

matlab 隐写分析,RS隐写分析.ppt

转载

卡休微卡 于 2021-03-21 11:48:53 发布 114 收藏 1

文章标签: [matlab 隐写分析](#)

RS隐写分析,rs隐写分析matlab,rs隐写分析算法,隐写分析,图像隐写分析matlab,图像隐写分析方法,rs网络分析仪,rs分析法,rs触发器分析,rs网络分析仪怎么用

RS analysis 高深 平滑度函数 我们用如下的公式来衡量一个图像的平滑度,平滑度函数体现了一个图像相邻像素的平滑程度: LSB的嵌入会给图像增加噪声,一般情况下f值也会随之增加。LSB嵌入操作函数F(x) F(x)分为如下三种情况 关于F翻转函数 在LSB中我们只用到了F1函数,我们进一步的看这个函数实际上是定义了二进制数字末位的0到1和1到零的变换 而F-1函数在相邻数对的翻转的过程中已经涉及到了二进制数字的倒数第二位。比如1到2,那么就是01到10的跳变了 对像素组G进行交换操作 我们把图像分成若干个像素组G 设掩码算子 $M(m_1, m_2, \dots, m_n)$, m取值为0, 1或-1 例: $G=(39,38,40,41)$, $M=(1,0,1,0)$, $-M=(-1,0,-1,0)$ 那么有: 对像素组G进行F操作 正则组(Regular) 奇异组(Singular) 不变组(Unusable) 把图像像素组G进行F操作, 如果: R_m , R_{-m} , S_m , S_{-m} R_m 为F1作用下正则组占有所有像素组的比例 R_{-m} 为F-1作用下正则组占有所有像素组的比例 S_m 为F1作用下奇异组占有所有像素组的比例 S_{-m} 为F-1作用下奇异组占有所有像素组的比例 判断是否经过LSB隐写 如果待检图像没有经过LSB, 那么无论用F1还是F-1从统计上来说, 一般都会同等程度的增加图像块的混乱程度, 也就是说 R_m 近似等于 R_{-m} , S_m 也近似等于 S_{-m} , 而且 $R_m > S_m$, $R_{-m} > S_{-m}$ 如果是经过LSB密写的(即部分像素用过F1操作), 用F1和F-1就会有明显的不同。判断是否经过LSB隐写 对密写图像分小块进行F1翻转 R_m 和 S_m 计算时, 所有像素可以分三类: 1.没有经过翻转的: 灰度值不变 2.经过一次翻转的: 灰度值变化幅度为1 3.经过两次翻转的: 两次翻转又回到了原始值 而对密写图像进行F-1翻转时, 也有一些像素经过两次翻转, 但是经历的是一次F1和一次F-1, 这样会与原始值差的更多。判断是否经过LSB隐写 判断是否经过LSB隐写 上图为大量实验得出的, 上图中嵌入率为0和嵌入率为1的地方为一组特殊点 那么如果是没有经过LSB隐写的图像应该满足嵌入率为0时, R_m 和 R_{-m} , S_m 和 S_{-m} 的关系。如果不满足即可认为已经嵌入信息。对密写率的估计 RS分析方法的核心是由于LSB密写仅用到了F1翻转, 而没有用到F-1翻转, 所以当用F-1翻转或F1翻转去处理密写图像时, 参数呈现了不对称性, 用这种不对称性, 可以进一步估计出密写率。前面我们已经算出了 R_m 和 R_{-m} , S_m 和 S_{-m} 四个值, 但是我们要确定四条直线, 所以还需要四个值才行, 怎么找出这四个值呢? 对密写率的估计 设密写率是a, 那么前面计算的一组值是在 $a/2$ 的像素被F1翻转情况下得到的。那么将待检测图像所有像素都用一次F1变换, 即 $M=\{1,1,1,1\}$ 。那么相对于原始 图像有 $1-a/2$ 的像素被F1翻转, 此时计算的一组 R_m 和 R_{-m} , S_m 和 S_{-m} 是 $1-a/2$ 的像素被F1翻转情况下得到的。对密写率的估计 那么根据简单的比例关系就可以估计出密写率为 $L1/(L1+L2) 1-a/2 a/2 a=1$, 即密写率为100%时 更进一步对密写率估计 为了更准确的对密写率进行估计, 我们可以对上面的图线进行拟合。根据大量实验结果得出: R_{-m} 和 S_{-m} 与嵌入比例a成线性关系, R_m 和 S_m 是a的二次曲线关系。这个图像是这样的: 更进一步对密写率估计 更进一步对密写率估计 那么应用我们之前得到的几个特殊点和数值, 为了简洁, 做一个变量代换: $z=(x-a/2)/(1-a)$, 即把 $a/2$ 点映射成0, 把 $1-a/2$ 映射成1, 再对图线进行拟合后, 根据嵌入率为0时的 R_m 和 R_{-m} , S_m 和 S_{-m} 的焦点横坐标相等, 可以解出交点的横坐标z: 那么嵌入率就应该为 $z/(z-0.5)$ $d_0=R_m(0)-S_m(0)$ $d_1=R_m(1)-S_m(1)$ $d_0=R_m(0)-S_m(0)$ $d_1=R_m(1)-S_m(1)$ 参考文献 [1] 张开文 张新鹏 王朔中. 数字密写和密写分析[M]北京: 清华大学出版社, 2008: 29-31 [2]王炳锡 彭天强. 信息隐藏技术[M]北京: 国防工业出版社, 2007: 248-254. [3]Fridrich J, Goljan M, Du R. Detecting LSB steganography in color and gray-scale images. Magazine of IEEE Multimedia. Special Issue on Security, Oct-Nov, 2001: 22-28 [4]丁晋晶等.