



## 07.4 基于视觉特性的 隐写与隐写分析

---

钮心忻、杨榆、雷敏  
北京邮电大学信息安全中心  
[yangyu@bupt.edu.cn](mailto:yangyu@bupt.edu.cn)

# 一类基于视觉特性的隐写方法

---

- 在视觉不敏感区域嵌入较多秘密信息
- 在视觉较敏感区域嵌入少量秘密信息
- BPCS隐写
- PVD隐写

# BPCS 隐写

- BPCS (bit-plane complexity segmentation)  
位平面复杂度分割
- 思想
  - 将载体数据的多个位平面分成小块
  - 人的视觉对变化剧烈、复杂度较高的位平面小块不敏感
  - 秘密信息可以加载在多个位平面

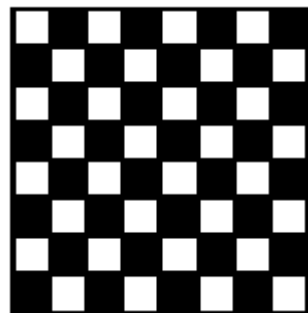
| 样点 |   |
|----|---|
| 3  | 2 |
| 1  | 0 |

| 第2比特平面 |   |
|--------|---|
| 1      | 1 |
| 0      | 0 |

| 第1比特平面 |   |
|--------|---|
| 1      | 0 |
| 1      | 0 |

# BPCS方法（嵌入）

- 将载体图像所有位平面分为小块，如 $8 \times 8$
- 计算每个小块的复杂度
  - 复杂度定义：所有相邻像素对中取值不同的像素对数目。复杂度最大可能值记为 $C_{\max}$
  - 例如， $8 \times 8$ ，复杂度0—112
  - 全0或全1：复杂度0
  - 0、1相间棋盘状：复杂度112



# BPCS方法（嵌入）

---

- 将复杂度大于 $\alpha C_{\max}$ 的位平面小块用于负载秘密信息
  - $\alpha$ 必须小于0.5
- 将秘密信息组成位平面小块，如果其复杂度大于 $\alpha C_{\max}$ ，则直接替换原位平面小块
- 如果其复杂度小于等于 $\alpha C_{\max}$ ，则需要作共轭处理
  - 共轭处理：将秘密信息小块与棋盘状小块作异或
  - 共轭处理后复杂度为 $C_{\max} - c$
  - 记录下哪些小块经过共轭处理

## 例

---

- 设分块大小为 $2 \times 2$ , 每像素值用3 比特表示,  $\alpha=0.4$ 。
- 1、 $C_{\max}$ 为多少?
- 2、若已知小块像素（行优先排列）为（7,5,4,6），其各个比特平面的复杂度为?
- 3、有哪些比特平面适于隐藏信息?

# 例

---

○ 解：

○ 1、  $C_{\max}=4, \alpha = 0.4,$

- $\alpha * C_{\max} = 0.4 * 4 = 1.6$

# 例

---

- 解：
- 2~3、
- 像素为7,5 即： 111 101
- 4,6           100 110
- 三个比特平面，从底到高分别为：
- (1 1)   (1 0)   (1 1)
- (0 0)   (0 1)   (1 1)
- 其复杂度为2, 4,0，因此可以隐藏秘密信息的是最低和次低比特平面



# 例

---

- 设秘密信息为
- 1 1 和 0 0
- 1 0    0 0
- 则隐藏秘密信息后，上述像素值变为？

## 例

---

- 解：
- 则第一块1 1复杂度为2，大于 $\alpha * C_{\max}$
- 1 0
- 可以直接替换，
- 第二块要0 0复杂度为0，小于 $\alpha * C_{\max}$
- 0 0
- 应与棋盘小块1 0异或，得：1 0
- 0 1                      0 1

## 例

---

- 解：
- 替换后，三个比特平面，从底到高分别为
- $(1\ 1)$      $(1\ 0)$      $(1\ 1)$
- $(1\ 0)$      $(0\ 1)$      $(1\ 1)$
- 还原为十进制值为：
- 7 5
- 5 6

## BPCS方法（提取）

---

- 将载体数据中所有复杂度大于 $\alpha C_{\max}$ 的位平面小块取出，即是秘密信息
- 做过共轭处理的小块：再作一次共轭，即是秘密信息

# BPCS 方法

---

- 一般不采用二进制形式划分位平面，而是采用循环码划分位平面
- 二进制划分位平面：会有许多小块复杂度大于 $0.5C_{\max}$
- 循环码划分位平面：绝大多数小块复杂度小于 $0.5C_{\max}$

# BPCS 方法

---

- 二进制码和循环码的互换
- B: 二进制
- G: 循环码

$$G_{N-1} = B_{N-1}, G_{N-2} = B_{N-1} \otimes B_{N-2}, G_{N-3} = B_{N-2} \otimes B_{N-3}, \dots, G_0 = B_1 \otimes B_0$$

$$B_{N-1} = G_{N-1}, B_{N-2} = B_{N-1} \otimes G_{N-2}, B_{N-3} = B_{N-2} \otimes G_{N-3}, \dots, B_0 = B_1 \otimes G_0$$

# BPCS 方法

表 6.1.1 数字 0~7 的二进制码和循环码

| 十进制 | 二进制 | 循环码 |
|-----|-----|-----|
| 0   | 000 | 000 |
| 1   | 001 | 001 |
| 2   | 010 | 011 |
| 3   | 011 | 010 |
| 4   | 100 | 110 |
| 5   | 101 | 111 |
| 6   | 110 | 110 |
| 7   | 111 | 100 |

# 对BPCS的隐写分析

---

- BPCS隐写的视觉隐蔽性较好，但会改变位平面小块复杂度的统计特性
- 思想
  - 对原始图像所有位平面小块的复杂度进行统计，将其直方图记为 $h(c)$ ， $c$ 取值： $0-C_{\max}$
  - 直方图特点：
    - 直方图集中于 $c$ 值较低的一侧（因为高位位平面相关性很强，复杂度很低）
    - 直方图连续性较好（因为对不同位平面许多小块的统计结果）



# 对BPCS的隐写分析

- $512 \times 512$  标准灰度图像Man按  $8 \times 8$  分块，得到的复杂度直方图

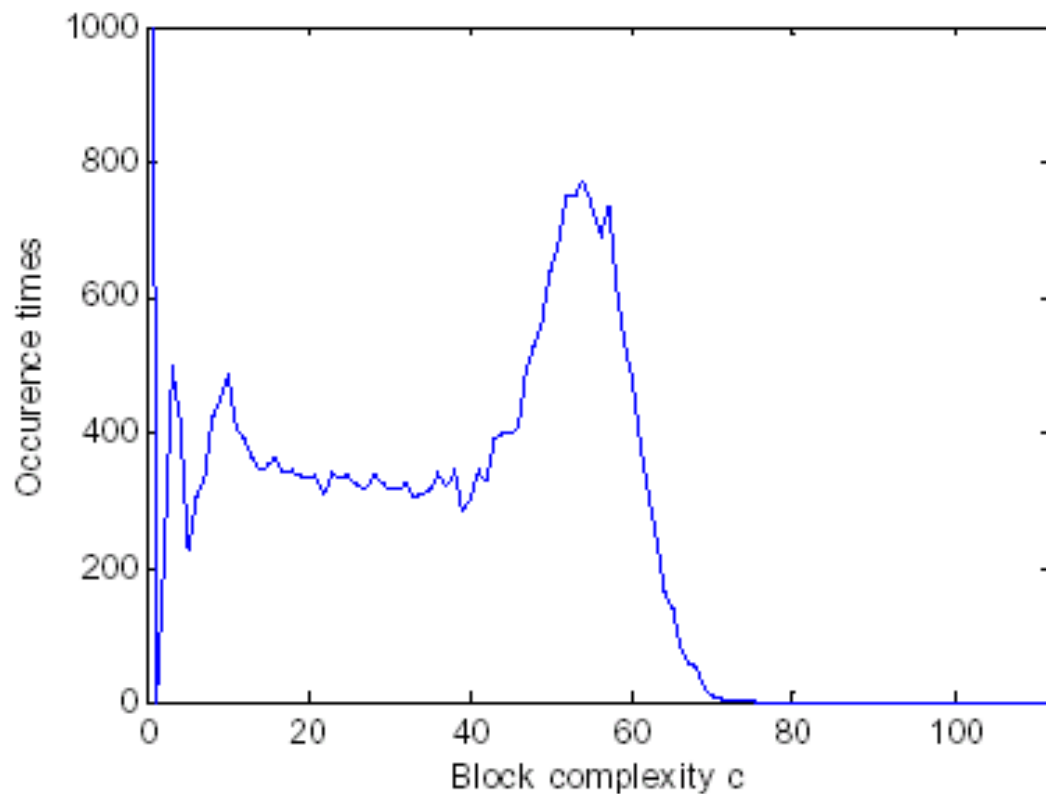


图 6.1.2 由原始图像 Man 得到的复杂度直方图

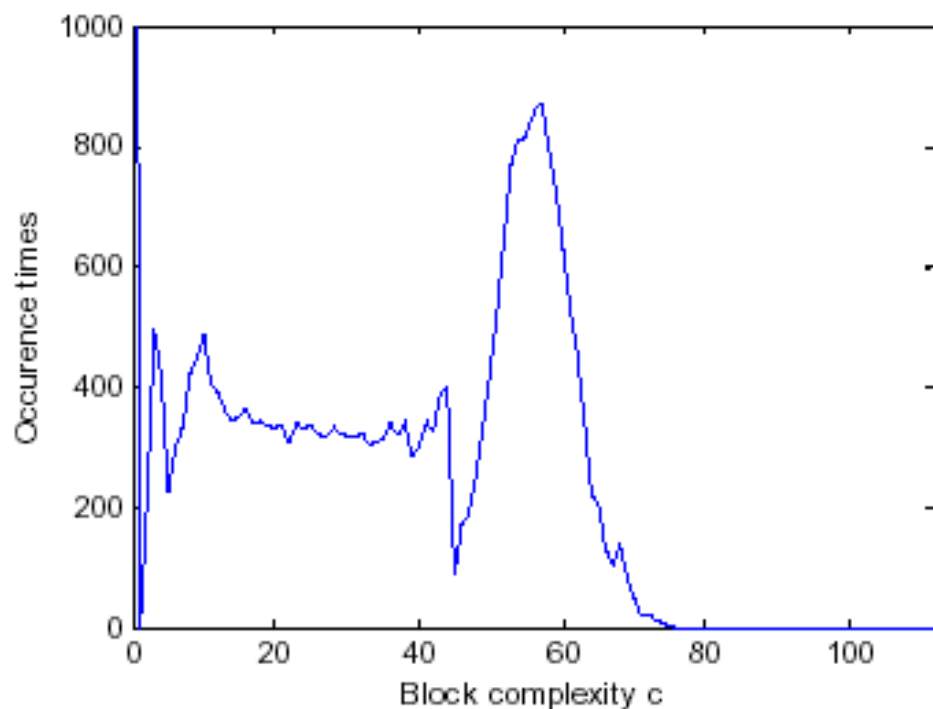
# 对BPCS的隐写分析

---

- BPCS隐写是将复杂度大于 $\alpha C_{\max}$ 的位平面小块置换成秘密信息组成的小块
- 秘密信息组成的小块
  - 随机信号，0/1概率0.5
  - 有文献对4096000个秘密信息小块进行了统计，秘密信息小块的复杂度近似正态分布
  - 均值 $0.5C_{\max}$ ，标准差 $0.047C_{\max}$
- 用秘密信息小块替换后，出现直方图不连续的现象

# 实验结果1

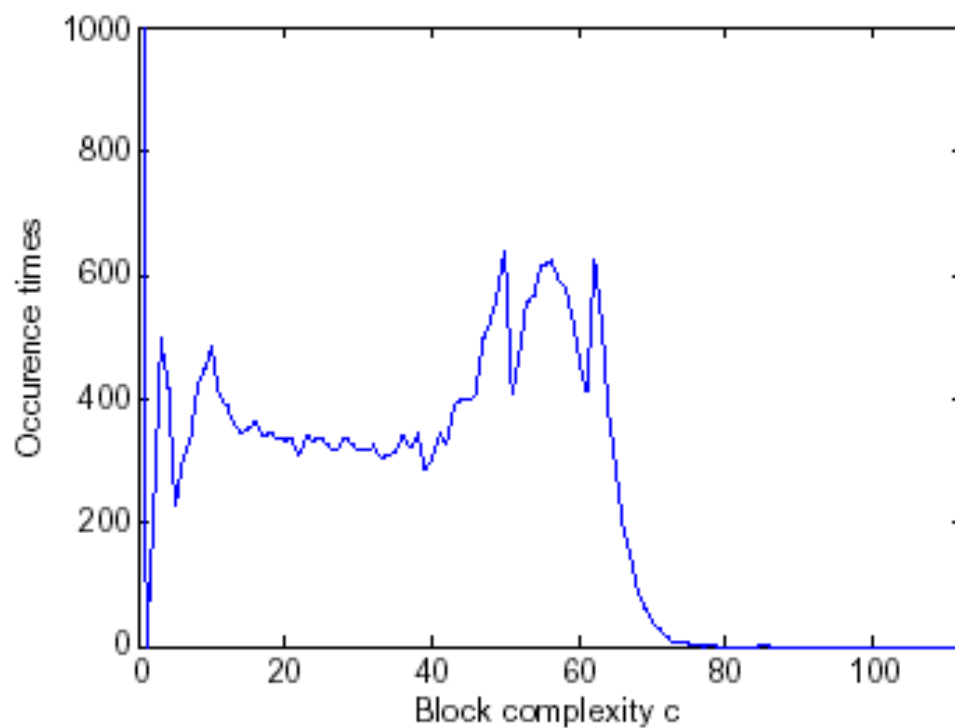
- 图像Man
- $8 \times 8$
- $\alpha=0.4$
- 嵌入量  $7.2E5$
- $PSNR=33.5\text{dB}$
- 在  $0.4C_{\max}=45$  处有明显不连续跳跃



(a) 复杂度直方图  $h(c)$

## 实验结果2

- ★ 图像Man
- ★  $8 \times 8$
- ★  $\alpha=0.45$
- ★ 嵌入量  $5.2E5$
- ★  $PSNR=36.7\text{dB}$
- ★ 在  
 $0.45C_{\max}=50$   
处有明显不连续  
跳跃



(a) 复杂度直方图  $h(c)$

# PVD 隐写

---

- PVD (pixel-value differencing)
- 基于图像中相邻像素对差异的隐写

# PVD隐写算法

---

- 将载体图像分成许多不交迭的小块，每个小块由两个相邻像素组成
  - 组成方式有很多种，如逐行或逐列 zigzag 扫描
- 秘密信息被隐藏在每个小块的两个像素灰度的差值中

# PVD隐写算法

---

- 像素灰度差值记为  $d=p(i+1)-p(i)$
- 将这个范围分成K个区域，每个区域的宽度是2的整数幂，记为 $2^n$ 
  - 例如分成6个区域：[0,7], [8,15], [16,31], [32,63], [64,127], [128,255]
- 如果差值  $|d|$  落在某一个区域，那么在这个小块中嵌入  $n$  比特
  - 嵌入方法：调整  $p(i+1)$  和（或）  $p(i)$  的值，使得它们的差值  $|d|$  等于  $n$  比特对应的十进制值

# PVD隐写算法

---

- 根据PVD算法，设 $k$ 值为6，即灰度被划分为6个区间 $[0,7]$ ,  $[8,15]$ ,  $[16,31]$ ,  $[32,63]$ ,  $[64,127]$ ,  $[128,255]$ ，则
  - 像素对  $(1, 8)$  和  $(8, 18)$  分别能隐藏多少比特秘密信息？
  - 若秘密信息为111和000，则隐藏像素值对变为？



# PVD隐写算法

---

- 根据PVD算法，设 $k$ 值为6，即灰度被划分为6个区间 $[0,7]$ ,  $[8,15]$ ,  $[16,31]$ ,  $[32,63]$ ,  $[64,127]$ ,  $[128,255]$ ，则
  - 解1：值对1差值 $|d|=7$ ，落入区间 $[0,7]$ ，区间宽度为8，因此可以隐藏 $\log_2(8)=3$ 比特信息。
  - 类似的，值对2差值 $|d|=10$ ，落入区间 $[8,15]$ ，区间宽度为8，因此可以隐藏 $\log_2(8)=3$ 比特信息。

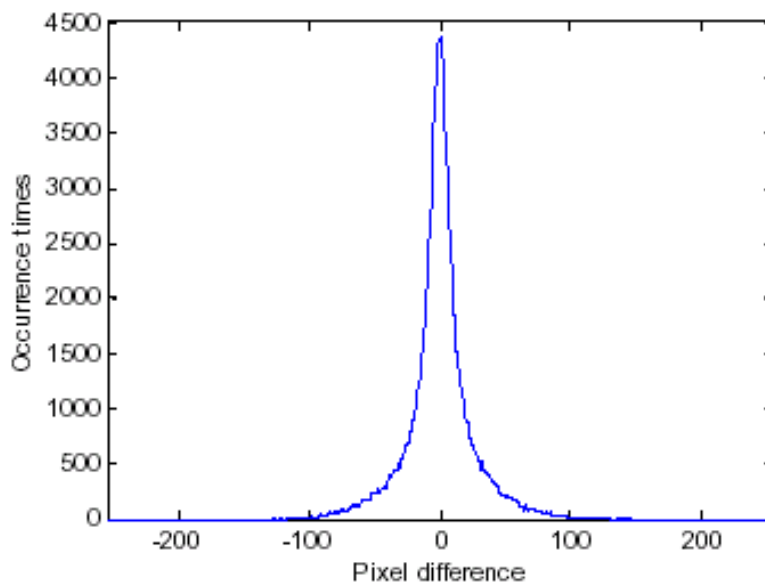
# PVD隐写算法

---

- 根据PVD算法，设k值为6，即灰度被划分为6个区间 $[0,7]$ ,  $[8,15]$ ,  $[16,31]$ ,  $[32,63]$ ,  $[64,127]$ ,  $[128,255]$ ，则
  - 解2：秘密信息为 $(111)_b$ ，因此差值应调整为 $(7)_d$ ，即隐藏秘密信息为，像素值仍为 $(1,8)$
  - 秘密信息为 $(000)_b$ ，对应十进制值为0，但差值区间为 $[8,15]$ ，因此要加上偏置8，差值应调整为 $(8)_d$ ，即隐藏秘密信息为，像素值为 $(8,16)$

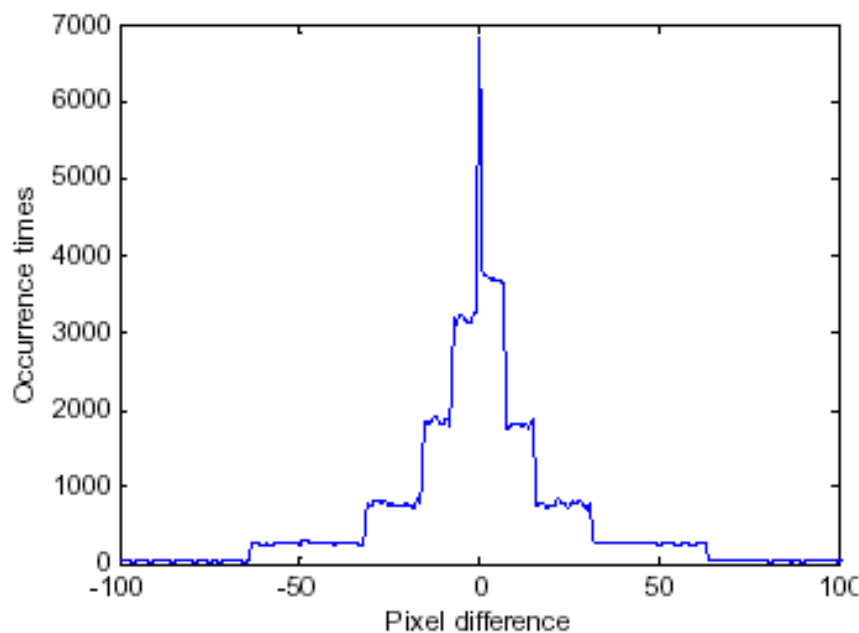
# 对PVD的隐写分析

- 考察图像小块像素差值的直方图
- 通常一幅正常图像的像素差值直方图应该比较平滑，并且随着  $|d|$  的增加而减小

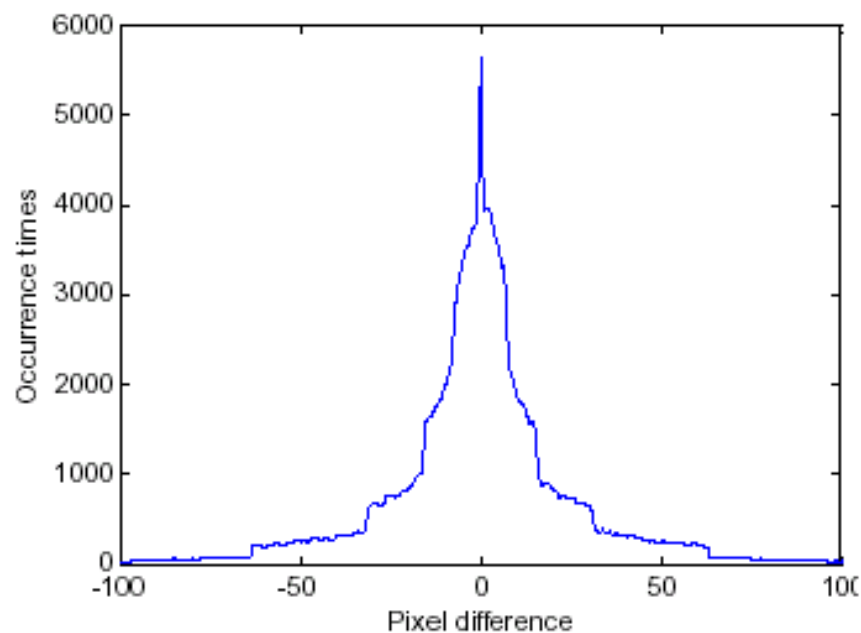


# 对PVD的隐写分析

## ○ 嵌入之后



(a)  $\alpha=99.6\%$



(b)  $\alpha=50\%$

图 6.2.2 由含秘图像 Baboon 得到的像素灰度差值直方图

# 隐写分析总结

---

- 找出隐写引起的特征变化
  - 直方图
  - 奇异点，等
- 与嵌入量有关
- 设计隐写算法时，从隐写分析的角度考虑其安全性
- 对已发表的各类隐写算法，有可能做一些深入的工作
  - 需要耐心细致的研究