 (8bit 灰度BMP图像)



## 1. 图像各个位平面的作用

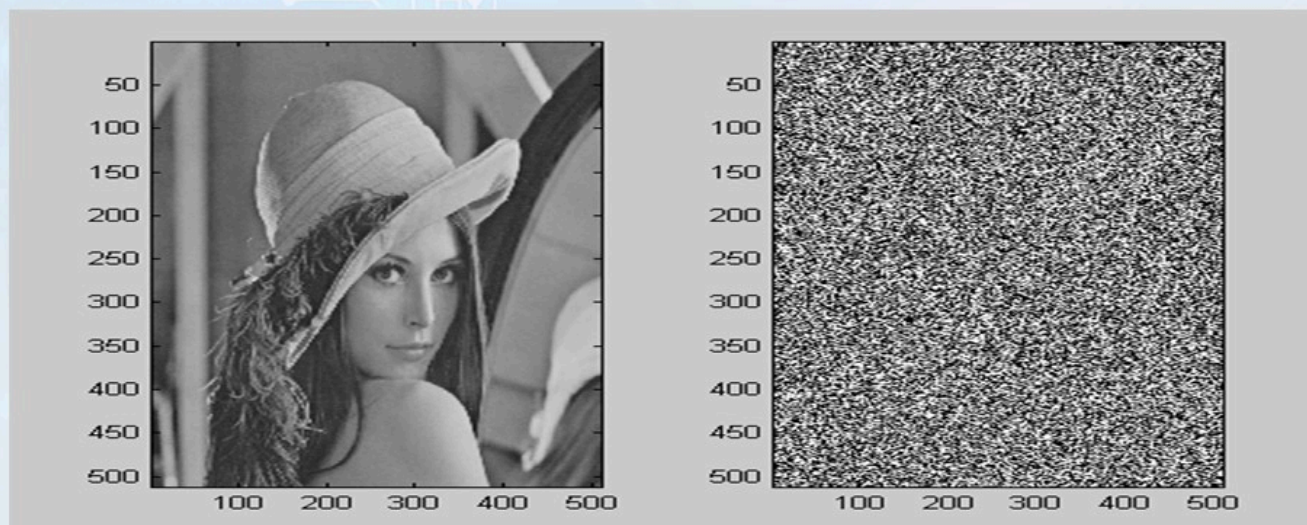
去掉第一个位平面的Lena图像和第一个位平面





## 2. 图像各个位平面的作用

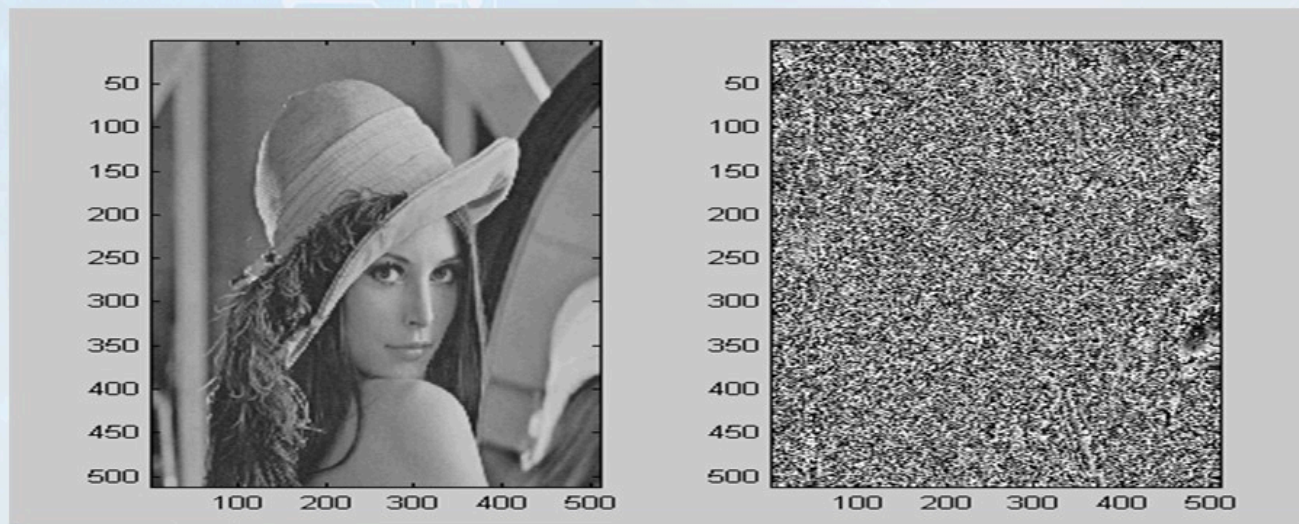
去掉第1 - 2个位平面的Lena图像和第1 - 2个位平面





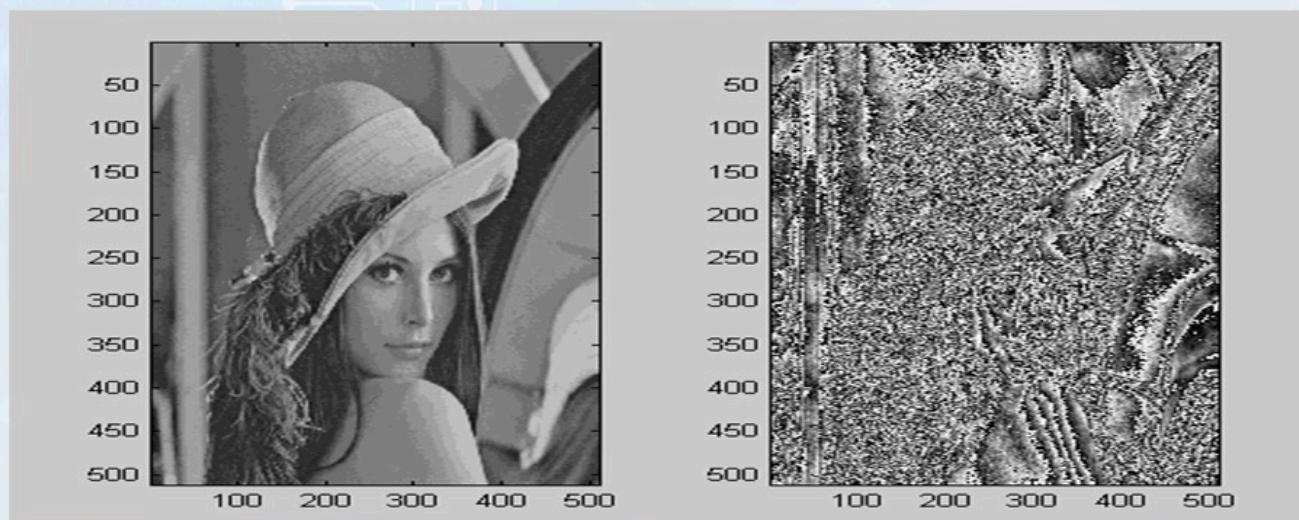
### 3. 图像各个位平面的作用

去掉第1 - 3个位平面的Lena图像和第1 - 3个位平面



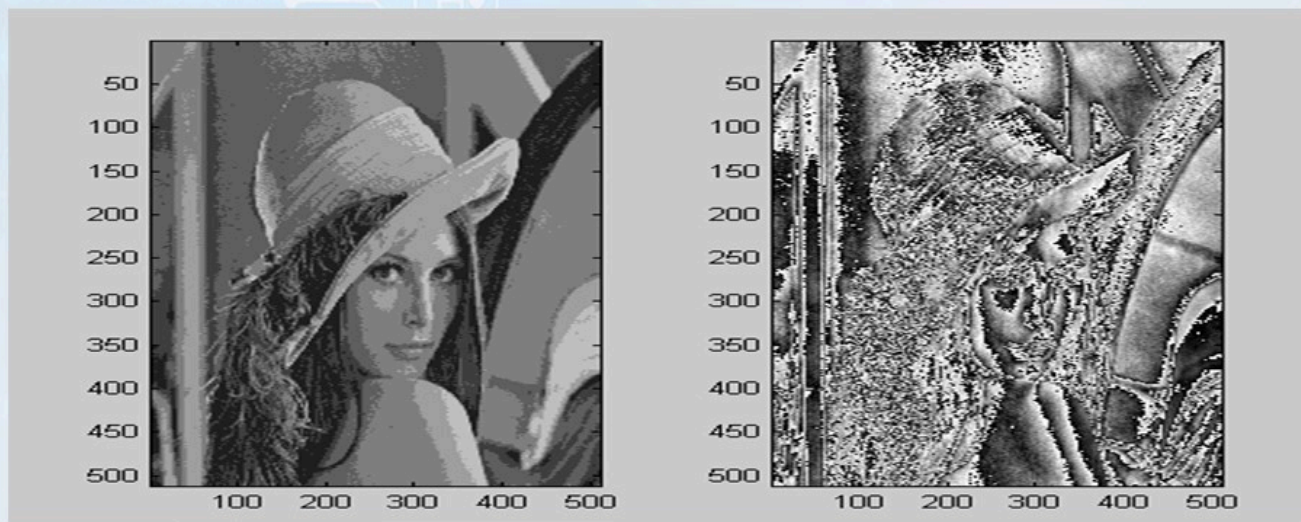
## 4. 图像各个位平面的作用

去掉第1 - 4个位平面的Lena图像和第1 - 4个位平面



## 5. 图像各个位平面的作用

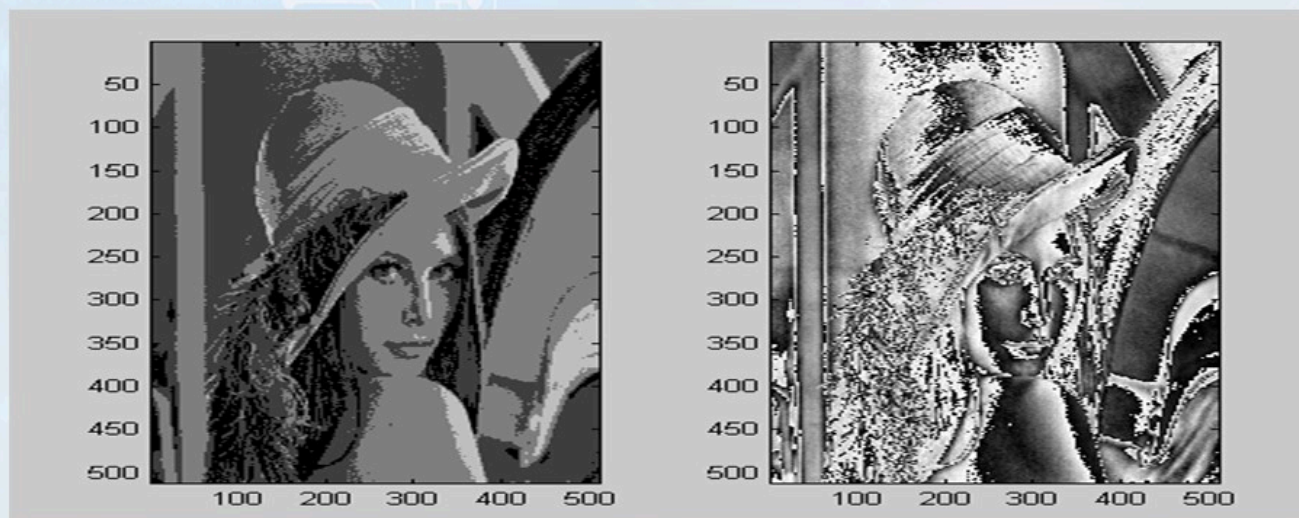
去掉第1 - 5个位平面的Lena图像和第1 - 5个位平面





## 6. 图像各个位平面的作用

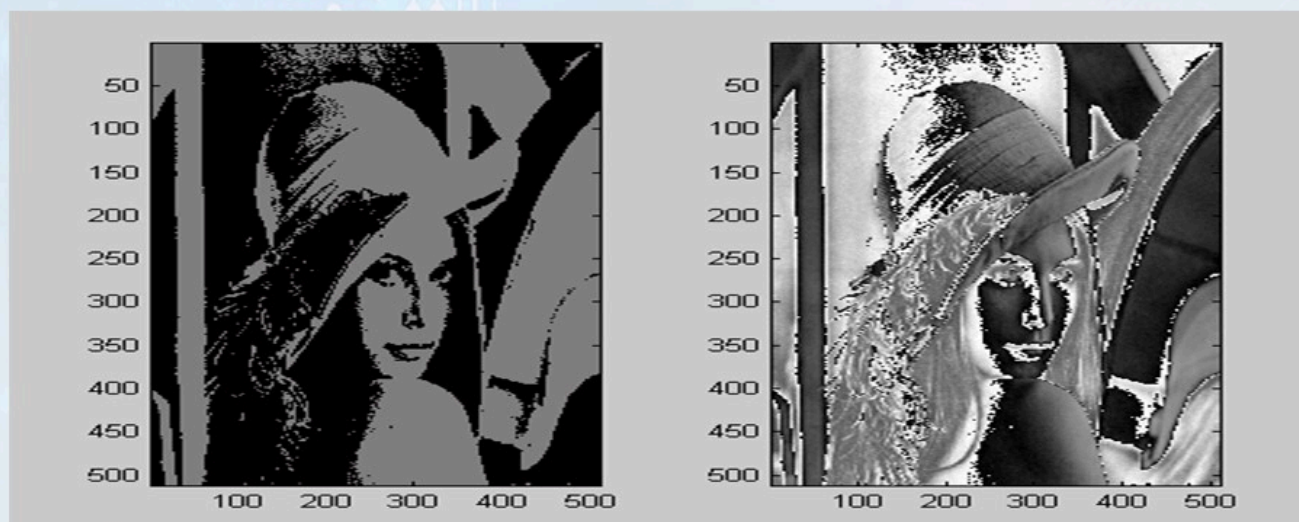
去掉第1 - 6个位平面的Lena图像和第1 - 6个位平面





## 7. 图像各个位平面的作用

去掉第1 - 7个位平面的Lena图像（即第八个位平面）  
和第1 - 7个位平面



# 位平面算法：相关函数

1、获取图像x的行数和列数：

行数  $m = \text{size}(x, 1)$ ; 列数  $n = \text{size}(x, 2)$ ;

也可以使用： $[m, n] = \text{size}(x)$ ;

2、bitget

$C = \text{bitget}(A, \text{bit})$

提取A的bit位的值

3、bitset

$C = \text{bitset}(A, \text{bit}, v)$

将A的bit位的值设为v，(v的取值为0或1)。

## 位平面算法: 提取第1位平面

```
x=imread('lenagray.bmp');  
[m,n]=size(x);  
for i=1:m  
    for j=1:n  
        c(i,j)=bitget(x(i,j),1);  
    end  
end  
figure  
imshow(c, []);  
title('第一位平面');
```



无法显示该图片。

常用的原型如下:

```
imshow(I)
imshow(I,[low high])
imshow(filename)
```

下面分别介绍常用的原型:

**imshow(I)**—在窗口中显示图像I, 图像I可以是灰度图像, 也可是RGB真彩色图像, 也可以是二值图像。对于二值图像, imshow将值为0(零)的像素显示为黑色, 将值为1的像素显示为白色。对于灰度图像和RGB图像, 则按0~255的灰度级显示。

**imshow(I,[low high])**—根据向量[low high]来显示灰度图像, 此时函数imshow()将小于等于low的值显示为黑色, 将大于等于high的值显示为白色, 介于low和high之间的值显示为不同程度的灰色。

如果不指定low和high的值, 则退变为imshow(I,[]), 此时low的值为图像I中的最小值, high为图像I中的最大值。相当于对I作了归一化处理后才显示。

**imshow(filename)**—filename为图像文件的路径和名字, 示例如下:

```
imshow('F:\material\images\P0003-view-01.jpg');
```

使用上面的语句可以直接显示图片“P0003-view-01.jpg”

由以上原型, 我们得到使用函数imshow()应该注意的地方。

- 1 如果图像是做运算得到的, 使其数据范围不以0~255的256个灰度等级为基准, 则应归一化后再显示图像, 即需要使用语句“imshow(I,[])”来显示图像, 否则图像的显示效果会因为对比度不够或数据截断而大打折扣, 甚至让自己误认为自己的运算有问题。
- 2 显示硬盘中的图片文件, 并不需要先使用函数imread()进行读取后再用函数imshow()显示, 直接使用imshow(filename) 即可, 即下面的代码:

```
I= imread('F:\material\images\P0003-view-01.jpg');
imshow(I);
```

1  
2

可直接替换为:

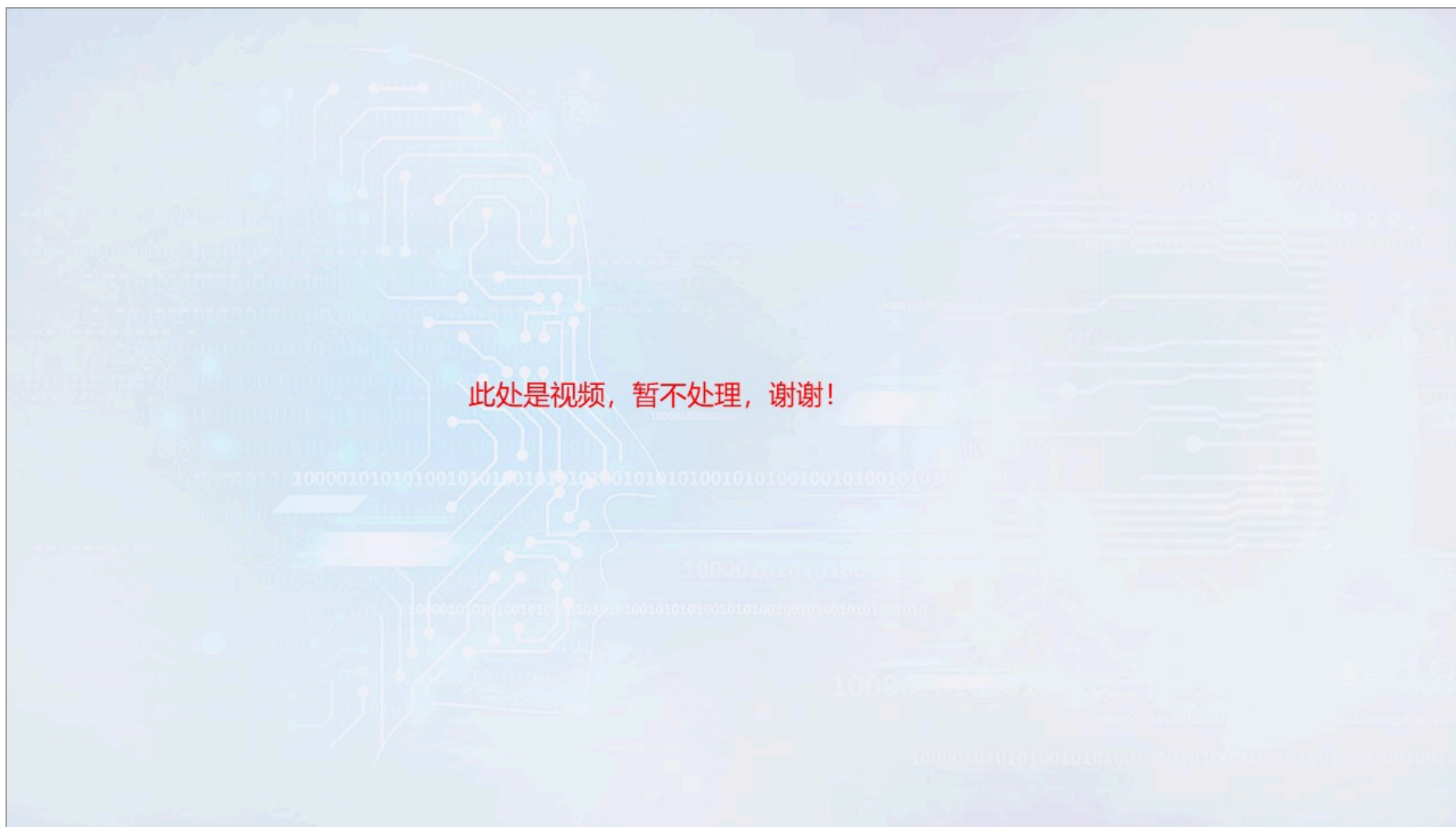
```
imshow('F:\material\images\P0003-view-01.jpg');
```



# 位平面算法

依次提取1 ~ 8个位平面

```
x=imread('lenagray.bmp');  
[m,n]=size(x);  
for t=1:8  
    for i=1:m  
        for j=1:n  
            c(i,j)=bitget(x(i,j),t);  
        end  
    end  
    figure; imshow(c,[]); title(t)  
end
```



此处是视频，暂不处理，谢谢！

```
c=imread('lenagray.bmp'); m=size(c,1); n=size(c,2);y=zeros(m,n);
for t=1:8
    for i=1:m
        for j=1:n
            x(i,j)=bitget(c(i,j),t);
        end
    end
    figure;imshow(x,[]);title(['第',num2str(t),'个位平面']);
    for i=1:m
        for j=1:n
            y(i,j)=bitset(y(i,j),t,x(i,j));
        end
    end
    figure;imshow(y,[]);title(['第1-',num2str(t),'个位平面']);
end
```

依次提取并显示第1、2、3、...、8个位平面。

依次显示1、1-2、1-3、1-4、...、1-8个位平面。

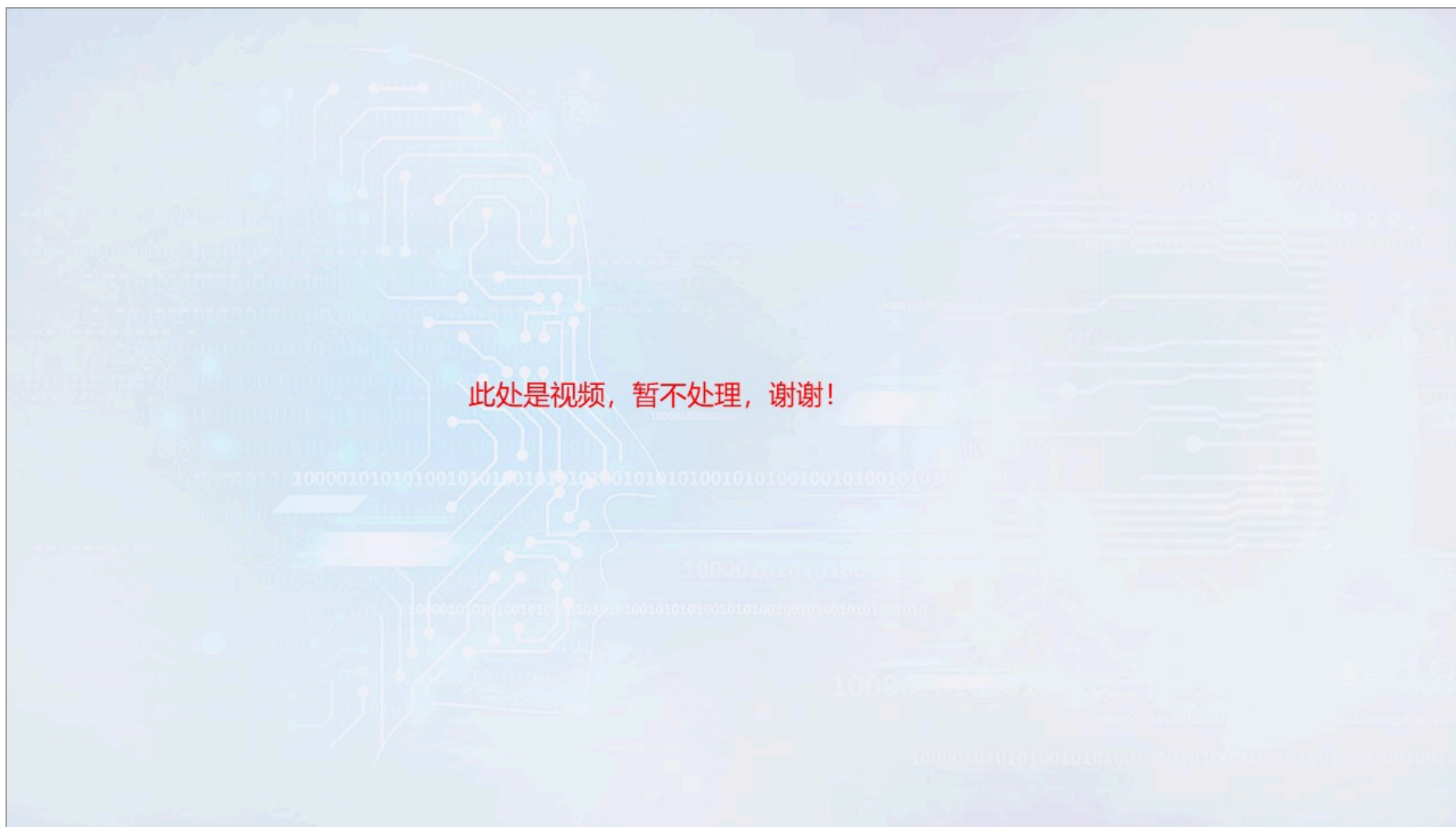




## 位平面算法

去除第1位平面

```
x=imread('lenagray.bmp');  
imshow(x); title('原图像');  
[m,n]=size(x);  
for i=1:m  
    for j=1:n  
        x(i,j)=bitset(x(i,j),1,0);  
    end  
end  
figure; imshow(x);  
title('去除第一位平面');
```



此处是视频，暂不处理，谢谢！

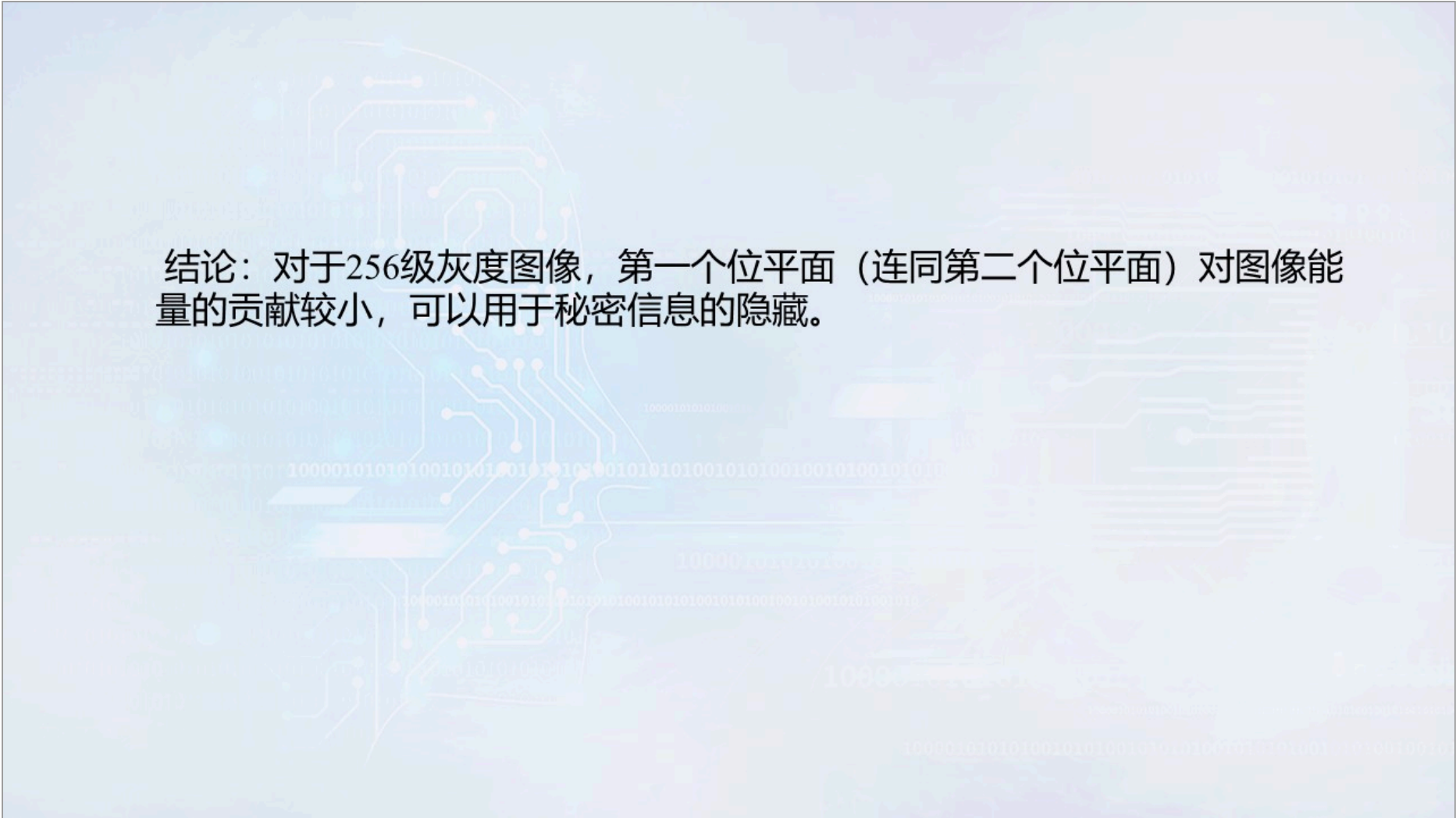
# 位平面算法

依次去除1 ~ t个位平面

```
x=imread(' lenagray.bmp' );
[m,n]=size(x);
for t=1:7
    for i=1:m
        for j=1:n
            x(i,j)=bitset(x(i,j),t,0);
        end
    end
    figure; imshow(x);
    title([' 去除第1-', num2str(t), ' 位平面' ]);
end
```







结论：对于256级灰度图像，第一个位平面（连同第二个位平面）对图像能量的贡献较小，可以用于秘密信息的隐藏。

