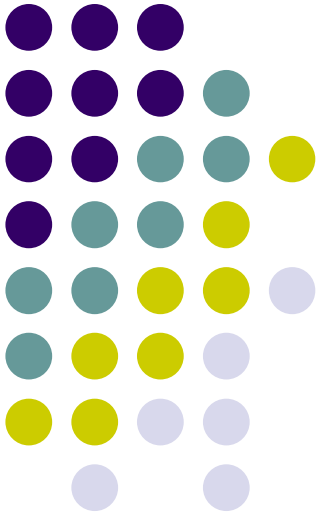
信息隐藏与数字水印

第9讲：图像隐密术(索引色图)

（Image Steganography）

周琳娜

2023.04.26

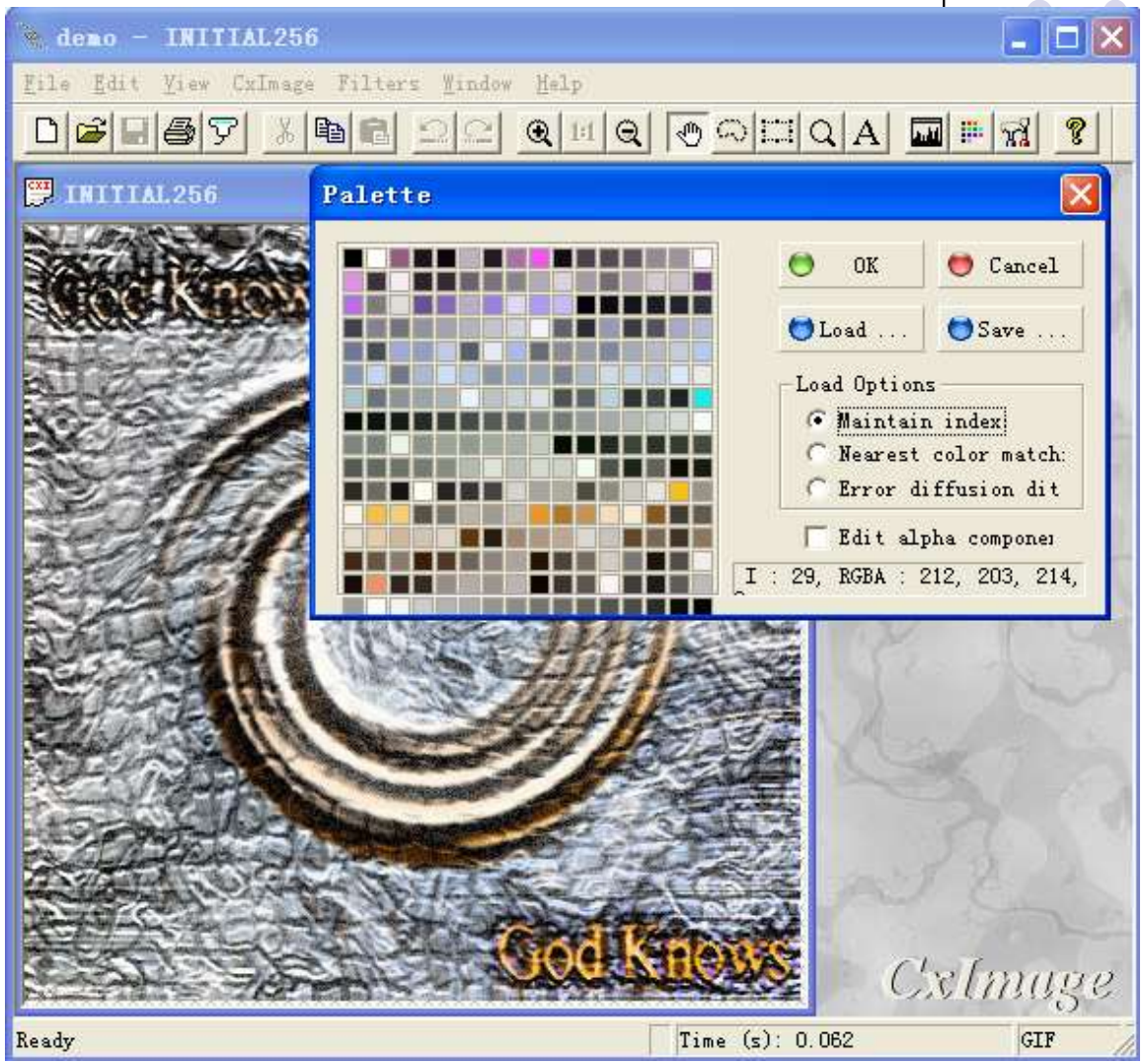


图像隐密术

- 1、图像基本知识
- 2、位图隐密术
- 3、索引图隐密术
- 4、变换域图隐密术
- 5、隐蔽通信实践（互动）

调色板图像格式特点（1）

- **调色板图像：**
- **图像数据中含有颜色数小于256的调色板，像素值不直接表示颜色，而是代表调色板某一颜色的索引值。**
- **数据量相对较小，便于在Internet上传输。**



调色板图像格式特点（2）

- 真彩色图像的颜色量化：(BMP,GIF支持)

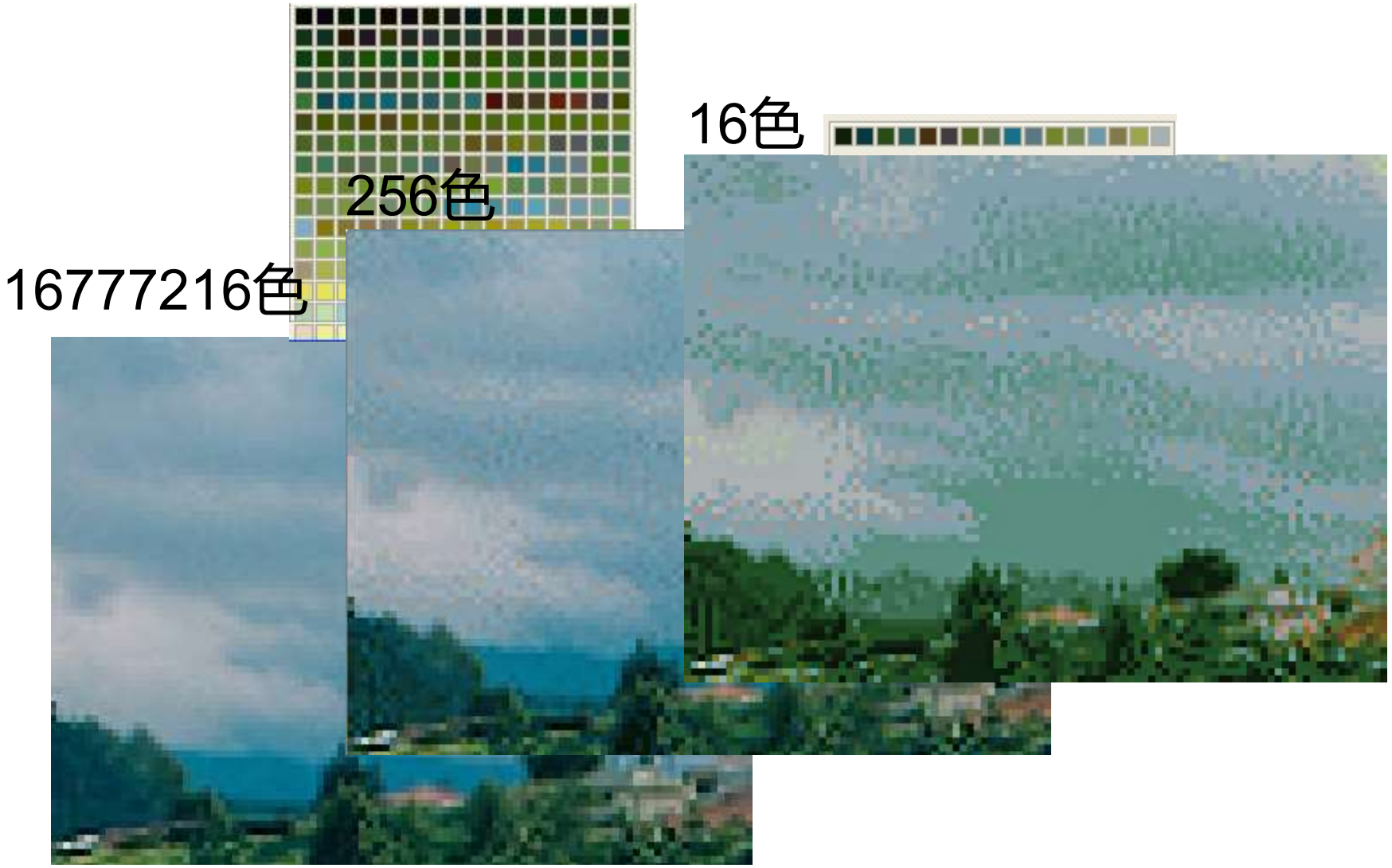
256色

16色

16777216色

调色板图像格式特点（2）

- 真彩色图像的颜色量化：(BMP,GIF支持)



调色板图像格式特点（3）

动画图像的应用：彩信、微博等丰富载体形式

调色板图像格式数据

索引图像是一种把像素值直接作为RGB调色板下标的图像。一幅索引图包含一个数据矩阵data和一个调色板矩阵map，data可以是unit8、unit16或双精度类型，而map矩阵总是一个m×3的双精度类型矩阵（其中，m表示颜色数目），该矩阵的元素都是[0，1]内的浮点数。map矩阵的每一行指定一个颜色的红、绿、蓝颜色分量。索引图像可以把像素值直接映射为调色板数值，每一个像素的颜色使用data的数值作为map的下标来获得：数值1表示map的第一行，数值2表示map的第二行，依此类推。

调色板图像格式数据

下面图中图像数据矩阵大小为256×256，表示有65535个像素点构成。调色板矩阵大小为256×3，表示有256种颜色。我们看到图像索引矩阵的(1,1)单元的内容为125，也就是说这一点像素的颜色就是调色板矩阵的第125行所定义的颜色。

调色板图像格式数据

Array Editor: woman

File Edit View Web Window Help

Numeric format: shortG Size: 256 by 256

	1	2	3	4	5	6	7	8	9	10
1	125	79	111	98	111	79	125	98	91	112
2	98	112	91	112	79	111	112	79	112	79
3	98	125	98	111	112	98	79	111	112	111
4	106	98	79	112	79	125	98	94	98	79
5	79	125	98	91	98	98	79	111	112	125
6	98	111	112	98	94	111	112	79	98	79
7	112	79	111	94	98	79	111	112	91	112
8	111	112	98	111	112	91	112	79	98	91
9	106	98	79	112	79	98	111	112	111	112
10	111	112	111	91	112	111	94	111	79	98
11	79	98	125	98	112	91	98	112	111	94
12	111	94	111	98	91	112	111	79	112	111

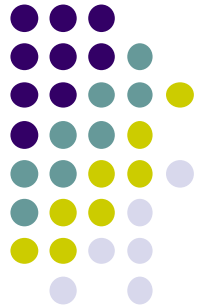
Array Editor: map

File Edit View Web Window Help

Size: 256 by 3

	1	2	3
123	0.59851	0.59851	0.59851
124	0.60194	0.60194	0.60194
125	0.60536	0.60536	0.60536
126	0.60877	0.60877	0.60877
127	0.61218	0.61218	0.61218
128	0.61557	0.61557	0.61557
129	0.61896	0.61896	0.61896
130	0.62234	0.62234	0.62234
131	0.62572	0.62572	0.62572
132	0.62908	0.62908	0.62908
133	0.63244	0.63244	0.63244
134	0.63579	0.63579	0.63579
135	0.63913	0.63913	0.63913
136	0.64247	0.64247	0.64247
137	0.64579	0.64579	0.64579
138	0.64911	0.64911	0.64911
139	0.65243	0.65243	0.65243
140	0.65573	0.65573	0.65573
141	0.65903	0.65903	0.65903
142	0.66232	0.66232	0.66232
143	0.66561	0.66561	0.66561

调色板图像格式数据



我们今后会大量使用以下三条对索引图像的操作语句：

读取索引图像：

[data,map]=imread(filename,permission);

显示索引图像：

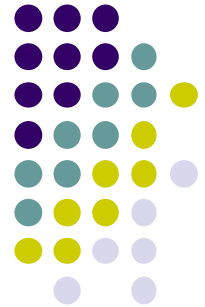
image(data),colormap(map);

存储索引图像：

imwrite(data,map,filename,permission)

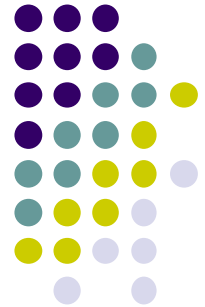
其中，data为像素索引矩阵，map为调色板矩阵，filename为文件路径，permission为图像文件格式。

索引色图像的典型代表-GIF介绍



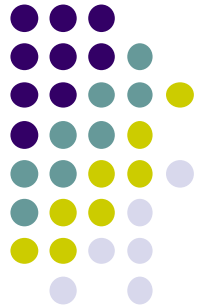
- BMP、JPEG、GIF是目前最为常见的三种数字图像格式，在互联网上得到广泛应用。
- GIF图像格式中主要包括调色板和图像数据，图像数据包含每个像素指向调色板的索引值。调色板最多包含256种颜色，图像数据采用了LZW无损压缩，因而图像文件较小，最大嵌入量较大。

调色板图像隐密术



- 典型的调色板图像格式—GIF。
- 一幅 GIF 图像最多有256种颜色。
- 可通过修改调色板嵌入隐蔽信息。
- 相邻颜色指数可能对应不同颜色，简单地用隐蔽数据替代LSB 可能引起明显的颜色异常。
- 为克服这一缺点，某些密写工具如EzStego先重新排列调色板，使连续的颜色指数代表相近颜色。

GIF流行嵌入算法的分类



- 只改动调色板

  Gif-shuffle
- 同时改动调色板和索引值

  S-Tools、Hide&Seek 、 CameraShy
- 只改动索引值

  EZ Stego、GIF-IT-UP

静态GIF典型隐藏算法（1）

- **按颜色重新排序调色板**GifShuffle

软件：GifShuffle

原理简介：把调色板中的颜色重新排列，用颜色顺序的不同组合隐藏信息。由于GIF图像调色板中最多有256种颜色，因而这种隐藏方法的最大隐藏容量仅为

log2(256!)=210个字节。

缺点：嵌入容量小，有些图像处理软件打开或保存图像会改变调色板排序，使得嵌入数据不能提取。

安全性：目前尚没有检测其算法的方法，在一定程度上较好，但由于其诸多缺点使得其应用不广。

静态GIF典型隐藏算法（2）

- **修改调色板颜色的LSB**

原理简介：根据嵌入数据分别改动R、G、B三分量的最低有效位，在调色板颜色数目为256的情况下，其隐藏信息的容量为256×3 = 768比特，即96个字节大小的信息。

缺点：嵌入容量小，改变调色板索引顺序也会引起嵌入数据丢失，LSB嵌入的对效应。

安全性：差

静态GIF典型隐藏算法（3）

- **修改调色板和图像数据**Hide&Seek

原理简介： Hide&Seek将调色板中各颜色分量进行划分，生成一个特殊的调色板（RGB中颜色值均为4的倍数），然后提取128个基色并经过细微的修改后扩展到256个颜色，产生可互相替换的相近颜色，嵌入时根据嵌入数据进行相应的替换。

缺点： 在调色板中留下了人为的痕迹,生成的调色板中存在成组的相近颜色，如右图所示。

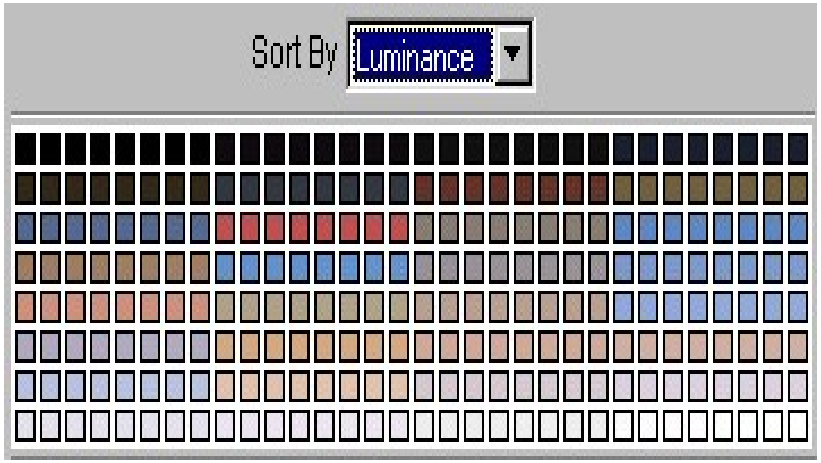
安全性： 视觉效果很差，无法抵抗调色板检测攻击。

静态GIF典型隐藏算法（4）

- **修改调色板和图像数据**Stools

原理简介： S-Tools算法首先根据嵌入数据大小从原图像的调色板中提取基色（按亮度），使得调色板颜色数量减少，最小基色数目为32色，嵌入数据越大其提取的基色数越少（提取32位基色扩展后每像素能嵌入3bit信息），然后新的基色经过细微的修改后被扩展到整个调色板中，扩展到256个颜色，产生可互相替换的相近颜色，嵌入时根据嵌入数据进行相应的替换。

缺点： 在调色板中留下了人为的痕迹,生成的调色板中存在成组的相近颜色，如右图所示。



安全性： 视觉效果差，无法抵抗调色板检测攻击。

静态GIF典型隐藏算法（5）

- **修改调色板和图像数据Camera/Shy**

原理简介： Camera/Shy只能针对使用颜色数较少的图像，当图像数据中使用到的颜色数为256种时，是不能成功嵌入信息的（只是用网络安全色替代该图像的调色板）。当使用到的颜色数较少时，该算法提出调色板中实际用到的颜色，根据嵌入容量选择部分颜色进行扩展但不一定扩展到256色，可扩展成32、128或256个颜色，嵌入时进行相近颜色索引的替换。

缺点： 在调色板中留下了人为的痕迹,生成的调色板中存在成组的相近颜色，对大部分自然图像不能成功嵌入信息。

安全性： 其视觉效果较前两种好，无法抵抗调色板检测攻击。

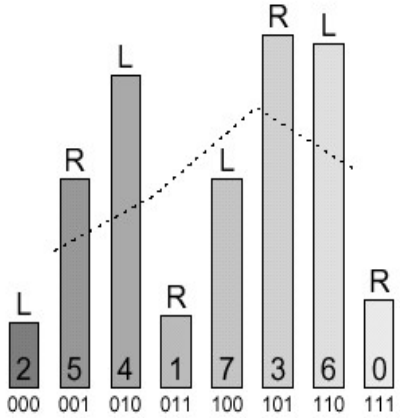
静态GIF典型隐藏算法（6）

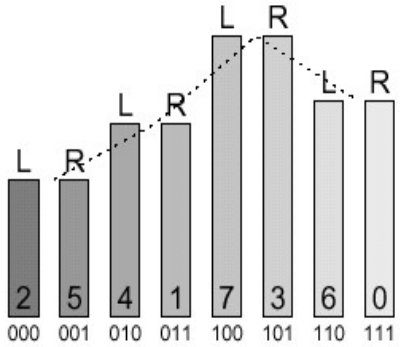
- **调色板颜色预分对Ez_Stego**

原理简介：将原始的调色板根据颜色相近度进行排序得到排序后的调色板的索引值；然后将排序后的调色板的索引值两两分组，信息嵌入过程中，如需改变索引值用同组索引值代替。

缺点：会造成“对效应”，如下图所示

安全性：对效应使得其安全性降低，无法抵抗χ^2统计攻击。





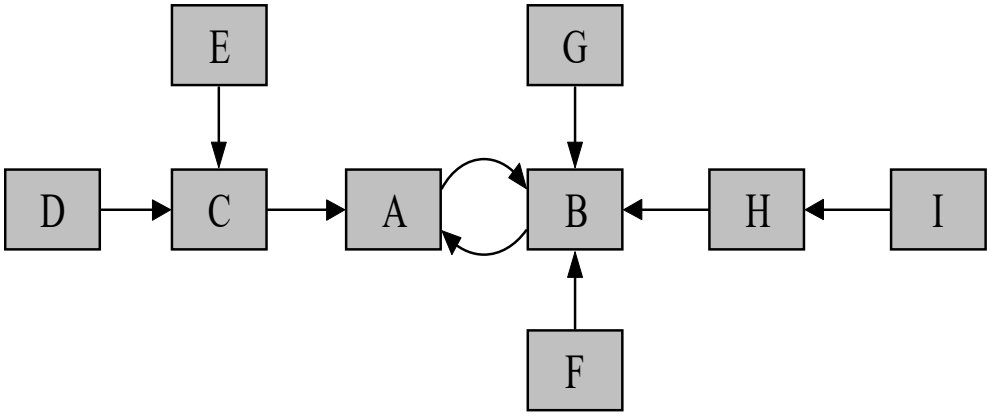
静态GIF典型隐藏算法（7）

- **调色板颜色最优分对法GIF-IT-UP**

原理简介：在寻找每个颜色的替换颜色时遵循最短距离原则，每个颜色索引可以同时成为其他几个颜色索引的替换索引，但没个颜色索引只能由唯一的一个颜色索引替换，其替换转移结构入图。

缺点：其替换转移结构仍会有统计特征。

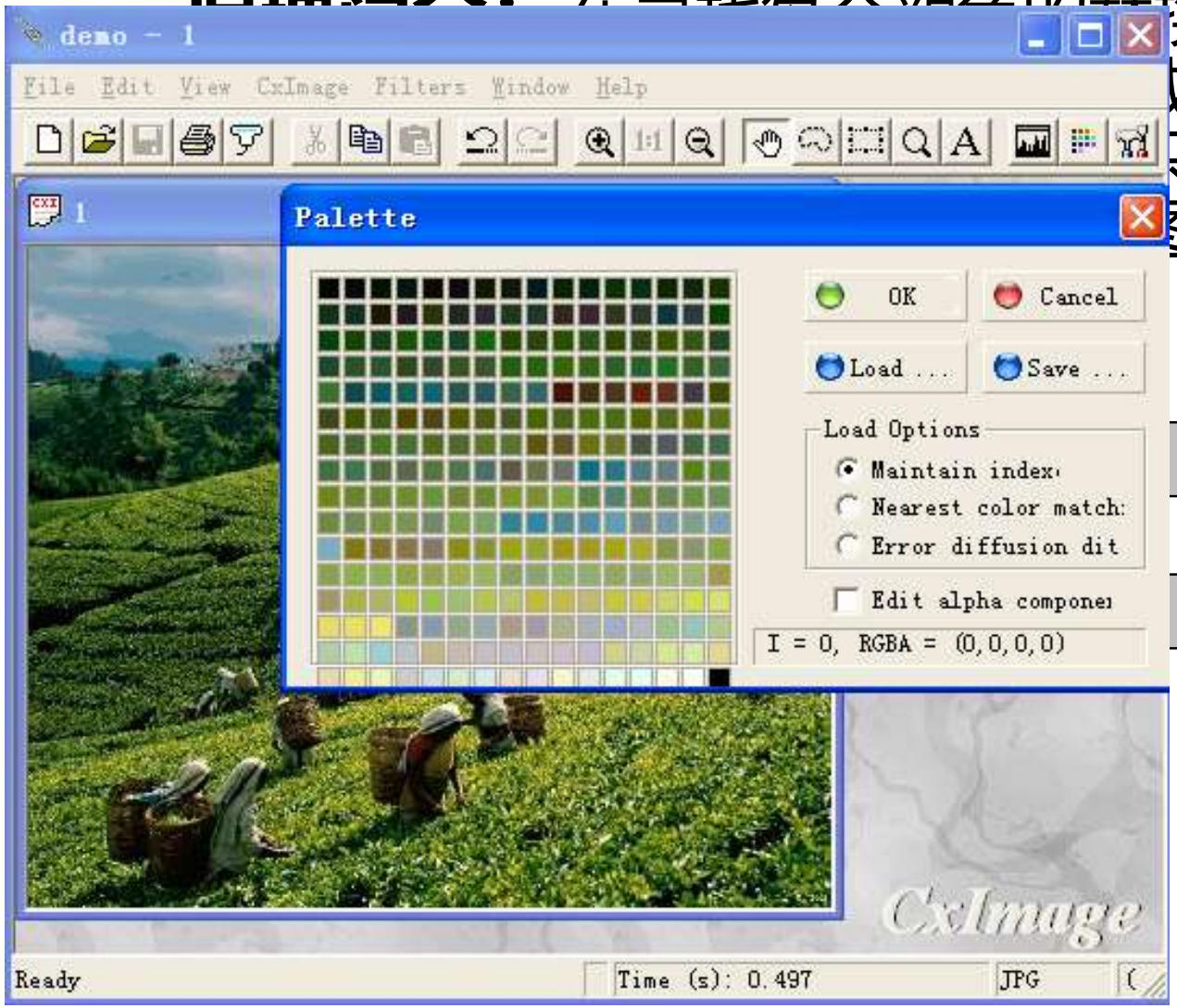
安全性：其视觉效果较好，不会产生“对效应”，统计安全性较高。

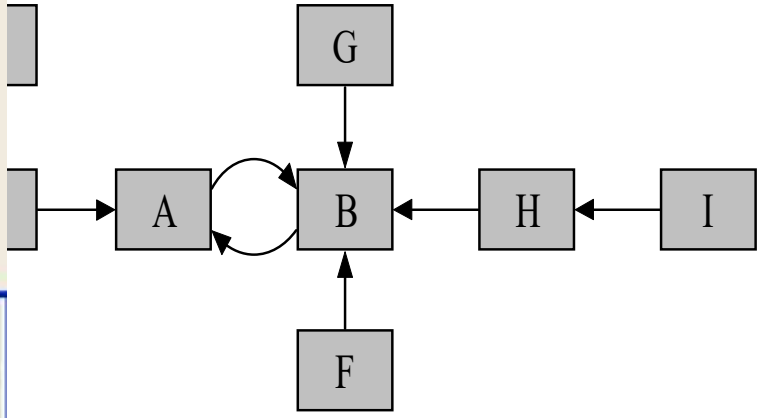


静态GIF典型隐藏算法（7）

- **调色板颜色最优分对法GIF-IT-UP**

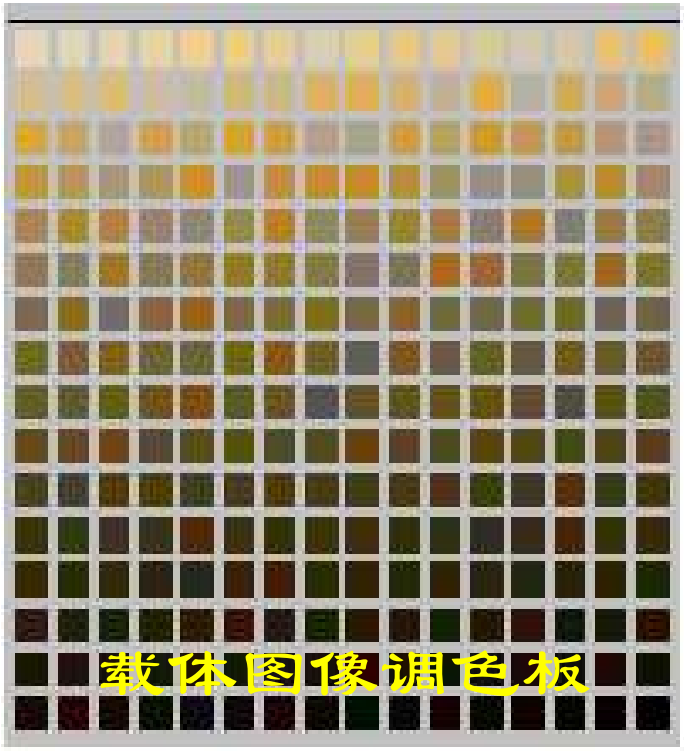
原理简介：在寻找每个颜色的替换颜色时遵循最短距离成为其他几个颜色索引能由唯一的一个颜色。

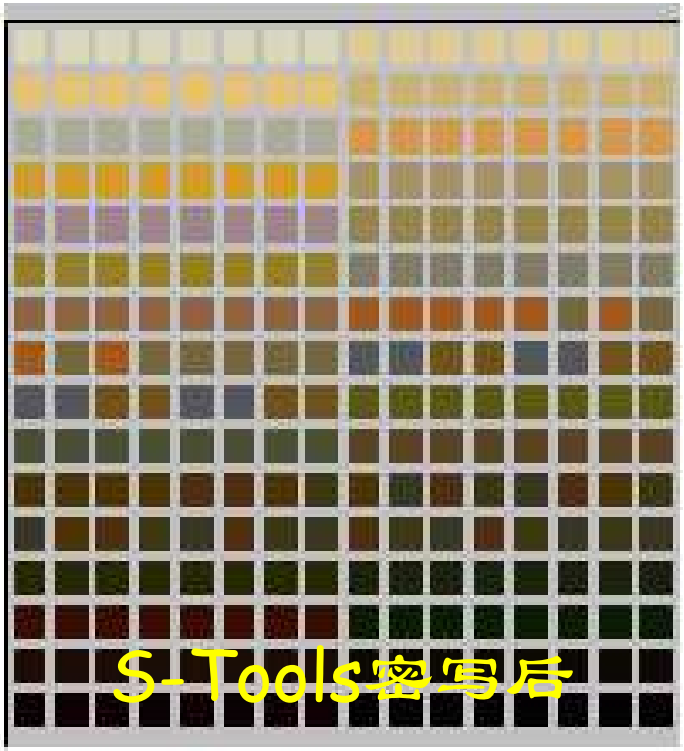




调色板图像隐密分析

- 有一些密写法根据密钥将调色板置乱，将隐蔽数据嵌入，而图像本身却没有任何变化。
- 异乎寻常的调色板足以引起密写分析者的警觉。

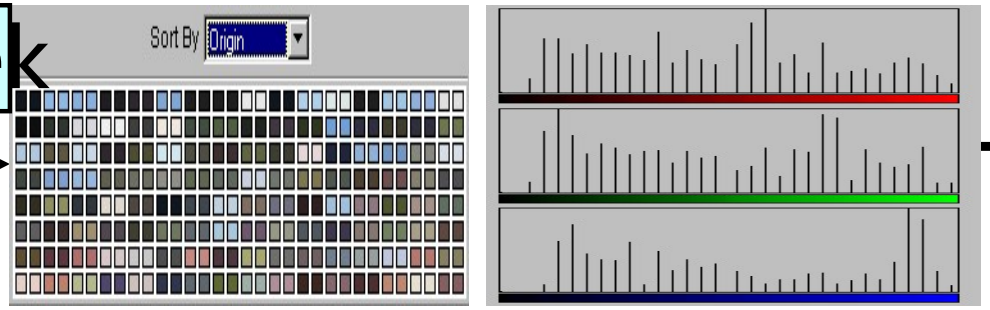




静态GIF典型分析算法（1）

- **针对Stools和Hide&Seek的一阶分析方法**

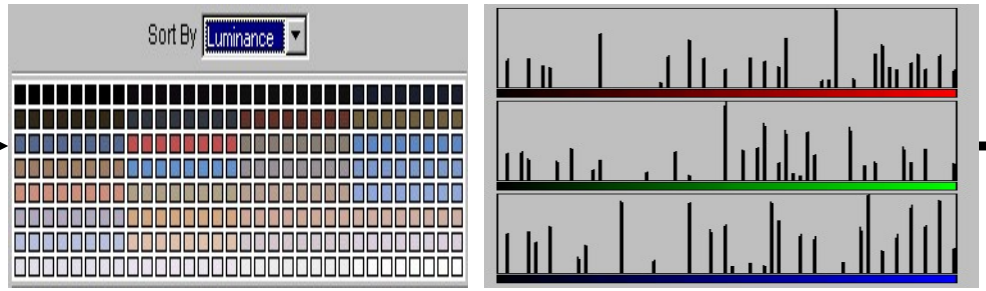
Hide&Seek



$$p = \frac{D_a^2}{D_h \cdot D_i}$$

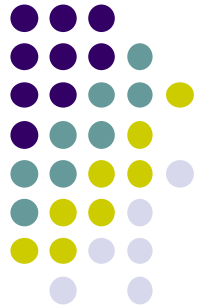
Da为各颜色分量直方图中的孤立成对数，Dh为各颜色分量直方图中的所有成对数，De为各颜色分量直方图中坐标为偶数的孤立非成对数，Do为各颜色分量直方图中坐标为奇数的孤立非成对数 。Di为各颜色分量直方图中的以4为步长的非孤立的色度级的个数

STools



$$p = \frac{D_a^3}{D_h^2 \cdot (D_e + D_o)}$$

静态GIF典型分析算法（2）



- **针对EzStego的分析方法**

  原理简介： 由于EzStego产生的对效应与LSB替换产生的对效应类似，因此可以按照RS的分析思路来设计针对EzStego的分析算法。

  算法步骤：

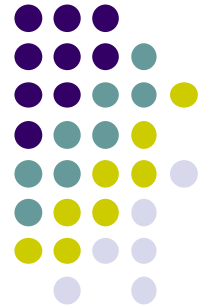
  （1）设图像颜色总数为P，按照EzStego算法中的排序方法对调色板排序，分成如下颜色对：

  $$E = \{(c\pi(0), c\pi(1)), \cdots, (c\pi(P-2), c\pi(P-1))\}$$

  （2）对E进行“shift”操作，得到E’：

  $$E' = \{(c\pi(1), c\pi(2)), \cdots, (c\pi(P-1), c\pi(0))\}$$

静态GIF典型分析算法（3）



算法步骤：

（3）用0代表cπ(2k)，1代表cπ(2k+1)对图像按行或列扫描得到Z(cπ(2k), cπ(2k+1))，对所有的对儿进行扫描把所有的二进制序列连接起来：

$$Z = Z(c\pi(0), c\pi(1)) \,\&\, \ldots \,\&\, Z(c\pi(P-2), c\pi(P-1))$$

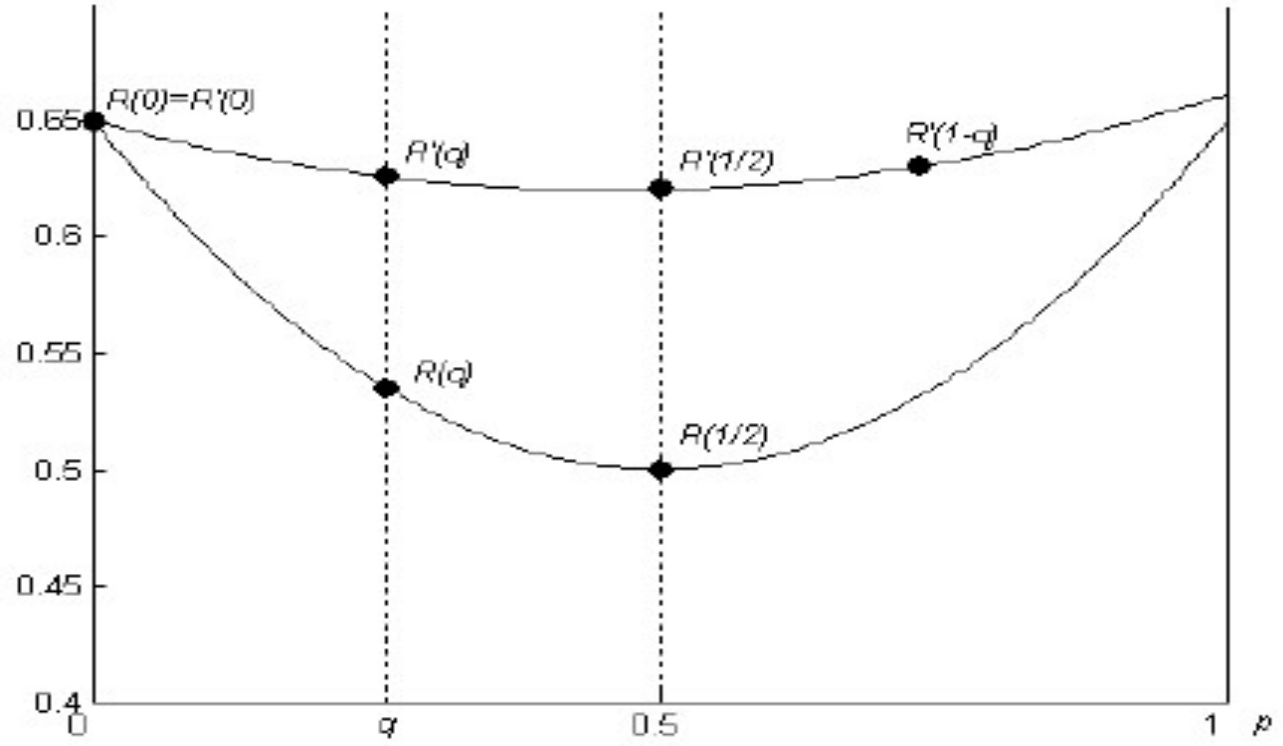
（4）用0代表cπ(2k-1)，1代表cπ(2k)对图像按行或列扫描得到Z(cπ(2k-1), cπ(2k))，对所有的对儿进行扫描把所有的二进制序列连接起来：

$$Z' = Z(c\pi(1), c\pi(2)) \,\&\,\&\, \ldots \,\&\, Z(c\pi(P-1), c\pi(0))$$

（5）计算序列中“同质对”所占的比列，所谓同质对即是序列中连着的00或11，如0001100001中同质 对的个数为6，比值为0.6

静态GIF典型分析算法（4）

- 根据实验得：
  - 当图像为载体图像时，R(0)=R’(0)；
  - 当图像为满嵌图像时，R(1/2)=0.5；
  - 两条曲线可近似为二次曲线；



静态GIF典型分析算法（4）

设$D(p) = R(p) - R'(p) = ap^2 + bp + c$

代入已知条件：

$$D(0) = R(0) - R'(0) = c = 0$$

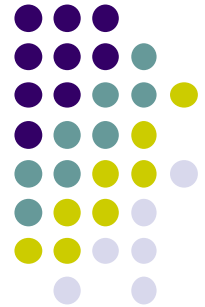
$$D(1/2) = R(1/2) - R'(1/2) = a/4 + b/2$$

$$D(1 - q) = a(1 - q)2 + b(1 - q)$$

推出$4D(1/2)q + [D(1-q) - D(q) - 4D(1/2)]q + D(q) = 0$

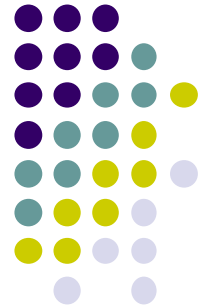
求解较小值为修改概率q

静态GIF算法设计关键问题（1）



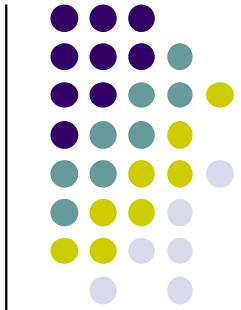
- **隐藏算法设计：**
  - 调色板颜色的替换方式应尽量考虑替换前后的视觉质量，即视觉上相近颜色的度量；
  - 尽可能不用固定的“颜色对”方式，而采用更复杂的颜色分组方式来避免出现“对效应”；
  - 真彩色图像的颜色量化方法不是唯一的，因而由同一幅真彩图像得到的调色板图像的数据特性可能不同，这使更安全的隐藏算法成为可能。

静态GIF算法设计关键问题（2）



- **分析算法设计：**
  - 调色板图像的宏观统计量源于自然图像统计特征和颜色量化过程；
  - 统计量的统计规律可通过对上述统计特征的来源分析有目的地进行实验和建模。

动态GIF典型隐藏算法（1）



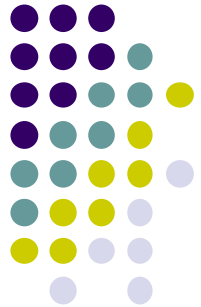
- 技术思路
  - ✓ 利用相邻帧对应位置的两个索引值的差的奇偶性来代表“0”和“1”。
  - ✓ 替换颜色准则

$$d_{i,j}=\sqrt{(R_j-R_i)^2+(G_j-G_i)^2+(B_j-B_i)^2}$$

  - ✓ 逐帧修改

n个索引值可以嵌入（n-1）个秘密信息比特

动态GIF典型隐藏算法（1）



- 技术分析
  - ✓ 替换颜色选择前提是颜色的索引值奇偶性不同；
  - ✓ 图像单调区域的修改视觉差异明显；
  - ✓ 奇异颜色、两种颜色对内替换，抗检测能力弱

动态GIF典型隐藏算法（1）

GIF图像的包装域

文本编程.gif

```
00000000h: 47 49 46 38 39 61 C0 03 D0 02 F7 00 00 00 32 98
00000010h: 00 30 90 00 2E 8C 00 32 94 00 2A 7C 00 0E 2C 00
00000020h: 0A 1C 00 28 78 00 12 38 00 2E 88 00 00 14 00 26
00000030h: 74 00 00 18 00 20 60 00 1E 5C 00 18 4C 00 18 48
00000040h: 00 0E 30 00 1E 58 00 32 90 00 2C 80 00 2A 80 00
```

avd_保健.gif

```
00000000h: 47 49 46 38 39 61 40 01 F0 00 77 00 00 21 FF 0B
00000010h: 4E 45 54 53 43 41 50 45 32 2E 30 03 01 00 00 00
00000020h: 21 F9 04 00 32 00 00 00 2C 00 00 00 00 40 01 F0
00000030h: 00 87 00 00 00 80 00 00 00 80 00 80 80 00 00 00
00000040h: 80 80 00 80 00 80 80 C0 C0 C0 C0 DC C0 A6 CA F0
```

0024.GIF

```
00000000h: 47 49 46 38 39 61 C8 00 2E 01 E7 00 00 00 00 00
00000010h: 80 00 00 00 80 00 80 80 00 00 00 80 80 00 80 00
00000020h: 80 80 C0 C0 C0 C0 DC C0 A6 CA F0 00 00 33 00 00
00000030h: 66 00 00 99 00 00 CC 00 33 00 00 33 33 00 33 66
00000040h: 00 33 99 00 33 CC 00 33 FF 00 66 00 00 66 33 00
00000050h: 66 66 00 66 99 00 66 CC 00 66 FF 00 99 00 00 99
00000060h: 33 00 99 66 00 99 99 00 99 CC 00 99 FF 00 CC 00
```

Fm.gif

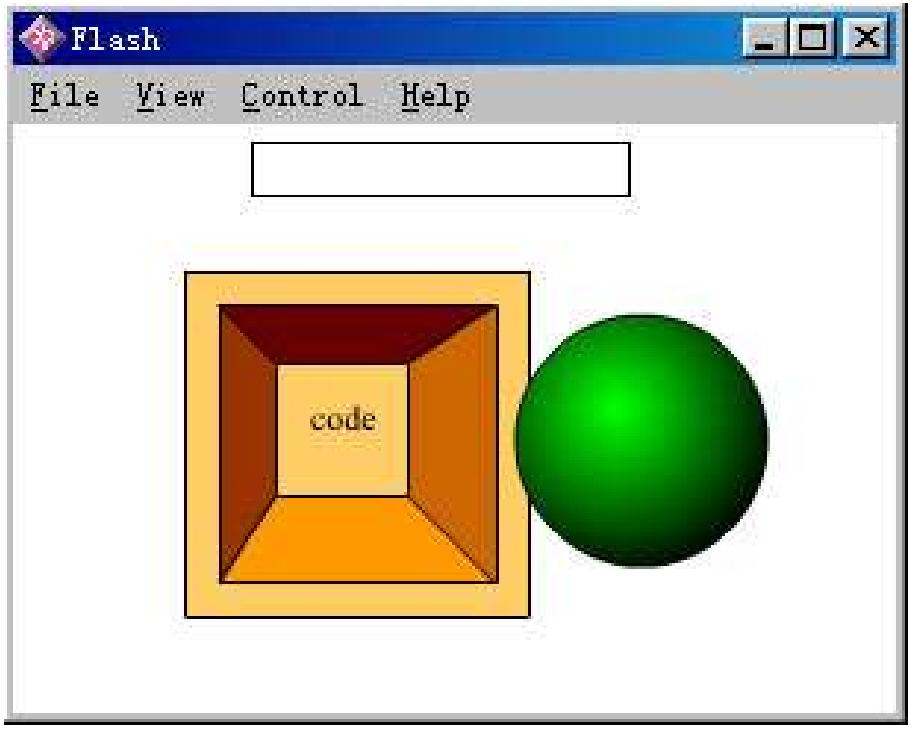
```
00000000h: 47 49 46 38 39 61 2C 01 90 01 D7 00 00 00 00 00
00000010h: 00 28 00 00 50 00 00 6E 00 00 96 00 00 BE 00 00
00000020h: DC 00 00 FF 00 00 00 6E 00 28 6E 00 50 6E 00 6E
00000030h: 6E 00 96 6E 00 BE 6E 00 DC 6E 00 FF 6E 00 00 BE
00000040h: 00 28 BE 00 50 BE 00 6E BE 00 96 BE 00 BE BE 00 ; .(?P?n?柧.掀.
```

```
void CxImageGIF::EncodeHeader(CxFile *fp)
{
    fp->Write("GIF89a",1,6);       //GIF Header

    Putword(head.biWidth,fp);              //Logical screen descriptor
    Putword(head.biHeight,fp);

    BYTE Flags;
    if (head.biClrUsed==0){
        Flags=0x11;
    } else {
        Flags = 0x80;
        Flags |=(head.biBitCount - 1) << 5;
        Flags |=(head.biBitCount - 1);
    }

    fp->PutC(Flags); //GIF "packed fields"
    fp->PutC(0);     //GIF "BackGround"
    fp->PutC(0);     //GIF "pixel aspect ratio"

    if (head.biClrUsed!=0){
        RGBQUAD* pPal = GetPalette();
        for(DWORD i=0; i<head.biClrUsed; ++i)
        {
            fp->PutC(pPal[i].rgbRed);
            fp->PutC(pPal[i].rgbGreen);
            fp->PutC(pPal[i].rgbBlue);
        }
    }
}
```

动态GIF典型隐藏算法（1）

嵌入结果对比

常用矢量图—卡通图


Flash
File View Control Help
code

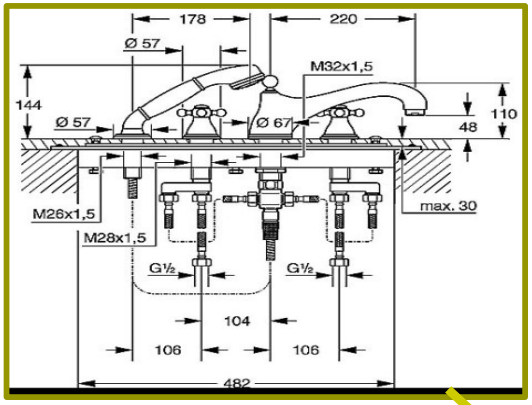

泡了一杯红茶
Steep a cup of black tea
倒掉
Pours out

常用矢量图—地图、PPT

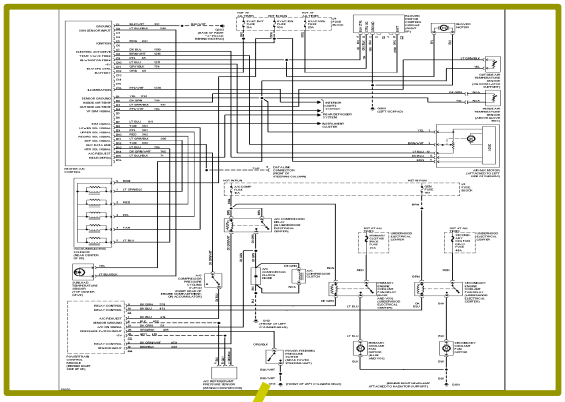

中国联通 3G 10:32
嘀嘀打车
国际关系学院
北宫门
国际关系学院
泉河快速路
颐和园路
地铁4号线
万寿山
同庆街
西苑
我在 新建宫门路21号院 附近
海淀区
颐东苑
明湖东路
中信所西苑
昆明湖路
公园路
海淀公园
六郎庄
六郎庄路
现在用车
预约

常用矢量图—工程图

机械工程图

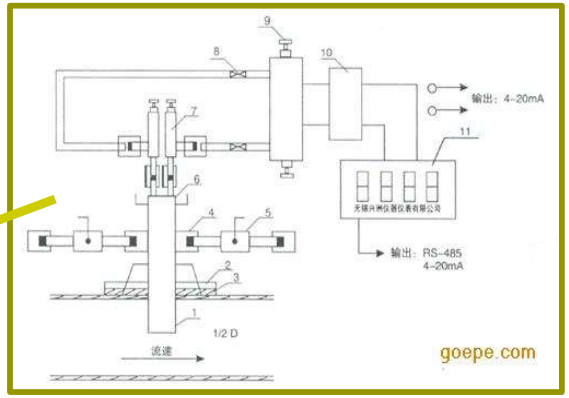


PCB工程图



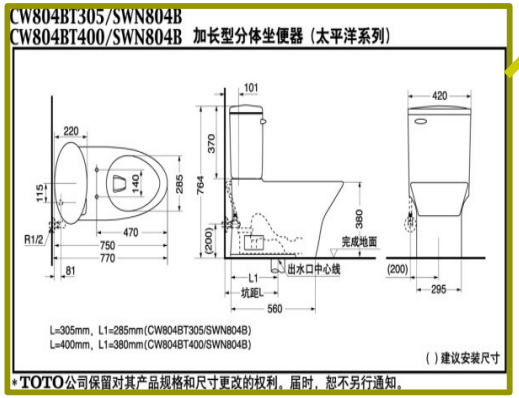
...

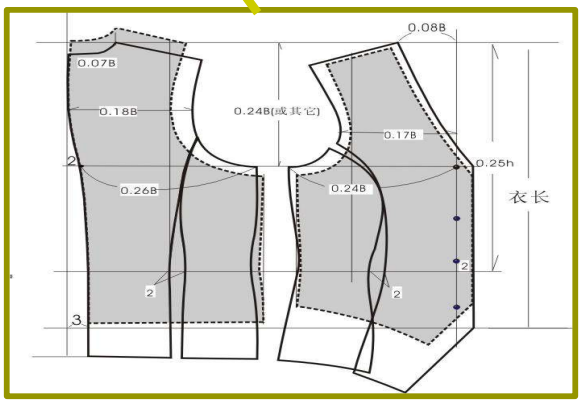
管道工程图



二维CAD工程图

家具工程图





服装工程图

矢量图信息隐藏

- 技术分析
  - ✓ 矢量微调的掩密方式：利用人类的视觉冗余，微小地改变物体的形状、颜色和线条粗细等来携带秘密信息。

矢量图信息隐藏

- 技术分析
  - ✓ 矢量微调的掩密方式：利用人类的视觉冗余，微小地改变物体的形状、颜色和线条粗细等来携带秘密信息。

矢量图信息隐藏

- 技术分析
  - ✓ 矢量微调的掩密方式：利用人类的视觉冗余，微小地改变物体的形状、颜色和线条粗细等来携带秘密信息。

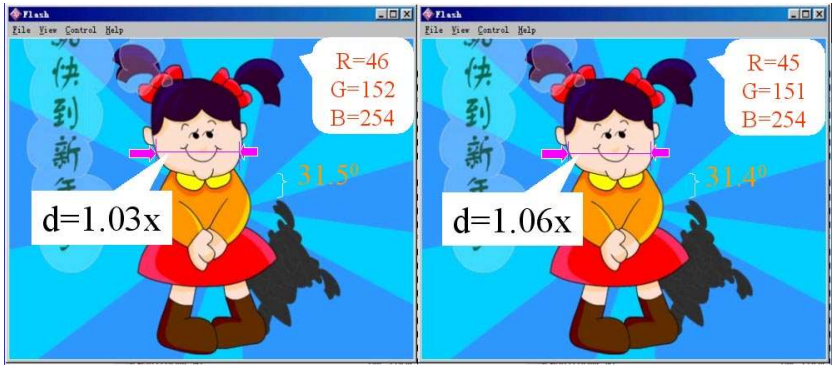
矢量图信息隐藏

- 技术分析
  - ✓ 矢量微调的掩密方式：利用人类的视觉冗余，微小地改变物体的形状、颜色和线条粗细等来携带秘密信息。

矢量原图　　携带了秘密信息后的矢量图

矢量图信息隐藏

- 技术分析
  - ✓ 矢量微调的掩密方式：利用人类的视觉冗余，微小地改变物体的形状、颜色和线条粗细等来携带秘密信息。



将式 $f'_i = f_i + \alpha |f_i| \omega_i + \beta \omega_i$ 中 β 先取0，在隐藏容量和载体的保真度之间权衡实验后得

$$\alpha = 0.15$$

再根据上幅矢量图的各个对象的形状、颜色和线条粗细等携密参数的不失真的信噪比（SNR）下限约束 和式

$$\beta \in \left[0, -\alpha \mu_{|f|} + \sqrt{\alpha^2 \mu_{|f|}^2 - \mu_{f^2}(\alpha^2 - \frac{1}{P})}\right]$$

⇒ $\beta = 0.013$

矢量图信息隐藏

- 技术分析
  - ✓ 替换颜色选择前提是颜色的索引值奇偶性不同；
  - ✓ 图像单调区域的修改视觉差异明显；
  - ✓ 奇异颜色、两种颜色对内替换，抗检测能力弱

需掌握的内容

- GIF图像数据分为几个部分?各有什么作用?
- GIF图像信息隐藏分为哪几类?
- 当载体为GIF图像时,图像数据能否用LSB嵌入?

休息中。。。

欢迎大家继续讨论！